UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN

MÁSTER UNIVERSITARIO EN
INGENIERÍA DE TELECOMUNICACIÓN

TRABAJO FIN DE MÁSTER

Design and Development of Gamification Techniques for Empowering Museum Experiences based on Interactive Machine Learning Algorithms

ALEJANDRO LÓPEZ MARTÍNEZ

2024

TRABAJO DE FIN DE MASTER

Título: Diseño y Desarrollo de Técnicas de Gamificación para la Potenciación de Experiencias Museísticas basadas en Algoritmos de Aprendizaje Automático Interactivos.

Título (inglés): Design and Development of Gamification Techniques for Empowering Museum Experiences based on Interactive Machine Learning Algorithms

Autor: Alejandro López Martínez

Tutor: Álvaro Carrera Barroso

Departamento: Departamento de Ingeniería de Sistemas Telemáticos

MIEMBROS DEL TRIBUNAL CALIFICADOR

Presidente: —

Vocal: —

Secretario: —

Suplente: —

FECHA DE LECTURA:

CALIFICACIÓN:

UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE
INGENIEROS DE TELECOMUNICACIÓN

Departamento de Ingeniería de Sistemas Telemáticos
Grupo de Sistemas Inteligentes

TRABAJO DE FIN DE MÁSTER

Design and Development of Gamification Techniques for Empowering Museum Experiences based on Interactive Machine Learning Algorithms

FEBRERO 2024

Resumen

La motivación de este proyecto radica en la importancia de los museos en la preservación y diseminación de patrimonio cultural. Estas instituciones aprovechan diversas tecnologías y estrategias digitales para impulsar el *engagement* de los visitantes. En este aspecto, los museos exploran cómo aprovechar el gran potencial del aprendizaje automático para mejorar las experiencias de los visitantes. Desde sistemas de recomendación a audioguías avanzadas y museos virtuales, estas aplicaciones de aprendizaje automático pueden desempeñar un papel fundamental en atraer audiencias a estas instituciones.

Las técnicas mencionadas anteriormente se basan en un enfoque de aprendizaje automático clásico, donde los algoritmos aprenden de los datos sin intervención humana. Cuando la complejidad de los problemas aumenta o no hay suficientes datos de entrenamiento, el rendimiento y resultados proporcionados por el enfoque de aprendizaje automático clásico se ven afectados negativamente. No obstante, estos inconvenientes se pueden superar con intervención humana durante el proceso de aprendizaje del algoritmo. Este enfoque se conoce como aprendizaje automático interactivo y a nuestro leal saber y entender, su potencial no ha sido explorado aún en el campo del patrimonio cultural.

Por tanto, este proyecto se adentra en la explotación del enfoque de aprendizaje automático interactivo para aplicarlo al aprendizaje de patrones de rutas de visitantes de museos. Este reto se basa en una aplicación del Problema del Viajante que se resolverá mediante un algoritmo Max-Min Sistema de Hormigas. En este caso no se pretende aprender las rutas necesariamente más cortas, sino las que mayor satisfacción pueden proporcionar al visitante. El objetivo es desarrollar un sistema que permita aprender de las rutas de visitantes. Así, se gamificará el entrenamiento del algoritmo para lograr una experiencia más atractiva y divertida.

Para ello, este trabajo propone una plataforma de gamificación impulsada por aprendizaje automático interactivo basada en un museo virtual que los usuarios podrán recorrer y que se utilizará para recoger los datos necesarios para ir proporcionando realimentación al algoritmo durante su aprendizaje.

Palabras clave: Aprendizaje Automático Interactivo, Algoritmos de Optimización por Colonia de Hormigas, Gamificación, Museos, Patrimonio Cultural.

Abstract

The motivation for this project is based on the importance of museums in the preservation and dissemination of cultural heritage. These institutions have been exploiting different technologies and digital strategies to boost visitor engagement. In this regard, museums are also exploring how to harness the vast potential of machine learning towards enhancing visitors' experiences. From recommendation systems to enhanced audio-guides and virtual museums, these machine learning applications can play a pivotal role in drawing audiences to their institutions.

The aforementioned techniques rely on classical ML, where algorithms learn from data without human mediation. When the complexity of problems increase or there is a lack of sufficient training data, the performance and results provided by the ML approach are negatively affected. However, in those cases, such hindrance can be overcome if human interaction is taken into account to facilitate machine learning. This approach is known as interactive Machine Learning (iML) and to the best of our knowledge, its potential has not been explored in the cultural heritage field.

Therefore, this project delves into the exploitation of the interactive Machine Learning (iML) approach to apply it to learning about route patterns from museum visitors. This challenge is based on an application of the Travelling Salesman Problem (TSP) which will be tackled using MAX-MIN Ant System (MMAS) algorithm. In this case, the aim is not to necessarily learn the shortest routes, but those that may provide visitors with greater satisfaction. Thus, the goal of this project is to develop a system capable of learning from visitors' routes. For this purpose, the learning phase of the algorithm will be gamified to offer a more entertaining and appealing experience.

In order to achieve this objective, this project proposes an iML-powered gamification platform based on a virtual museum that visitors can navigate and that will be used to collect the necessary data to feed the algorithm during its learning phase.

Keywords: Interactive Machine Learning, Ant Colony Optimisation Algorithms, Gamification, Museums, Cultural Heritage.

Agradecimientos

Quisiera agradecer a todas aquellas personas cercanas y amigos que siempre han estado para apoyarme, tanto en los buenos como en los malos momentos. En especial a mis padres, Raquel y Alejandro, a mi hermano Mario y a mi pareja Claudia. Sois los pilares de mi vida y sin vosotros no sería la persona que soy hoy.

También quiero dar las gracias al Grupo de Sistemas Inteligentes y en especial a mi tutor, Álvaro Carrera Barroso, por haberme brindado la oportunidad de realizar este trabajo. Gracias por tu orientación y apoyo durante todo el desarrollo de este proyecto de fin de máster.

Llega el final de un camino, que pese a haber estado lleno de desafíos, también ha traído muchas alegrías. No tengo ninguna duda que lo volvería a recorrer.

Contents

Resumen **IV**

Abstract **V**

Agradecimientos **VI**

Contents **VII**

List of Figures **X**

List of Tables **XIII**

1 Introduction **1**

1.1 Context and Motivation . . . 2

1.2 Project goals . . . 4

1.3 Structure of this document . . . 5

2 Theoretical Background **6**

2.1 Machine Learning . . . 7

2.1.1 Supervised Learning . . . 8

2.1.2 Unsupervised Learning . . . 9

2.1.3 Reinforcement Learning . . . 10

2.1.4 Interactive Machine Learning . . . 11

2.2 Travelling Salesman Problem . . . 12

2.2.1 Problem Formulation . . . 12

2.2.2 Algorithms for Solving the TSP 13

2.2.2.1 Exact Algorithms 13

2.2.2.2 TSP Heuristics 14

2.3 Ant Colony Optimisation 16

2.3.1 Ant Social Foraging 16

2.3.2 Ant Colony Optimisation (ACO) 17

2.3.2.1 Ant System 18

2.3.2.2 Ant Colony System 20

2.3.2.3 Rank-Based Ant System 21

2.3.2.4 MAX-MIN Ant System 22

3 State of the Art **25**

3.1 Applications of the Travelling Salesman Problem 26

3.2 Applications of Ant Colony Optimisation Algorithms 28

3.3 Review of Interactive Machine Learning Approaches 31

3.4 Solving the Museum Visitor Routing Problem 34

4 Enabling Technolgies **37**

4.1 Virtual Museum Visualization Technologies 38

4.1.1 THREE.js 38

4.1.2 RAMEN 40

4.1.3 Flask 41

4.2 Simulation and Algorithm Controller Technologies 43

4.2.1 MESA 43

4.2.2 Shapely 44

4.2.3 NetworkX 45

4.2.4 MongoDB 48

5 Architecture 50

5.1 Museum Environment Interaction System . . . 53

5.2 Museum Controller System . . . 54

5.2.1 Museum Application Controller . . . 54

5.2.2 Museum Map Manager . . . 55

5.2.3 iML Routing System . . . 57

5.2.4 Gamification System . . . 60

6 Virtual Museum Case Study Development 63

6.1 Virtual Museum Map Construction . . . 64

6.2 Navigation in the Virtual Museum . . . 66

6.3 Virtual Museum Implementation Details . . . 70

6.3.1 Movement Mapping . . . 71

6.3.2 Virtual Museum: Movement Verification . . . 76

7 Evaluation 78

7.1 MMAS Baseline Validation . . . 79

7.2 Interactive MMAS (iML) Evaluation in the Virtual Museum . . . 80

7.2.1 Scenarios Setup . . . 80

7.2.2 Model Testing Stage . . . 83

7.2.2.1 Classic MMAS Model Training . . . 83

7.2.2.2 iML-Enhanced MMAS Model Training . . . 88

7.2.3 Model Evaluation Stage . . . 96

8 Conclusions and Future Work 102

8.1 Conclusion . . . 103

8.2 Future Work . . . 104

A Project Impact **106**

A.1 Social Impact 106

A.2 Economic Impact 107

A.3 Environmental Impact 107

A.4 Ethical Impact 108

B Project Budget **110**

B.1 Project Structure 110

B.2 Physical Resources 112

B.3 Human Resources 112

B.4 Total Cost 113

Glossary **114**

Bibliography **115**

List of Figures

2.1 Illustration of the training framework of supervised Machine Learning . . . 8

2.2 Classification vs regression 8

2.3 Illustration of the operational framework of unsupervised Machine Learning 9

2.4 Difference between clustering and association problems in unsupervised Machine Learning. 10

2.5 Agent-environment interaction in reinforcement learning. 10

2.6 iML human-in-the-loop (glass-box) approach. 12

2.7 Illustration of ant colony convergence to a common and shortest path from their nest to the food source 17

3.1 Illustration of the Vehicle Routing Problem (VRP). 26

3.2 Flexible resource allocation in Elastic Optical Networks (EONs) 27

3.3 Team Orienteering Problem (TOP) illustrated with a use-case of a delivery fleet. 29

3.4 Protein folding example into a two-dimensional configuration. 30

3.5 Screen capture of the Travelling Snakesman iML game. 33

3.6 Museum exhibition topology of Tsiropoulou et al. and route including the exhibits chosen by the visitor. 36

4.1 Perspective and orthographic cameras 38

4.2 Light types in Three.js 39

4.3 Construction of a Three.js mesh Consisting of a geometry and a material. . 39

4.4 RAMEN architecture. 40

4.5 Flask app architecture for a local environment 42

4.6 MESA model components 44

4.7 Creation of a graph with NetworkX, calculation of shortest path between two nodes and visual representation of graph and shortest path. 47

5.1 Methodology for the gamification and training of the iML-powered museum routing platform. 51

5.2 Architecture of the proposed nteractive machine learning-powered museum routing platform. 52

5.3 Flowchart of the Museum App Controller's data distribution and exchange process. 55

5.4 Tasks performed by the Museum Map Manager when exhibit and position data are received. 56

5.5 iML Routing System's Class Diagram 57

5.6 User points awarding mechanism used by the Museum Routing Gamification Subsystem. 60

5.7 Architecture of the Smart Object Gamification Subsystem. 62

6.1 Structure in JSON format of the museum blueprint used in the project, including some examples. 64

6.2 Overhead view of the proposed three-dimensional virtual museum scenario. 66

6.3 Screenshot of virtual museum's login screen. 67

6.4 Screenshot of virtual museum's registration screen. 67

6.5 Virtual museum welcome screen. 68

6.6 Warning displayed to users in the virtual museum when they click on an exhibit that is out of their interaction range. 69

6.7 Interaction with an exhibit in the virtual museum. 69

6.8 Communication sequence used to validate visitor movement. 71

6.9 Coordinate system in the virtual museum. 72

6.10 Representation of the rotation of a vector 74

6.11 Decision and action process to validate the next move of a visitor 77

7.1 Average iteration to yield MMAS best solutions in the chosen TSP reference instances. 79

7.2 Percentage deviation from the optimum solution in the selected TSP instances. 80

7.3 Two-dimensional representation of the museum map with labelled nodes. . 82

7.4 Pheromone distribution by intervals at selected iterations during classic MAX-MIN Ant System (MMAS) model training. 84

7.5 Histograms of pheromone distribution at selected iterations during classic MMAS model training. 86

7.6 Comparison of pheromone densities at the selected iterations. 87

7.7 Comparison of pheromone value distributions and statistics at the selected iterations. 87

7.8 Pheromone distribution by intervals at selected iterations during iML-enhanced MMAS model training. 93

7.9 Histograms of pheromone distribution at selected iterations during iML-enhanced MMAS model training. 94

7.10 Comparison of pheromone densities at the selected iterations during iML-enhanced MMAS model training. 95

7.11 Comparison of pheromone value distributions and statistics at the selected iterations during iML-enhanced MMAS model training. 95

List of Tables

6.1 Visitor movement mappings with respect to their perceived coordinate system 73

7.1 Calibration of MMAS implementation with TSP reference instances 79

7.2 Settings of MMAS typical parameters 81

7.3 Pheromone Intervals 83

7.4 Comparison of artworks visited for each type of reference route used to train the iML-enhanced model. 91

7.5 Route suggested by classic and iml-enhanced mmas models for target artworks in the first test. 97

7.6 Route suggested by classic and iml-enhanced mmas models for target artworks in the second test. 98

7.7 Route suggested by classic and iml-enhanced mmas models for target artworks in the third test. 99

7.8 Route suggested by classic and iml-enhanced mmas models for target artworks in the fourth test. 100

B.1 Project Task Breakdown 111

CHAPTER 1

Introduction

This chapter will describe the project's context and provide a quick summary of all the different aspects that will be explored throughout the project. It will also include a list of objectives to be achieved during the project's execution. In addition, it will introduce the document's structure and provide a summary of each chapter.

1.1 Context and Motivation

Cultural heritage refers to the tangible and intangible legacy of a society inherited from past generations, ranging from traditions and literature to historical monuments and artefacts [1], which are considered fundamental to humanity [2][3]. The concept of cultural heritage also encompasses the idea that civilisation has a major duty to preserve, conserve, and pass on heritage to future generations [4]. In this regard, museums play an important role in the preservation of cultural heritage [5] by curating items that recount events of our history [6].

Thoughtful consideration should be given to how museums approach their cultural heritage exhibitions [7] to effectively enhance visitor engagement [8]. In order to accomplish this goal and attract a wider audience, museums are increasingly promoting immersive and first-hand experiences. This is because research has shown that interactive exhibits can boost visitor engagement and satisfaction [9][10]; favour learning and understanding [11], and overall produce more memorable experiences [12].

A wide range of technologies are being harnessed by museums to tailor interactive visits, making them more appealing. Virtual reality (VR), for instance, enables museum visitors to acquire stimulating and immersive information about museum collections[13]. Similarly, by exploiting augmented reality (AR) technologies, real locations may be integrated with rich media material, allowing museums to bring their treasures to life. [14].

Another promising technology to help museums engage with visitors is the Internet of Things (IoT) [15]. Physical items that have been given digital capabilities through IoT are referred to as ”smart objects” [16]. These can be coupled with applications on users' mobile devices. To keep users motivated and interested, these applications frequently incorporate gamification [17] tactics. Given their popularity and appeal, quiz-based gamification techniques are frequently used[18]. However, a drawback of such games is that questions must be periodically regenerated, and this is a time-consuming task. In addition, the upkeep of smart objects can be taxing and time-consuming.

Therefore, to confront these challenges, I proposed a solution as part of my undergraduate thesis, which was later further developed and refined to be presented as a research paper [19] at Intelligent Environments 2020 conference. The solution consists of a low-maintenance gamified smart object platform that automates the creation of questions by exploiting semantic web technologies.

A prototype was implemented at Joaquin Serna Telecommunication Museum, where QR codes were generated for the exhibits. When a visitor scans one of these QR codes with the designated application, more information is neatly presented to them, and additionally they will be prompted with some automatically generated questions about the item of interest. Points can only be earned by answering questions correctly within the given time limit. Visitors can compare their scores against other users through a leader board, motivating them to proactively interact with more exhibits and overall offer a memorable and enjoyable experience.

In the aforementioned project, the displays and interactions with cultural heritage in museums were explored. However, it did not delve into how and where to find items of interest within the museum.

Despite the diversity of visitor profiles[20], there appears to be a similar problem for the majority of the public: determining where to begin the visit and what to view. Choosing the initial path to navigate is challenging, and visitors may experience a disheartening sense of disorientation anc uncertainty, particularly if it is their first time or the museum is expansive. Additionally, deciding which exhibits to explore can be a difficult task, so visitors tend to focus on the exceptional items as a sample of the broad collection. These inconveniences hinder their museum experience, preventing people from making the most out of their visit.

Hence, the topic of this master's thesis is borne out of this context: the gamified smart object platform must be complemented by a system that aids visitors in the enjoyable and satisfactory navigation of the museum.

In this regard, audio guides are one of the most popular and adopted technologies in museums [21]. Typically, a selection of exhibits is chosen, and each is allocated a numbered stop. These, in turn, are usually arranged according to a predetermined path. Such route should attempt to be appealing and interesting for the visitor, and achieving this can be challenging. In fact, museums have been redefining their audio guides in an effort to improve their tours. To do so, they are employing several strategies, including audience segmentation [22] and an emphasis on audio guides as a value-added service, as opposed to a product[23]. Nonetheless, these approaches require significant time and financial commitments to adapt a tour to the preference of visitors.

The task of tailoring a preferred route can also be tackled from a mathematical perspective. There are many algorithms to find the shortest route from one point to another, such as the well-known Dijkstra [24] algorithm. In these cases there is no restriction on which nodes to traverse, just that they must overall yield the shortest path. Nevertheless, when the restriction of having to visit every node once is applied, finding an optimal solution to the problem becomes computationally infeasible for situations with a large number of nodes. This then becomes an optimisation issue well exemplified by the Travelling Salesman Problem (TSP) [25], which necessitates the application of heuristic algorithms to arrive at computationally feasible solutions in an efficient manner. Among the most successful algorithms for this purpose is the Min-Max Ant System [26], based on Ant Colony Optimization (ACO) techniques.

However, in the museum space, the objective is not to discover the fastest route to the exhibits, but rather to develop routes that the public may find more entertaining. Given that the cost measure is stated in terms of user satisfaction, there is yet no data available to train an algorithm to seek not the shortest but the most entertaining path. This paucity of data, coupled with the complexity of the problem, might significantly impair the efficacy of Machine Learning algorithms [27].

Thus, in order to be able to tackle this computationally hard problem and find the most satisfactory routes for visitors, we will resort to an interactive Machine Learning (iML) [28] approach. This is, in the project at hand we will delve into the implementation of MMAS where visitors will intervene in the learning loop of the algorithm to account for their preferences towards the computation of the most satisfactory path.

1.2 Project goals

As aforementioned, the ultimate objective of this project is to design a system that suggests museum visitors the most interesting routes in order to increase their satisfaction throughout the visit. To achieve this, interactive machine learning techniques will be applied to figure out visitors' route patterns. Gamification tactics will be utilised in a virtual museum scenario to make the algorithm's learning process more interesting and engaging to users. Concretely, the system will be built based upon the Smart Object gamification platform implemented as part of my end of degree project. It will be expanded to conceive an integral system that provides an enhanced museum experience from start to end. In short, the goals set for this project are the following:

- Augmentation of the Smart Object Gamification Platform to include a full gamification system for museums with the iML-powered routing module.
- Devise a virtual museum scenario that users can navigate and hence collect visitor feedback to train the iML algorithm.
- Train the iML routing algorithm to find routes that provide increased visitor satisfaction.

1.3 Structure of this document

This master's thesis is organised as follows:

- **Chapter 1 - Introduction**: within this chapter (the current one) an overview of the context surrounding the project is presented. Furthermore, the goals for this project are elucidated.
- **Chapter 2 - Theoretical Background**: the purpose of this chapter is to provide the reader with a clear outline of the fundamental concepts that underpin this work. It aims to offer a foundation for understanding the implementation of the interactive machine learning algorithm employed in this project.
- **Chapter 3 - State of the Art**: a succint analysis of real-world cross-domain applications of the algorithms employed in this project is offered. Additionally, a literature review is conducted that focuses on related works within the museum domain.
- **Chapter 4 - Enabling Technologies**: this chapter will describe and justify the different technology choices that have brought the project to fruition.
- **Chapter 5 - Architecture**: this chapter elaborates on the architecture of the proposed interactive machine learning routing system, offering a comprehensive description of all the modules involved.
- **Chapter 6 - Interactive Machine Learning System Development**: this chapter focuses on detailing the development process of the virtual museum simulation. Moreover, within the chapter a description is given on the implementation of the proposed interactive machine learning algorithm for the construction of museum routes. Also, an overview of the various gamification strategies employed within the virtual museum is outlined.
- **Chapter 7 - Evaluation**: results of the training phase of the algorithm are shown, and the effectiveness of the algorithm will be assessed based on the objectives set.
- **Chapter 8 - Conclusion**: the outcomes and achievements of the project are presented in this chapter, as well as potential venues for future research.
- **Appendix A - Project Impact**: this appendix reflects on the impact of this project in various dimensions, including the social, environmental, econonomic and ethical implications.
- **Appendix B - Project Budget**: an assessment of the required resources and human labour together with its pertaining costs will be analysed in this appendix.

CHAPTER 2

Theoretical Background

In this chapter we will examine the primary concepts that form the foundation of this work. First, Machine Learning techniques will be described, followed by an introduction to the concept of iML. Then, a comparative analysis of both approaches will be presented so that the reader can identify their key differentiating characteristics. Afterwards, we will focus on stating the Travelling Salesman Problem (TSP), which serves as the basis for the proposed museum route optimisation strategy. An overview of the most commonly used algorithms to tackle this problem will be provided. Amongst these, we will delve into the core of the algorithm employed in this project - Ant Colony Optimisation (ACO) - and thus conclude this chapter.

2.1 Machine Learning

Prior to delving into the concept of “Machine Learning”, it is worthwhile taking into consideration that this term is not new. In fact, it was coined in 1959 by Arthur Samuel who defined Machine Learning as the *“field of study that gives computers the ability to learn without being explicitly programmed”* [29].

Sixty years ago, Artificial Intelligence and Machine Learning were envisioned as revolutionary technologies that would have huge impact on everyday lives. There was significant enthusiasm and lots of funding and research was destined to this field. However, due to the lack of training data and the limitations in computing power [30], progress made was far from the high expectations set at that time. There have been ever since periods of heightened interest and further research, followed by phases of disillusionment and waning enthusiasm due to unmet expectations (AI winters). [31].

In the last decade, AI reflourished and has boomed due to great technological advances made, including the rise of Big Data and significant improvements in computing power at reduced costs [32]. Just to put the advances in perspective, in an interview in 2018 an IBM project manager that worked alongside NASA to take Apollo 11 to the moon mentioned that *“the software that controls what happens when you move your mouse on your PC–the mouse driver for Windows–takes more memory than all the NASA supercomputers put together had for Apollo”* [33].

Following a brief overview of the historical background and factors contributing to the ongoing AI surge, our attention now turns to the concept of Machine Learning. It is a discipline of Artificial Intelligence that is concerned with empowering systems to make predictions using input data, as well as revealing insights and patterns that would be challenging to identify through manual observation alone [34].

The aforementioned predictions and patterns occur as a result of the algorithm’s “learning” process, in which a mathematical model is tuned based on the data provided, thus generating outputs that align with the trained system. This is done without the need for human involvement during the learning process, earning them the name automatic Machine Learning (aML). The term *automatic* highlights their ability to function autonomously [35]. The learning mechanism depends on the characteristics of the available data and the intended application. The three traditional aML paradigms are: supervised, unsupervised and reinforcement learning. These will be described next, as well as interactive Machine Learning (iML), which breaks away from the conventional notion of Machine Learning.

2.1.1 Supervised Learning

Supervised learning is distinguished by the presence of labelled input data for training. This is, each data sample is manually assigned meaningful labels that provide context for the models. The name of this technique reminds of a 'supervisor' who provides instructions for establishing connections between labels and examples [36]. Put simply, in supervised learning, an input variable x is used to learn a mapping function f that generates an output $y = f(x)$, where the output corresponds to the provided label. Figure 2.1 showcases the operational process of supervised learning.

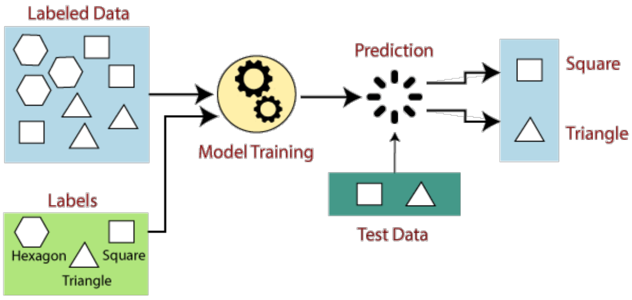


Figure 2.1: Illustration of the training framework of supervised Machine Learning[1]

It is natural to question the type of output variables, or labels, that can be produced in supervised learning, as it plays a pivotal role in determining the applicable models. When the output is discrete, the algorithm can classify input data into specific classes or groups (such as distinguishing between "cat" and "dog" in a set of images). This scenario is known as a classification problem. On the other hand, when the output is continuous, it is named a regression problem, like, for instance, predicting house prices. Figure 2.2 is presented as a means to provide a visual example of the difference between classification and regression.

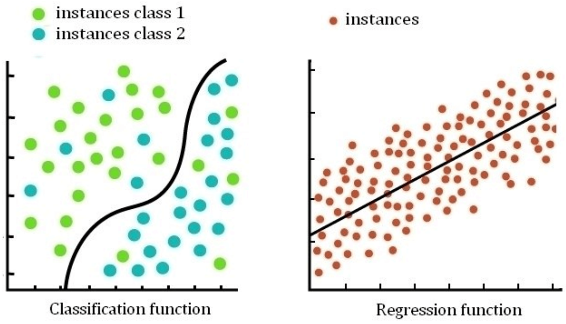


Figure 2.2: Classification vs regression[2]

[1] Source: https://www.javatpoint.com/supervised-machine-learning

[2] Source: Belkacem (2021) [37]

2.1.2 Unsupervised Learning

In many cases, obtaining labeled data can be challenging or laborious, or there may be an insufficient quantity of labeled data available. This requires the use of unsupervised learning techniques, where the algorithm endeavours to discover relationships and patterns among the input data that would otherwise be very difficult to extract manually. The procedure involves providing the model with unlabelled data and allowing it to analyse and interpret the information [38]. Similar to supervised learning, the selection of an appropriate algorithm depends on the characteristics data. Figure 2.3 portrays the outlined operational framework.

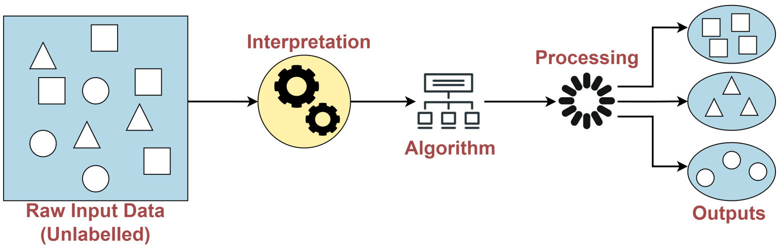


Figure 2.3: Illustration of the operational framework of unsupervised Machine Learning

Unlike supervised learning, unsupervised learning cannot make predictions due to the absence of labels, rendering it unsuitable for classification and regression problems. As mentioned before, the selection of an unsupervised learning algorithm also relies on the data provided. Consequently, two distinct problem types emerge in this scenario: clustering and association.

Clustering aims to identify a set of similarities and differences amongst data, facilitating the grouping of data points that exhibit greater similarity.

On the other hand, association aims to discover connections between the data, identifying elements that frequenctly occur together, or, in other words, establishing relationships between the occurrence of one particular element and the presence of another. For instance, people who purchase a pencil often tend to buy an eraser as well. Figure 2.4 depicts a visual example of the difference between clustering and association.

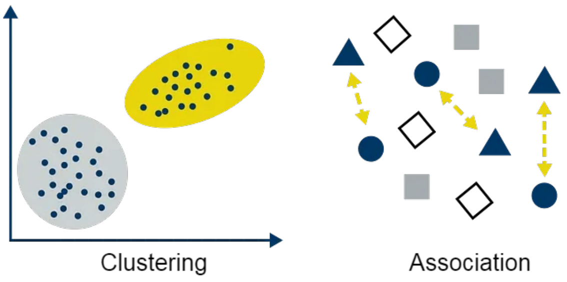


Figure 2.4: Difference between clustering and association problems in unsupervised Machine Learning.[3]

2.1.3 Reinforcement Learning

The third and last Machine Learning technique is known as Reinforcement Learning. Unlike supervised and unsupervised learning, it does not rely on an initial dataset. Instead, it acquires knowledge through trial and error to discern which actions result in the highest rewards [39].

Reinforcement learning consists of five primary components: i) agent, ii) environment, iii) action, iv) policy and v) the reward. The agent serves as the learner or decision-maker, actively engaging with the environment to optimize its actions. Through this interaction, the environment - the situation in which the agent is present - provides valuable feedback to the agent, evaluating the consequences of its chosen actions. Guided by a policy, which is a mapping function linking environmental states to actions, the agent strategically selects actions in pursuit of maximising rewards. These rewards, in turn, represent the numerical feedback given by the environment to signal the desirability of the agent's actions. The ultimate goal is to calculate the better policy that maximises rewards obtained. Figure 2.5 provides a visual representation of the outlined learning procedure.

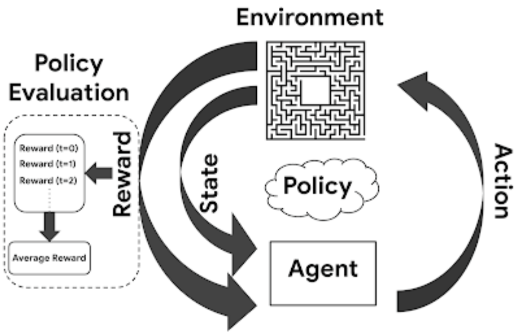


Figure 2.5: Agent-environment interaction in reinforcement learning.[4]

[3] Image adapted and edited from original source: https://medium.com

[4] Source: Google AI Research Blog

2.1.4 Interactive Machine Learning

In the previous sections an overview on aML was provided, where it was explained that it refers to algorithms that learn automatically without human intervention during the learning process. Furthermore, we explored the various types of machine learning techniques and which one to apply depending on the use-case. Also, it was mentioned that the current boom in AI can be traced back to two crucial factors: the rise of Big Data and significant advancements in computing power.

Nevertheless, this abundance of data and computing power also poses a drawback. These algorithms require substantial amounts of training data and consume considerable computational resources. Also, gaining access to a vast dataset is not always feasible, as it may prove laborious to obtain such data. Consequently, when there is a scarcity of training data, uncertainty regarding the available data, or an escalation in the complexity of the problem to be solved, implementing the aML approach becomes more challenging.

Furthermore, a dichotomy emerges between the progress of these algorithms and the increasing emphasis on green computing policies [40]. These policies strive to achieve a more efficient and sustainable use of resources in order to mitigate the environmental impact of computers.

Another drawback of aML is its “black-box” [41] nature, referring to the difficulty or complexity of understanding the internal workings of the algorithm. This lack of transparency in the algorithm’s decision-making process can present issues, particularly in fields such as medicine. Consequently, the inability to justify the reasoning behind decisions may foster distrust and give rise to legal and privacy concerns.

This lack of explainability and transparency conflicts with the General Data Protection Regulation (GDPR), which introduces regulations governing automated decision-making processes, including profiling [42]. The GDPR ensures the inclusion of a human element to review and challenge decisions made by an algorithm. Under the GDPR, individuals have the right to receive an explanation of how and why an automated process makes a series of decisions that significantly impact them. This right seeks to ensure transparency, accountability, and the capacity to question or seek remedies for potential instances of unfairness or discrimination.

Nevertheless, if we involve humans not only in the data preprocessing phase - as is done in aML but also in the actual algorithm’s learning process, we arrive at the concept known as interactive Machine Learning (iML). It is defined as *“algorithms that can interact with agents and can optimize their learning behavior through these interactions, where the agents* *can also be human"* [43]. In other words, by including a *"human in-the-learning-loop"*, we transition from a "black-box" system to a "glass-box" [44] (see Figure 2.6), granting greater transparency to the algorithm.

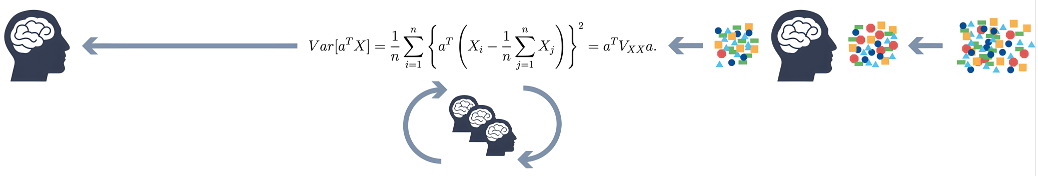

$Var[a^TX] = \frac{1}{n}\sum_{i=1}^{n}\left\{a^T\left(X_i - \frac{1}{n}\sum_{j=1}^{n}X_j\right)\right\}^2 = a^TV_{XX}a.$

Figure 2.6: iML human-in-the-loop (glass-box) approach.[5]

Moreover, iML offers an additional advantage worth highlighting. It can prove effective in scenarios where data is scarce or when contextual information is lacking, such as in NP-hard problems [45]. By involving humans, we can leverage their intuition and generalisation abilities. This approach strives to harness human expertise and capabilites to enhance the effectiveness of automated methods. Thus, iML aims to attain results that are on par with or surpass those achieved by aML, while optimising the use of computational resources.

2.2 Travelling Salesman Problem

The Travelling Salesman Problem (TSP) was initially proposed by mathematicians Hamilton and Kirkman in the 19th century [46]. The main aim of this problem is to determine the shortest route, in terms of distance, that enables a traveller to visit a given set of cities only once before returning to the starting city.

Since its origin, TSP has gained significant popularity among optimisation problems, primarily due to two key factors. Firstly, it presents a problem with a relatively straightforward formulation, yet it poses considerable computational challenges [47]. Secondly, the TSP serves as a foundation for modelling numerous real-world applications in various domains, including transport and logistics [48], electronics [49], and medicine [50].

This section will present the mathematical formulation of the Travelling Salesman Problem, providing an overview of its underlying principles. Subsequently, various algorithms available for addressing the TSP will be reviewed.

2.2.1 Problem Formulation

The formal mathematical definition of the TSP based on Graph Theory is the following:

[5] Source: Holzinger (2016) [43]

Given a complete weighted graph $G = (V, E, w)$, where $V = \{v_1, v_2, ..., v_n\}$ is the set of vertices and $E = \{(v_i, v_j)|v_i, v_j \in V, v_i \neq v_j\}$ is the set of edges, with each edge associated with a weight w, find a permutation π of the vertices such that the sum of the edge weights along the Hamiltonian cycle[6] formed by π is minimized.

Initially, a seemingly straightforward solution emerges: one could generate all of the possible different paths and calculate their respective distances to identify the path with the minimum distance. However, this approach quickly becomes computationally intractable. For example, with 5 cities, there are 60 potential paths to consider, while with 10 cities, the number increases to 181,440. It becomes evident that the number of potential paths grows exponentially, and even with just 61 cities, the possibilities surpass the estimated number of atoms in the universe [51].

Consequently, TSP is categorised as and proven to be NP-complete [52], indicating that it is not possible to find the optimal solution for large instances using exact methods within a polynomial time frame. This implies that the computational complexity of the problem increases exponentially as the problem size grows. Therefore, optimisation algorithms are employed to find suboptimal but reasonably satisfactory solutions within a feasible time-frame.

2.2.2 Algorithms for Solving the TSP

To solve the Traveling Salesman Problem (TSP), algorithms can be categorised into two main groups: exact and approximation algorithms. Exact algorithms ensure the attainment of the optimal solution; however, they become computationally intractable for a larger number of cities. On the other hand, approximation algorithms - known as heuristics - focus on handling the scalability of the problem at the expense of not guaranteeing the best solution. An overview of both families of algorithms will be offered next.

2.2.2.1 Exact Algorithms

As previously mentioned, a naïve approach to exact resolution would involve employing brute force, which entails calculating all possible paths and selecting the one with the lowest cost. However, this method exhibits a complexity of $O(n!)$. There are other exact resolution techniques that enhance computation time, but they still remain exponential.

One notable algorithm is the Bellman-Held-Karp algorithm [53], which exploits the

[6] A Hamiltonian cycle is a cycle that visits each vertex exactly once and returns to the starting vertex.

following property of the TSP problem: each subpath of a minimum distance route is itself a minimum distance path. By breaking down the problem into subtasks, the algorithm uses the computed optimal paths of smaller subroutes to determine optimal paths for larger subroutes. This yields a complexity of $O(n^2 2^n)$, which improves factorial time complexity, although it remains inadequate for a significant number of cities.

Another algorithm for exact TSP resolution is Concorde TSP Solver [54], based on a variation of the branch-and-cut method [55]. This algorithm established the record for the longest optimal tour, comprising 85,900 objectives and requiring an approximate resolution time of 136 CPU years [56]. Currently, it stands as the most effective algorithm for exact resolution. Nevertheless, it requires substantial computational capacity and time to address larger TSP instances. Hence, as previously indicated, it is generally preferable to employ approximation resolution methods, which yield suboptimal yet acceptable solutions in practice.

For instance, these methods can yield an approximate solution to an instance of nearly 2 million nodes within 30 minutes [57], surpassing the Held-Karp Lower Bound (HKLB) by just about 15%. The HKLB serves as a quick and cheap means of obtaining a highly accurate estimation of the optimal tour length, which holds important practical value because it allows to assess the quality of the suboptimal solutions found by approximation approaches.

With respect to these algorithms, we will now explore some of the most commonly employed non-optimal approaches (heuristics) for solving the TSP.

2.2.2.2 TSP Heuristics

Nearest Neighbour algorithm: this approach follows the principle of always selecting the closest city as the next destination. With a complexity of $O(n^2)$, this simple heuristic generally keeps its best tour within 25% [58] of the HKLB. It starts with a random city selected as initial location. Then, the next city to be chosen is the one with the minimum cost (i.e. nearest) of all the neighbours. This process is repeated until all cities have been visited, so that the algorithm must return to the first city to complete the tour.

Greedy heuristic: it is similar to the Nearest Neighbour algorithm in the sense that it also selects as next destination the vertex whose edge has the least weight amongst the adjacent vertices. However, the starting city is not randomly chosen and instead, all N edges are sorted prior to selecting them, being the first selected edge the one with least weight. The algorithm stops when a tour of length N has been constructed. Greedy heuristic usually provides slightly better results than the simpler Nearest Neighbor approach, keeping solutions within 15-20% [58] of the HKLB. The initial sorting contributes towards the improvement of results obtained, but also adds to the time complexity, so this algorithm is $O(n^2 log_2 n)$.

Tabu Search: this algorithm is based on a neighbourhood-search principle, and in order to avoid getting stuck in local minima, it implements memory structures that hold data about previously visited solutions that are not good enough and thus should be disregarded (i.e. tabu list). The idea is to start by generating an initial tour and then applying small modification to the current tour by swapping nodes and evaluate the quality of the solution. This is performed iteratively with the aid of the aforementioned memory structures that gradually contribute to intensify the search and explore the solution space. The heuristic terminates when the stop criteria is met, which is typically after a given number of tour modifications have been performed. While tabu search can produce better results than simpler heuristics like the nearest-neighbour approach or greedy algorithm, it does have a drawback of having a time complexity of $O(n^3)$ [58].

Genetic algorithms: these meta-heuristics draw inspiration from the natural selection process, where organisms that are better adapted to the environment have a higher likelihood of survival and reproduction [59]. In Genetic algorithms, the initial set of candidate solutions - solutions are referred to as individuals -, is randomly chosen. These are evolved using biologically inspired operators such as mutation, crossover and selection. The fittest individuals, determined by a fitness function, are selected amongst the population and their genome undergoes modifications through crossover (i.e. recombination) and random mutation, yielding a new set of candidate solutions. This process is repeated iteratively until the stop criterion is satisfied, typically a target fitness level or a maximum number of iterations. For TSP, the evaluation of fitness can be compared against the weight of the constructed tours [60].

Simulated Annealing: this heuristic is inspired by the annealing process in metallurgy, where a material is heated and slowly cooled to achieve a modification in its micro-structure and metallic properties [61]. In the context of simulated annealing, the temperature parameter regulates the level of randomness employed when altering the path, with the objective of minimising this randomness. At higher temperatures, a greater number of random changes occur (i.e. node exchanges), thereby increasing the probability of accepting longer distance tours. This strategic approach facilitates a broader exploration of the solution space, effectively preventing early entrapment in local minima. Conversely, as the temperature decreases step by step, the amount of random changes is reduced and there is less tolerance for worse tours, until a stop criterion is met.

The previously described algorithms provide a concise illustration of various optimisation techniques used to address the NP-complete TSP. There are other methods, including Ant Colony Optimisation (ACO). Since the iML algorithm implemented for the museum tours in this project is founded on ACO, we will present a comprehensive explanation of this algorithm in the upcoming section.

2.3 Ant Colony Optimisation

This section delves into the exploration of a family of metaheuristics that draw inspiration from the behaviour exhibited by ants when they seek out the shortest paths from their nest to food sources. Firstly, a succint analysis of ant social foraging patterns will be provided to understand the foundation of ACO. Subsequently, several common implementations and variations of ACO will be provided.

2.3.1 Ant Social Foraging

Foraging refers to the actions performed by animals to acquire sustenance, such as exploring, selecting and manipulating food sources [62]. This concept also applies to ants, which are social insects living in colonies. Through their cooperative behaviour, ants can accomplish tasks that would otherwise be very difficult - if not impossible - for an individual ant. One of these collective tasks is foraging.

Some species of ants release a substance known as *pheromone* to mark the trail between their nest and food sources. Pheromones are olfactory signals that allow other ants to follow the trail [63]. In the absence of pheromone trails, ants tend to move in a random manner. However, when ants detect pheromones deposited by other ants, they are more likely to follow the path with pheromone markings, as supported by research [64, 65, 66].

When ants are searching for food and encounter multiple paths, their decision-making process is influenced by the amount of pheromone present in each path. Therefore, the stronger the trail, the higher the likelihood that ants will follow it. As ants travel along this path, they also deposit pheromones, further increasing the chances of other ants choosing the same route.

The conducted experiments have also demonstrated that ants not only eventually converge on a common path from the nest to the food source but also tend to choose the shortest or one of the shortest paths. Illustrative examples are the double bridge experiments described in [64, 65]. Essentially, there are bifurcations that result in two possible paths ants can take to reach the food source, one being shorter than the other. Initially, with no pheromone trail present, when an ant reaches the intersection, it lacks any criteria to preferentially select one path over the other, so it randomly chooses one of them.

As more ants explore the environment, a trend emerges where ants increasingly opt to follow the shorter path. The reason behind this pattern lies in the fact that ants taking the shorter path can reach the food source more quickly and hence return to the nest sooner compared to those that choose the longer path. Consequently, the shorter path becomes enriched with a higher concentration of pheromones as ants continuously deposit them during their expeditions.

Also, over time, the pheromones on the longer path start to evaporate gradually. With respect to the shorter path, since a larger number of ants are choosing it, the impact of pheromone evaporation is diminished there. As a result, the ant colony's chosen path progressively converges towards the shorter one. This behaviour is depicted in Figure 2.7, which illustrates the outcomes of an experiment conducted in [64]. After 8 minutes, a significant majority of ants followed the shortest route from the nest to the foraging area.

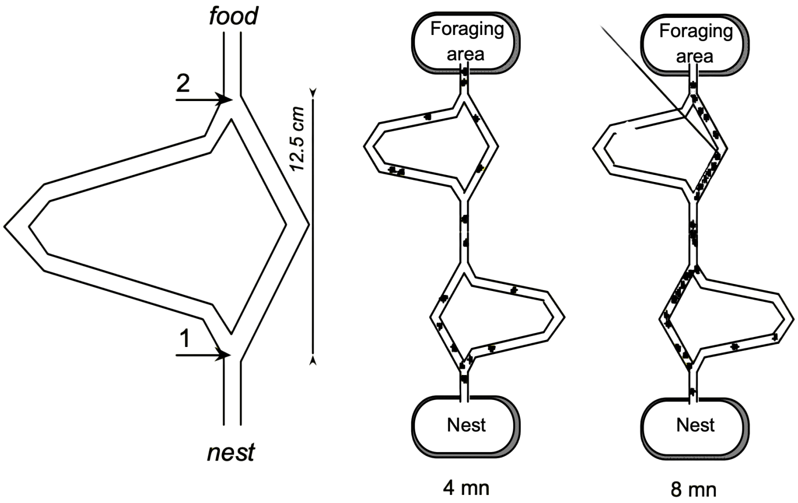


Figure 2.7: Illustration of ant colony convergence to a common and shortest path from their nest to the food source[7]

2.3.2 Ant Colony Optimisation (ACO)

The ACO [67] metaheuristic is inspired by the aforementioned behavior of real ants and is employed to create artificial ant colonies that explore a graph representing a given problem.

[7] Source: Bonabeau et al. (1999) [64]

In the case of TSP, these ant agents iteratively construct Hamiltonian cycles in a complete graph, aiming to minimise the cumulative weights of the chosen edges. The fundamental principles employed to achieve this objective are elaborated below.

Each artificial ant explores the graph and has a probability of selecting a node that is proportional to the weight of the corresponding edge. In this context, pheromones serve as a representation of the edge weights. Initially, all edges are assigned an equal concentration of pheromones to encourage exploration. Every ant undergoes a set number of iterations, constructing a given number of tours. Over time, the tours are influenced by the amount of pheromone deposited by other ants. Consequently, an edge with a higher pheromone concentration is favoured probabilistically.

Furthermore, pheromone evaporation occurs, simulating the environmental impact on these olfactory signals within a natural context as outlined in section 2.3.1). The rationale behind implementing this characteristic is that if an edge is consistently excluded from potential favourable tours, the reduction in pheromone value along this arc ensures that the algorithm does not have memory of bad choices over time [26].

Ultimately, each ant reports the best tour it has discovered, and from this subset, the one with the shortest length is selected as the best solution obtained.

The first ACO algorithm, known as Ant System (AS), was proposed in 1992 by Dorigo [67]. An overview of this algorithm will be provided next, as well as subsequent extensions that were developed to enhance previous approaches, namely Ant Colony System (ACS) [68]; Ranked-Based Ant System (AS_{rank}) [69] and MMAS [26].

2.3.2.1 Ant System

When applying AS to the TSP, each edge (a,b) of the graph has a given fixed cost metric d_{ab}, as well as a variable pheromone trail ϕ_{ab} that will be updated by the ants. The first step is to initialise the pheromone trails for all edges with an equal concentration. Afterwards, there are two main processes that take place: ant tour construction and pheromone update. The way these steps are carried out are the key differentiating factors between different ACO algorithms.

Once the initial pheromone trails have been set, ant agents are randomly placed in cities (i.e. nodes). In each step of the tour construction, an ant j in a node a moves to a neighbour unvisited node b according to the probability rule defined in Equation 2.1:

$$p_{ab}^{j} = \frac{[\phi_{ab}]^{\lambda}\,[\omega_{ab}]^{\mu}}{\sum_{c\in N_a^j}[\phi_{ac}]^{\lambda}\,[\omega_{ac}]^{\mu}}, \text{ if b} \in N_a^j \tag{2.1}$$

Taking into consideration that

- $\omega = 1/d$ the inverse of the cost metric (e.g distance).
- λ is a constant that regulates the probabilistic influence of the pheromone trail.
- μ is a constant that regulates the probabilistic influence of the heuristic information (i.e. cost metrics)
- N_a^j represents the neighbour nodes of a that ant j has not yet visited.

With respect to parameter λ, an excessively high value would result in the probability rule solely considering pheromone trails. Consequently, the search process could quickly reach a state of stagnation, favouring paths with higher pheromone concentration to a much greater extent than others. Conversely, if μ is set too high, the algorithm would exhibit a substantial probabilistic bias towards selecting paths with lesser cost metrics. This would cause again entrapment in local minima (premature convergence), since exploration would be limited to a small set of solution nodes. Thus, it is crucial to strike a suitable balance between these parameters. This balance would allow for more paths to be explored while enabling the algorithm to gradually converge towards a potentially superior solution by expanding the search.

Moreover, an ant j possesses knowledge of which neighbouring nodes of a (N_a^j) remain unexplored. These data are stored as an array of visited nodes that form the tour being constructed. This memory feature is important as it allows for backtracking, thereby facilitating the appropriate update of pheromone trails ϕ after each iteration. In other words, the adjustment of pheromone levels takes place after all ants have constructed a tour.

Regarding the process of updating the pheromone trails, the first action is to trigger pheromone evaporation. Essentially, this entails multiplying all pheromone levels by a constant decay parameter. Following the global update of pheromone levels, each ant will then increment the concentration of pheromones on the edges belonging to their constructed tour solution. These two modifications are expressed by the pheromone update rule in Equation 2.2:

$$\phi_{ab}\prime = (1-\rho)\cdot\phi_{ab} + \sum_{j=1}^{m}\Delta\phi_{ab}^{j} \tag{2.2}$$

where

- $(1-\rho)$ is the pheromone evaporation constant and $0 \leq \rho < 1$.
- $\Delta\phi_{ab}^{j} = \frac{1}{L_j}$ if edge (a,b) is in the solution constructed by ant j; otherwise $\Delta\phi_{ab}^{j} = 0$. It represents the amount of pheromone deposited by ant j for each edge it has travelled through. L_j is the length of the tour constructed by ant j.
- m is the number of ants participating in the generation of tours.

This equation underscores the aforementioned principle of providing more reward to arcs belonging to shorter tours.

2.3.2.2 Ant Colony System

Ant Colony System (ACS) [68] introduces three key differentiating aspects from AS: i) a different probability rule to select nodes that aims to find a compromise between the exploration of new arcs and the exploitation of information available at that moment; ii) a modified scheme to perform global pheromone update and iii) the addition of a local pheromone update rule that makes ants deposit pheromone after each transitioning to a new node.

With respect to the probability rule, it introduces a uniformly distributed random variable $q \sim U(0,1)$. Depending on whether it is greater or lower than a parameter q_0 (adjustable) $\in [0,1]$, the function defining the probability rule varies. The chance for an ant j to transition from node a to node b is determined by the probability distribution in Equation 2.3.

$$p_{ab}^{j} = \begin{cases} \arg\max_{u \in N_j^a} \left\{ [\phi(a,c)] \cdot [\omega(a,c)]^{\mu} \right\}, & \text{if } q \leq q_0 \\ B, & \text{otherwise} \end{cases} \tag{2.3}$$

where B is a random variable that follows the probability rule defined in Equation 2.1, which is the state transition rule of AS.

Regarding the global pheromone update, unlike in AS, only the global best ant is allowed to deposit pheromone after all ants have constructed their tours. This elitist update (refer to Equation 2.4) approach together with the described state transition rule aim to focus the search around the neighbour nodes of the iteration-best tour.

$$\phi_{ab}\prime = (1-\rho) \cdot \phi_{ab} + \Delta\phi_{ab}^{globalbest} \tag{2.4}$$

where the amount of pheromone deposited by the global best ant is the inverse of the best tour's length.

Finally, in relation to the local pheromone update procedure, each time an ant travels through an arc (a,b), it deposits pheromone in the edge according to Equation 2.5. The desired effect of this rule is to reduce the amount of pheromone on visited trails and therefore favour better exploration of other nodes, avoiding to be entrapped in local minima.

$$\phi_{ab}\prime = (1-\epsilon) \cdot \phi_{ab} + \epsilon \cdot \Delta\phi_{ab} \tag{2.5}$$

The local update rule is very similar to the global update rule, but the difference is that evidently the information added comes from the local ant. $(1-\epsilon)$ is another pheromone evaporation value and $\Delta\phi_{ab}$ is a parameter that can be tweaked, although there are recommended values that come from empirical experience [68].

2.3.2.3 Rank-Based Ant System

AS$_{\text{rank}}$ [69] is similar to AS because it is based on the same state transition rule and it also performs an offline pheromone update (i.e. global update after construction). The differentiating feature is that it proposes a modification to the global pheromone update procedure of AS. Specifically, this change involves sorting the constructed tours based on their length. Afterwards, only the ξ ants that generated shorter routes are allowed to lay pheromone ϕ. By implementing this sorting mechanism, each of these ants is assigned a rank value r, where lower means better. There is a particularity for edges which are part of the best ant's tour, because these will receive and additional amount of pheromone, proportional to ξ . This global pheromone update mechanism is described by Equation 2.6, for an ant j in an arc (a, b).

$$\phi_{ab}\prime = (1-\rho) \cdot \phi_{ab} + \xi \cdot \Delta\phi_{ab}^{globalbest} + \sum_{j=1}^{\xi-1}(\xi - r)\Delta\phi_{ab}^{j} \tag{2.6}$$

where:

- $\Delta\phi_{ab}^{g} lobalbest = \frac{1}{L_{globalbest}}$ if edge (a, b) is contained in the global-best tour constructed.

- $\Delta\phi_{ab}^{j} = \frac{1}{L_j}$ if edge (a, b) is part of the tour constructed by tour j.

The primary objective of the proposed global pheromone update procedure is is to achieve a balance between exploitation and exploration. Although the search becomes more directed because more importance is given to better tours, utilising multiple good ants contributes towards widening the scope of the search, reducing the chance of early stagnation.

2.3.2.4 MAX-MIN Ant System

MAX-MIN Ant System (MMAS) is one of the best performing [26] variants of AS. The key to this better performance is that MMAS enhances the exploitations of the best solutions while also introducing a way to prevent early entrapment in local minima. The followed strategy consists of three differentiating features from AS:

1. *Global pheromone update*: Only a single ant is permitted to deposit pheromones, which can be either the one generating the best tour in the current iteration or the ant that has produced the best tour overall. Empirical evidence demonstrates that using a mixed strategy yields superior outcomes. Initially, iteration-best ants are preferred and gradually the global-best ant is more frequently selected to update pheromone trails [26]. This is because at the start, the best solutions in each iteration are expected to differ significantly, resulting in reinforcement of diverse paths and widening the scope of the search. Over time, selecting the global-best ant allows to gradually produce a more directed search.

   Selecting the ant with the best overall tour to initially update pheromones can prematurely narrow down the search. This is due to the considerable concentration of pheromones on edges associated with the global best tour, which often leads to early stagnation and thus the attainment of bad quality solutions. In any case, the pheromone update rule for MMAS is expressed in Equation 2.7:

$$\phi_{ab\prime} = (1-\rho)\cdot\phi_{ab} + \Delta\phi_{ab}^{best} \tag{2.7}$$

   where $\Delta\phi_{ab}^{best} = \frac{1}{L_{best}}$ if edge (a,b) is in the solution constructed by the (global or iteration) best ant; otherwise $\Delta\phi_{ab}^{best} = 0$. So, if edge (a,b) is part of the best solution it gets reinforced with pheromone; if not, the pheromone trail will be weakened due to evaporation $(1-\rho)$.

2. *Pheromone upper and lower bounds (Max-Min)*: the meta-heuristic imposes limits to the range of possible values pheromone levels in each edge can lie in: $\phi_{min} \leq \phi \leq \phi_{max}$. These bounds contribute towards avoiding situations where there are edges with substantially more pheromone than others, which could lead again to early stagnation of the search. The expressions for the values (refer to [26] for proof) of these upper and lower bounds are given in 2.8:

$$\phi_{max} = \frac{1}{L_{best}}, \qquad \phi_{min} = \frac{\phi_{max}(1 - \sqrt[n]{p_{best}})}{(0.5n - 1)\sqrt[n]{p_{best}}} \tag{2.8}$$

   Taking into consideration that n refers to the total number of cities present in the problem. p_{best} describes the probability that upon convergence, an ant is able to select at each city the edge with the most concentration of pheromone ϕ_{max}. This parameter can be tweaked to improve performance, and experimental evidence [26] suggests that $p_{best} \sim 0.05$ is a good value to use.

3. *Pheromone trail initialisation*: in order to increase the exploration of solutions during the first iterations, pheromone trails are initialised to ϕ_{max}. This means that at the start the relative difference between edges will not be as great as if the edges were initialised with a small value of pheromone.

There is an additional optional feature introduced by MMAS, useful in high iteration instances, which consists in smoothing pheromone trails if the meta-heuristic is close to convergence (indicated by the average branching factor [70]). The objective is to allow arcs with low chance of being selected to be probabilistically favourable again and hence achieve a more efficient exploration of the search space. Equation 2.9 describes this mechanism, where δ $(0 < \delta < 1)$ is the smoothing factor.

$$\phi_{ab}\prime = \phi_{ab} + \delta \cdot (\phi_{max} - \phi_{ab}) \tag{2.9}$$

In order to provide a more complete overview of the workings of MMAS, Listing 2.1 describes in pseudo-code the logic for the implementation of this meta-heuristic. The probability (state transition) rule is the same as the one defined in AS (Equation 2.1, and the global pheromone update rule has been defined in Equation 2.7.

Listing 2.1: MMAS algorithm for TSP

```
Initialise pheromone trails to $\phi_{max}$
Do:
    Randomly place ants in starting nodes
    For every ant until all ants have generated a tour do:
        Chose next node according to the probability rule
    Apply global pheromone update
    Ensure phermone trails are bounded:
        If  $\phi > \phi_{max}$ → set $\phi = \phi_{max}$
        If  $\phi < \phi_{min}$ → set $\phi = \phi_{min}$
        Else do nothing
    [Optional] Apply pheromone trail smoothing if close-convergence/stagnation criteria met.
While stop condition not met
```

CHAPTER 3

State of the Art

This chapter will conduct a review of existing research that also explores the central concepts of the present work. First, real-world applications of the Travelling Salesman Problem will be analysed. Second, an overview of applications of Ant Colony Optimisation will be offered. Subsequently, we will present a succinct analysis of areas of research that have also delved into employing interactive machine learning techniques. Finally, the review will shift towards works that have employed optimisation algorithms to devise effective itineraries for museum visits.

3.1 Applications of the Travelling Salesman Problem

Extensive research has been conducted on the practical implementation of the TSP in various real-world scenarios, and it finds wide-ranging applications in different industries. Examples are logistics and transportation; manufacturing and production planning; network design and optimisation; and health informatics. A concise review of some TSP applications will be provided next.

In the field of logistics and transportation a generalisation of TSP is introduced, the Vehicle Routing Problem (VRP) [71]. This problem is typically defined within the context of delivering goods located at a central depot to customers who have placed orders for them, while trying to optimise a set of criteria (time windows, total distance travelled, fuel consumptions, carbon dioxide emmisions, etc). Figure 3.1 offers a graphical representation of the VRP.

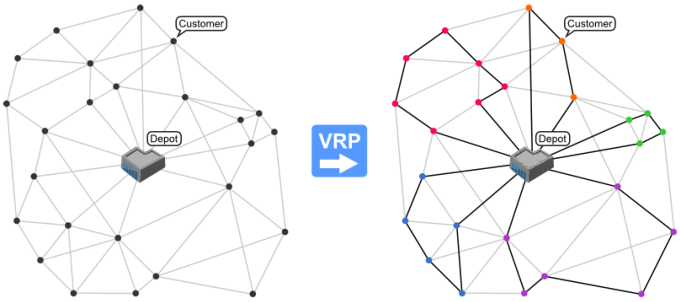


Figure 3.1: Illustration of the Vehicle Routing Problem (VRP).[1]

Researchers are actively exploring methods to minimize total travel distance while adhering to time constraints, particularly in scenarios where timely delivery of goods is critical. For instance, in the case of perishable food delivery [73, 74, 75], it is essential to ensure that nourishments remain fresh, requiring prompt delivery to prevent deterioration. Another application of VRP is seen in the delivery of goods purchased through e-commerce platforms [76, 77, 78]. Minimising travel distance is crucial for trying to ensure on-time delivery, which greatly contributes towards increasing customer satisfaction [79].

Additionally, there is a growing interest on sustainable mobility due to increasing concerns about climate change. Consequently, numerous research avenues are dedicated to addressing the VRP from the standpoint of reducing greenhouse gas emissions and the overall environmental impact of transportation [80, 81, 82, 83]. Introducing environmentally friendly restrictions to the VRP categorises the problem as a Green Vehicle Routing

[1] Source: Ochelska-Mierzejewska et al. (2021) [72]

Problem [84].

In the context of manufacturing and production industries, the application of TSP can be exemplified through the batch production of items like printed circuit boards (PCBs) [85, 86]. When drilling holes of various sizes in a circuit board, the need for drill head changes arises, resulting in time-consuming processes. This situation can be addressed by treating it as a collection of TSP instances, with each instance corresponding to a specific set of hole diameters. The primary goal is to minimize the travel time of the drilling machine by consecutively drilling holes of the same diameter before transitioning to the next diameter. Ultimately, this approach contributes to maximizing production output and reducing unit production costs.

TSP has also found its practical application in the design and planning of telecommunication networks. For example, [87] focuses on EONs [88], where unlike WDM, each signal is assigned a variable number of frequency slots according to their bandwidth requirements and demands at a given time (refer to Figure 3.2 for an illustration). Also, the allocation of guard-bands between signals is flexible to accommodate traffic variations [89]. An optimisation problem stemming from the TSP is proposed here, where each node is a signal and the edges are the guard-band requirements between adjacent lightpaths. The main goal is to minimise the total amount of spectrum used to allocate a given set of signals while satisfying a set of constraints such as the guard-band requirements for each pair of adjacent lightpaths.

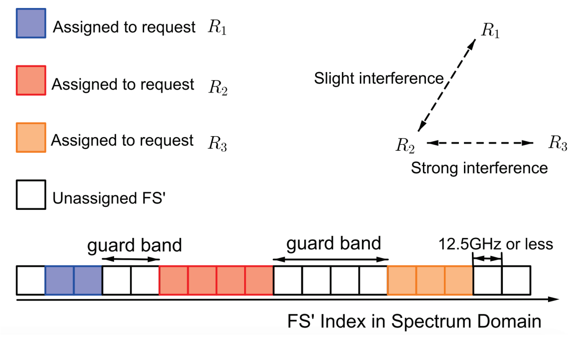

Assigned to request R_1
Assigned to request R_2
Assigned to request R_3
Unassigned FS'
R_1
Slight interference
R_2
R_3
Strong interference
guard band
guard band
12.5GHz or less
FS' Index in Spectrum Domain

Figure 3.2: Flexible resource allocation in EONs [2]

Another variation of TSP in this domain is the Black and White Travelling Salesman Problem (BWTSP). Here, nodes are categorised as either "black" or "white". The goal is to construct a minimum cost Hamiltonian cycle taking into account that the number of white nodes between two consecutive black vertices is limited. Additionally, the distance between black nodes must be less than a specified value. This can be applied to SONET

[2] Source: Wu et al. (2017) [87]

ring networks, where self-healing rings [90] require two consecutive ring offices (i.e. black nodes) to be separated by a maximum of H hubs (i.e. white nodes) within a distance less than D [91].

Finally, one more practical use of the TSP is found in the field of DNA sequencing. For instance, in the study in [92], oligosaccharides (oligos) are regarded as nodes, and edges are inversely proportional to the overlapping factor between each pair of oligos. Therefore, the aim is to construct from a set of oligos and a given DNA sequence, a combination of oligos that yields the greatest overlapping score. This will in turn contribute to obtaining the deepest sequencing of that DNA region.

3.2 Applications of Ant Colony Optimisation Algorithms

Ant Colony Optimisation (ACO) metaheuristics are used to provide approximate solutions to optimisation problems. They are inspired in the behaviour of ants, which are represented as agents that iteratively explore a graph aiming to find the shortest paths. They adapt pheromone trails based on the fitness of the edges they traverse.

ACO algorithms are primarily used to tackle NP-hard optimisation problems (such as the TSP, as mentioned in Section 3.1), and these manifest in different fields. Consequently, ACO has a large number of applications for various types of NP-hard problems, such as routing, scheduling or bioinformatics. An overview will be presented regarding works that focus on the aforementioned NP-hard problems types. This review draws inspiration from [93], which presents a compilation of ACO applications.

It is important to highlight that there are also some works that have applied ACO heuristics within the context of museums, although there is little research on the topic. Therefore, the review of these works will be carried out in the next section, which will elaborate related works that have applied optimisation algorithms to calculate efficient museum routes.

In addition to the traditional TSP, ACO algorithms have been applied to various routing problems that involve variations of the TSP, such as the VRP mentioned in the previous section. In the context of tackling the VRP, a pioneering study [94] employed an ACO algorithm, specifically AS, to produce shorter routes for vehicles.

Another line of research, exemplified in [95], focuses on minimising additional constraints in the VRP. In their work, they exploit ACS to minimise both the number of vehicles and the total travel time. To achieve this, two ant colonies are utilised, with each colony dedicated to one objective. The colonies exchange feedback with each other through pheromone trails, facilitating mutual reinforcement.

The last example that will be provided in the realm of routing problems is about the application of ACO algorithms for solving the Team Orienteering Problem (TOP) [96]. It is another variant of VRP, where a team comprising M vehicles is tasked with visiting multiple locations, each associated with a cost metric (score). The fundamental objective is to determine M routes that commence from an initial point and terminate at an endpoint, while maximising the cumulative score achieved. Figure 3.3 depicts the TOP in a practical use-case with a delivery fleet. The work in [97] proposes the first ACO approach towards addressing the TOP, referred to as ACO-TOP. It introduces several modifications in order to make it suitable for tackling this problem. Nevertheless, certain elements of ACO-TOP remain consistent with other ACO algorithms. For instance, the pheromone updating rule employed is the same as the one proposed by MMAS.

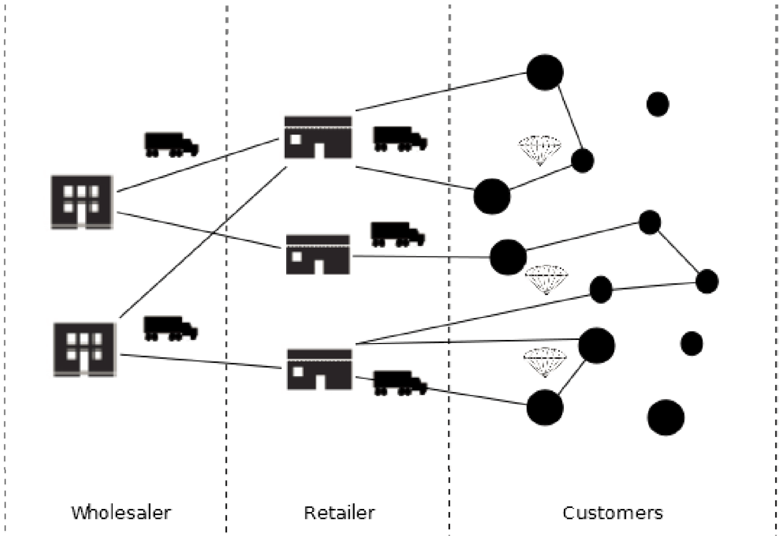


Figure 3.3: Team Orienteering Problem (TOP) illustrated with a use-case of a delivery fleet.[3]

The focus now shifts towards exploring the utilisation of ACO algorithms in addressing scheduling problems, where the core objective is to allocate tasks to machines in an efficient manner. Successful task scheduling requires consideration of various factors including machine power, setup times, task deadlines, or the order in which jobs must be processed.

A typical scheduling problem is the Single Machine Total Weighted Tardiness (SMTWT) problem. There is only one machine available to process a set of tasks, each having a given processing time and deadline. Notably, the problem handles sequence-dependent [99] setup times, meaning that it depends on both the next job to be processed as well as

[3] Source: Reyes-Rubiano et al. (2018) [98]

the previous task. Consequently, the goal is to determine an optimal task sequence that minimises the overall processing delays (i.e. tardiness). This holds significant relevance in the manufacturing industry, as it is crucial to reduce as much as possible the costs resulting from a failure to complete a given set of tasks in time.

Various research endeavours have applied ACO algorithms to tackle the SMTWT problem, as evident in [100, 101]. The work in [100] introduces a modified version of ACS that incorporates local search and employs an heterogeneous colony of ants. In this approach, all ants engage in the loal search steps, with half of them executing the steps in reverse order, contributing to improve the achieved results. Similarly, the algorithm proposed in [101] also builds upon ACS and applies local search mechanisms. Additionally, it adds several modifications that are key to its good performance. Notably, it introduces a new pheromone model that draws inspiration from the trail limits concept in MMAS, and a fresh global pheromone update mechanism.

Shifting focus to the field of bioinformatics, it is an area that has given rise to numerous NP-hard optimisation problems. Particularly noteworthy is the protein folding [102] problem. Polypeptide chains naturally arrange into structures (i.e they are folded into shapes). It is crucial for them to fold correctly to ensure proper functionality. Otherwise, misfolded or unfolded proteins can contribute to various human diseases, including neurodegenerative conditions like Alzheimer's disease [102]. The configurations with the lowest energy tend to be the most stable and likely represent the correct folded structures. However, the sheer number of potential configurations presents a combinatorial explosion problem. Therefore, being able to predict how a protein folds poses a challenge. In this sense, ACO algorithms can be harnessed to efficiently find minimal-energy conformations, as reflected by bodies of research such as [103, 104, 105]. Figure 3.4 provides an illustrative example of the conformation of a structure, which can help into putting into perspective the sheer number of possible conformations and the utility of heuristics to predict the most likely configuration.

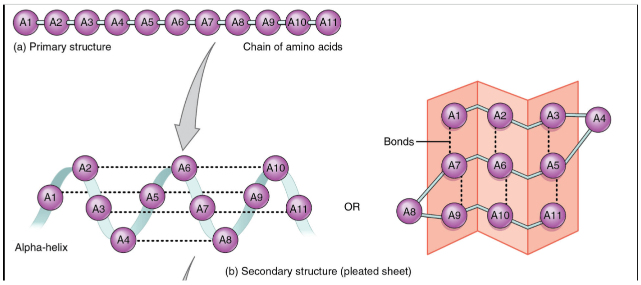


Figure 3.4: Protein folding example into a two-dimensional configuration.[4]

[4] Source: Wikipedia https://en.wikipedia.org/wiki/Protein_folding

Another prominent challenge in the realm of bioinformatics where ACO algorithms have proven effective is DNA sequencing. As discussed in the previous section, the problem of DNA sequencing can be reduced to solving a TSP instance. Hence, employing ACO algorithms here is well-suited. For instance, [106] proposes a state-of-the-art multi-level framework based on MMAS to address DNA sequencing by hybridisation.

3.3 Review of Interactive Machine Learning Approaches

Numerous iML systems have been implemented across several domains, aiming to overcome the limitations of conventional ML algorithms. Noteworthy fields of application of iML approaches encompass health, finance, games, robotics and military or cybersecurity. Some works in each of these fields will be reviewed next. The following review draws inspiration from [107], which extensively examines numerous bodies of research employing iML techniques. However, their analysis focuses on evaluating works using a merit-oriented taxonomy, rather than classifying them according to the domain of application, which is how the review will be structured in this section.

Before delving into the analysis of the aforementioned works, it is worth mentioning that, to the best of our knowledge, no prior lines of research have explored the practical applications of iML in the cultural heritage domain.

Nevertheless, certain works have also addressed the generation of optimised museum routes, albeit without employing iML techniques. These endeavours will be explored in Section 3.4.

With respect to iML applications in the field of health, [108] develops a system capable of recognising entities and relations of biomedical terms. The particularity of this approach is that experts in the medical field (concretely doctors) iteratively annotated medical texts, incrementally processing the dataset of medical documents, providing quality annotations to the ML model during its training. The algorithm provided suggestions based on the annotations given, and these are iteratively refined by the doctors, making the model progressively produce more and better medical annotations and relations. This interactive and iterative doctor-in-the-loop approach proves that it speeds up the generation of good quality annotations by the ML algorithm.

Another relevant application is presented in [109], which demonstrates the advantages of human-AI collaboration within the medical field to generate more reliable models that can assist doctors in making more accurate diagnoses. The work introduces an already trained medical image deep neural network as a starting point, which retrieves similar images to the one doctors have provided. The idea is that the images returned help doctors in identifying any potential pathologies based on the input image. However, a significant challenge arises in ensuring that the ML model generates medically relevant images that align with a doctor's perception of similarity, as the concept of similarity from an algorithmic perspective may differ considerably from a medical point of view. To address this, a refinement system is implemented to iteratively enhance and guide the search process for medically similar and pertinent images. This refinement process is incorporated into an interface that allows doctors to crop regions of images, isolating specific relevant features and conveying their importance to the system. Additionally, they are able to select a subset of matches from the retrieved set, indicating the most relevant images and prompting the system to search for images on the fly that are similar to the selected ones.

Also, [110] introduces a novel suite for detecting fraudulent activities for healthcare insurance companies. Although this application could pertain to the medical sector due to its relation to healthcare, it is more appropriately classified as an implementation of iML within the financial domain. The framework itself is trained using a combination of data provided by medical and insurance experts, who offer objectives and hypotheses. Based on this guidance and a dataset of insurance transactions, the ML algorithm assesses the likelihood of a claim being fraudulent. The results are then presented to experts through a visualisation tool that also attempts to establish connections to provide supporting evidence. This aids experts in their process of gathering facts, enabling them to make more informed judgements regarding the authenticity of specific circumstances. Furthermore, these experts have the ability to adjust the weights assigned to certain attributes, which allows the ML algorithm to incorporate human judgment for the next training iterations, therefore adopting a human in the learning loop approach.

In regard to the use of iML approaches in games, [111] presents a system aimed at enhancing the performance of a deep reinforcement learning model that is tasked with the delivery of a quest in a map of the popular *"Minecraft"* game. The paper addresses challenges arising from perceptual aliasing in three-dimensional environments, which can hinder the convergence of deep RL algorithms towards a policy. The reason is that the mentioned aliasing makes many states of the algorithm to share almost the same visual features. Consequently, this issue makes it difficult for the algorithm to establish a clear cause-and-effect relationship between visual traits and actions taken. Therefore, to overcome this obstacle, the work introduces a framework that incorporrates human advice to assist the deep RL algorithm. Whevenever the algorithm takes an action with low confidence, it consults a pool of pending advices provided by the user, and it will refer to this advice as needed. This synergy between human and AI is leveraged and proven to be successful, as reflected by the results obtained in the paper. The deep RL model trained with human advice outperforms the baseline RL, even when the number of human suggestions is low or some of the advice provided is not entirely accurate.

Additionally, [45] presents a simple game in their work to exemplify the advantages of integrating iML approaches by involving a human in the learning process of ML models. The game proposed, named *"The Travelling Snakesman"* (see Figure 3.5) is inspired in the renowned *"Snake"* game. However, in this implementation the snake does not grow and the goal is to eat all apples as fast as possible. Behind the scenes, the game is actually an implementation of the TSP using ACO metaheuristic that has been tailored to accept external pheromone update. This is, each time the user-controlled snake eats an apple, the edge travelled by the user is reinforced with an additional amount of pheromone. Therefore, the algorithm is subtly taking into consideration human contribution. The results obtained in this work also prove that the classic ML approach is significantly outperformed by the iML approach, making the algorithm converge faster to constructing shorter paths to eat all apples.

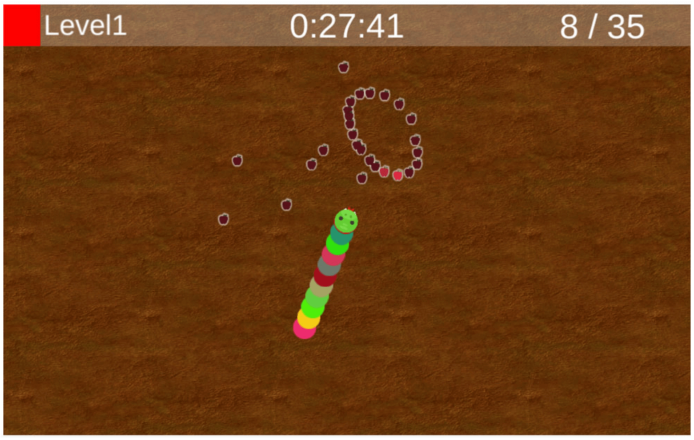


Figure 3.5: Screen capture of the Travelling Snakesman iML game.[5]

Shifting focus towards the military sector, [112] explores neuro-evolutionary ML algorithms coupled with human intervention to enhance the performance of robots in an area search task. In the military context, it is not only important to carry out the mission but to do so in a way that fits with the overall plan of the commander. Therefore, a commander must be able to trust their team to make their own decisions, expecting them to be in line with what the commander needs to accomplish. Bearing this in mind, in this work, some robots are trained using the iML approach, where people could choose a suitable behaviour

[5] Source: Holzinger et al. (2019) [45]

for the algorithm to evolve further in the next iterations of the search. The other robots are trained in the same environment, but with no human contributions, referred to as *"black-box"* ML. When the training phase finished, both types of robots were tasked with a search mission, and users were asked to discern between the robots that use classic ML approaches and the iML-based robots. Their results show that users more often discerned these two correctly, because they could recognise their behaviour.

Finally, in concluding the review of the various examples of applications of iML algorithms, we will explore iML contributions in the domain of cybersecurity. The paper in [113] focuses on the development of an iML algorithm to enhance data exfiltration detection systems. Active Detection system governed by classic ML models are very useful as they possess the capability to identify suspicious activities across vast amounts of data. However, one downside is their tendency to generate numerous false alarms. The work also reflects on the importance of cybersecurity experts, who excel at detecting malicious attacks. Unfortunately, they cannot process large datasets. Therefore, they hypothesise that synergies between human and AI could be leveraged in this area. As they prove throughout their work, by allowing cybersecurity experts to add more annotations to alarms raised by the ML model during the training phase and throughout several iterations, the resulting algorithm raises less false positives than the approach with no human contributions.

3.4 Solving the Museum Visitor Routing Problem

As mentioned in earlier chapters, the construction of museum routes can be seen as a variant of the TSP, where nodes are the exhibits to visit, and the edges represent the cost function to be minimised. This section will explore different bodies of research that also address this problem with approaches that stem from heuristics.

The research paper in [114] develops an ACS-based algorithm for the generation of routes. In their case, a multi-objective approach is followed, where an ant colony per cost function is used. The objective functions to minimise are based on similarity, diversity, novelty, user preferences and progresiveness. The "similarity" metric pursues the generation of a tour through the exhibits which are most similar to the ones already explored by the visitor. On the other hand, an antagonistic objective to "similarity" is also used, "diversity". The aim in this case is to try to find the route that maximises the diversity score so that the most dissimilar exhibits to the ones already visited by the user are taken into account, in an attempt to offer variety to avoid a monotonous experience. In order to try to keep visitors engaged, the "novelty" objective aims to find a route that searches for exhibits not yet discovered by the visitor. Another ant colony focuses on obtaining a route favouring elements that the visitor likes. The last objective, “progresiveness”, aims to find the most well-crafted sequence of exhibits, because it makes sense to view them in a particular order. Finally, a sixth ant colony is employed to merge all of this objectives and therefore be able to generate a weighted metric based on them for the generation of the final route that will be proposed to a visitor.

Also in the realm of ACO algorithms, [115] present a multimedia guidance application in PDAs that visitors can use to obtain more details on exhibits as well as follow a recommended route throughout the museum. Given that the computing power of PDA is very limited, the construction of routes is divided into subinstances that each PDA can process in parallel using also an ACS-based approach. In this case, the cost function to optimise is the satisfaction score of the user, as in this master's thesis project. A satisfaction measure is obtained from the semantic distance between metadata values of an artwork (i.e. a similarity score) and the preference values manifested by a visitor.

Another study, [116] also endevours towards offering museum tours that maximise visitor's quality of experience. In their approach, they aim to find the path of minimum cost between the exhibition's entrance and exit. The Quality of Experience metric that is used as edge weight results from the composition of three metrics. The first one is a metric to take into account spatial parameters, the distance between two exhibits, where exhibits in closer vicinity to the currently visited one will be favoured. The second metric aims at capturing the physical limitations of the environment, which is represented by crowd density at a given area of interest. The third metric represents the time spent listening to a facilitator providing useful explanations about the exhibit. For the museum tour construction algorithm, two methods are used. For lower Ns, the Held-Karp exact method is used, guaranteeing to find the best solution. Nevertheless, to tackle the scalability of the scenario to a real-world museum environment, the Nearest Neighbour heuristic is employed to produce approximations to good tours in terms of satisfaction score.

An interesting note here is that this work introduces the concept of the “human in the loop', which is also addressed in this master thesis. However, they refer to different ideas, as they mean with it that they follow an user-centric approach where visitors select N exhibits of interest and then the algorithm will calculate the path between these items of choice that maximises the overall perceived satisfaction for the visitor. This is better represented in Figure 3.6 which is taken directly from their work. It portrays the topology of their museum exhibition, where only the subset of exhibits chosen by the user is taken into account for calculating a route between the entrance and exit.

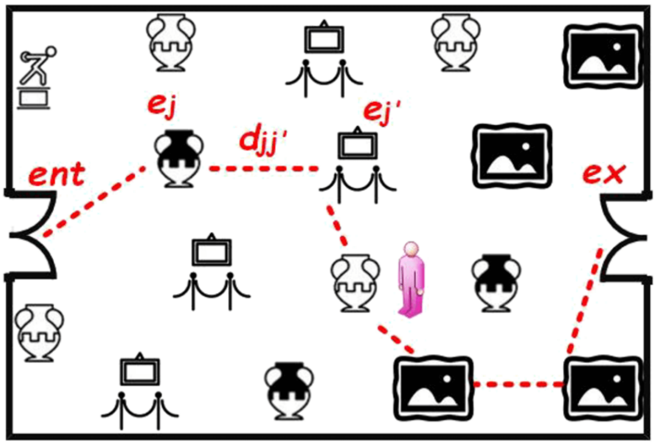


Figure 3.6: Museum exhibition topology of Tsiropoulou et al. and route including the exhibits chosen by the visitor.[6]

As opposed to the aforementioned bodies of research that focus on maximising users' satisfaction during their museum visit, [117] approaches the routing problem from a different perspective. They propose a system that implements a route guidance function that schedules routes to reduce congestion and prolonged touring time. Exhibits forming part of the tour are assumed to be of interest to the visitor, so the goal is for a museum to be able to efficiently balance concurrent visitors and make the flow of people as smooth as possible.

[6] Source: Tsiropoulou et al. (2017) [116]

CHAPTER 4

Enabling Technolgies

This chapter provides a summary of the key technologies that made the project possible. It is divided into two sections. The first section describes the technologies that have brought about the creation of the client-side part of the system, which is the virtual museum users will be able to explore and interact with. The second section focuses on the technologies employed to implement the iML algorithm and the controller logic for the virtual museum simulation interface.

4.1 Virtual Museum Visualization Technologies

4.1.1 THREE.js

THREE.js [118] is an open-source JavaScript library[1] that enables the use of 3D graphics in the web browser through Web Graphics Library (WebGL) [119]. Before providing an overview of the different components that THREE.js consists of, it is worth offering a brief description of what WebGL is.

WebGL is an application programming interface (API) that provides developers with a platform-agnostic method for generating interactive two- and three-dimensional graphics applications on the web. Given that almost every modern web browser provides WebGL support[2], users can enjoy GPU-accelerated content on web pages without having to resort to third party extensions.

Returning to the description of THREE.js, its core elements are the following: i) renderer; ii) scene; iii) camera; iv) lights and v) objects. The renderer is more specifically a WebGL renderer, and it is used by the browser to be able to display the designed scenes. In turn, a scene represents a three-dimensional (x,y,z) space where items are positioned.

The camera is the element that dictates how the scene will be portrayed. Two camera types are available depending on the desired type of element projection: the perspective and the orthographic camera. In the perspective vision distant objects appear to be smaller, whilst in the orthographic view elements' size do not vary with respect to the distance from the camera. Although our eyes are accustomed to the perspective vision, an orthographic camera can be useful when it comes to modelling and estimating object's proportions in the scene. Figure 4.1 further depicts the distinction between these two cameras.

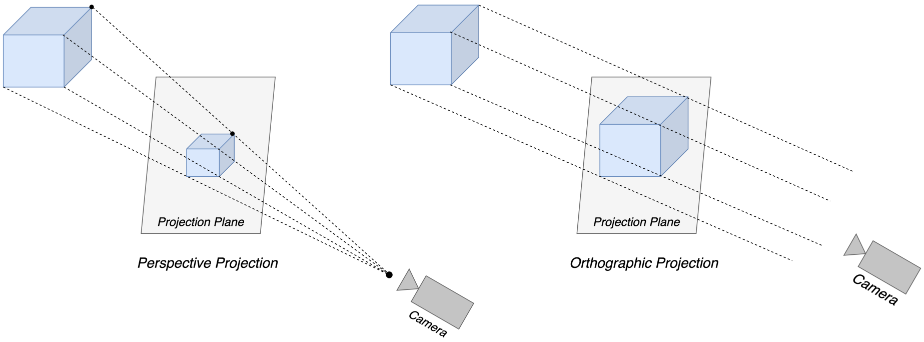


Figure 4.1: Perspective and orthographic cameras

[1] https://github.com/mrdoob/three.js/

[2] https://caniuse.com/webgl

In addition, THREE.js has the ability to integrate different illumination techniques to the scene, so that elements appear brighter or dimmer according to the light type chosen and the illumination angle. There are four principal light sources that can be used. Directional light is characterised by parallel beams emerging from a common direction, whereas ambient light is defined by beams that come from all directions and therefore items are evenly illuminated. Another type is point light, which emanates radially from a source. The remaining source is spot light, which travels in one specific direction and brightens a solid angle (cone-shaped). The reader may better understand the different illumination effects achieved by these light sources through the inspection of Figure 4.2.

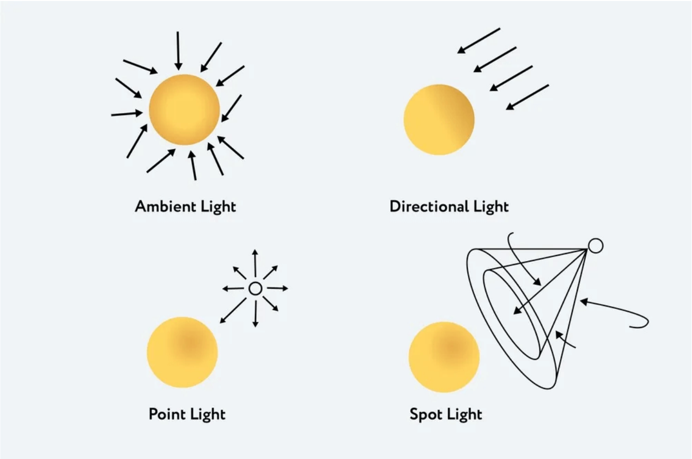


Figure 4.2: Light types in Three.js

The last main element left to describe are objects. In three-dimensional graphics, objects are modelled with (poly)meshes. These are surfaces whose vertices and edges are shared. Almost every 3D object can be approximated using triangular polygon meshes given the simplicity of this structure, and this is how THREE.js constructs meshes. They are made up of two elements: a geometry and a material. The geometry represents the shape of the object (edges, vertices...), while the material defines the object's appearance and its behaviour with respect to light sources in a scene. Figure 4.3 illustrates a mesh as the composition of these two aforementioned elements.

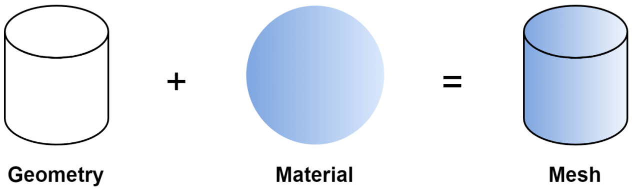


Figure 4.3: Construction of a Three.js mesh Consisting of a geometry and a material.

4.1.2 RAMEN

Representation of **A**nimated **M**ultitudes in **EN**vironments (RAMEN)[3] is a three-dimensional visualization tool for social simulations. It employs THREE.js as its core library to generate indoor designs, and it is the tool that has been used as the starting point to generate the virtual museum scenario of this project. RAMEN consists of three principal subsystems:

- Batch module - accepts input data where the floorplan is described, rooms are delimited and items are distributed across rooms. It also allows to define the movement patterns of agents.
- Model generation subsystem - is nurtured with data given to the batch module and renders a scene representing the input data received.
- Visualization subsystem - allows users to interact with the simulation and control certain represented elements.

However, in order for this tool to serve the needs of this project, it needed to be modified in a number of ways. For instance, a first-person viewpoint has been added, allowing users to explore the museum in an immersive manner using the mouse and movement keys. Additionally, the tool comes with an in-built python visualization server, but as a new server was engineered for this project, just the essential JavaScript code and utilities were taken from RAMEN. Figure 4.4 illustrates the architecture of the tool, and the submodules that were utilized and modified for the virtual museum visualization are shown in blue.

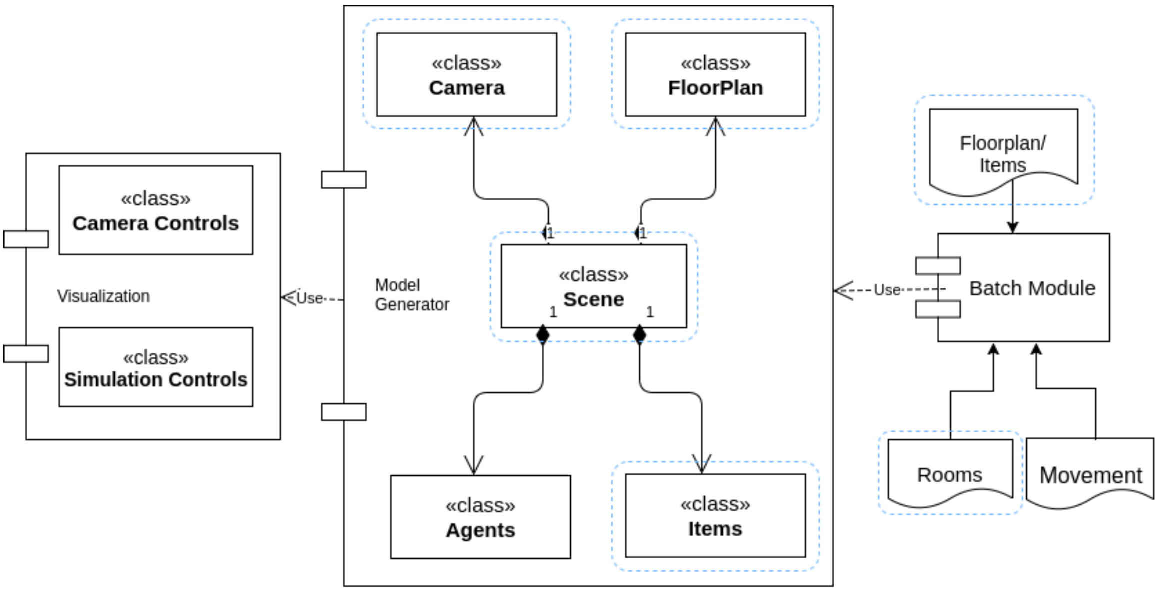


Figure 4.4: RAMEN architecture.

[3] https://gsi.upm.es/software/projects/ramen/

Therefore, the methodology for applying the modified version of RAMEN is the following: given a JSON file describing which (x,y) positions delimit each room and which items belong to each room, a THREE.js scene is generated with a set of meshes that represent the specified input floor plan. Wall and floor materials used are the default ones applied by RAMEN, whilst for paintings an image of them is used which RAMEN can add depth to so they are bestowed a three-dimensional appearance. The camera is placed at the entrance of the museum in a first person perspective, and RAMEN coupled with the visualisation server are awaiting user movement to update the camera view accordingly. A comprehensive explanation of RAMEN's implementation for the virtual museum can be found in Chapter 6. Meanwhile, the next part will explain the technology used for the aforementioned visualisation server.

4.1.3 Flask

Flask [120] is a Python framework for developing web applications. It is well-liked for its simplicity and lightness, as it includes only the essential tools for rapidly creating web applications. In light of this basic approach, Flask is more accurately described as a modular micro-framework, making it an excellent option for developing simple web applications or prototypes. If extra utilities are necessary, such as user authentication, REST implementation, or database integrations, among others, there is a vast selection of third-party libraries available to facilitate the application's development.

Flask can be used to develop the server-side logic as well as the user interface for an application. Business logic is written within Flask's controller classes, but in order for a client to communicate with the application, Flask relies on Werkzeug [121] as the principal tool to manage and process HTTP connections. Regarding the front-end side, Jinja [122] is the core library used.

Werkzeug allows for the creation of Web Server Gateway Interface [123] (WGSI)-compliant Python web applications. Web server software is used to manage sophisticated low-level tasks such as handling HTTP connections, processing requests, applying security protocols,etc; hence, the question arises as to how a Python programme can communicate with web servers to deliver a Python application via HTTP. For this purpose, the WSGI standard was conceived, which acts as a bridge between the two previously described components by providing a common communication framework.

Taking into account Flask's focus on lightness and quickness, Werkzeug is a convenient WSGI tool to integrate with Flask given that it provides a simple built-in WSGI HTTP server suitable for local development and testing. However, simplicity comes at the expense of poor performance under heavier loads. As a result, since Werkzeug is less powerful than other WSGI tools such as uWSGI [124] or Gunicorn [125], it would be essential to switch to one of these alternatives when deploying the application to production. These other technologies also include more complex, robust, and standalone HTTP servers. Nevertheless, it is still recommended to couple them in production with a web server that handles static files and manages requests. In turn, a lot of the load can be relieved from the WSGI HTTP server and, therefore, boost performance.

Regarding content rendering, Flask uses view templates to enhance the application's functionalities while separating the presentation layer from the business logic. In essence, Jinja permits data to be transferred as Python code and processed server-side, allowing the layout and behaviour of the final HTML page to be dynamically adjusted before being sent back to the client. Moreover, utilising a template engine like Jinja has the benefit of allowing template inheritance, which encourages code reuse and maintains application-wide consistency.

Figure 4.5 depicts the architecture of a Flask web application at a high level to better show how a Python Flask web application operates. It also illustrates the connection between Flask and its two key libraries, Werkzeug and Jinja. No standalone web server is included, as this figure illustrates how Flask is originally distributed (suitable for fast local development). The production architecture would consist of a separate WSGI server (Gunicorn, for instance) and a web server such as NGINX acting as reverse-proxy [126].

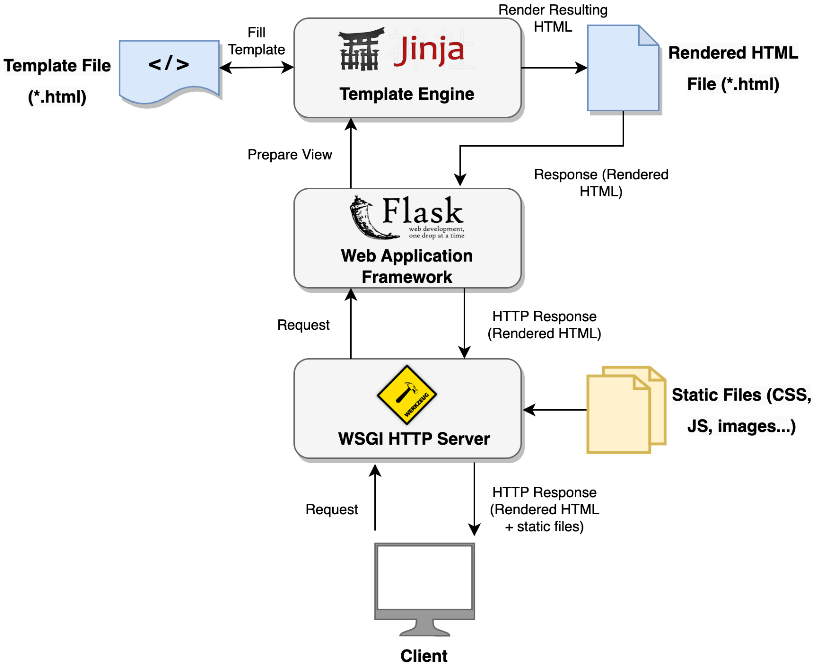


Figure 4.5: Flask app architecture for a local environment

4.2 Simulation and Algorithm Controller Technologies

4.2.1 MESA

MESA [127] is a framework written in Python that allows building agent-based models. These are computer simulations that represent a system as a group of agents that interact with each other and their environment [128]. On account of the simulation, it is possible to observe and analyse the behaviour of a system.

The three main modules that MESA consist of are: i) the model, ii) the analysis subsystem, and iii) the visualisation feature. An overview of these three components and their respective submodules will be provided in the following paragraphs.

The modelling module provides the necessary tools to construct a model that represents the desired environment and conditions to observe. There is a Model class that represents the model itself and allows users to define and adjust model parameters. In the project at hand, the museum model's parameters will be primarily those of MMAS. Also, there are Agent classes, which will be used to represent visitors and exhibits. Additionally, a Scheduler class is provided for controlling the temporal order of agents' actions. Finally, the Space class enables users to define the area in which agents can move and interact with their surroundings. This is in essence a matrix grid where each element is a position in which an agent can be. In order to make the simulation appear more genuine, visitor movement is constrained by walls, and to travel to different rooms, they must necessarily go through the grid location containing the doors. These aforementioned obstacle items are not provided by MESA and have been developed entirely as part of this thesis, along with the logic to evaluate whether a visitor can advance to the desired position.

Regarding the analysis subsystem, it offers a set of tools to gather and access data generated by the running of the model, not only after the execution has concluded, but also at each iteration (DataCollector class). In addition, the BatchRunner class allows users to execute the model several times and optionally alter the values of the specified parameters.

With respect to the last submodule, the visualisation system, MESA puts at our disposal a number of instruments to create and manage an interactive visualisation of the model. It comes with a visualisation server that handles the generation of a web server containing the logic to interact with the visualisation interface, which paints in a canvas element a two-dimensional representation of the model. It is designed in such a way that it is easy to modify the way elements are portrayed, and values of the model parameters can be dynamically adjusted through the user interface. For the virtual museum, we have leveraged this module to have a simple and quick visual representation of the simulation. However, given that real users will act as visitors, this two-dimensional representation is not suitable to offer a realistic and immersive virtual museum experience. Hence, an advanced three-dimensional visualisation module decoupled from MESA has been tailored for this project.

Figure 4.6 presents a diagram summarising and illustrating the operation of the three main MESA modules that have been described, as well as the interconnection between them.

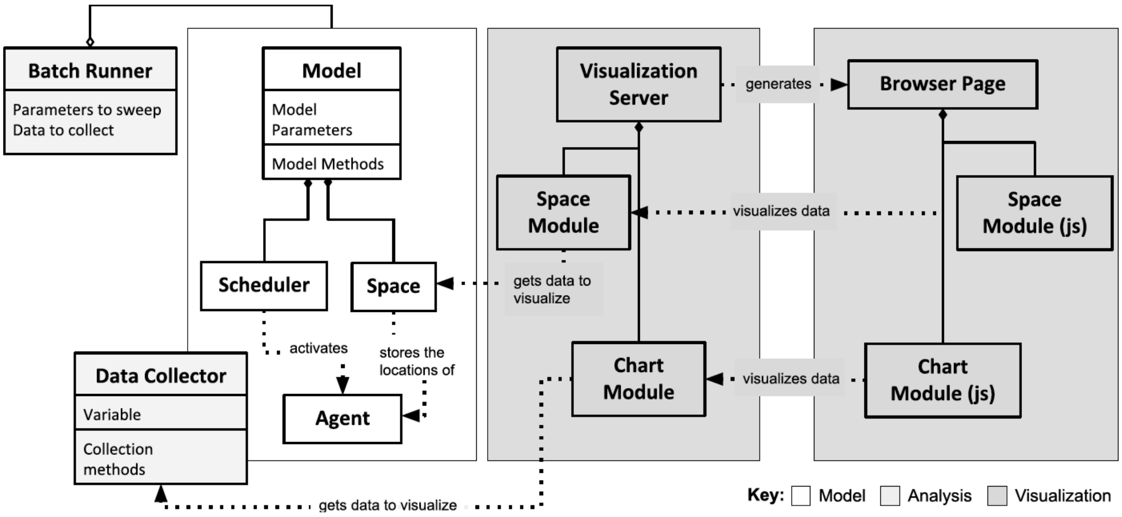


Figure 4.6: MESA model components

Source: *"Utilizing Python for Agent-Based Modeling: The Mesa Framework"* [129]

4.2.2 Shapely

Shapely [130] is a Python library focused on providing users with computational geometry tools to handle and analyse vector data. It has been the main piece of software used to control the validity of visitor movement, so that the system is able to prevent visitors from moving through walls and allowing only to travel between rooms through doors.

Shapely works with geometrical elements of type *Point*, *Curve*, and *Surface*. Regardless of the object type, all are linked to three mutually exclusive sets of points in the real coordinate plane $\mathbb{R}^2$ [131] (Shapely does not perform three-dimensional geometrical computations). These groups of points are i) *interior* points, which are contained within the object; ii) *boundary* points that demarcate the borders of the element; and iii) *exterior points*, which are essentially all those points outside the object, not belonging to any of the other two sets.

Returning to the aforementioned geometrical elements handled by the library, a *Point* object (*Point* class) - being the most fundamental geometrical element since it only represents a deterministic location (x,y) - is itself an interior point. Given that it has no length, no boundary points can be delineated, because the point itself cannot simultaneously belong to both interior and boundary sets, as per the aforestated principle of mutual exclusivity.

The next geometric object, *Curve*, is represented as a concatenation of open straight lines (*LineString* class). Provided that a straight line is essentially an infinite succession of points, these are all considered to be part of the interior set, except for the points that determine the start and end of the curve which belong to the boundary set.

If the curves were to end at the same point where they started, a region would be enclosed by the closed curve, resulting in a *Surface*, the remaining geometric object left to describe (denoted by the *Polygon* class). The infinite collection of points forming the enclosed area belongs to the interior set of points of the object, whereas the curves defining the borders of the surface are encompassed in the boundary set of points.

Amongst other features, Shapely allows users to seamlessly carry out geometrical analyses. These can be based on properties and positions of involved elements, or based on the spatial relationships between objects given as binary predicates. The former includes calculations such as distances, centroids, areas, perimeters, buffers (a way of representing a circle based on an area around a center point to which the buffer is applied)... The latter includes calculations to determine whether two objects are intersecting, touching, overlapping crossing, one contained within the other, etc. In order to use the correct predicates for the required use case, it is important to know how they take into consideration the interior, boundary, and exterior sets.

For the development of this project, museum rooms have been modelled as Shapely Polygons, doors as LineStrings, and the visitor as a buffered Point i.e., circle whose radius is the field of interaction (or field of view).

4.2.3 NetworkX

Networkx [132] is a Python package that allows to create, modify and analyse network graphs. This tool has been used in this thesis to carry out the implementation of Min-Max Ant System heuristic, since the problem can be interpreted as a graph of exhibit nodes and the cost metric of the edges is a pheromone trail.

With NetworkX, both simple and directed graphs can be built with a few commands, as it takes care of abstracting the underlying difficulty and intricacies of constructing graphs.

For example, when adding an edge A-B between two nodes A and B in a simple graph, NetworkX is aware that edges are not directed and therefore it automatically adds the “return” link B-A.

Furthermore, although the graph instance created belongs to a custom NetworkX class, nodes and edges do not. These can be any hashable Python object, meaning that such flexibility is very convenient for representing complex networks in a wide range of fields. Nodes and edges can be complemented with any attributes of choice. Some of these attributes are reserved by NetworkX. For instance, node “weight” attribute is used by the library to build weighted graphs, and the shortest path between two nodes taking into account edge weight can be then calculated using the built-in implementation of Dijkstra’s algorithm. Another special property is “color” of nodes/edges, used to modify the default graph colors when a graph representation is portrayed. Excluding these exceptions, attributes can be anything the user desires, and as aforestated, in the case of this thesis, the amount of pheromone has been added as an attribute of the edges.

Apart from custom attributes and built-in implementations of the most typical network analysis algorithms, NetworkX offers a visualisation module to draw and export graphs. Additionally, the package offers interoperability with other well-known algebraic and statistical calculation Python libraries such as Numpy or Scipy [133]. This is done by NetworkX through the transformation of graphs into (sparse) matrices.

Overall, it is a flexible and easy-to-use tool that allows users to rapidly create and analyse graphs, hence its suitability for this project. Figure 4.7 provides an example of the ease and quickness of creating, analysing and visualizing a graph with NetworkX. In a very few steps it illustrates how to create a network graph, calculate the shortest path from node A to node F using Dijkstra’s algorithm, draw the graph and highlight in it the shortest path from A to F.

```python
import networkx as nx
graph = nx.Graph()
nodes_pos = nx.spring_layout(graph)
graph.add_edges_from([(’A’,’B’, {’weight’: 4}), (’A’,’C’, {’weight’: 2}),
 (’B’,’C’, {’weight’: 4}), (’C’,’D’, {’weight’: 3}),
 (’C’,’E’, {’weight’: 1}), (’C’,’F’, {’weight’: 6}),
 (’D’,’F’, {’weight’: 2}), (’E’,’F’ {’weight’: 3})])

# Calculate shortest path A->F from weighted path using built-in Dijkstra’s
    algorithm implementation
dijkstra_nodes = nx.shortest_path(graph,’A’, ’F’, weight=’weight’) #[’A’,’C’,
     ’E’, ’F’]
dijkstra_edges = [(node, dijkstra_nodes[i+1]) for i, node in enumerate(
    dijkstra_nodes) if i < len(dijkstra_nodes) - 1] # [(’A’, ’C’),(’C’,’E’),
    (’E’, ’F’)]

# Draw graph nodes, those in the shortest path A -> F will be coloured in
    blue
nx.draw(graph, nodes_pos, node_color=["lightblue" if n in dijkstra_nodes else
     "lightgray" for n in graph.nodes()], with_labels=True)

# Draw graph edges, those in the shortest path A->F will be thicker than the
    rest and highlighted in blue
fig = nx.draw_networkx_edges(graph, nodes_pos,
    edge_color=["blue" if e in dijkstra_edges else "darkgrey" for e in graph.
        edges()],
    width=[3 if e in dijkstra_edges else 1.0 for e in graph.edges()])

# Show weights of each node
fig = nx.draw_networkx_edge_labels(graph, nodes_pos,
    nx.get_edge_attributes(graph, ’weight’), font_color=’red’)
```

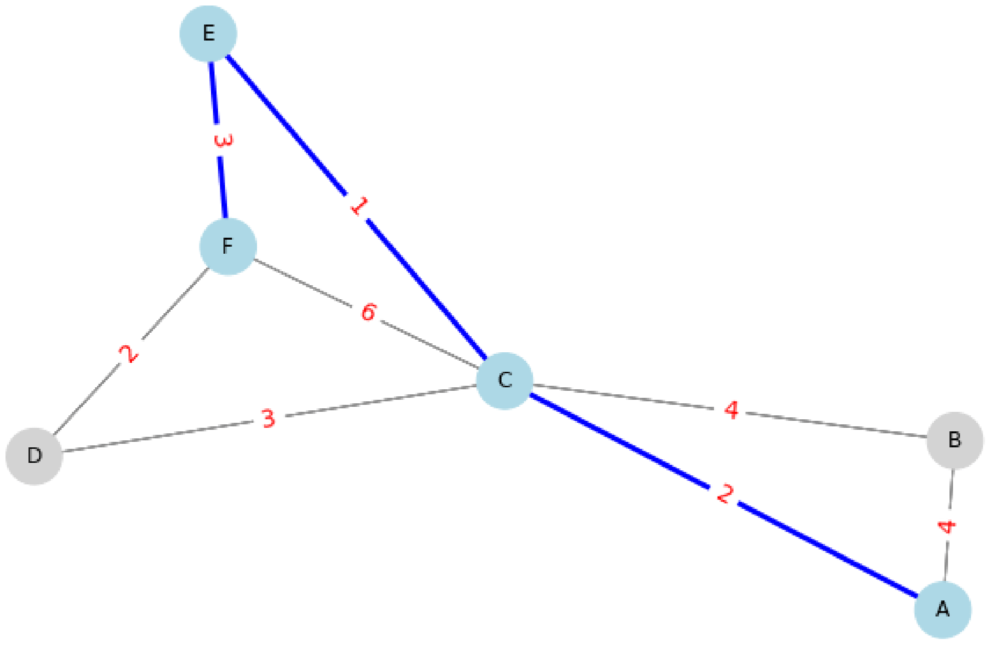


Figure 4.7: Creation of a graph with NetworkX, calculation of shortest path between two nodes and visual representation of graph and shortest path.

4.2.4 MongoDB

MongoDB [134] is an open-source document-oriented NoSQL database that, given its power and flexibility, has become one of the database systems of choice in the technical stacks of many high-profile business[4]. It has been ranked as the fourth[5] most popular database management system worldwide, only behind propietary relational database systems (Oracle, MySQL and Microsoft SQL Server). This means that amongst NoSQL management systems, MongoDB is the most popular technology.

In a document-oriented database, a document refers to a record (analogous to a RDBMS row). A group of documents is stored in a collection, which would be the equivalent to the concept of tables in SQL. Each document can have one or more key-value pairs, these keys are known as fields (comparable to SQL columns). MongoDB has the following key features:

- **JSON Documents:** MongoDB's document model is based on JSON objects - concretely, Binary JSON (BSON) - , which has become one of the most popular data format standards across APIs. Since many programming languages can handle JSON objects, the implementation of MongoDB as the database layer becomes quicker and easier.
- **Schemaless:** MongoDB, unlike relational database management systems, does not need to define a structure for the data it holds. This implies that the deployment of changes to data can be carried out faster and more efficiently, since changes do not need to apply to the rest of documents.
- **Ad-hoc queries:** searches can be done by field, ranges and regular expressions. It is possible to select which fields to return from the query, but MongoDB also allows storing JavaScript functions.
- **Indexing:** MongoDB supports indexing to enhance performance and sorting for frequently executed queries. An index can be created for any field from a document, and secondary indices are also supported.
- **Replication:** master-slave replication is supported, whereby the master MongoDB node can perform read and write operations, while slave nodes can copy data from the master and only execute read operations.

[4] MongoDB customer success stories: https://www.mongodb.com/who-uses-mongodb

[5] Ranking of the most popular database management systems worldwide, as of February 2023: https://www.statista.com/statistics/809750/worldwide-popularity-ranking-database-management-systems/

- **Sharding and Load Balancing:** MongoDB has been designed to scale horizontally by means of shards. A shard is just a configuration of a master node with one or more slaves, and the database must choose a sharding key that determines how data in a collection will be distributed across different machines. This also allows MongoDB to balance the workload and duplicate data in order to keep the system functional should a hardware failure occur.
- **File Storage:** in virtue of MongoDB's load balancing and replication capabilities, it can be used to store large objects and files via GridFS functionality.
- **Aggregation Framework:** MongoDB supports an implementation of MapReduce [135] paradigm and it offers the aggregate() operator that can be employed to construct a pipeline with different aggregation stages.

Taking into consideration the list of highlighted features MongoDB offers, it is a very convenient database technology choice for this project. As outlined earlier in this section, the virtual museum floorplan is in JSON format, meaning it can be inserted as a document and queries can be made against it to easily retrieve data about room distribution and exhibits placed in each room. Additionally, the gamified smart-objects platform developed as part of my undegraduate thesis also uses MongoDB to store the automatically generated questions as they are in JSON format. Since the system developed as part of this master's thesis is linked to the smart-objects platform by means of QR codes, it also makes sense to use the same database for both cases.

CHAPTER 5

Architecture

This chapter presents the methodology used in this work. It describes the overall architecture of the project, with the connections between the different components involved on the development of the project.

Prior to outlining the architecture of the suggested iML-driven gamified routing platform for museums, it is essential to have an understanding the methodology used to train the algorithm and include gamification tactics into its learning process.

Upon the start of a visitor's museum tour, the proposed system will begin tracking the visitor's movements. When the visitor comes across an exhibit of interest, the platform will record it and adjust the points granted to the visitor based on both the exhibit itself and the distance travelled to reach it. Simultaneously, the iML routing subsystem will factor in this exhibit as part of its learning process, assigning greater importance to the path between the previous and current exhibits. This cycle continues as the visitor moves towards another exhibit, and the system will continue taking this feedback into account until the visitor concludes their visit. This methodology is portrayed in Figure 5.1.

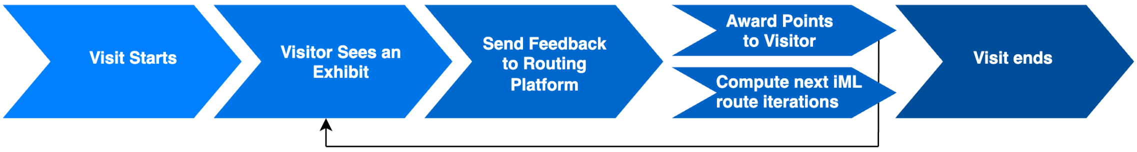


Figure 5.1: Methodology for the gamification and training of the iML-powered museum routing platform.

Based on the proposed methodology, we will now proceed to introduce the platform architecture, designed to facilitate the gamification of the learning process for the algorithm responsible for calculating routes to improve visitors' satisfaction within museums.

The platform can be divided into two principal systems:

- **Museum Environment Interaction System**: offers a gamified visual interface through which visitors interact with the system so that the platform can keep track of the routes taken by visitors to see exhibits. Relevant visitor data collected here is passed on to the museum controller system.
- **Museum Controller System**: receives data gathered by the environment interaction system and processes it accordingly. Visitor paths to exhibits are fed to the iML routing system, so that visitor's patterns are considered in the training phase of the algorithm. Also, based on the feedback provided by the environment interaction system, it awards or deducts points to users as part of the gamification elements provided, and the result is sent back to be displayed to the users via the interaction system.

The previous explanation about the two main systems that conform the platform is just a concise outline of their core functions. In fact, they consist of different modules interconnected with each other as depicted in Figure 5.2. Subsequent sections will offer a comprehensive analysis of the different elements constituting the platform.

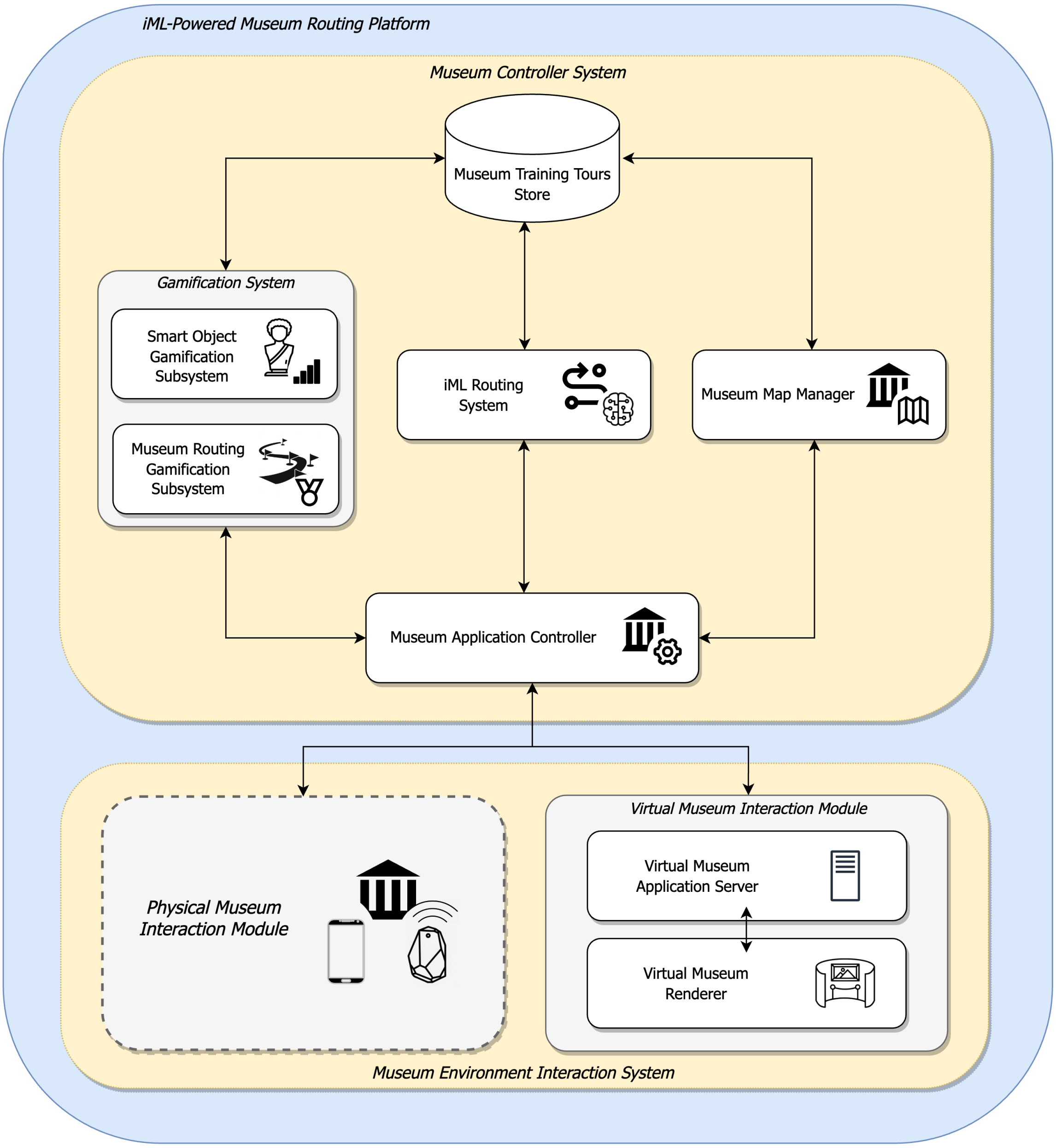

iML-Powered Museum Routing Platform
Museum Controller System
Museum Training Tours Store
Gamification System
Smart Object Gamification Subsystem
Museum Routing Gamification Subsystem
iML Routing System
Museum Map Manager
Museum Application Controller
Physical Museum Interaction Module
Virtual Museum Interaction Module
Virtual Museum Application Server
Virtual Museum Renderer
Museum Environment Interaction System

Figure 5.2: Architecture of the proposed nteractive machine learning-powered museum routing platform.

5.1 Museum Environment Interaction System

Visitors interact with the proposed platform through this system. For the scope of this master's thesis, a virtual museum interaction module has been developed. This is, a three-dimensional virtual museum has been built that visitors can explore using a web browser. Nevertheless, Figure 5.2 also depicts a *physical museum interaction module*. It consists of two fundamental elements:

- **Virtual Museum Application Server**: manages the virtual museum application. It handles the registration of new users, as well as the processing of logins and logouts. Once a user is logged in, the server invokes the renderer. This enables users to access the virtual museum and navigate it through a point-of-view experience. Visitor moves and exhibits seen information is stored by the server, which is provided by the renderer. These data are sent to the museum controller system for further processing and to award points for exhibits seen and travelled paths.

- **Virtual Museum Renderer**: creates a three dimensional virtual museum interface for users. The application server hosts a floor plan, which the renderer interprets to construct the virtual museum according to the specification defined in said document. The floor plan essentially serves as a blueprint for building the virtual museum and arraging exhibits. Therefore, the virtual museum renderer component can generate any three-dimensional virtual museum of choice, as long as the floor plan document is correctly defined.

  The renderer is also responsible for recording visitors' movements and relaying this information to the application server. Furthermore, it provides additional pertinent exhibit information. Upon interaction with an exhibit, a QR code is also displayed. This code enables users to access a Trivia quiz game served by the Smart Object Gamification Subsystem (which is part of the platform designed in my end of degree project and now expanded). This subsystem will be elaborated later in subsequent sections.

It is worth noting that the physical interaction module is outlined with a dashed border in the Architecture Figure 5.2. This is due to the fact that this module has not been implemented as part of the initial design of this platform. The decision not to include this module was influenced by the challenges associated with the implementation within a physical museum setting: asking for permission, spending time in the museum to implement a viable technological solution, etc. This would most likely have implied a much greater number of hours invested in the project than the estimated ones in the scope of a master thesis.

However, it is important to highlight it as part of the architecture because the Museum Controller System is designed to remain agnostic and adaptable to various user interaction methods. The data emitted by the interaction module would be very similar or the same whether in a virtual or physical museum environment. Consequently, connecting a physical museum to the system would be straightforward, as the system's design allows for scalability.

For instance, in a physical museum, an implementation could be devised based on beacons to locate users and a gamification application to transmit this information to the Museum Controller System. Further elaboration on this idea will be provided in the section discussing future lines of work in the final chapter of this work.

5.2 Museum Controller System

This system receives data collected by the *museum interaction system*, processes it, acts upon it and returns feedback to the interaction system. In essence, intelligence resides within the *museum controller system*. As it can be seen in Figure 5.2, it is made up of different submodules, each serving a specific purpose. These are: i) *museum application controller*, ii) *museum map manager*, iii) *iML routing system* iv) *gamification system* and iv) the *museum training tours store* as the database persistence layer. These elements will be analysed in the following sections.

5.2.1 Museum Application Controller

This module provides an interconnection layer between the Museum Controller System's other subsystems and the interaction system. It receives data collected by the Environment Interaction System and appropriately distributes it to each module for processing. If a module provides feedback, it will communicate this to the Museum Controller System, which will then forward it to the interaction system. The main data sent to the Museum Controller System from the interaction System includes notifications about newly viewed exhibits and the user's position (movement).

Initially, the controller facilitates this information to all modules simultaneously, except when it originates from a virtual museum scenario. In this case, it is necessary to first validate the position the user desires to advance to, which is handled by the *museum map manager*. Consequently, in this scenario, the information must only first be provided to the Manager. If the movement is indeed valid, the received data is then distributed to the remaining modules for processing, as it can now be deemed as correct. This process of information exchange and distribution by the Museum Application Controller is depicted in the flowchart shown in Figure 5.3.

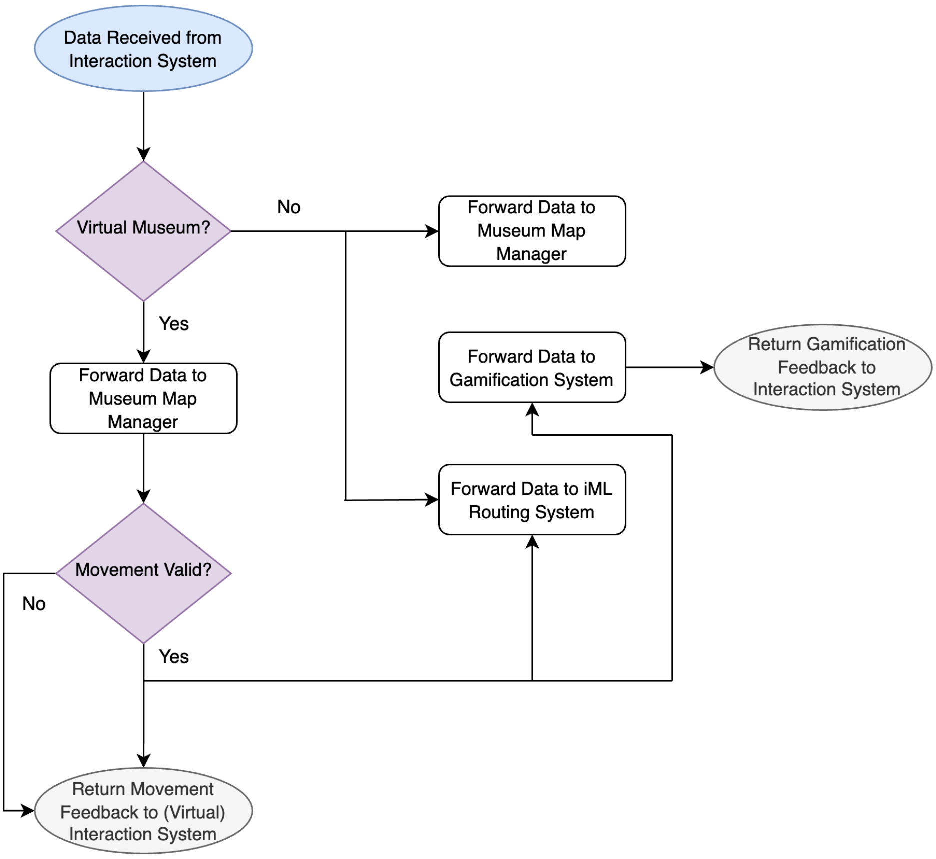

Data Received from Interaction System
Virtual Museum?
No
Yes
Forward Data to Museum Map Manager
Forward Data to Museum Map Manager
Forward Data to Gamification System
Return Gamification Feedback to Interaction System
Forward Data to iML Routing System
Movement Valid?
No
Yes
Return Movement Feedback to (Virtual) Interaction System

Figure 5.3: Flowchart of the Museum App Controller's data distribution and exchange process.

In the scope of this project, the museum controller system needs only to notify the interaction system of the gamification feedback and of the validity of the movement in the case of a virtual museum scenario. There is no feedback sent from the iML Routing System because this prototype focuses on training the algorithm, so it is not ready yet to provide route recommendations. This is logically the next stage of evolution the system, which will be discussed in the future lines of work section found in the final chapter of this work.

5.2.2 Museum Map Manager

The map manager is tasked with registering visitors' activity. It gathers information about user movements (positions) and logs them, saving them to a list of positions. This means that this module stores the paths followed by visitors throughout the museum. When a new position or movement is registered, the module also associates to them a timestamp. Additionally, when notified of a new exhibit visited, the module will calculate the time taken for the visitor to reach the exhibit since the last one seen. This feature extends to recording the overall duration of the visit as well. Figure 5.4 provides a depiction of the sequence of tasks performed by the module when new data arrives.

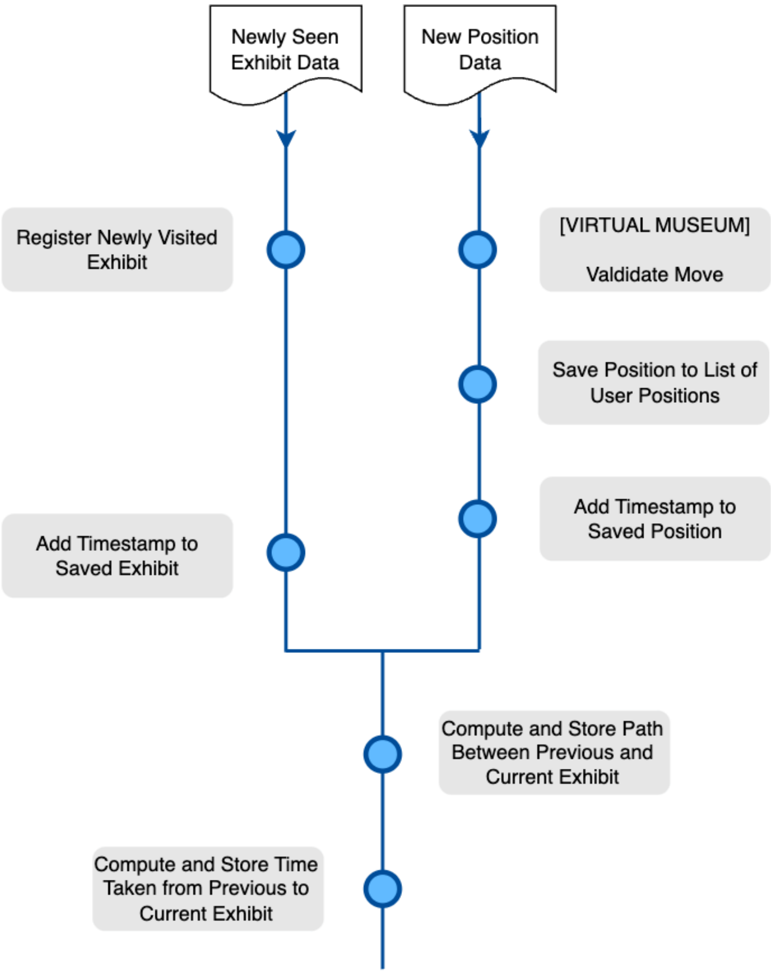

Newly Seen Exhibit Data
New Position Data
Register Newly Visited Exhibit
[VIRTUAL MUSEUM]
Valdidate Move
Save Position to List of User Positions
Add Timestamp to Saved Position
Add Timestamp to Saved Exhibit
Compute and Store Path Between Previous and Current Exhibit
Compute and Store Time Taken from Previous to Current Exhibit

Figure 5.4: Tasks performed by the Museum Map Manager when exhibit and position data are received.

While it is true that in the case of interacting with a real physical museum the tasks of this module would only be limited to those previously described, this subsystem has more relevance in a virtual museum scenario. This is because in a virtual museum, validity of movements must be verified in order to prevent users from going straight through walls and not through doors. Hence, the map manager is also capable of performing obstacle detection to verify requested positions are valid. In order to do this complex task, it bases on the museum blueprint the virtual interaction uses to create a matrix representation of the museum's main elements (walls, doors and exhibits) and their respective locations.

5.2.3 iML Routing System

The interactive Machine Learning (iML) Routing System holds the MMAS-based algorithm used learn the most satisfactory routes for visitors. When the visit starts, it executes N iterations of the algorithm. Afterwards, it remains in standby waiting to be notified that the visitor has seen an exhibit. Once it receives this data, the algorithm takes the exhibit into account into its learning process by providing an extra pheromone increment to the edge the visitor has travelled.

The relationship between the different elements that compose the iML Routing System is represented in the Class diagram of Figure 5.5.

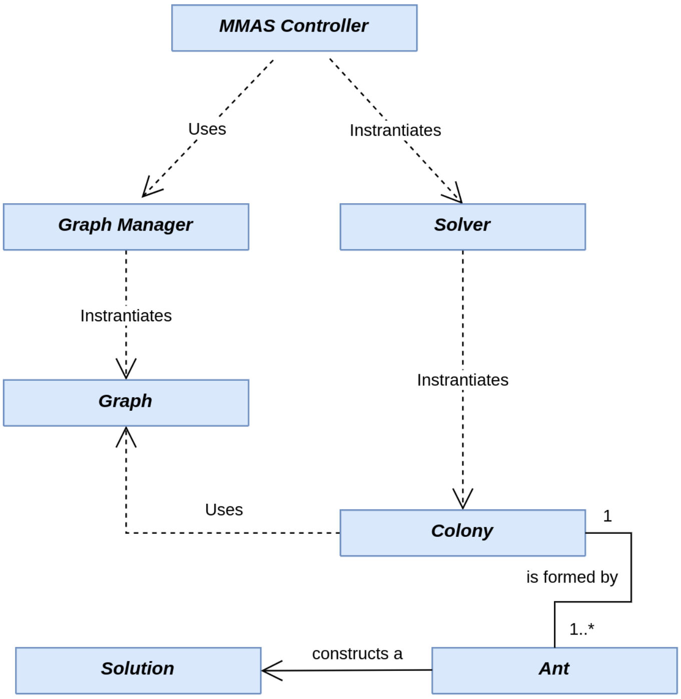


Figure 5.5: iML Routing System's Class Diagram

A description of each core element composing the algorithm is offered next:

- **MMAS Controller**: this is the main structure that initialises and controls the algorithm. It has been designed to be able to compute a number of iterations specified by the user and afterwards wait for further instructions. Basically, when a user has visited an exhibit, the MMAS Controller will receive the seen exhibit and grant a pheromone trail boost to the edge conformed by the previous exhibit seen and this new one. This is in essence the part where user feedback is being actively taken into account in the learning loop. MMAS Controller takes as input the following parameters:

  - *Graph Manager instance:* this element is used to handle the graph that the algorithm uses. It is responsible for converting the input museum blueprint into a valid graph that can be interpreted by the algorithm. This implies that it takes care of assigning weights to each node (initially based on euclidean distance). The Graph Manager will also be in charge of assigning the initial pheromone trails to every edge. The prepared graph will then be passed on to the MMAS Controller to act accordingly.
  - *ρ:* the pheromone factor to use which will dictate the pheromone evaporation rate $(1 - rho)$
  - *λ, μ:* the parameters that regulate the relative importance of pheromones and weight in each edge when computing node transition states.
  - *Number of ants (N):* the number of ants that will be used to build solutions. In essence, there will be N tours generated in each iteration. A greater number of ants means that there are more constructed possibilities, at the expense of increased computational cost. This number must be a tradeoff figure. For instance, using a number of ants equal to the number of total nodes (exhibits + doors) is a sensible approach.
  - *Start node:* The choice of starting node is usually a random choice in the classic ACO approach. However, in this case the visit always starts at the entrance door, therefore this node will be passed as parameter. Nevertheless, the start node parameter has been included because it is useful to have this element parametrised as it can then allow to explore test subgraphs to calibrate the algorithm, as we have done.

- **Colony:** A colony of N ants will be generated using this element. Basically, it is a convenient and organised way to generate articial ants with the λ and μ parameters that it receives from the MMAS Controller, which is responsible for invoking this component.

- **Ant**: an Ant component is responsible for building a candidate solution (i.e. tour) of the given graph. Ant elements hold the logic of retrieving all unexplored nodes and construct a tour by iteratively transitioning to unvisited nodes (exhibits and doors in the project at hand). They have implemented the probability state rule that is defined for MMAS (refer back to Section 2.3.2). When a node is selected, ants take care of adding it to the list of visited nodes, until no unexplored nodes are left and the tour has been constructed. This tour entity, named *Solution* is in fact another class of the algorithm.

- **Solution**: as explained above, this entity represents a tour that an ant has built. This element stores the path produced as well as the calculated total cost of this path. It also contains other useful elements that are very convenient to keep track of during the construction of the tour. This has the advantage that ants are only basically tasked with choosing next nodes and these results are decoupled of ants to reduce complexity.

- **Solver:** the solver is the main responsible for orchestrating the *Colony* to generate *Ants* and thus instruct them to build *Solutions*. The MMAS Controller invokes this component passing as input the graph to solve, the *Colony* instance and the aforementioned parameters. The Solver is able to organise tour constructions up to a limit of iterations, parameter which is governed by the MMAS Controller. For each iteration, the Solver is in charge of managing the following four tasks (in order):

    1. Obtain candidate solutions by instructing *Ants* to construct tours (as explained above).
    2. Order the candidate solutions in ascending order of total cost.
    3. Retrieve the best solution (first result from the ordered list produced in the previous step). This is the iteration-best solution, which will be used to try to update the global best record. If this new solution has a total lower cost than the tour that is currently considered as the global-best, it will be assigned as the new global best.
    4. Update pheromone trails according to MMAS principles. As it has been proven by research (refer to Section 2.3.2 for further details), choosing iteration-best solutions to update pheromone trails is better for lower number of iterations, and as the iteration count increases it is preferable to select the global best. The *Solver* uses a Sigmoid function to perform a weighted random decision of which type of best solution to use, which will be incrementally biased towards the global best as the iteration count grows. After this has been decided, the component has to perform three pheromone update functions:

        (a) Evaporate pheromone trails of all edges of the graph by the pheromone evaporation factor of $(1 - \rho)$.

(b) Increase pheromone trails of those edges that belong to the best solution (iteration or global best, depending on the decision of the aforementioned Sigmoid function).

(c) Finally, review all pheromone trails in order to detect which edges have pheromone values that are out of the allowed bounds (please refer to Section 2.3.2 for further information on these Max-Min bounds). Those that are below the minimum allowed concentration will be assigned the minimum value. Similarly, those beyond the maximum established concentration will be capped at this maximum threshold value.

5.2.4 Gamification System

The Gamification System of this platform consists of two components: i) the museum routing gamification subsystem and ii) the smart object gamification subsystem.

With respect to the museum routing gamification subsystem, its main goal is to offer gamification elements for the learning process of the iML algorithm. In order to do so, the module will deduct points every time a user/visitor moves and has not visited an exhibit. This penalisation is put in place to motivate visitors to visit the museum with purpose and make their paths efficient. When they visit an artwork, points will be awarded based on a satisfaction score of the exhibit. These points will be therefore added and subtracted respectively to the user's total points, which will be visible to other visitors. Figure 5.6 provides an illustration of the described point awarding mechanism.

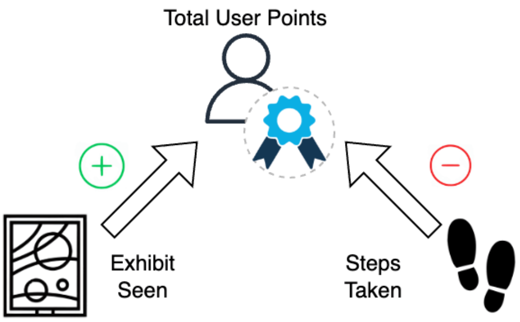


Figure 5.6: User points awarding mechanism used by the Museum Routing Gamification Subsystem.

Moving on to the other module, the smart object gamification subsystem, this component originates from the Smart Object Gamification platform devised as part of my end of degree thesis and subsequently published in [19]. The gamification platform has not only been integrated in this new system, in fact it is more appropriate to say that it has been further developed and grown to offer a complete system aimed at enhancing visitors' museum experiences.

In the Smart Object Gamification Platform, questions about exhibits are automatically generated by exploiting Linked Data and semantic web technologies. These questions pop up to users in the form of a Trivia quiz game whenever they use the devised mobile application to scan a QR code linked to an exhibit of interest. This application includes several gamification tactics such as:

- *Time-based points system:* Users earn points each time they answer a question correctly. The amount of points awarded depends on the time taken to provided an answer. The faster, the more points that can be obtained. If the timer reaches zero, the user will miss the question and earn no points.
- *Ranks*: Users can 'level up' if they earn enough points, making them earn badges and helper tokens.
- *Tokens*: When users level up, they are granted helper tokens. These are 'lifelines' they can use to rule out one incorrect choice in a question.
- *Leaderboard*: The mobile application also displays to users a leaderboard, where the best players appear. The comparison is based on the rank users have achieved and number of points earned in their rank.

Additionally, the mobile application also neatly displays additional information about the exhibit when the QR Code is scanned. This information can be useful to answer correctly the questions about the exhibit.

Figure 5.7 presents the partial architecture of the Gamification System of the Smart Object Gamification Platform. The rest of the architecture is not presented here because it represents the modules that take care of extracting exhibit information and automatically generating questions, whilst the component shown here is in charge of displaying the data generated by the other modules.

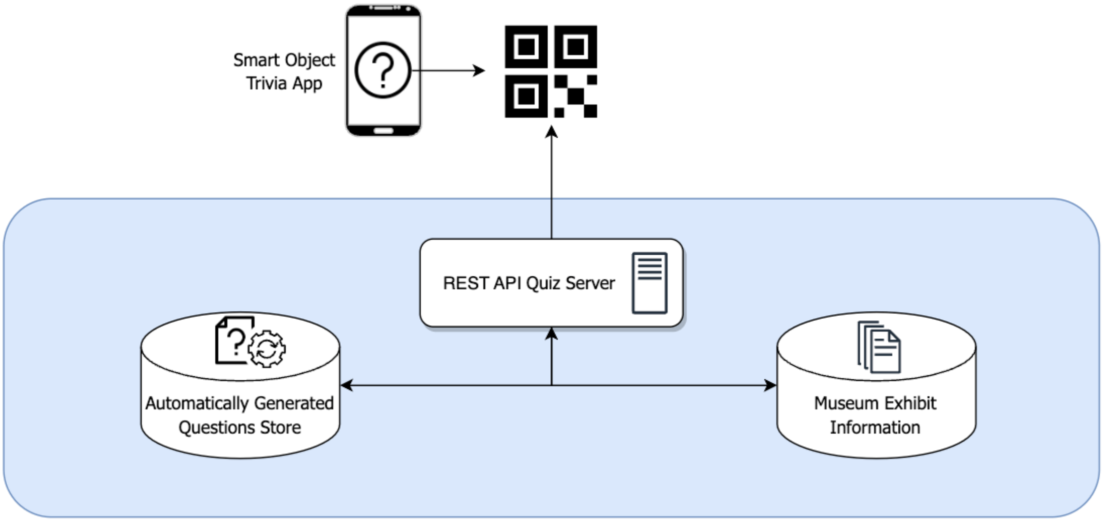


Figure 5.7: Architecture of the Smart Object Gamification Subsystem.

CHAPTER 6

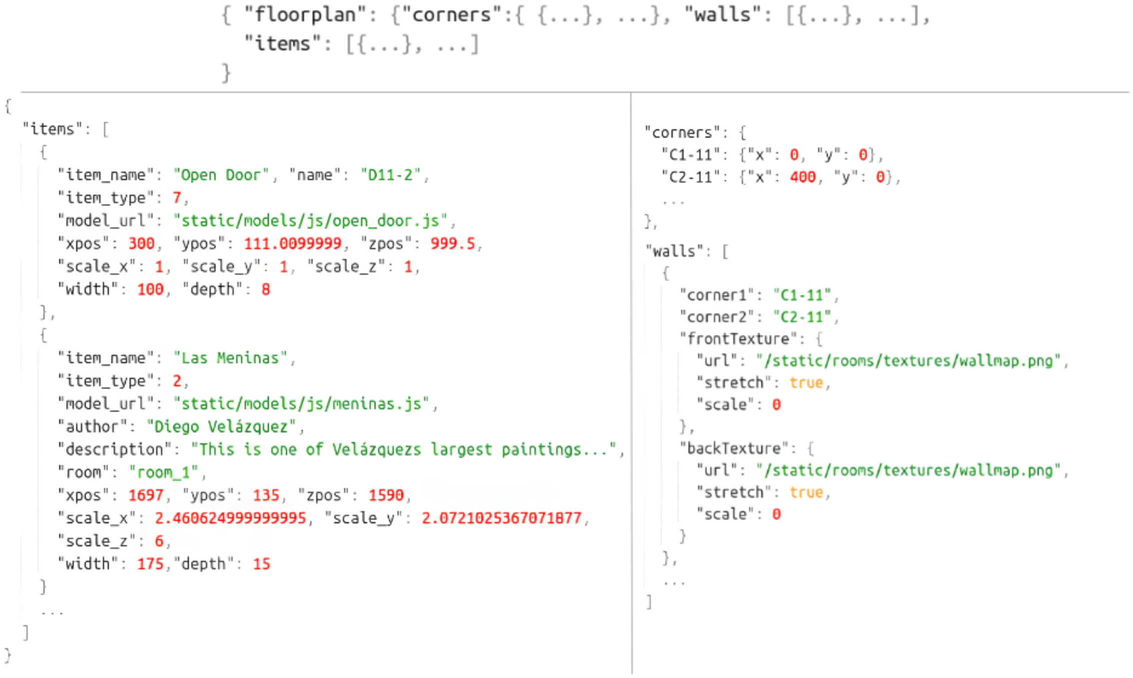
Virtual Museum Case Study Development

This Chapter delves into the development and implementation of a virtual museum that will be used as a means to produce data to aid the iML algorithm during its training phase. First, Section 6.1 describes the construction of the 3D virtual museum based on an input specifications file, the blueprint. Then, Section 6.2 presents the rendered virtual museum and explains how can users navigate the museum and interact with paintings. An important part of the development has been ensuring that the experience is realistic, and hence implementing obstacle detection processes was imperative to prevent users from going through walls. These implementation details are provided in Section 6.3.

6.1 Virtual Museum Map Construction

Taking into account that, as mentioned in the introductory section of this thesis, the primary objective of this project is to gamify the learning process of the algorithm for calculating the most satisfactory museum route, a straightforward two-dimensional visualisation module could have been used to create a bird's-eye view of the museum in order to accomplish this. However, this would not be an immersive experience and would not contribute to the gamification process. On the other hand, a three-dimensional virtual museum is significantly more engaging and interactive, allowing users to immerse themselves more fully in the experience. They will be able to examine artworks from a first-person perspective while navigating the various museum rooms.

In Chapter 5 it was described that the *virtual museum renderer* module is able to build a three-dimensional virtual museum based on an input blueprint. Figure 6.1 portrays the structure of the blueprint with some examples taken from the virtual museum implementation.



```
{ "floorplan": {"corners":{ {...}, ...}, "walls": [{...}, ...],
  "items": [{...}, ...]
}
```

```
{
  "items": [
    {
      "item_name": "Open Door", "name": "D11-2",
      "item_type": 7,
      "model_url": "static/models/js/open_door.js",
      "xpos": 300, "ypos": 111.0099999, "zpos": 999.5,
      "scale_x": 1, "scale_y": 1, "scale_z": 1,
      "width": 100, "depth": 8
    },
    {
      "item_name": "Las Meninas",
      "item_type": 2,
      "model_url": "static/models/js/meninas.js",
      "author": "Diego Velázquez",
      "description": "This is one of Velázquezs largest paintings...",
      "room": "room_1",
      "xpos": 1697, "ypos": 135, "zpos": 1590,
      "scale_x": 2.460624999999995, "scale_y": 2.0721025367071877,
      "scale_z": 6,
      "width": 175,"depth": 15
    }
    ...
  ]
}
```

```
"corners": {
  "C1-11": {"x": 0, "y": 0},
  "C2-11": {"x": 400, "y": 0},
  ...
},
"walls": [
  {
    "corner1": "C1-11",
    "corner2": "C2-11",
    "frontTexture": {
      "url": "/static/rooms/textures/wallmap.png",
      "stretch": true,
      "scale": 0
    },
    "backTexture": {
      "url": "/static/rooms/textures/wallmap.png",
      "stretch": true,
      "scale": 0
    }
  },
  ...
]
```

Figure 6.1: Structure in JSON format of the museum blueprint used in the project, including some examples.

At the top side of Figure 6.1 the base structure of the JSON blueprint is presented, which basically includes two keys: *floorplan* and *items*.

The value paired to *floorplan* is a JSON object containing data about all edges (*corners*) of the museum. Corners are given an identifier and their (x,y) position is also indicated. They are the basic building blocks. Corner data is used for the other key of the floorplan object, *walls*. A wall is represented by a start corner and an end corner, which is in essence a straight line (or a bounded plane in 3D). Additionally, wall objects also contain data about the front and back textures that will be used when rendering them, so that they look realistic. The right part of Figure 6.1 depicts the described structure of corners and walls and provides some real examples.

With respect to the *items* key, it contains an array of the objects that will be placed around the museum. In the case of this implementation, items are mainly the exhibits. Data keys that can be found are the position (x,y,z) and scaling properties; an identifier given to each item; and a link to the model (JavaScript export file from Blender [1] used to render the shape of the item. There is another special type of item, doors. When an item of this type is added to the map, it must be placed in an intersection with a wall plane, and it will be rendered in such a way that the frame of the door can be seen but the inside not. These are open doors and so this allows users to see other rooms through doors. An example of how to include doors and exhibits in the blueprint can be seen on the left side of Figure 6.1.

For the proposed virtual museum in this project, a blueprint has been created consisting of 15 different rooms in which artworks are distributed. Figure 6.2 portrays an overhead shot of the virtual museum, resultant from feeding the input blueprint to the *virtual museum renderer*. Labels are included to help the reader identify how the fifteen rooms have been distributed.

Ideally, the implemented virtual museum should be a digital representation of a real physical museum. However, real museum maps can be particularly complex and they contain thousands of exhibits. A greater complexity (i.e. an increase in the number of graph nodes) implies that more computational resources are needed to train the algorithm (within acceptable time frames). Since the platform has been developed in my personal computer, it would not have been feasible to use a real representation of a museum. Nevertheless, we consider that the proposed map has an adequate level of complexity for the purpose of an experimental proof-of-concept to validate the platform.

Almost every painting displayed in this museum belongs to the Prado Museum[2], because the museum's online collection is very comprehensive and well-documented. There are thorough descriptions of the artworks that have been used in the virtual museum. More-

[1] Blender Open Source 3D graphics renderer: https://www.blender.org/

[2] Prado Museum (Madrid, Spain) official website: https://www.museodelprado.es/

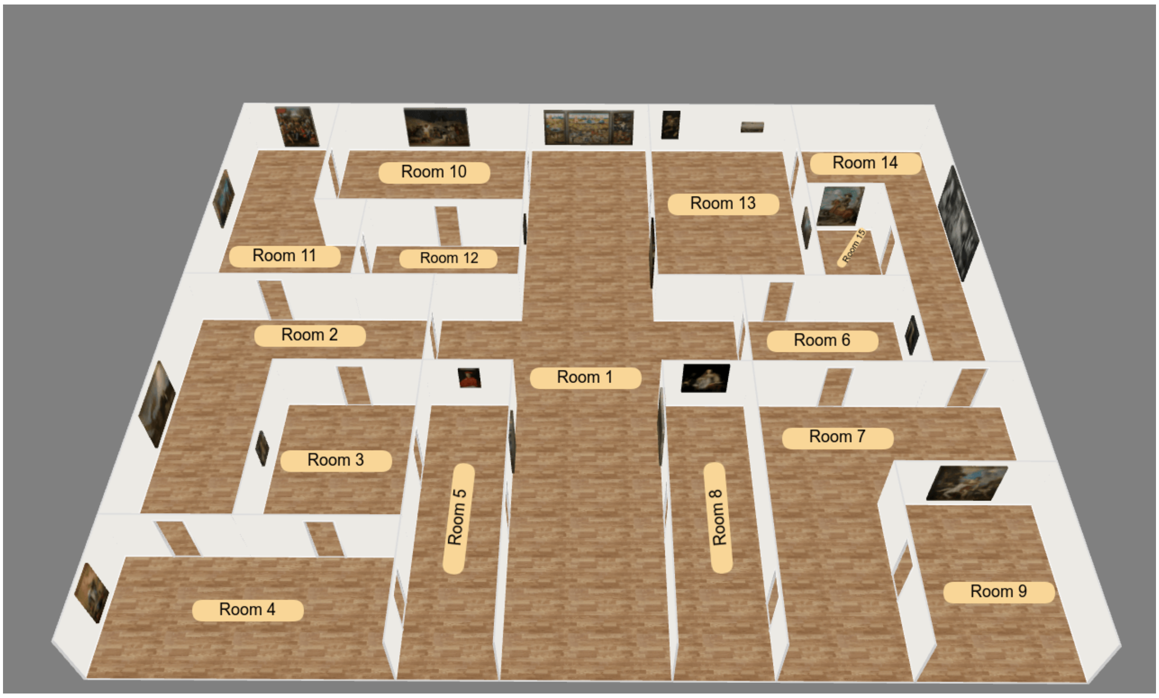


Figure 6.2: Overhead view of the proposed three-dimensional virtual museum scenario.

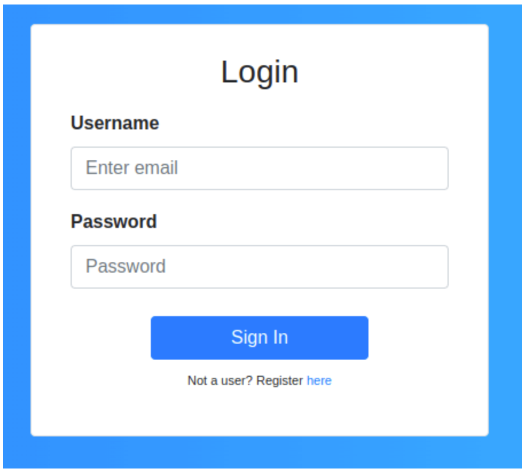
over, and most importantly, the online collection of Prado Museum contains high-resolution images of their artworks, making it ideal for our goal of achieving a three-dimensional effect on the displayed artworks in the virtual museum.

Although the implemented virtual museum does not represent a real-world museum that users might already know of, the artworks that have been used in the virtual museum are all amongst the most renowned paintings in the Prado Museum. The objective is to offer a scenario where users feel familiarised with the vast majority of the exhibits included, despite being unacquainted with the layout.

6.2 Navigation in the Virtual Museum

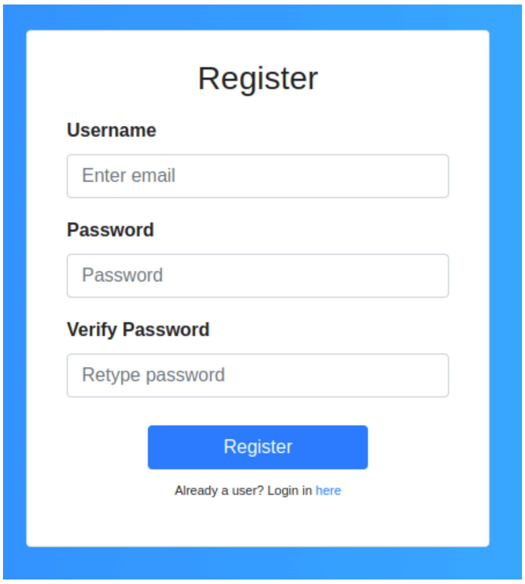
In the previous Section an overhead view of the rendered museum (without HUD) was given, but that view is not how a user would see it. This Section will show how visitors see the museum and the elements of interaction in it.

First, when users open their browsers to access the virtual museum, they must log in. If it is their first time they must register to create a new user account. This registration and login process has been implemented to provide a minimum access control layer to the virtual museum. Hence, the registration form is very simple, as for the purpose and scope



Not a user? Register here

Figure 6.3: Screenshot of virtual museum's login screen.



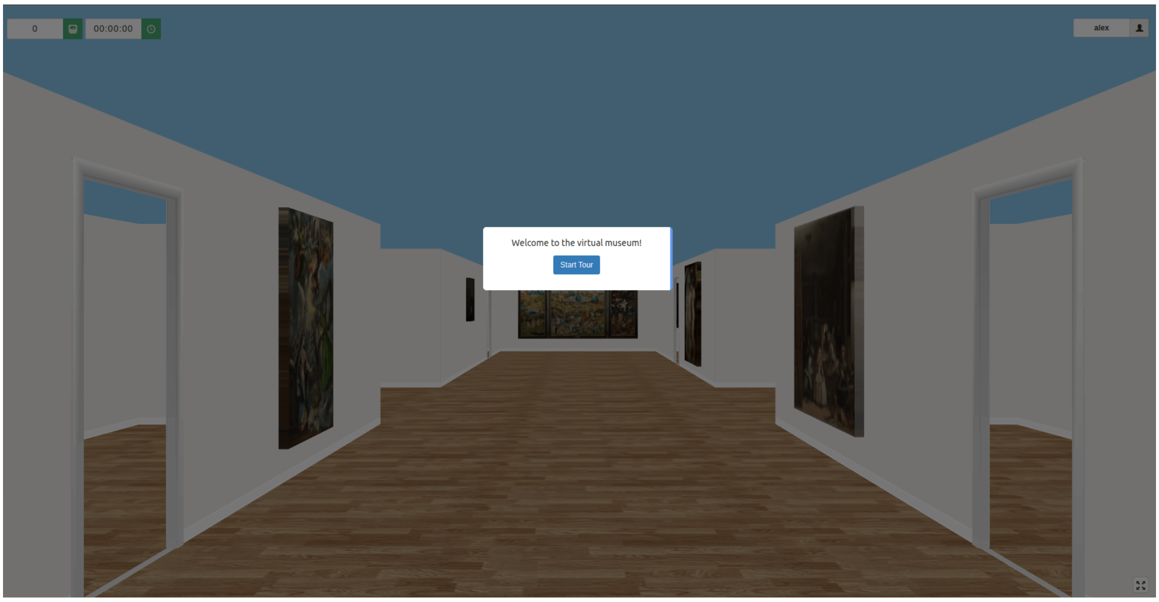
Already a user? Login in here

Figure 6.4: Screenshot of virtual museum's registration screen.

of the project there is no need to store any personal data. Figure 6.3 and Figure 6.4 display screenshots of the login and registration interfaces of the virtual museum.

Once a visitor has successfully logged in, they will be redirected to the virtual museum view. Just before starting the tour there is a pop-up window welcoming users and prompting a '*Start Tour*' button (refer to Figure 6.5). At this point, users will only be able to commence their visit upon clicking this button.

The timer starts running as soon as the tour commences. The timer is displayed to the user at the top left side of the screen, together with their score. These are gamification elements to motivate and engage visitors. Points are awarded when an artwork is visited and deducted for every step taken. This feature has been added to encourage visitors to think the best way of going from one point to the other, so that routes are efficient, and the iML algorithm learns better.



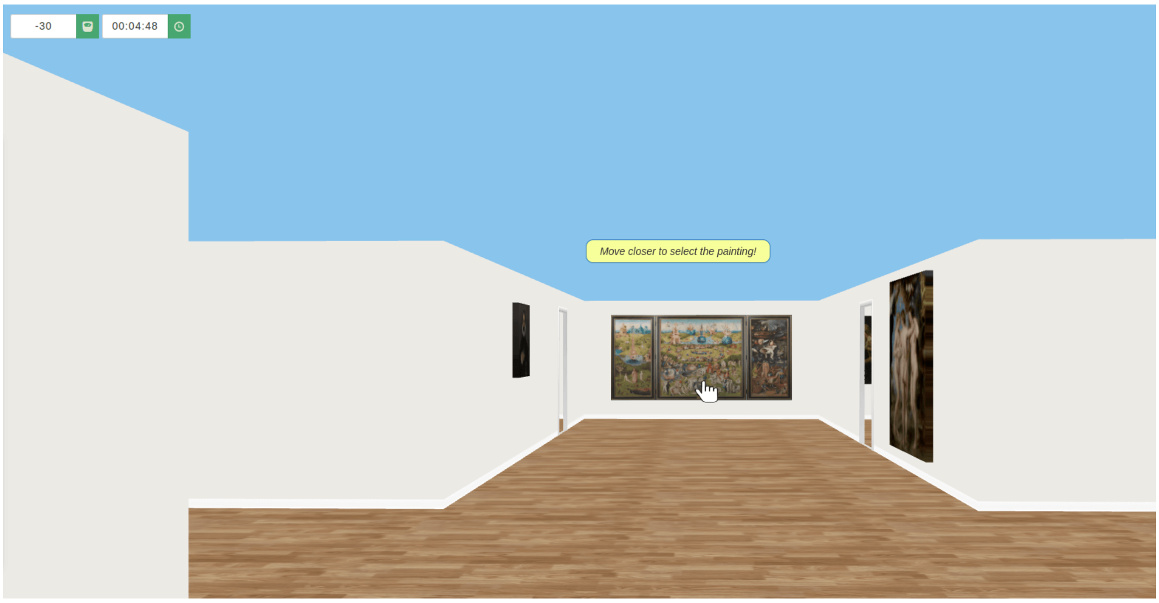
Welcome to the virtual museum!

Figure 6.5: Virtual museum welcome screen.

As in a real museum, visiting an exhibit implies that people get close to them to adequately contemplate them. Therefore, this behaviour has also been implemented into this virtual museum. Artworks glow when the mouse is hovered over them, letting users know that they can interact with them. If a users clicks on a painting that is too far away, a floating message appears with the following text: "*Move closer to select the painting*"; thus the interaction is blocked. Figure 6.6 displays a screenshot of the virtual museum showing the aforementioned message after clicking on the renowned *Garden of Earthly Delights* painting and being far away. Additionally, it can be seen how the timer is reflecting the total tour time and that the total score is negative because the user has moved but no paintings have been visited yet, so still no points have been awarded.

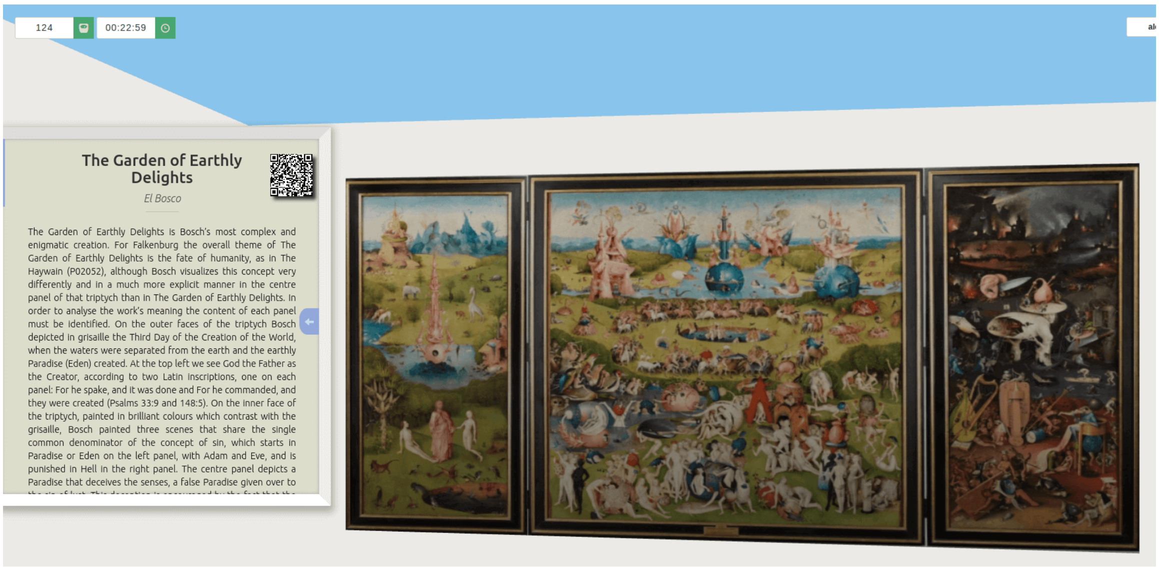
When the visitor comes in close range of an exhibit, they will be able to interact with it. This is portrayed in Figure 6.7. The user has now moved close to the *Garden of Earthly Delights* painting and they have been able to interact with it. Upon clicking on the exhibit, a white-framed information pane slides in from the left of the screen. It presents information about the exhibit, concretely its name, author and a full description so that a more complete and immersive experience is offered. Points have been awarded for this interaction and hence the score is now positive.

Also, it is worth highlighting that a QR code is displayed. By scanning this QR code with the *Smart Object Gamification App*, users will be able to access the Trivia Quiz of automatically generated questions. This is part of the gamification system of the iML routing platform, concretely belonging to the subsystem developed as part of my end of degree project and that has now been expanded and integrated into this project, as described in Chapter 5.



Move closer to select the painting!

Figure 6.6: Warning displayed to users in the virtual museum when they click on an exhibit that is out of their interaction range.



The Garden of Earthly Delights

El Bosco

The Garden of Earthly Delights is Bosch's most complex and enigmatic creation. For Falkenburg the overall theme of The Garden of Earthly Delights is the fate of humanity, as in The Haywain (P02052), although Bosch visualizes this concept very differently and in a much more explicit manner in the centre panel of that triptych than in The Garden of Earthly Delights. In order to analyse the work's meaning the content of each panel must be identified. On the outer faces of the triptych Bosch depicted in grisaille the Third Day of the Creation of the World, when the waters were separated from the earth and the earthly Paradise (Eden) created. At the top left we see God the Father as the Creator, according to two Latin inscriptions, one on each panel: For he spake, and it was done and For he commanded, and they were created (Psalms 33:9 and 148:5). On the inner face of the triptych, painted in brilliant colours which contrast with the grisaille, Bosch painted three scenes that share the single common denominator of the concept of sin, which starts in Paradise or Eden on the left panel, with Adam and Eve, and is punished in Hell in the right panel. The centre panel depicts a Paradise that deceives the senses, a false Paradise given over to

Figure 6.7: Interaction with an exhibit in the virtual museum.

6.3 Virtual Museum Implementation Details

Actions performed by the visitor through the front-end visualisation module often require to be verified against the simulation controller. This is because the aforementioned component must know at every moment the state of the visitor: current position; current room; paintings watched, and other metrics required by the interactive machine learning subsystem, which will be described in subsequent sections.

Although the virtual museum view elegantly represents a 3D space with walls, doors, and paintings, RAMEN was devised to just be an enhanced visualisation component for agent-based simulations, thus delegating custom logic to the actual modelling framework. Therefore, if no further adaptations were applied, a visitor would be able to walk through walls, which is undesirable behaviour for the proposed simulation.

In other cases such as the occupancy simulation scenario -also employing RAMEN for visualisation- defined in [136], controlling collisions with walls is not necessary, since agent movement is dictated by an implementation of the A* algorithm[3]. However, in the case at hand, the human agent is actually controlled by real users, so there must be total freedom of movement. This factor makes it imperative to determine at all times the validity of the step that the visitor is taking.

For the aforementioned reasons, obstacle detection has been developed outside of RAMEN and within the simulation controller. Shapely [130] has been used to offload the elaboration of the logic to perform geometrical calculations, as this package puts a wide range of computational geometry utilities at developers' disposal.

Therefore, each time a user wants to move, RAMEN must make a request to the simulation controller (via REST API), which in turn will verify whether such movement is valid. If the response is affirmative, the visitor will advance to the desired position; otherwise, the visitor will stay at the last valid position. To be more specific, visitors can use movement keys and camera rotations to travel the museum, analogously to three-dimensional games.

Verification of movements involves several checks and calculations per request. The first to be made is to compute the visitor's next position. Once the new coordinate has been obtained, the intersections with other floorplan elements are checked. If this new position implies that the visitor is colliding with a wall, this movement will be rendered invalid, and the controller will instruct the visualisation module to block this new position. If, in contrast, no wall collisions are detected, the visitor will be able to advance.

[3] https://en.wikipedia.org/wiki/A*_search_algorithm

Figure 6.8 depicts the communication sequence between the front-end visualisation module and the simulation controller each time a visitor presses a movement key. In the example provided in the Figure, when a user presses the *'A'* movement key the front-end visualisation component must translate this to a valid direction, 'left' in this case. This is sent together with the camera rotation angle (radians) in the request payload, as the simulation controller requires these two items to correctly calculate the next position. A detailed explanation of the movement mapping procedure will be provided in Section 6.3.1.

Additionally, although only wall intersections have been outlined in the obstacle detection phase, there are other factors to bear in mind when validating movement. Section 6.3.2 will expand on how movement verification is performed.

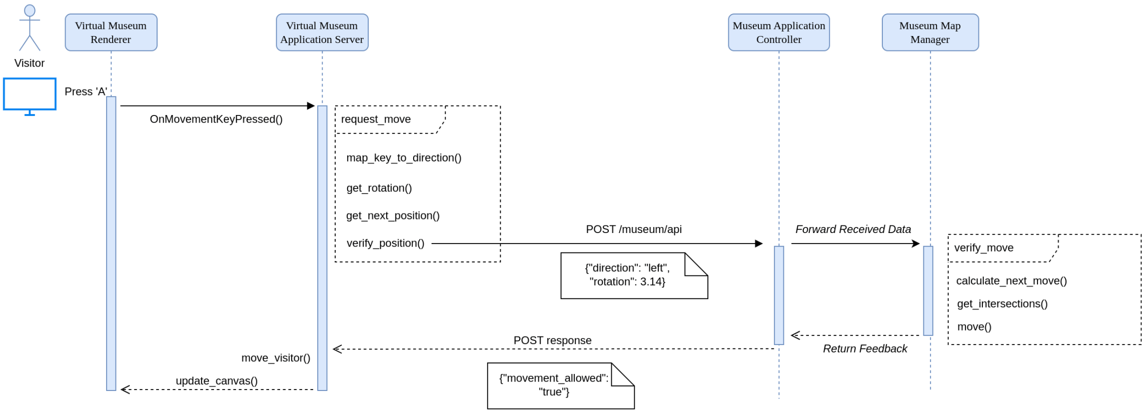


Figure 6.8: Communication sequence used to validate visitor movement.

6.3.1 Movement Mapping

The virtual museum is based on a *html* canvas element which is defined by a left-handed Cartesian coordinate system. This means that the origin is in the top left corner and the x-axis extends to the right of the page (as in the right-handed Cartesian coordinate system). However, in this case, the y-axis runs down the screen instead of extending upward.

In the case of this visualisation module, the museum has been placed in such way that $-\vec{y}$ is the north reference (any positive movement is considered towards $-\vec{y}$) in the left-handed Cartesian coordinate system. According to three-dimensional games terminology, the museum can be considered the world space [137], because it is a global coordinate space used as reference to describe other coordinate spaces - as it will be shortly explained-. Figure 6.9 depicts a snapshot of the starting point of the tour and displays the axes of the aforementioned world space.

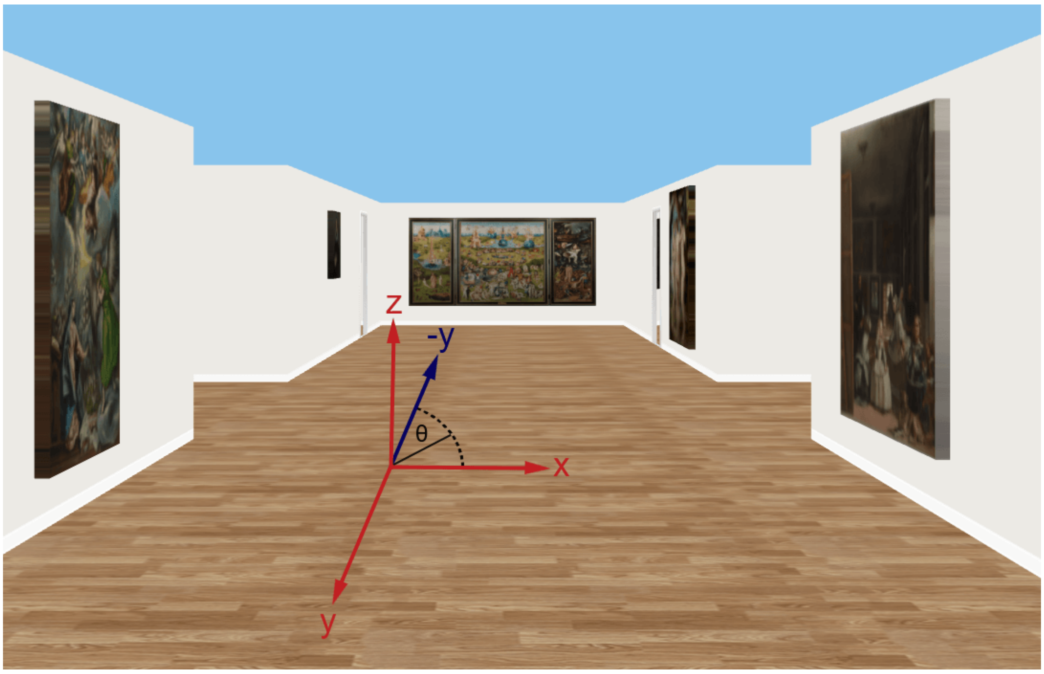


Figure 6.9: Coordinate system in the virtual museum.

Visitors can travel along the world space by means of the typical WASD [138] key bindings, which can be expressed as direction vectors with respect to their perceived coordinate system. Table 6.1 shows the translation between the available movement keys and the directions they represent.

With these four available movements, visitors could explore the whole museum map. However, the experience would be far from satisfactory because in their first-person perspective they would always be looking towards a fixed direction, $-\vec{y}$). Hence, allowing visitors to turn and adjust their view will improve their visual three-dimensional experience [139]. In other words, the camera space [137] must also be factored in so that users can modify the orientation of what they are looking at.

Referring back to Figure 6.9, there is an angle portrayed, (θ), that represents the camera rotation. This has significant implications in the visitor's perceived coordinate system, because the new direction faced will now be the visitor's north, i.e. forward. In other words, regardless of camera rotations, movement keys will still represent the same instruction (see Table 6.1). From the previous statement it can be inferred that rotating the observer's view by an angle θ essentially means applying a rotation transformation of the camera coordinate space by that same angle.

For example, factoring out rotation $(\theta = 0)$, if the visitor wishes to move forward, they will press '*W*'. It was defined that in the observer's (camera) space this movement represents

Key	Movement Instruction	Direction Vector
W	Forwards	(0,1)
A	Left	(-1,0)
S	Backwards	(0,-1)
D	Right	(1,0)

Table 6.1: Visitor movement mappings with respect to their perceived coordinate system

positive movement, so $\vec{d}$ = (0,1) with respect to the camera space, which has been defined as a right-handed Cartesian coordinate system. However, the museum coordinate space is left-handed, so in that space the direction vector translates to $\vec{d}'$ = (0,-1).

If the visitor rotates the camera 90 degrees to the right ($\theta = \pi/4\, rad$), they will now be facing $+\vec{x}$. Therefore, if they press *'W'* again, they will perceive that they are moving forward, although with respect to the museum grid they are moving right, so $\vec{d}''$ = (1,0).

The described behaviour can be more conveniently and formally expressed as a linear transformation representing the desired unit direction vectors of movement in the museum space (see Expression 6.1):

Let S: $\mathbb{R}^2 \rightarrow \mathbb{R}^2$ be a linear transformation defined for any $(x, y) \in \mathbb{R}^2$ as:

$$S((x,y)) = (xcos\beta + ysin\beta, xsin\beta - ycos\beta) \qquad \beta \in [0, 2\pi) \tag{6.1}$$

These equations can be obtained by applying the following transformations to the vectors defined in Table 6.1: *1) rotation around origin* and *2) reflection over x-axis*. It is necessary to apply these transformations so that the camera space can be mapped to the global space, since all internal calculations (including MMAS) use as reference the museum coordinate system. Proof for this derivation is provided below:

First, the linear transformation matrix pertaining to a rotation around the origin will be derived. Let α be the clockwise angle defined between a vector $\vec{r}$ and the y-axis. Let β be the angle formed between $\vec{r}$ and vector $\vec{r}'$. Both vectors have the same modulus ρ (refer to Figure 6.10 for a graphical representation).

These vectors can be expressed in terms of their modulus ρ, and the angles α and β, as shown in Equation 6.2.

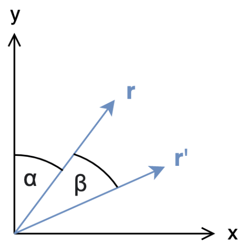


Figure 6.10: Representation of the rotation of a vector

$$\vec{r} = \begin{pmatrix} \rho sin(\alpha) \\ \\ \rho cos(\alpha) \end{pmatrix} \qquad \vec{r}' = \begin{pmatrix} \rho sin(\alpha+\beta) \\ \\ \rho cos(\alpha+\beta) \end{pmatrix} \qquad \alpha, \beta \in [0, 2\pi) \tag{6.2}$$

By further expanding $\vec{r}'$ and taking into consideration that $x = \rho sin(\alpha), y = \rho cos(\alpha)$, we obtain the transformation matrix $\boldsymbol{R}$ dictating the rotation of a vector by a clockwise angle β, as expressed in Equation 6.3.

$$\vec{r}' = \begin{pmatrix} \rho sin(\alpha)cos(\beta) + \rho cos(\alpha)sin(\beta) \\ \\ \rho cos(\alpha)cos(\beta) - \rho sin(\alpha)sin(\beta) \end{pmatrix} = \begin{pmatrix} cos(\beta) & sin(\beta) \\ \\ -sin(\beta) & cos(\beta) \end{pmatrix} \begin{pmatrix} x \\ \\ y \end{pmatrix} = R \begin{pmatrix} x \\ \\ y \end{pmatrix} \tag{6.3}$$

Second, proof will be provided for the linear transformation matrix corresponding to a reflection on the x-axis. A possible approach could consist in obtaining the generic 2x2 reflection matrix for any straight line in the xy-plane and then substituting for y=0. However, since we are only interested in the x-axis reflection transformation, the generic demonstration is beyond the scope of this project. It is sufficient to obtain the transformation matrix only for the required case.

Reflecting a vector on y=0 results in a vector with the same abscissa component, while the coordinate component is multiplied by a factor of -1. This can be more formally expressed as shown in Expression 6.4:

Let F: $\mathbb{R}^2 \rightarrow \mathbb{R}^2$ be a linear transformation defined as:

$$F((x, y)) = (x, \text{-}y) \tag{6.4}$$

Applying this transformation to the canonical basis of $\mathbb{R}^2$, $\{\vec{v_1}, \vec{v_2}\} = \{(1,0),(0,1)\}$

results in:

$$F((1,0)) = (1,0)$$
$$F((0,1)) = (0,-1)$$

Thus, the canonical matrix of F, named T, is provided in Expression 6.5:

$$\boldsymbol{T} = [F(\vec{v_1})F(\vec{v_2})] = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{6.5}$$

Third, in order to arrive at the final transformation matrix used to map movement in the museum space, we must compute the composition of the two aforementioned transformations: rotation around origin (denoted by function r) followed by reflection on the x-axis (denoted by function t), for any $\vec{v} \in \mathbb{R}^2$:

$$\mathrm{t}(\mathrm{r}(\vec{v}))) = \mathrm{t}(\boldsymbol{R}(\vec{v})) = \boldsymbol{TR}\vec{v}$$

$$\mathbf{S} = \mathbf{TR} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} cos(\beta) & sin(\beta) \\ -sin(\beta) & cos(\beta) \end{pmatrix} = \begin{pmatrix} cos(\beta) & sin(\beta) \\ sin(\beta) & -cos(\beta) \end{pmatrix} \tag{6.6}$$

Therefore, multiplying $\vec{v}$ by matrix $\boldsymbol{S}$ yields the required transformation vector, which is shown in Expression 6.7:

$$\vec{s} = \mathbf{S}\vec{v} = \begin{pmatrix} cos(\beta) & sin(\beta) \\ sin(\beta) & -cos(\beta) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} xcos\beta + ysin\beta \\ xsin\beta - ycos\beta \end{pmatrix} \qquad Q.E.D \tag{6.7}$$

It must be highlighted that in this case, from this vector transformation a unit direction vector is obtained, since movement keys are mapped to the canonical unit direction vectors (refer back to Table 6.1). Hence, without any additional modifications, these movement changes are quite small and it would visually not look smooth. Therefore, to offer a more fluid experience, the virtual museum scenario allows customising movement speed. That is, speed is nothing else than a scalar quantity that increases the magnitude of the movement direction vector. Taking this into consideration results in the final expression (Equation 6.8) to calculate the next position of a visitor (x_2, y_2) from their previous position (x_1, y_1) and the derived unit direction vector $(\overrightarrow{x, y})$ multiplied by the speed of movement k:

$$
\begin{gathered}
x_1 + k(xcos\beta + ysin\beta) = x_2 \\
y_1 + k(xsin\beta - ycos\beta) = y_2 \\
(x_1, y_1), (x_2, y_2) \in I\!R^2; \quad k \in I\!R_{>0}; \quad \alpha, \beta \in [0, 2\pi); \\
(\vec{x, y}) = \{(0,1), (1,0), (0,-1), (-1,0)\}
\end{gathered} \tag{6.8}
$$

6.3.2 Virtual Museum: Movement Verification

The previous section delved into the calculation of the visitor's next position given their choice of movement, but did not contemplate the validity of the new position. Therefore, this section describes the methodology used to check whether a given position (x_2, y_2) can be occupied by a visitor and the consequences derived from this movement.

The proposed virtual museum scenario consists of three main environment elements: walls, doors, and exhibits. Walls are spatial boundary elements that are used to delimit the different rooms in the museum. They are considered physical obstacles as opposed to doors, which allow movement between rooms. Also, visitors may interact with exhibits if they are in close proximity to them.

With the objective to assess the validity of the next movement of a visitor, it must be checked whether the point or set of points that describe the location of any of the aforementioned elements coincides with the visitor's new target position. To achieve this in a faster and more efficient manner, only the subset of coordinates enclosed by the room the visitor is currently in is studied. With the aid of Shapely library (refer to Chapter 4 for details on this Python package), geometrical calculations can be easily performed to determine whether the visitor's next position intersects or not with a wall, a door, or an exhibit.

Shapely's *"within"*[4] predicate assesses that the new position coordinate is completely contained in the room, without intersecting with its boundaries (i.e., room walls). On the one hand, a positive response to this calculation implies that there is no contact with any of the aforementioned environment elements. Therefore, this movement will be allowed. On the other hand, if the next move is not within the polygon, then it means that it is intersecting with the room's boundaries. At first glance, one could think that in this case the movement will be rendered invalid because boundaries are walls. However, it must be borne in mind that doors and exhibits lie in these boundaries, meaning that a given position

[4] Documentation of Shapely's *within* binary predicate: https://shapely.readthedocs.io/en/stable/manual.html#object.within

can be both in the subset of points defining a wall and a door (or exhibit).

To establish which element the new target position intersects with at the room's boundary, it must first be determined whether it is a door or an exhibit. This is because these elements take precedence over walls. If the contact is with the position of a door, the movement is valid, and additionally the system must calculate the new room the visitor is going to. Also, should the new position be in close proximity to an exhibit, interaction with the painting must be allowed. This interaction consists in allowing visitors to click on the painting, and a description about the artwork is displayed, together with a QR code granting access to automatically generated questions about the exhibit (served by the smart object gamification platform developed as part of my end-of-degree project).

Otherwise, the remaining case to be handled is the one in which no door or exhibit is found to intersect with the new position at the boundary. Therefore, this directly implies that the collision is with a wall, so the movement is not valid.

Finally, every valid movement is stored in the system, so it can be fed into the module that computes the iterations of the MMAS algorithm. Before concluding this part, Figure 6.11 is presented to the reader as a means of summarising the movement validation decision process described.

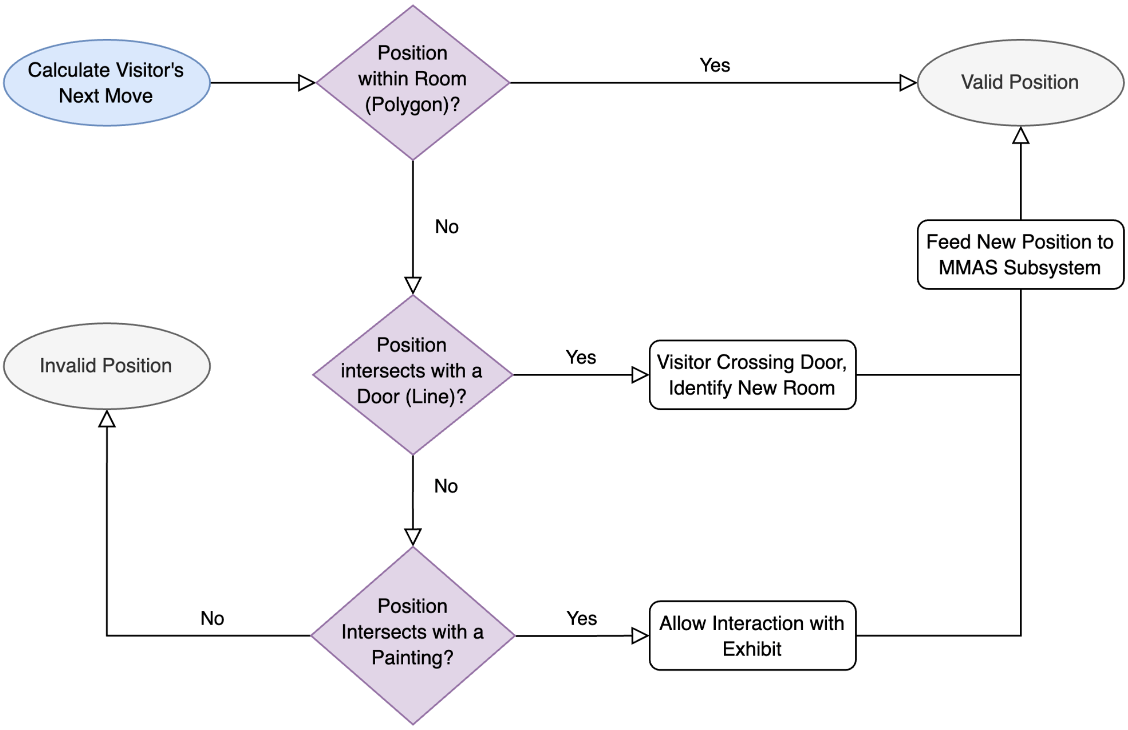

Calculate Visitor's Next Move
Position within Room (Polygon)?
Yes
Valid Position
No
Feed New Position to MMAS Subsystem
Invalid Position
Position intersects with a Door (Line)?
Yes
Visitor Crossing Door, Identify New Room
No
No
Position Intersects with a Painting?
Yes
Allow Interaction with Exhibit

Figure 6.11: Decision and action process to validate the next move of a visitor

CHAPTER 7

Evaluation

This Chapter delves into the evaluation phase of the platform. Concretely, the MMAS-based iML algorithm will be validated. A calibration process against known TSP solutions is introduced first. Then, the Chapter explores the evaluation of the implemented algorithm in the virtual museum.

7.1 MMAS Baseline Validation

In order to verify that the MMAS algorithm implementation has been correct and the results obtained are consistent with those in bodies of research [26], a validation process against four reference TSP instances has been carried out.

The optimum solution to these TSP can be found in the *Tsplib95* project library [140], where contributions can be made to upload TSP instances and their proven optimun solution. They are considered the benchmark results and therefore it is convenient to have some of them at our disposal for this validation.

TSP Reference Instance	Optimum (Best Minimum Cost)	MMAS Best Average Minimum Cost	% Deviation	Avg Iterations to yield MMAS Best Cost
ulysses16	6859	6859.0	0.00%	120
gr24	1272	1272.4	0.03%	283
berlin52	7542	7543.5	0.02%	384
eil76	538	539.1	0.20%	441

Table 7.1: Calibration of MMAS implementation with TSP reference instances

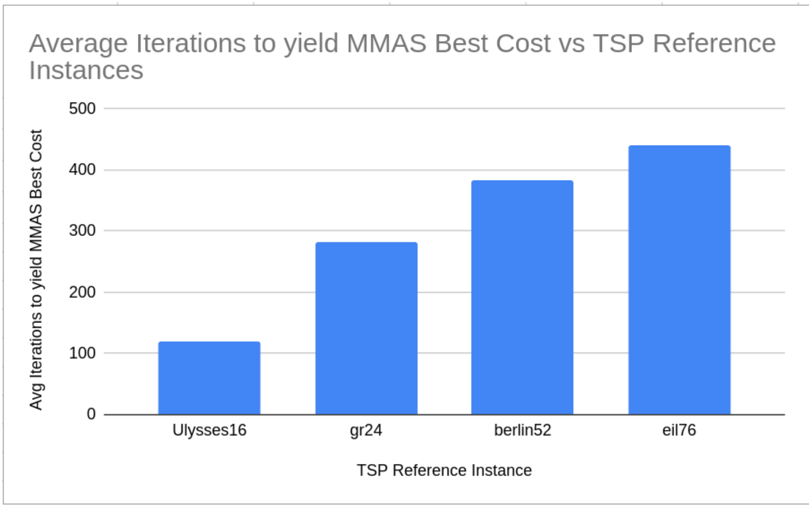


Figure 7.1: Average iteration to yield MMAS best solutions in the chosen TSP reference instances.

These calibration process has been done by setting N = 1000 iterations averaged over 10 runs. Table 7.1 reflects the calibration results obtained against four reference TSP problems. Figure 7.1 presents the average iterations needed by the MMAS implementation done in this project to obtain the best cost. Figure 7.2 illustrates the percentage deviation from the optimum solution of the reference instances. Overall, the results obtained are consistent with the expected MMAS performance for the number of nodes in each case. The algorithm needs more iterations on average to obtain its best solution as the number of nodes increase.

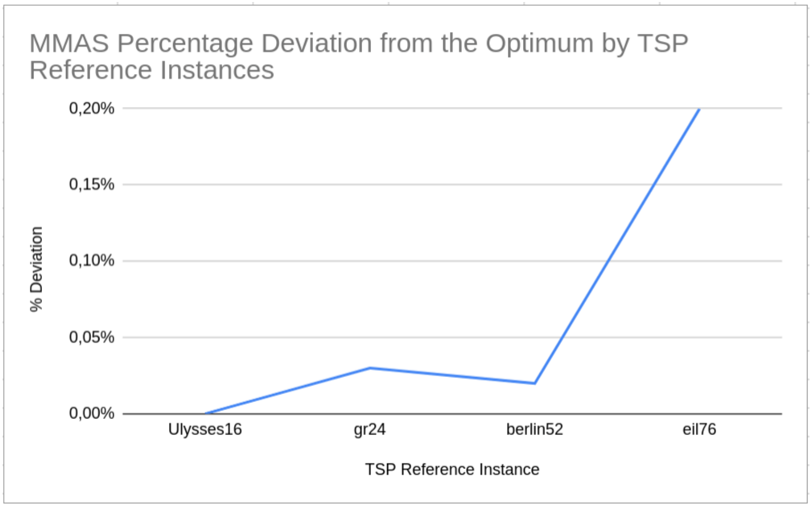


Figure 7.2: Percentage deviation from the optimum solution in the selected TSP instances.

This calibration, of course, allows us to validate our implementation of MMAS against the classic distance objective. Based on this baseline validation, we can thus consider the algorithm ready for receiving external human input.

7.2 Interactive MMAS (iML) Evaluation in the Virtual Museum

In this section we will delve into the training and analysis of the iML model. For this purpose, we will first explain the evaluation methodology and then describe the required setup and configuration parameters used to initialise the instances. Subsequently, the evaluation scenarios will be executed and assessed.

7.2.1 Scenarios Setup

The proposed evaluation procedure consists of two main stages: i) model training and ii) model testing.

In the model training phase, the aim is to obtain two trained versions of the model. One will correspond to the traditional application of MMAS - classic ML - in the museum graph; the other will be generated using the iML approach by providing human feedback into the learning loop. Having these two versions is required in order to compare the outcomes of iML with those of the original method.

It must be remembered that in this project, the primary interest does not lie in identifying the algorithm's best route in terms of minimising the total distance travelled. Instead, the focus will be shifted towards observing how the distribution and concentration of pheromones evolves throughout the iterations, aiming to see the influence of human feedback in this regard.

In order to assess the final pheromone concentrations more fairly, the initial configurations in both cases will be equivalent. Specifically, the parameters employed to initialise the algorithm will be the same, as well as the total number of iterations for which the algorithm is trained. These initialisation values are reflected in Table 7.2.

I_{start}	$I_{interactive}$	**phi multiplier**	λ	μ	τ	ρ_{best}	M_{ants}	**Start Node**	**Start Room**
15	5	150	**1**	**2**	**0.02**	**0.05**	**N**	**D1-1 (Entrance)**	**Room 1**

Table 7.2: Settings of MMAS typical parameters

With respect to the parameter representing the number ants, it has been set to a number equal to the total number of nodes, which is N=47 in this context. Unlike a classic TSP problem, the starting point is not random here. Users will start their tour as soon as they enter the museum. Hence, it is reasonable to construct tours taking into account this fixed starting position, which is the entrance door, identified as *D1-1*.

Regarding the consideration of human feedback into the model, a user navigates through the virtual museum, behaving as they would in a real museum. In the background, the pheromone trail in edges travelled by users will be boosted, significantly increasing the probability of selecting these edges in the algorithm's state transition rule. Each time a user goes through a door or interacts with an exhibit, the amount of pheromone on that edge will be multiplied by a factor of **1500**. While this amount may seem large, the value is justified as the pheromone concentration values generally fall within rather low ranges because of the graph's characteristics ($10^-5 - 10^-7$). After this pheromone update, **5** more iterations are executed. Additionally, an initial computation of **15** iterations occurs before the user starts their route.

In the second stage - model testing -, the two versions of the models (ML and iML) will be evaluated and its outcomes compared. At this step, each model will posses a pheromone distribution derived from all the iterations executed during their training phases. Guided by these learning experiences, the models in each case will be used to construct a route containing a selection of target artworks. The proposed routes will then be assessed through a satisfaction metric to provide a comparative analysis.

In order to better understand the information about the proposed and generated routes in the following sections, a two-dimensional map of the museum with labelled nodes is provided in Figure 7.3.

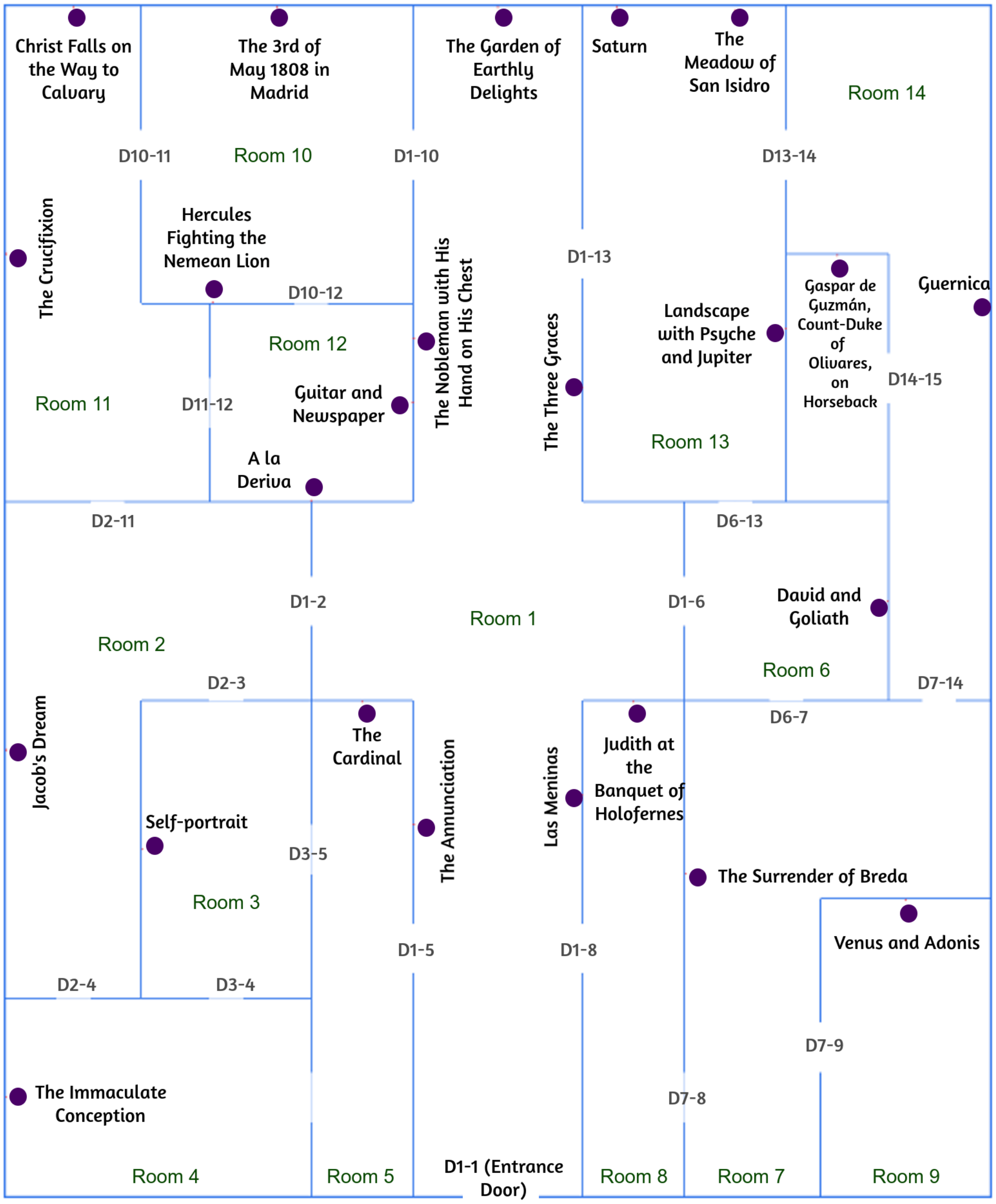


Figure 7.3: Two-dimensional representation of the museum map with labelled nodes.

7.2.2 Model Testing Stage

This section covers the training of the iML-boosted and the classic MMAS models. First, the training of the algorithm without human input will be analysed. Afterwards, the training phase with the *human-in-the-loop* will executed. With respect to the iML approach, the procedure to obtain routes for model feedback will be comprehensively described.

In both cases, the model learning phase will last $\mathbf{I = 3000}$ iterations. This value is derived from the number of iterations that are executed based on the human feedback provided, as is explained in Section 7.2.2.2.

7.2.2.1 Classic MMAS Model Training

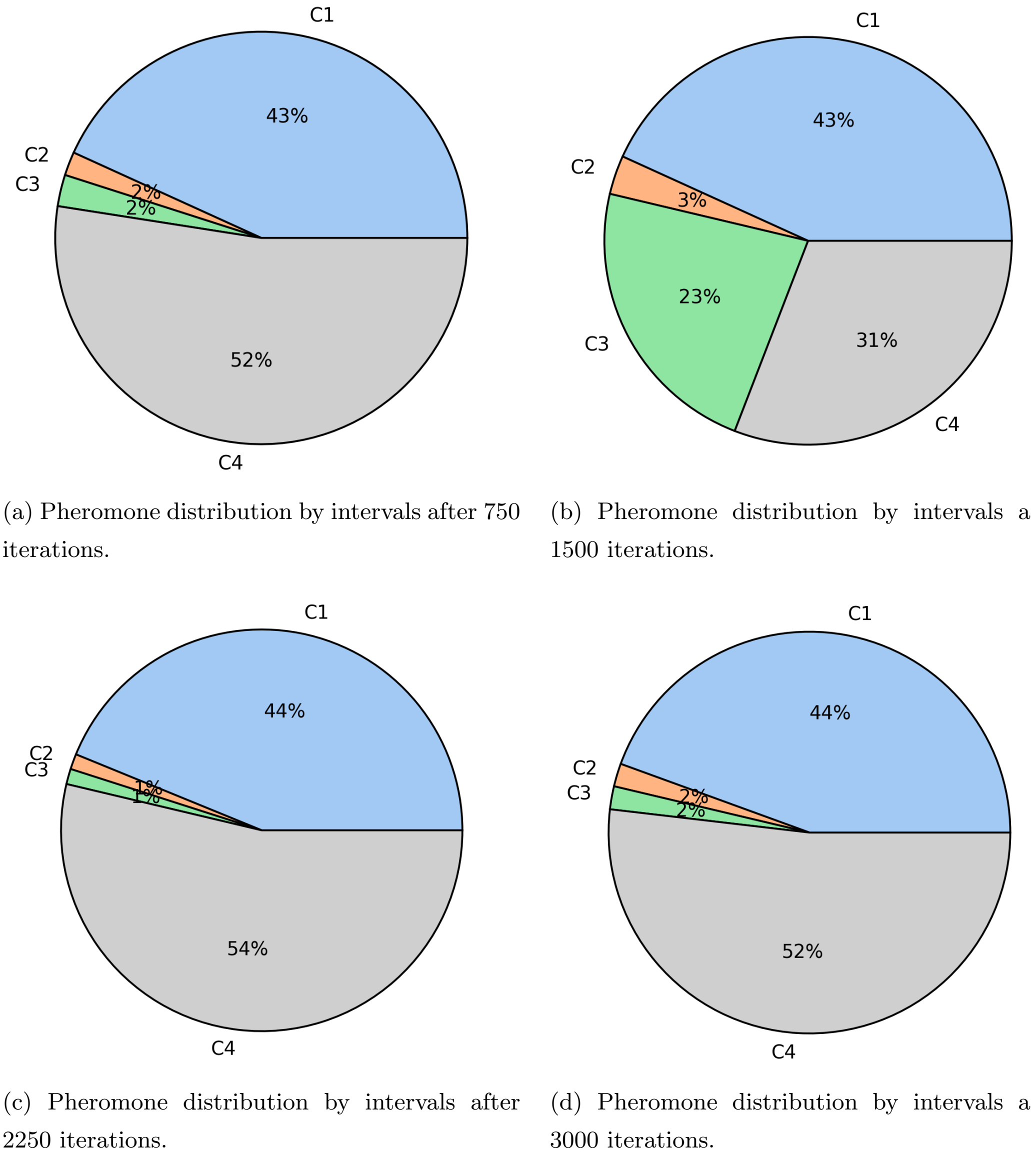
In this Section we will delve into training the classic version of the model, this is, without human input. During the training phase we will analyse the evolution of the pheromone trails. In order to facilitate the analysis, and exploiting the fact that pheromones are upper and lower bounded in MMAS, they will be also classified into different strata. First, we will examine the variation in pheromone concentrations within these specified categories as iterations progress. Afterwards, we will inspect the distribution of pheromones across their range and across iterations. Finally, we will offer a comparative overview of pheromone trails over the course of the training.

With respect to the aforementioned categories, we have classified pheromones into four different intervals. Each interval represents the 25% of the the range τ_{min_i} - τ_{max_i} at any given iteration i. Reference to the iteration number is necessary as the maximum and minimum pheromone values are updated at every iteration. These intervals are displayed in Table 7.3. The magnitude of the range at a given iteration, τ_{max_i} - τ_{min_i}, is represented by the symbol $\Delta\tau_{range_i}$.

Interval Name	Pheromone Values (τ)
C1	$\tau_{min_i} \leq \tau \leq \tau_{min_i} + 0.25\Delta\tau_{range_i}$
C2	$\tau_{min_i} + 0.25\Delta\tau_{range_i} < \tau \leq \tau_{min_i} + 0.50\Delta\tau_{range_i}$
C3	$\tau_{min_i} + 0.50\Delta\tau_{range_i} < \tau \leq \tau_{min_i} + 0.75\Delta\tau_{range_i}$
C4	$\tau_{min_i} + 0.75\Delta\tau_{range_i} < \tau \leq \tau_{max_i}$

Table 7.3: Pheromone Intervals

We will now proceed to analyse the evolution of pheromones in this training phase. Since there are 3000 iterations, we have decided to simplify the observations by focusing on four iterations: 750, 1500, 2250 and 3000.



(a) Pheromone distribution by intervals after 750 iterations.

(b) Pheromone distribution by intervals after 1500 iterations.

(c) Pheromone distribution by intervals after 2250 iterations.

(d) Pheromone distribution by intervals after 3000 iterations.

Figure 7.4: Pheromone distribution by intervals at selected iterations during classic MMAS model training.

In regards to the pheromone intervals, Figure 7.4 portrays them at the selected iterations. We can observe that intervals C1 and C4 tend to be predominant, usually containing more than 90% of the pheromones. It can be seen that interval C4 is the most frequent interval. In contrast, the middle intervals, C3 and C4, have an insignificant presence, typically less than 4% in total. A possible explanation for these observations is that, since this implementation of MMAS uses the iteration-best solution to perform pheromone update, the search space is broadened. This means that it is more likely that the algorithm may select paths with less pheromone concentration, therefore boosting them, and making them closer to τ_{max}. Also, due to the nature of the algorithm, those paths with much lower pheromone trail values will not be so likely to be chosen, so their pheromone concentration will be very close to τ_{min}.

It is also worth highlighting the 20% increase in pheromone interval C3 in iteration 1500. The previous explanation also helps towards understanding this notable trait. It appears that at iteration 1500 the best solution found may consist of edges with a relatively low pheromone concentration. This means that by boosting these edges, their pheromone concentrations have likely increased from interval C1 or C2 to interval C3. It seems more likely that it was a little explored path (C1), and that would explain why the extra pheromone added is not enough to place pheromones at interval C4, leaving them at C3.

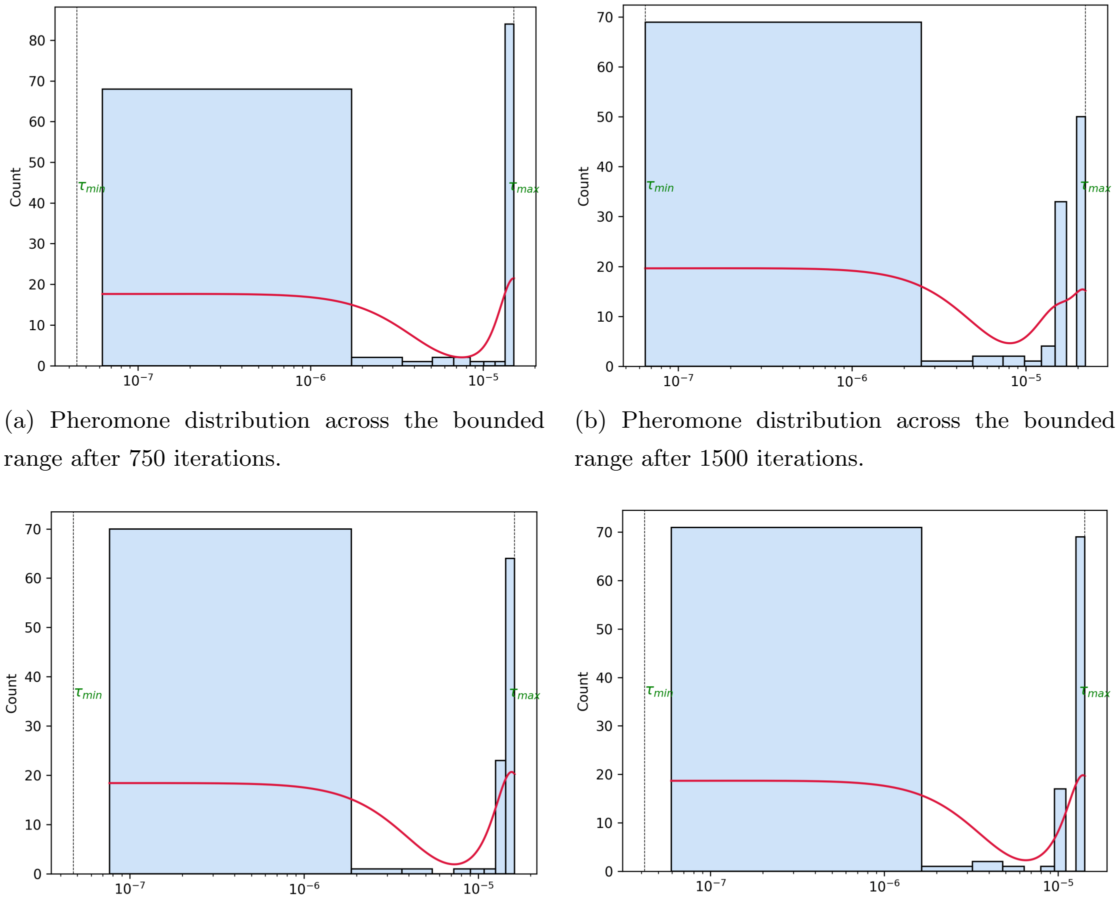
We will delve deeper into the pheromone distributions at each of the analysed iterations. Figure 7.5 is presented to better appreciate how pheromone trails are dispersed throughout the bounded range. This Figure provides more fine-grained details to the observations made in the intervals displayed in Figure 7.4.

We can see that the pheromone values lie between $10^{-8} - 10^{-5}$, so the difference between the maximum and minimum values is three orders of magnitude, thus a logarithmic scale for pheromone values has been chosen in this case to better represent the results. In all four of the displayed iterations, we can distinguish two predominant peaks. There is a high but more widespread concentration of pheromones at the lower end of the range, whereas the pheromones at the higher end of the range are very tightly distributed.

The used evaporation factor parameter, which is 0.02 (please refer back to Table 7.2), can explain the observed patterns for the lower values. This small evaporation factor causes pheromone trails on unexplored paths to gradually decrease with each iteration. However, this very gradual decrease can turn into an exponential decrease (Equation 2.7) if a given edge is not chosen in successive iterations, hence significantly lowering the values. The variations are significantly less for pheromone-boosted edges because the addition rule (Equation 2.7) will only gradually increase them.

The remaining two charts, depicted in Figures 7.6 and 7.7, aim at providing a comparative view of all iterations of interest at once. With respect to Figure 7.6, the pheromone densities portrayed show that the trend is overall quite similar in iterations 750, 2250, and 3000. It clearly shows a bimodal distribution of pheromone values. Also, we can clearly see the fall of the density peak that would correspond to the higher pheromone values (C4) in iteration 1500, as we already observed and explained in previous figures.

Figure 7.7 also helps to show some statistical aspects of the bimodal pheromone distributions that were seen at the different iterations. Given that, as aforementioned, the difference between the maximum and minimum values is three orders of magnitude and that pheromones are bimodally distributed (C1, C4), it is reasonable to see that the median lies within the high end of the range. Nevertheless, it also shows that at 1500 iterations, the median is located at what would be interval C3.



(a) Pheromone distribution across the bounded range after 750 iterations.

(b) Pheromone distribution across the bounded range after 1500 iterations.

(c) Pheromone distribution across the bounded range after 2250 iterations.

(d) Pheromone distribution across the bounded range 3000 iterations.

Figure 7.5: Histograms of pheromone distribution at selected iterations during classic MMAS model training.

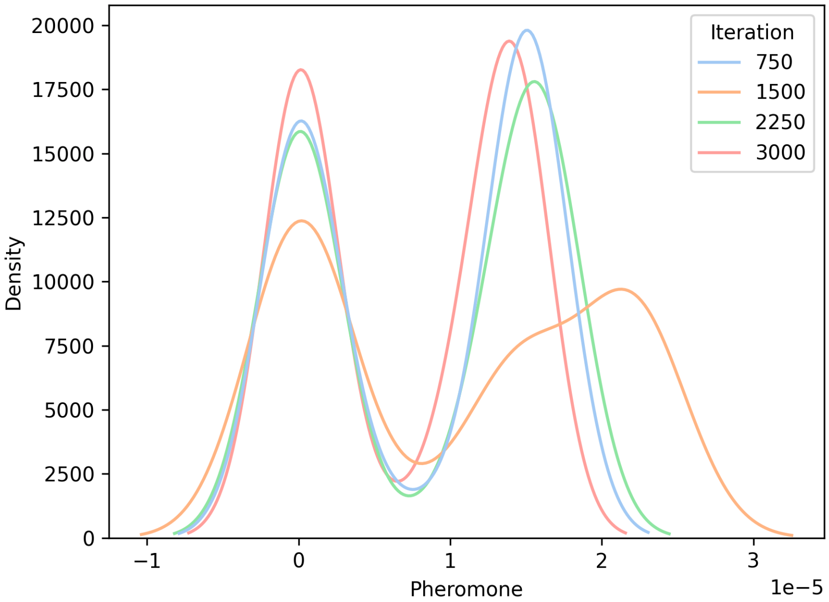


Figure 7.6: Comparison of pheromone densities at the selected iterations.

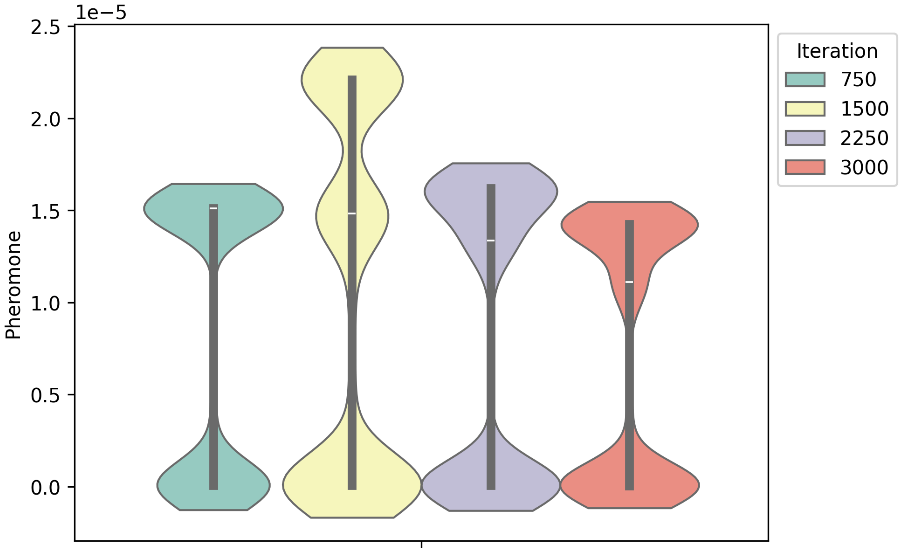


Figure 7.7: Comparison of pheromone value distributions and statistics at the selected iterations.

7.2.2.2 iML-Enhanced MMAS Model Training

For the case of training the model using the iML approach, it was not possible to carry out the training of the model using real users, or at least the number of users required to appreciate behavioural and trajectory patterns in the virtual museum would be too large for the scope of this evaluation. In a real museum, every exhibit is carefully and thoughtfully placed and the greater affluence of people gives rise to observable patterns [141].

Therefore, visitor routes will be simulated for the scenario, backing on existing literature to justify the choice of routes. More precisely, the training phase will be carried out based on four different types of reference routes, which we will consider to be not from four users but hypothetically the four most common routes that could have been observed in the virtual museum. Based on this premise, we can simulate a scenario with several visits and observe visitor patterns, which will facilitate in assessing the learning process of the model.

The choice of the most common routes has been made based on the visitor profiles identified by the research conducted in [11]. The author identifies five visitor roles:

- **Explorer**: these visitors want to learn new information. They look to engage in a process of discovery, and therefore they do not have set goals in the museum. Explorers read the labels in exhibits and take their time to carry out the visit. To better understand this profile, the author compares explorers to shoppers that like wandering around the store viewing items but without any particular element in mind.
- **Facilitator**: the main driver for these visitors is to satisfy the needs of their companions and help maximise their quality of experience. In essence, they tend to adopt more the role a of tour guide navigating the museum. A real example of this type of visitor is shown in the research conducted by the author, where a parent had devised a specific learning agenda for his children.
- **Experience Seeker**: these visitors aim to make memorable museum experiences, and they like getting an overview of the museum rather than a deeper understanding. To achieve this, they choose to focus their time on the most popular exhibits of the museum. Visiting the highlights is one of the main factors for them to deem the visit as satisfactory.
- **Professional/Hobbyist**: these visitors arrive at the museum with a clear goal in mind. They know what the want to see and study, and know how to best get there. However, if they find it relevant enough, they may be occasionally sidetracked by elements outside of their initial goals. The author compares them to shoppers that

head straight to the desired section: they know what they want to buy and tend to ignore everything else.

- **Recharger**: these visitors come to the museum to relax and disconnect. The museum provides a peaceful and quiet environment for them and they are characterised by tending to sit down and spend a lot of time in galleries.

Our initial intention was to obtain five different types of routes, each justified by one of the described profiles. However, the reason why only four types of routes have been identified is because simulating a Recharger's route was challenging due to the nature of their motivations. Their route choices are not directly related to exhibits themselves, making it more difficult to justify a possible choice of artworks to visit.

Hence, considering the aforementioned visitor types, the proposed routes are the following:

1. **Route 1**
    - **Path:** *D1-1 → Las Meninas → The Annunciation → D1-5 → The Cardinal → D3-5 → Self-portrait → D3-4 → The Immaculate Conception → D2-4 → Jacob's Dream → D2-11 → The Crucifixion → Christ falls on the Way to Calvary → D10-11 → Hercules fighting the Nemean Lion → The 3rd of May 1808 in Madrid → D10-12 → Guitar and Newspaper → A la deriva → D10-12 → D1-10 → The Garden of Earthly Delights → The Nobleman with his Hand on his Chest → The Three Graces → D1-13 → Landscape with Psyche and Jupiter → Saturn → The Meadow of San Isidro → D13-14 → Guernica → D14-15 → Gaspar de Guzmán, Count-Duke of Olivares, on Horseback → D14-15 → D7-14 → The Surrender of Breda → D7-8 → Judith at the Banquet of Holofernes → D1-8 → D1-1*
    - **Explorer Route:** considering their motivation to discover and explore the museum, and bearing in mind that the proposed virtual museum scenario is not very large, it may be reasonable to suggest a route that navigates through mostly all of the rooms and visits many exhibits.
2. **Route 2**
    - **Path:** *D1-1 → D1-5 → The Cardinal → D3-5 → Self-portrait → D3-4 → The Immaculate Conception → D2-4 → Jacob's Dream → D2-11 → The Crucifixion → D10-11 → The 3rd of May 1808 in Madrid → D1-10 → The Garden of Earthly Delights → D1-13 → D13-14 → Guernica → D7-14 → D6-7 → D1-6 → The*

Three Graces → *The Nobleman with his Hand on his Chest* → *The Annunciation* → *Las Meninas* → *D1-1*

- **Experience Seeker Route:** the priority in the tailored route is to visit the most popular or renowned elements. Therefore, and remembering that the virtual museum mainly contains artworks from Prado Museum, we have largely based the selection of exhibits in the most recognised pieces of Prado [1].

3. **Route 3**

- **Path:** *D1-1* → *Las Meninas* → *The Annunciation* → *D1-5* → *The Cardinal* → *D3-5* → *Self-portrait* → *D2-3* → *D2-11* → *The Crucifixion* → *Christ falls on the Way to Calvary* → *D10-11* → *Hercules fighting the Nemean Lion* → *The 3rd of May 1808 in Madrid* → *D1-10* → *The Garden of Earthly Delights* → *The Three Graces* → *D1-13* → *Saturn* → *D13-14* → *Guernica* → *D7-14* → *D6-7* → *David with the Head of Goliath* → *D6-7* → *D7-9* → *Venus and Adonis* → *D7-9* → *D7-8* → *D1-8* → *D1-1*

- **Facilitator Route:** this route has been created based on the role of a parent preparing a route for their children. It is reasonable to suppose that one of the main goals is to tailor visits that help them gain general knowledge, so a good starting point is including some of the most renowned artworks (similar to explorer in this sense). Also, it would be interesting to consider as well pieces that represent elements from history and mythology, again with the goal of improving the general knowledge acquired by the children throughout the visit.

4. **Route 4**

- **Path:** *D1-1* → *Las Meninas* → *D1-8* → *D7-8* → *The Surrender of Breda* → *D7-14* → *Guernica* → *D14-15* → *Gaspar de Guzmán, Count-Duke of Olivares, on Horseback* → *D14-15* → *D13-14* → *The Meadow of San Isidro* → *Saturn* → *D1-13* → *The Garden of Earthly Delights* → *D1-10* → *The 3rd of May 1808 in Madrid* → *D1-10* → *D1-1.*

- **Professional/Hobbyist Route:** for this case we have theorised that the visitor is deeply interested in both Goya's and Velázquez's pieces, so the visitor will head mostly straight to them. Only a very few paintings that were not in their original plan drew their attention.

[1] 14 masterpieces from the Museo del Prado: https://www.museodelprado.es/en/whats-on/new/14-masterpieces-from-the-museo-del-prado-in/220f2fe9-beec-44a2-afdb-8c67ff37256c

As a means of summarising and comparing the selection of artworks in each base route, Table 7.4 is presented.

	Route Type				
Artworks	**Route 1**	**Route 2**	**Route 3**	**Route 4**	
The Annunciation	X	X	X		**3**
Las Meninas	X	X	X	X	**4**
The Nobleman with his Hand on his Chest	X	X			**2**
The Three Graces	X	X	X		**3**
The Garden of Earthly Delights	X	X	X	X	**4**
Jacob's Dream	X	X			**2**
Self-portrait	X	X	X		**3**
The Immaculate Conception	X	X			**2**
The Cardinal	X	X	X		**3**
David with the Head of Goliath			X		**1**
The Surrender of Breda	X			X	**2**
Judith at the Banquet of Holofernes	X				**1**
Venus and Adonis			X		**1**
The 3rd of May 1808 in Madrid	X	X	X	X	**4**
Hercules fighting the Nemean Lion	X		X		**2**
The Crucifixion	X	X	X		**3**
Christ falls on the Way to Calvary	X		X		**2**
Guitar and Newspaper	X				**1**
A la deriva	X				**1**
Saturn	X		X		**2**
The Meadow of San Isidro	X			X	**2**
Landscape with Psyche and Jupiter	X				**1**
Guernica	X	X	X	X	**4**
Gaspar de Guzmán, Count-Duke of Olivares, on Horseback	X			X	**2**

Table 7.4: Comparison of artworks visited for each type of reference route used to train the iML-enhanced model.

The studies conducted in [11] do not reflect metrics for the frequency of the different route types, so we will assume an even distribution. This is, we will traing the algorithm with **20** visits, where there will be **5** visits of each type of these routes. It must be noted that this does **not** mean that each type of route will be repeated five times. The listed routes will serve as the reference paths, and each visit will be **slightly different to their respective reference routes** by including **slight variations** of these.

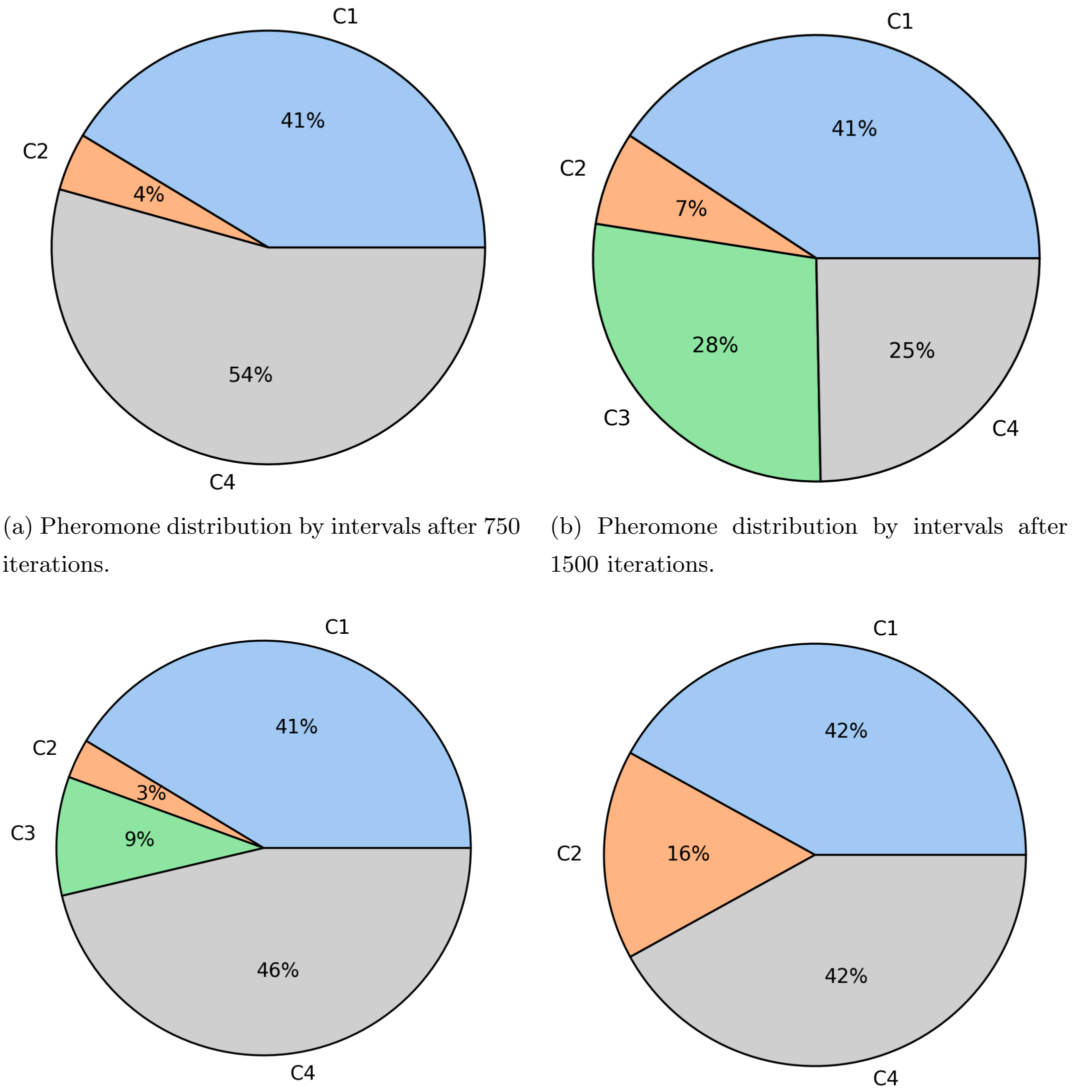
Recalling $I_{interactive}$ and I_{start} from Table 7.2 presented in the evaluation setup section, and taking into account that the reference route types produce 112 edges to add feedback to, this yields an approximate (due to the aforementioned path variations) amount of **3000** iterations.

We will now proceed to analyse the evolution of pheromones in this training phase. It will be carried out analogously to the classic MMAS model, comparing the same four iterations: 750, 1500, 2250, and 3000.

Figure 7.8 portrays the pheromone distribution across the selected intervals (C1,C2,C3 and C4). It can be seen that it resembles the results shown for the classic MMAS model, as categories C1 and C4 tend to be the predominant ones. However, in this case, except for iteration 750, the contributions of intervals C2 and C3 are now not negligible, especially for interval C3, where there are even more pheromones in C3 (28%) than in C4 (25%) in iteration 1500.

The observed predominant C1 and C4 intervals can be explained with the same reasoning as the one provided for the classic MMAS model. Additionally, the greater presence of C2 and C3 occurs because of the human feedback provided. This is because there is a significant pheromone boost to those edges traversed by a visitor, and these edges are not necessarily the “best” choice (in terms of minimising distance), so human input is reinforcing paths that would otherwise remain practically unexplored by the algorithm due to the low probability of node transition. This boost is significant enough to bring pheromones from the low range (C1) to C3. However, the cost metric of some of these edges traversed by human will sometimes outweigh the pheromone increases, therefore achieving a stabiliser effect (as can be seen in iterations 750 and 2250). These observations make sense, as the core of the algorithm is tailored to favour shorter distances, but the human-in-the-loop approach is trying to add user experience as another factor.

Moving on to the fine-grained view of each of the selected iterations, it can be seen in Figure 7.9 that the behaviour is again similar to what was observed for the classic MMAS case. The widespread pheromone values at the low end of the range is due to the previously explained effect of the pheromone evaporation constant. From this Figure it is also interesting to point out that, except for iteration 1500, there are no pheromones equal to τ_{min}, but there are for τ_{max}. This could have occurred due to the exceptional boost of the less likely edges, preventing them from decreasing to the minimum, as there is a greater likelihood they will be favoured in subsequent iterations. Thus, the evaporation factor for



(a) Pheromone distribution by intervals after 750 iterations.

(b) Pheromone distribution by intervals after 1500 iterations.

(c) Pheromone distribution by intervals after 2250 iterations.

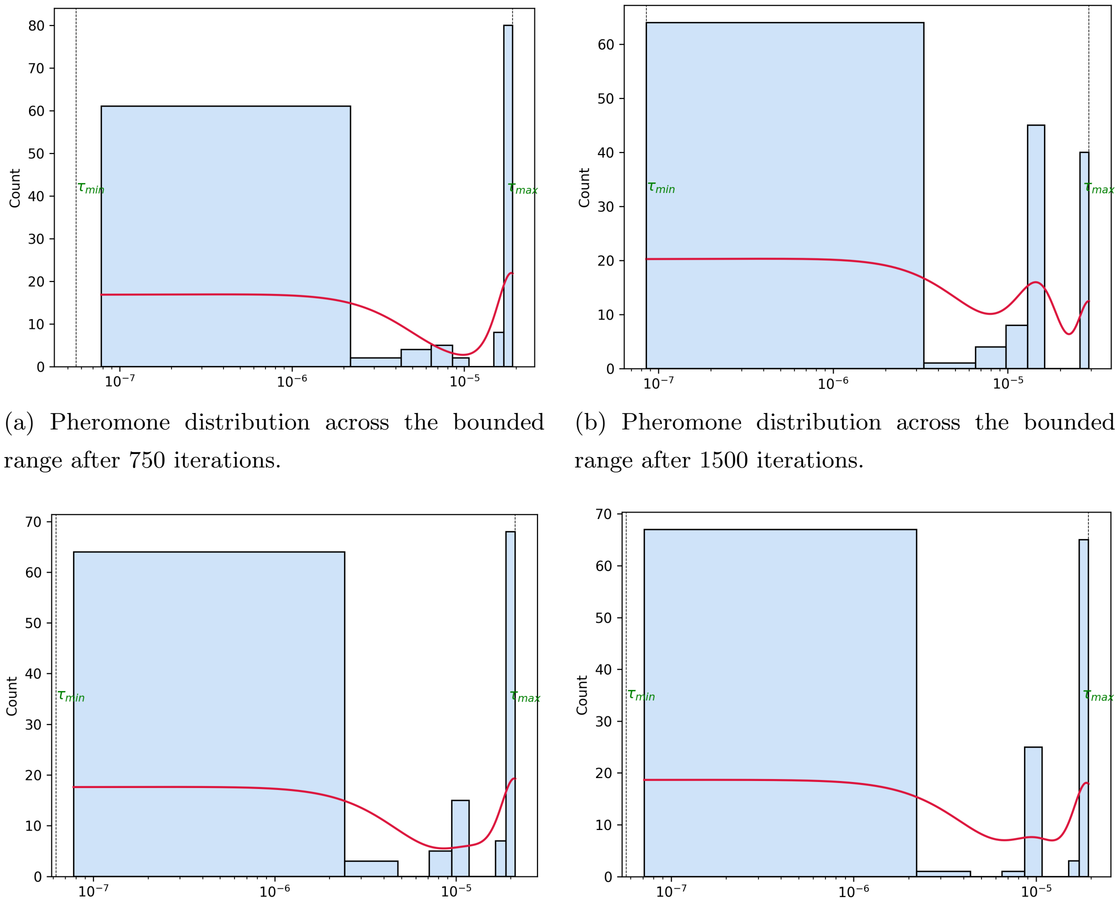
(d) Pheromone distribution by intervals after 3000 iterations.

Figure 7.8: Pheromone distribution by intervals at selected iterations during iML-enhanced MMAS model training.

these edges in these cases in outweighed by the user-driven pheromone boost.

With respect to Figure 7.10 representing pheromone distribution density across the iterations of interest, it can be appreciated that, as backed by the other Figures, iterations 750 and 2250 are very similar to the behaviour observed in classic MMAS. A bimodal distribution is clearly seen in these two iterations, event at iteration 3000, although in this case the trough presents a local minimum due to 16% of pheromones being located in interval C3 as we have observed. However, for iteration 1500 the bimodal distribution turns into a multimodal one with three peaks of decreasing density, being in line with the previous observations.

Finally, the shape of the violin plots depicted in Figure 7.11 again resemble those obtained for the classic MMAS approach, although, as we have seen, the middle intervals are more relevant in the iML case. For the last two iterations (2250 and 3000), the median appears to be more centered, i.e. at approximately the middle of the range of the pheromone values.



(a) Pheromone distribution across the bounded range after 750 iterations.

(b) Pheromone distribution across the bounded range after 1500 iterations.

(c) Pheromone distribution across the bounded range after 2250 iterations.

(d) Pheromone distribution across the bounded range 3000 iterations.

Figure 7.9: Histograms of pheromone distribution at selected iterations during iML-enhanced MMAS model training.

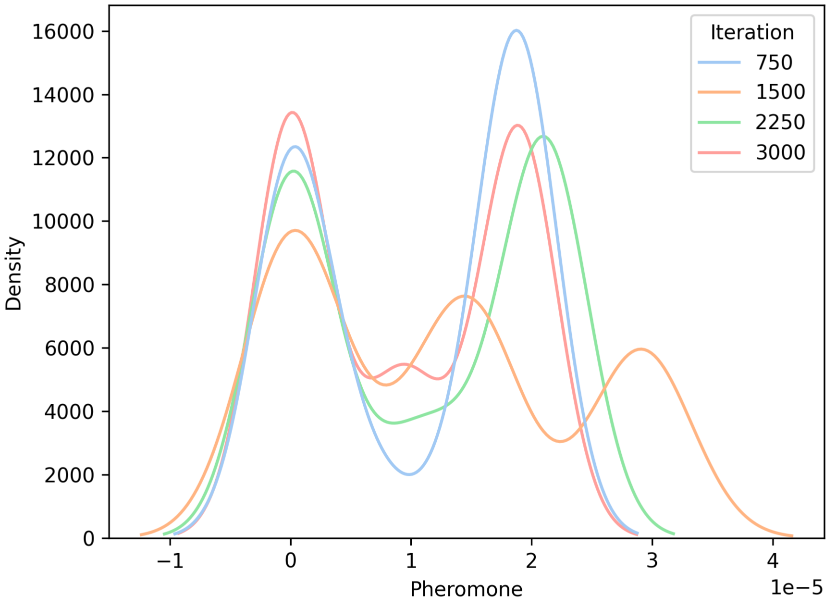


Figure 7.10: Comparison of pheromone densities at the selected iterations during iML-enhanced MMAS model training.

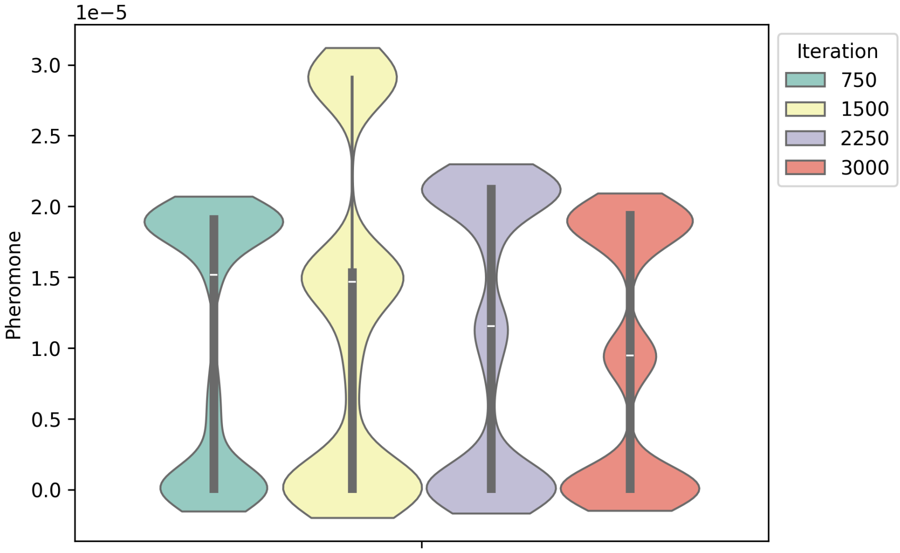


Figure 7.11: Comparison of pheromone value distributions and statistics at the selected iterations during iML-enhanced MMAS model training.

7.2.3 Model Evaluation Stage

Once the training phase has concluded, the resulting models will be tested. The tests will consist in selecting some target artworks and letting both the classic MMAS model and the iML-enhanced model suggest a route to include them in the visit. Provided that the scenario is set in a virtual museum and, as explained, there are no real users involved in the evaluation (despite having simulated their possible behaviour), it is difficult to obtain a real satisfaction metric. It is a personal and subjective measurement, and ideally, if real users could have taken part in the evaluation, their satisfaction could have been measured by means of questionnaires.

Hence, for the purpose of the proof-of-concept prototype developed for this master's thesis, we propose a simple yet unequivocal satisfaction metric that allows us to quantitatively evaluate and compare the performance of the basic MMAS and the iML-enhanced models. For this purpose, we will assume that satisfaction is directly proportional to the number of artworks visited. Thus, routes that take visitors through more artworks will be more attractive. However, the distance travelled must also be factored in. It may have a negative impact on visitor satisfaction if the route takes too long. For instance, recalling one of the works explored in Chapter 3 - Section 3.4 [116], their satisfaction metric in their museum setting also takes into account route distance.

Considering these two factors, we propose a straightforward satisfaction metric, S (Equation 7.1): a ratio of metres required to travel to reach an artwork. In essence, this indicator reflects that visitors will achieve greater satisfaction if they can explore more artworks whilst walking less.

$$S(artworks/m) = \frac{Exhibits\,Visited}{Total\,Distance\,Travelled\,(m)} \tag{7.1}$$

In reality, it is challenging to use a unique satisfaction metric across different visitor types, as there are different elements that affect their satisfaction depending on the profile type. However, the visitor-profile-independent satisfaction metric proposed can allow us to more easily evaluate the outcomes of the models for this prototype, thus reducing the high complexity of quantifying such a metric.

The models will be tested against four route suggestions. The selection of target artworks will be based on the four visitor profiles described earlier in this Chapter. For each of these cases, three target artworks will be selected, and the proposed route will be awarded a score based on the defined satisfaction metric. It must be noted that the scores will be multiplied by a factor of **100** to facilitate ease of interpretation, as they would otherwise be fractional decimals (< 1).

1. **Test 1**

   - **Target artworks:** i) *Jacob's Dream*, ii) *The 3rd of May 1808 in Madrid* and iii) *David with the Head of Goliath.*
   - **Suggested Route:**

Model	Suggested Route	Distance Travelled (m)	Score
Classic MMAS	*D1-1 → D1-5 → D1-2 → Jacob's Dream →* *D1-2 → D1-10 → The 3rd of May 1808 in Madrid → D1-10 →* *D1-13 → D1-6 → D6-7 → David with the Head of Goliath*	123.95	**2.420**
iML-Enhanced MMAS	*D1-1 → Las Meninas → The Annunciation →* *The Nobleman with his Hand on his Chest →* *The Garden of Earthly Delights → D1-10* *→ The 3rd of May 1808 in Madrid → D10-11 →* *Christ falls on the Way to Calvary → The Crucifixion → D2-11 →* *Jacob's Dream → D2-11 → D2-3 → Self-portrait → D3-5 →* *The Cardinal → D1-5 D1-6 → David with the Head of Goliath*	211.56	**6.144**

Table 7.5: Route suggested by classic and iml-enhanced mmas models for target artworks in the first test.

   - **Route Analysis:** In this first case, we can see that, attending to the satisfaction metric obtained, the iML version suggests a far better route (+154%) than that which has been generated by the classic model. The reason for this is because although the iML route is 70% longer than the one provided by the classic model, it includes 11 artworks as opposed to 3 in the classic's route, which is a 267% difference.

     Further analysing the classic model's route, we can see that the artworks are limited to those defined as part of the route's objectives. This is coherent with the nature of the classic MMAS algorithm, which aims to minimise the total distance travelled to visit the nodes. We can also see in this route some atypical transitions, such as the suggestion to go to door *D1-5* from *D1-1* instead of travelling directly to *D1-2*. Of course, from a mathematical point of view, the most favourable transition is that one, but in reality, it does not make much sense as a visitor would perform such movement.

The iML version, in contrast, suggests transitions where these anomalies are much less common as a result of human feedback during the training phase and the model having learned from user input. Hence, the suggested route is inherently more natural because the model achieves a compromise between distance and user experience.

2. **Test 2**

- **Target artworks:** i) *Las Meninas*, ii) *The Surrender of Breda* and iii) *Landscape with Psyche and Jupiter.*
- **Suggested Route:**

Model	Suggested Route	Distance Travelled (m)	Score
Classic MMAS	*D1-1 → Las Meninas → D1-8 →* *D7-8 → The Surrender of Breda → D6-7 →* *D6-13 → Landscape with Psyche and Jupiter*	50.09	**5.989**
iML-Enhanced MMAS	*D1-1 → Las Meninas → D1-8 →* *Judith at the Banquet of Holofernes → D7-8 →* *The Surrender of Breda → D7-8 → D7-14 →* *Guernica → D13-14 → Landscape with Psyche and Jupiter*	97.31	**5.138**

Table 7.6: Route suggested by classic and iml-enhanced mmas models for target artworks in the second test.

- **Route Analysis:** The obtained metrics in this test show that the route offered by the classic MMAS model yields a higher artworks per metre ratio (+16.56%) than the one recommended by the iML-enhanced MMAS model. The iML path explores two additional exhibits not contemplated in the objectives, but this “detour” comes at the expense of almost double the distance, thus hindering the defined satisfaction score.

  We can see that the classic path, as also occurred in the previous test, only includes the target artworks. In this case there are no anomalous transitions, as it seems that the higher pheromone trails connecting those nodes correspond to edges that make sense in a real scenario.

3. **Test 3**

- **Target artworks:** i) *The Three Graces*, ii) *Saturn* and iii) *Gaspar de Guzmán, Count-Duke of Olivares, on Horseback.*
- **Suggested Route:**

Model	Suggested Route	Distance Travelled (m)	Score
Classic MMAS	*D1-1 → D1-6 → The Three Graces → D1-13 → Saturn → The Meadow of San Isidro → D13-14 → D14-15 → Gaspar de Guzmán, Count-Duke of Olivares, on Horseback*	64.96	**6.158**
iML-Enhanced MMAS	*D1-1 → Las Meninas → The Annunciation → The Nobleman with his Hand on his Chest → The Garden of Earthly Delights → The Three Graces → D1-13 → Saturn → The Meadow of San Isidro → D13-14 → Guernica → D14-15 → Gaspar de Guzmán, Count-Duke of Olivares, on Horseback*	105.00	**8.570**

Table 7.7: Route suggested by classic and iml-enhanced mmas models for target artworks in the third test.

- **Route Analysis:** The obtained satisfaction scores for the classic and iML routes are 6.158 and 8.570 artworks/m respectively, which is 39.2% greater. Again, the classic MMAS model, by being only distance-oriented, provides rather straightforward paths.

  Remarkably, in this case, there is an additional artwork included in the route suggested by the classic model, *"The Meadow of San Isidro"*. Evidently, there is no human influence and it just happens to be the path with highest pheromone concentration to reach the objectives.

  With respect to the route proposed by the iML model, it includes 5 more artworks (129% increase) at a 61% greater distance travelled compared to the classic model, so the outcomes are considerably better. The suggested path is also noteworthy, as it contains similar elements to the routes in previous tests. This observation is coherent with the expected results, because the paths most influenced by visitors will be selected if the defined target artworks are close to them.

  For instance, visiting first *Las Meninas* is suggested in all previous analysed cases. The important presence of this edge can be attributed to the popularity of this artwork, and indeed, the four main route types include *Las Meninas*.

4. **Test 4**

- **Target artworks:** i) *The Annunciation*, ii) *The Garden of Earthly Delights* and iii) *A la deriva*.
- **Suggested Route:**

Model	Suggested Route	Distance Travelled (m)	Score
Classic MMAS	*D1-1 → D1-5 → The Annunciation → The Nobleman with his Hand on his Chest → The Garden of Earthly Delights → D1-10 → D10-12 → A la deriva*	55.77	**7.172**
iML-Enhanced MMAS	*D1-1 → Las Meninas → The Annunciation → The Nobleman with his Hand on his Chest → The Garden of Earthly Delights → D1-10 → The 3rd of May 1808 in Madrid → D10-12 → A la deriva*	76.26	**7.868**

Table 7.8: Route suggested by classic and iml-enhanced mmas models for target artworks in the fourth test.

- **Route Analysis:** In this final test scenario, the route provided by the iML model only achieved a satisfaction score 9% better than the route generated by the classic MMAS model. Although the classic model is able to include an additional artwork, the iML version considers two artworks more by just walking 20 more metres.

  The observation on frequent edges manifesting in the iML-generated routes holds true in this case as well. It also portrays one example where the algorithm has discarded an artwork. This is, Room 12, which is where one of the targets is located (*A la deriva*), also holds another artwork. However, due to its low popularity amongst the visits in the training phase, the pheromone trails from the other elements in the room to *Guitar and Newspaper* are low, and so the final route does not contemplate it as an artwork of interest.

Overall, the results we have presented allow us to confirm that the iML algorithm works as intended and that its final pheromone distribution — the product of a learning process with human feedback — serves to suggest routes through the museum that incorporate additional points of interest besides those that have been set as objectives.

The goal of reducing distance is weakened in light of the user experience, and overall, the suggested routes score higher on satisfaction than those produced by the traditional version of MMAS.

However, as we have already seen, this does not mean that there are no situations in which the classic version is more noticeable because having too many turns can negatively impact satisfaction, even in cases where you view more artwork.

CHAPTER 8

Conclusions and Future Work

This final chapter presents the conclusions drawn from this project. Additionally, a section is dedicated to future lines of research that could stem from the results produced in this work.

8.1 Conclusion

As part of my end of degree project, a Smart Object Gamification platform for museums was developed and validated. This system offers an interactive and entertaining way of seeing exhibits and learning more about them. In essence, the platform addresses the interaction with exhibits to enhance visitor's satisfaction and overall achieve a memorable museum experience.

However, the system does not provide visitors with guidance as to how to reach the exhibits. This is a crucial aspect in museums and particularly more in larger ones, where one might be at a loss if there is a vast collection. Visitors might not be sure of what to see or how to best walk the museum. This is detrimental to visitors' museum experience.

To address this issue, in this master thesis a iML-powered gamified routing platform has been devised that has been trained to provide the best routes in terms of potential satisfaction attained by visitors. The notion of most satisfactory route is based on visitor route patterns, so the algorithm learns which are the most frequented paths. For this project it has been assumed that, overall, paths that are followed by most visitors (and the hence are highly likely to contain the most renowned exhibits) generally will provide greater satisfaction to new visitors.

The developed platform expands the Smart Object Gamification system in order to offer visitors a full gamification experience from start to end. A module has been developed to train the iML algorithm. This module receives data from the interaction system we have developed for this project. In order to make users enjoy more a daunting task like helping train an algorithm, a virtual museum has been developed. Users navigate the virtual museum through a first-person immersive perspective and can interact with exhibits. Gamification elements have been introduced such as awarding points upon seeing exhibits. In virtue of having built the platform expanding the Smart Object Gamification platform, users can scan a QR code that appears when they interact with an exhibit. In this way they can access the Trivia Quiz game of automatically generated questions.

The evaluation results show how the routing algorithm's training phase is influenced by users' choices of paths and exhibits. Overall, this work yields promising results for offering new alternatives towards enhancing visitors' quality of experience in museums. However, this would still need to be validated in a real museum, and therefore several additional developments and implementations must be carried out in order to fully exploit the advantages that this system could bring to museums or any other cultural heritage institution that may benefit from its implementation.

8.2 Future Work

The ideas and developments made for this project open up several new opportunities to further expand the proposed system. Some would have been great to implement as part of this project, for instance a museum route suggestion module in the virtual museum after having trained the algorithm. However, the scope of the project and time constraints did not make it possible. Therefore, we will ellaborate on these proposals in this section.

- **Implementation of route suggestions in the virtual musem**: the next evident step of evolving the iML routing platform is to develop a module capable of making the already trained iML algorithm suggest routes that can increase visitors' satisfaction with the museum experience.

- **Adjustment of route suggestions according to visitors' profiles**: once the basic route suggestion module has been implemented, the iML algorithm could be adjusted and retrained to take into account visitors' profiles. The system could make route suggestions based on a combination of several factors, including age group of visitors, personal preferences and the scenario. Apart from the personal data that visitors might input to the platform, the system could ask visitors before starting the visit a short list of questions that help in hopefully building the most satisfactory route for them given all the provided input.

- **Implementation of the platform in a physical museum**: as explained throughout the work, for the scope of this project it was more convenient to develop the system using a virtual museum. However, the proposed system has been designed with a controller system decoupled from the interaction system, which means that there would be very few adaptations needed to make the system work in a real museum. It would be necessary, of course, to develop a *physical museum interaction* module (as we have called it in the project) to be able to provide the integral museum experience in a real museum.

  A proposal for this implementation could consist in the deployment of a mesh of BLE Beacons strategically placed around the museum that can be used to track visitors' location within the museum. Data transmitted by BLE Beacons would be received by a mobile application that ommunicates with the *Museum Controller System.* For the Smart Object Gamification platform, a gamification app was already released. Hence, it could be advantageous to enhance such app so that users can access both the Trivia Quiz game of automatically generated questions about exhibits, and the route suggestions made by the iML algorithm.

- **Expansion of gamification tactics:** this is another future line of research to be highlighted. Providing enriching museum experiences to visitors can prove to be a challenge if they are not engaged. To make visitors enjoy their visit as much as possible, it is important to implement new gamification strategies. Gamification will provide the most gratifying reward in the short term, so it is a very convenient way to increase visitor engagement and keep them interested and entertained throughout the visit.

  One possible gamification tactic could be the devisal of a scavenger hunt game. This game proposal could be used both during the training phase and the final route suggestions. For instance, a number of clues could be hidden around the museum next to exhibits, and visitors need to find them. They would compete against each other and maybe the museum could offer a prize to the winner to further encourage them to participate and engage. Visitors might stop at exhibits that draw their attention to search for clues in them, so this could be harnessed by the algorithm to take into account which exhibits are the ones that attract visitors and therefore suggest routes that include these.

APPENDIX A

Project Impact

This appendix delves into the (potential) impact of this work and its future lines of development. Concretely, it will reflect on the social, economic, environmental and ethical implications of the system proposed in this work.

A.1 Social Impact

Museum visitors are the main stakeholders that are affected by the implementation of the iML-powered gamified routing platform for museums. This platform has been designed to enhance visitors' museum experiences with the end goal to increase their satisfaction with the visit. Therefore, it has a positive impact on visitors, which hopefully makes them enjoy the museum and in turn increase the likelihood of revisiting it. This also has more connotations, as the promotion of cultural heritage - besides the monetary component of museum revenues - is of high importance to our society. The reason for this is that it allows people to better understand our past and present, as well as to better grasp and interpret culture and overall the world we live in.

A.2 Economic Impact

In regard to the economic impact of this project, museums could potentially enjoy increased revenues as a result of the increase in visitors that this platform could bring about. This is a potential benefit that could happen if the physical museum interaction module were to be developed and implemented in a real museum.

It is also worth noting the financial viability of deploying the full system in a real museum. As mentioned in the future lines of development discussed in this work, the physical interaction module could consist in a solution based on a mobile gamification application that can read data sent by a mesh of BLE Beacons strategically located around the museum. With the beacons it would be possible to triangulate the position of a visitor and therefore feed this data to the iML routing algorithm. Thus, there would be costs involved in the development of such application as well as the cost of purchasing and configuring BLE Beacons.

Another element to bear in mind is the increase in computer power (and thus cost) necessary to process algorithm iterations for all visitors using the platform (probably a horizontal scale out topology would be required).

Taking all this into consideration, it is difficult to provide at this point an estimate of the total cost of implementing the platform into a real world scenario, since it will largely depend on the dimensions of the museum and the extent to which the platform will be implemented in the museum. This would mean that the project's cost could be adapted according to the budget of the museum to make it financially viable.

A.3 Environmental Impact

The environmental impact of the project developed for this master's thesis is low in terms of energy consumption. To be more precise, the virtual museum scenario has been developed in my personal computer, therefore the impact is low. The computer runs on a 500W power supply with a 80 Plus Gold efficiency rating, meaning that at least the electrical energy wasted is reduced.

If the project were to be further expanded to include the physical museum implementation, then the environmental impact would be significantly higher with respect to the current development. BLE Beacons would be needed which are made of a plastic casing, so this contributes to the circulation of more plastics. Although they are small gadgets, the number required in a museum could quickly become considerably large, and with that the amount of plastic used. It would become very important to recycle all this plastic once Beacons reach the end of their useful life. A possible idea to ensure these plastics are indeed recycled is to offer a free of charge service to museums to pick up ‘dead’ Beacons so that they do need to deal with their disposal.

Additionally, the implementation of the platform in a museum would imply the use of more computing resources, which means an increase in energy consumption. In that case it could be very advantageous to exploit the advantages of having the infrastructure on the cloud, enjoying the benefits of elastic computing (efficiently using only the resources we need). Most importantly, nowadays cloud services are increasingly environmentally aware and many are transitioning to green computing policies. For example, as of 2022, 90% of the electricity consumed by AWS cloud was attributable to renewable energies. They have set the objective to increase this number to 100% by 2025[1].

A.4 Ethical Impact

The virtual museum development done in this work does not have any significant ethical impact that should be taken into consideration. At this stage, it does not gather any important personal data of users (other than a username), so privacy is not an issue.

However, it is worth reflecting on the impact that future development of this work could have were it to be implemented in a physical museum. The idea would be to be able to segment visitors in the platform according to their preferences in order to adapt tour recommendations to each visitor. This means that the platform will gather data to do user profiling. In that case privacy concerns will start having greater importance. These data would not be shared with any third party whatsoever. However, personal data could be leaked due to a cyberattack, so if the platform were to be implemented in a real-world museum it would be highly advisable to carry out data anonymization procedures to protect visitors’ data.

There is another ethical aspect to consider in the future implementation of the platform in physical museums. It involves a stakeholder group that could be negatively affected if museums were to use this platform: tour guides. The adoption of this technology could overshadow the role of tour guides, since the platform can would provide information on the exhibits as well offer appealing routes to visitors. This means that the implementation of the

[1] https://sustainability.aboutamazon.com/products-services/the-cloud?energyType=true

platform could derive in job cuts of tour guides. With the the coming of new technologies usually comes this type of dilemma. However, it must be also be noted that a balance could be obtained by creating new job opportunities for these tour guides. Since they are well versed in the museum environment, their knowledge could be useful to further refine the routes patterns learned by the iML algorithm as well as keeping up to date the information displayed about the exhibits through the platform.

APPENDIX B

Project Budget

This appendix offers a succinct economic analysis of the project. A list of tasks that were necessary for the realisation of this master's thesis is provided, together with the time taken to complete them. The cost of material and human resources is evaluated with the aim of providing a final figure for the total budget of the project.

B.1 Project Structure

This section presents the structure of the project in terms of tasks and their durations. A total of 900 hours of work have been dedicated to the design and development of the platform, including writing this master's thesis report. This project has been done alongside my full-time job as full-stack software engineer, therefore the average time allotted per day to this thesis has been of 2 hours. This adds up to a total of **450 days** taken to complete the project. Table B.1 presents a breakdown of the tasks involved and the respective time taken to carry them out.

Task	**Duration (days)**
1. Conduct review of State of the Art and of related works	25
2. Study the foundations of the Travelling Salesman Problem and Ant Colony Optimisation Heuristics	18
3. Carry out a comprehensive analysis of Max-Min Ant System (MMAS) Metaheuristic	7
4. Capture Requirements for the Development of the gamified iML Routing Platform	15
5. Conceptualisation of methodology and architecture based on the requirements gathered	17
6. Study available viable technologies for developing the project and choose the most suitable ones	8
7. Learn how to use the chosen enabling technologies (tutorials, short courses)	10
8. Develop first iteration of the standalone Python implementation of traditional MMAS	30
9. Test and refine MMAS implementation with a reference TSP graph	25
10. Refine MMAS first iteration to accept user input to dynamically modify edge weights	15
11. Implement the Museum Controller System (without iML algorithm)	45
12. Adapt MMAS into the iML Routing module	15
13. Tailoring and constructing a reasonable museum blueprint and selecting exhibits from real museum's online collections	25
14. Develop virtual museum interaction module capable of flexibly rendering blueprints into 3D environments	90
15. Use the developed scenario to feed travelled paths in the museum to the training phase of the MMAS algorithm (iML) and compare against algorithm learning on its own with no human intervention	25
16. Write Master's Thesis Report	80
Total	**450**

Table B.1: Project Task Breakdown

B.2 Physical Resources

The material resources that have been used to bring about this project are detailed in this section. These comprise software and hardware used.

- *Hardware:* the project has been developed in a desktop computer with the following configuration:

  - CPU: Intel Core i7-7700 @ 3.6Ghz
  - RAM: 2x8GB @ 2333MHz
  - GPU: GTX 1060 6GB
  - PSU: 550W 80 Plus Gold
  - Storage: 480GB SSD

  The performance of the computer used has been more than sufficient to accomplish all of the developments required. Provided that the designed virtual museum has a bounded size, the number of nodes to compute in the MMAS algorithm is kept low enough to allow the project to run smoothly under this hardware. The estimated cost of a computer with the aforementioned configuration is about **1200€**.

- *Software:* all of the software used (mainly python libraries or frameworks) is either open-source or free to use. For the development of the project, PyCharm Community IDE for Python programming language was initially used. The project was 'dockerised' at a very early stage, and debugging an application using Docker containers is not supported in the Community edition. Therefore, a PyCharm Pro license was purchased to be able to debug code even when placed in Docker containers. This has a cost of 99€ per year. Considering that the project took 450 days to be completed, the total prorated cost of the license is of **122.05€** for the whole duration of the project.

B.3 Human Resources

A person with a strong technical background was needed to undertake the project. A telecommunication engineer is well suited to accomplish this. Taking into account that the development of the project has been done alongside a full-time job as software engineer, it makes sense to estimate the cost based on the salary earned in this job. The gross annual salary is 28000€, which results in 2333.33€/month gross. Given that 40 hours are worked in a week, then a total of 160 hours are worked in a month. Thus, the gross wage based on these figures is 14.58€/h. Extrapolating this to the duration of this project, 900 hours, the total cost of human resources adds up to **13124.98€**. This represents the total amount that the company would have to pay for the worker, including Personal Income Tax and Social Security. In Spain, Personal Income Tax withholding for the mentioned annual gross salary is 4376.4€ (15.63%), and the Social Security contributions are 1778€. Therefore, the engineer would perceive a net salary of 21845 €/year or 11.37€/h.

B.4 Total Cost

After having analysed and identified the different cost layers incurred in this project, the total cost is **14447.03€** for its duration of **900 hours**.

Glossary

ACO Ant Colony Optimisation. 6, 16–18, 28–31, 33, 35, 58

ACS Ant Colony System. 18, 20, 28, 30, 34, 35

aML automatic Machine Learning. 7, 11, 12

AS Ant System. 18, 20–23, 28

$\mathbf{AS_{rank}}$ Ranked-Based Ant System. 18, 21

BWTSP Black and White Travelling Salesman Problem. 27

EON Elastic Optical Network. XI, 27

HKLB Held-Karp Lower Bound. 14, 15

iML interactive Machine Learning. V, XI–XIV, 4, 6, 7, 12, 16, 31–34, 57, 78, 80, 81, 83, 88, 91, 93–101, 103

MMAS MAX-MIN Ant System. V, XIII, 18, 22, 23, 29–31, 59, 78–80, 83–86, 92–101

SMTWT Single Machine Total Weighted Tardiness. 29, 30

TOP Team Orienteering Problem. XI, 29

TSP Travelling Salesman Problem. V, 6, 12, 13, 15, 16, 18, 26–28, 31, 33, 34, 78, 79, 81

VRP Vehicle Routing Problem. XI, 26, 28, 29

Bibliography

[1] UNESCO. Heritage. https://en.unesco.org/creativity/sites/creativity/files/cdis/heritage_dimension.pdf. (Accessed 24/01/2023).

[2] David Lowenthal. Environmental perception: preserving the past. *Progress in Geography*, 3(4):549–559, 1979.

[3] Council of europe framework convention on the value of cultural heritage for society, 27 october 2005. *International Journal of Cultural Property*, 14(4):431–440, 2007.

[4] Richard Kurin et al. Safeguarding intangible cultural heritage: Key factors in implementing the 2003 convention. *International journal of intangible heritage*, 2(8):9–20, 2007.

[5] Albina Absalyamova, Timur Absalyamov, and Svetlana Absalyamova. Private museums as a form of preservation of cultural heritage. *Procedia-Social and Behavioral Sciences*, 188:218–221, 2015.

[6] Muhammad Nawaz Khan, Haseeb Ur Rahman, Mohammad Faisal, Faheem Khan, and Shabir Ahmad. An iot-enabled information system for smart navigation in museums. *Sensors*, 22(1):312, 2021.

[7] Edward P Alexander, Mary Alexander, and Juilee Decker. *Museums in motion: An introduction to the history and functions of museums*. Rowman & Littlefield, 2017.

[8] Beverly Serrell. *Exhibit labels: An interpretive approach*. Rowman & Littlefield, 2015.

[9] John H Falk and Lynn D Dierking. *The museum experience revisited*. Routledge, 2016.

[10] Sue Allen. Designs for learning: Studying science museum exhibits that do more than entertain. *Science education*, 88(S1):S17–S33, 2004.

[11] John H Falk. *Identity and the museum visitor experience*. Routledge, 2016.

[12] EunJung Chang. Interactive experiences and contextual learning in museums. *Studies in Art Education*, 47(2):170–186, 2006.

[13] Hyunae Lee, Timothy Hyungsoo Jung, M Claudia tom Dieck, and Namho Chung. Experiencing immersive virtual reality in museums. *Information & Management*, 57(5):103229, 2020.

[14] Mandy Ding et al. Augmented reality in museums. *Museums & augmented reality–A collection of essays from the arts management and technology laboratory*, pages 1–15, 2017.

[15] Shancang Li, Li Da Xu, and Shanshan Zhao. The internet of things: a survey. *Information systems frontiers*, 17:243–259, 2015.

[16] Gerd Kortuem, Fahim Kawsar, Vasughi Sundramoorthy, and Daniel Fitton. Smart objects as building blocks for the internet of things. *IEEE Internet Computing*, 14(1):44–51, 2009.

[17] Andrzej Marczewski. *Gamification: a simple introduction.* Andrzej Marczewski, 2013.

[18] Tara Khairiyah Md Zali, Nor Samsiah Sani, Abdul Hadi Abd Rahman, and Mohd Aliff. Attractiveness analysis of quiz games. *International Journal of Advanced Computer Science and Applications*, 10(8), 2019.

[19] Alejandro López-Martínez, Álvaro Carrera, and Carlos A Iglesias. Empowering museum experiences applying gamification techniques based on linked data and smart objects. *Applied Sciences*, 10(16):5419, 2020.

[20] Massimo Zancanaro, Tsvi Kuflik, Zvi Boger, Dina Goren-Bar, and Dan Goldwasser. Analyzing museum visitors' behavior patterns. In *User Modeling 2007: 11th International Conference, UM 2007, Corfu, Greece, July 25-29, 2007. Proceedings 11*, pages 238–246. Springer, 2007.

[21] Nancy Proctor and Chris Tellis. The state of the art in museum handhelds in 2003. 2003.

[22] Shelley Mannion, Amalia Sabiescu, and William Robinson. An audio state of mind: Understanding behaviour around audio guides and visitor media. *MW2015: Museums and the Web 2015*, 2015.

[23] Museums and the Web 2015. A new look at an old friend: Reevaluating the Met's audio-guide service. https://mw2015.museumsandtheweb.com/paper/a-new-look-at-an-old-friend-re-evaluating-the-mets-audio-guide-service/. (Accessed 05/02/2023).

[24] Edsger W. Dijkstra. *A Discipline of Programming.* Prentice-Hall, 1976.

[25] Bezalel Gavish and Stephen C Graves. The travelling salesman problem and related problems. 1978.

[26] Thomas Stützle and Holger H Hoos. Max–min ant system. *Future generation computer systems*, 16(8):889–914, 2000.

[27] Momin M Malik. A hierarchy of limitations in machine learning. *arXiv preprint arXiv:2002.05193*, 2020.

[28] Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. Power to the people: The role of humans in interactive machine learning. *Ai Magazine*, 35(4):105–120, 2014.

[29] A. L. Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3(3):210–229, 1959.

[30] Maad M Mijwel. History of artificial intelligence yapay zekânın t arihi. *no. April*, 2018, 2015.

[31] Luciano Floridi. Ai and its new winter: From myths to realities. *Philosophy & Technology*, 33:1–3, 2020.

[32] Jahanzaib Shabbir and Tarique Anwer. Artificial intelligence and its role in near future. *arXiv preprint arXiv:1804.01396*, 2018.

[33] Nick Heath. Nasa's unsung heroes: The apollo coders who put men on the moon. https://www.techrepublic.com/article/nasas-unsung-heroes-the-apollo-coders-who-put-men-on-the-moon/, 2018.

[34] Batta Mahesh. Machine learning algorithms-a review. *International Journal of Science and Research (IJSR).[Internet]*, 9:381–386, 2020.

[35] Alan M Turing. *Computing machinery and intelligence*. Springer, 2009.

[36] Pádraig Cunningham, Matthieu Cord, and Sarah Jane Delany. Supervised learning. *Machine learning techniques for multimedia: case studies on organization and retrieval*, pages 21–49, 2008.

[37] Sami Belkacem. *Machine learning approaches to rank news feed updates on social media*. PhD thesis, 04 2021.

[38] Zoubin Ghahramani. Unsupervised learning. *Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2-14, 2003, Tübingen, Germany, August 4-16, 2003, Revised Lectures*, pages 72–112, 2004.

[39] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[40] Patrick Kurp. Green computing. *Communications of the ACM*, 51(10):11–13, 2008.

[41] Davide Castelvecchi. Can we open the black box of ai? *Nature News*, 538(7623):20, 2016.

[42] Christopher Kuner, Dan Jerker B Svantesson, Fred H Cate, Orla Lynskey, and Christopher Millard. Machine learning with personal data: is data protection law smart enough to meet the challenge? *International Data Privacy Law*, 7(1):1–2, 2017.

[43] Andreas Holzinger. Interactive machine learning for health informatics: when do we need the human-in-the-loop? *Brain Informatics*, 3(2):119–131, 2016.

[44] Andreas Holzinger, Markus Plass, Katharina Holzinger, Gloria Cerasela Crisan, Camelia-M Pintea, and Vasile Palade. A glass-box interactive machine learning approach for solving np-hard problems with the human-in-the-loop. *arXiv preprint arXiv:1708.01104*, 2017.

[45] Andreas Holzinger, Markus Plass, Michael Kickmeier-Rust, Katharina Holzinger, Gloria Cerasela Crişan, Camelia-M Pintea, and Vasile Palade. Interactive machine learning: experimental evidence for the human in the algorithmic loop: A case study on ant colony optimization. *Applied Intelligence*, 49:2401–2414, 2019.

[46] Gilbert Laporte. A short history of the traveling salesman problem. *Canada Research Chair in Distribution Management, Centre for Research on Transportation (CRT) and GERAD HEC Montréal, Canada*, 2006.

[47] Karla L. Hoffman, Manfred Padberg, and Giovanni Rinaldi. *Traveling Salesman Problem*, pages 1573–1578. Springer US, 2013.

[48] E. Filip and M. Otakar. The travelling salesman problem and its application in logistic practice. 8:163–173, 10 2011.

[49] DONALD CHAN and Daniel Mercier. Ic insertion: an application of the travelling salesman problem. *International Journal of Production Research - INT J PROD RES*, 27:1837–1841, 10 1989.

[50] Marco Caserta and Stefan Voß. A hybrid algorithm for the dna sequencing problem. *Discrete Applied Mathematics*, 163:87–99, 2014.

[51] Antoine Dutot and Damien Olivier. 5 - swarm problem-solving. In *Agent-based Spatial Simulation with NetLogo, Volume 2*, pages 117–172. Elsevier, 2017.

[52] Michael R Garey, Ronald L Graham, and David S Johnson. Some np-complete geometric problems. In *Proceedings of the eighth annual ACM symposium on Theory of computing*, pages 10–22, 1976.

[53] Michael Held and Richard M Karp. A dynamic programming approach to sequencing problems. *Journal of the Society for Industrial and Applied mathematics*, 10(1):196–210, 1962.

[54] David Applegate, Robert Bixby, William Cook, and Vasek Chvátal. On the solution of traveling salesman problems. *Documenta Mathematica*, pages 645–656, 1998.

[55] Manfred Padberg and Giovanni Rinaldi. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM review*, 33(1):60–100, 1991.

[56] David L Applegate, Robert E Bixby, Vašek Chvátal, William Cook, Daniel G Espinoza, Marcos Goycoolea, and Keld Helsgaun. Certification of an optimal tsp tour through 85,900 cities. *Operations Research Letters*, 37(1):11–15, 2009.

[57] Keld Helsgaun. An effective implementation of the lin–kernighan traveling salesman heuristic. *European journal of operational research*, 126(1):106–130, 2000.

[58] David S. Johnson and Lyle A. McGeoch. The traveling salesman problem: A case study in local optimization. 2008.

[59] Egbert Giles Leigh Jr. Natural selection and mutability. *The American Naturalist*, 104(937):301–305, 1970.

[60] Rajesh Matai, Surya Prakash Singh, and Murari Lal Mittal. Traveling salesman problem: an overview of applications, formulations, and solution approaches. *Traveling salesman problem, theory and applications*, 1, 2010.

[61] Christopher C Skiscim and Bruce L Golden. Optimization by simulated annealing: A preliminary computational study for the tsp. Technical report, Institute of Electrical and Electronics Engineers (IEEE), 1983.

[62] Gad Perry and Eric R Pianka. Animal foraging: past, present and future. *Trends in Ecology & Evolution*, 12(9):360–364, 1997.

[63] M.D. Ginzel. Olfactory signals. In *Encyclopedia of Animal Behavior*, pages 584–588. 2010.

[64] Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *Swarm intelligence: from natural to artificial systems*. Number 1. Oxford university press, 1999.

[65] M. Dorigo, V. Maniezzo, and A. Colorni. Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 26(1):29–41, 1996.

[66] Simon Goss, Serge Aron, Jean-Louis Deneubourg, and Jacques Marie Pasteels. Self-organized shortcuts in the argentine ant. *Naturwissenschaften*, 76(12):579–581, 1989.

[67] Marco Dorigo. Optimization, learning and natural algorithms. *Ph. D. Thesis, Politecnico di Milano*, 1992.

[68] Marco Dorigo and Luca Maria Gambardella. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on evolutionary computation*, 1(1):53–66, 1997.

[69] Bernd Bullnheimer. A new rank based version of the ant system: A computational study. *Central Eur. J. Oper. Res. Econ.*, 7:25–38, 1999.

[70] Luca M Gambardella and Marco Dorigo. Ant-q: A reinforcement learning approach to the traveling salesman problem. In *Machine learning proceedings 1995*, pages 252–260. Elsevier, 1995.

[71] George B Dantzig and John H Ramser. The truck dispatching problem. *Management science*, 6(1):80–91, 1959.

[72] Joanna Ochelska-Mierzejewska, Aneta Poniszewska-Marańda, and Witold Marańda. Selected genetic algorithms for vehicle routing problem solving. *Electronics*, 10(24), 2021.

[73] Huang Mei, Yang Jingshuai, MA Teng, LI Xiuli, and Wang Ting. The modeling of milk-run vehicle routing problem based on improved c-w algorithm that joined time window. *Transportation Research Procedia*, 25:716–728, 2017. World Conference on Transport Research - WCTR 2016 Shanghai. 10-15 July 2016.

[74] Chaug-Ing Hsu, Sheng-Feng Hung, and Hui-Chieh Li. Vehicle routing problem with time-windows for perishable food delivery. *Journal of food engineering*, 80(2):465–475, 2007.

[75] G Bocewicz, P Nielsen, and Z Banaszak. Declarative modeling of a milk-run vehicle routing problem for split and merge supply streams scheduling. In *Information Systems Architecture and Technology: Proceedings of 39th International Conference on Information Systems Architecture and Technology–ISAT 2018: Part II*, pages 157–172. Springer, 2019.

[76] Sami Serkan Özarık, Lucas P. Veelenturf, Tom Van Woensel, and Gilbert Laporte. Optimizing e-commerce last-mile vehicle routing and scheduling under uncertain customer presence. *Transportation Research Part E: Logistics and Transportation Review*, 148:102263, 2021.

[77] Stef Moons, Katrien Ramaekers, An Caris, and Yasemin Arda. Integration of order picking and vehicle routing in a b2c e-commerce context. *Flexible Services and Manufacturing Journal*, 30(4):813–843, 2018.

[78] Ling Liu, Kunpeng Li, and Zhixue Liu. A capacitated vehicle routing problem with order available time in e-commerce industry. *Engineering Optimization*, 49(3):449–465, 2017.

[79] Martin Savelsbergh and Tom Van Woensel. 50th anniversary invited article—city logistics: Challenges and opportunities. *Transportation Science*, 50(2):579–590, 2016.

[80] Maurizio Bruglieri, Simona Mancini, Ferdinando Pezzella, and Ornella Pisacane. A path-based solution approach for the green vehicle routing problem. *Computers & Operations Research*, 103:109–122, 2019.

[81] Ángel Felipe, M Teresa Ortuño, Giovanni Righini, and Gregorio Tirado. A heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges. *Transportation Research Part E: Logistics and Transportation Review*, 71:111–128, 2014.

[82] Anurag Tiwari and Pei-Chann Chang. A block recombination approach to solve green vehicle routing problem. *International Journal of Production Economics*, 164:379–387, 2015.

[83] Paulo Roberto de Oliveira da Costa, Stefano Mauceri, Paula Carroll, and Fabiano Pallonetto. A genetic algorithm for a green vehicle routing problem. *Electronic notes in discrete mathematics*, 64:65–74, 2018.

[84] Sevgi Erdoğan and Elise Miller-Hooks. A green vehicle routing problem. *Transportation research part E: logistics and transportation review*, 48(1):100–114, 2012.

[85] Godfrey C. Onwubolu. *Optimizing CNC Drilling Machine Operations: Traveling Salesman Problem-Differential Evolution Approach*, pages 537–565. Springer Berlin Heidelberg, 2004.

[86] Martin Grötschel, Michael Jünger, and Gerhard Reinelt. Optimal control of plotting and drilling machines: a case study. *Zeitschrift für Operations Research*, 35:61–84, 1991.

[87] Haitao Wu, Fen Zhou, Zuqing Zhu, and Yaojun Chen. On the distance spectrum assignment in elastic optical networks. *IEEE/ACM Transactions on Networking*, 25(4):2391–2404, 2017.

[88] Bijoy Chand Chatterjee, Nityananda Sarma, and Eiji Oki. Routing and spectrum allocation in elastic optical networks: A tutorial. *IEEE Communications Surveys & Tutorials*, 17(3):1776–1800, 2015.

[89] Adaptive routing and spectrum allocation in elastic optical networks. *Optical Switching and Networking*, 24:12–20, 2017.

[90] O.J. Wasem, R.H. Cardwell, and T.-H. Wu. Software for designing survivable sonet networks using self-healing rings. In *[Conference Record] SUPERCOMM/ICC '92 Discovering a New World of Communications*, pages 425–431 vol.1, 1992.

[91] Steven Cosares, David N Deutsch, Iraj Saniee, and Ondria J Wasem. Sonet toolkit: A decision support system for designing robust and cost-effective fiber-optic networks. *Interfaces*, 25(1):20–40, 1995.

[92] Jacek Błażewicz, Piotr Formanowicz, Marta Kasprzak, Wojciech T Markiewicz, and J Węglarz. Dna sequencing with positive and negative errors. *Journal of Computational Biology*, 6(1):113–123, 1999.

[93] Thomas Stützle, Manuel López-Ibáñez, and Marco Dorigo. A concise overview of applications of ant colony optimization. *Wiley encyclopedia of operations research and management science*, 2:896–911, 2011.

[94] Bernd Bullnheimer, Richard F Hartl, and Christine Strauss. An improved ant system algorithm for the vehicle routing problem. *Annals of operations research*, 89(0):319–328, 1999.

[95] Luca Maria Gambardella, Éric Taillard, and Giovanni Agazzi. Macs-vrptw: A multiple ant colony system for vehicle routing problems with time windows, 1999.

[96] I-Ming Chao, Bruce L Golden, and Edward A Wasil. The team orienteering problem. *European journal of operational research*, 88(3):464–474, 1996.

[97] Liangjun Ke, Claudia Archetti, and Zuren Feng. Ants can solve the team orienteering problem. *Computers & Industrial Engineering*, 54(3):648–665, 2008.

[98] Lorena S. Reyes-Rubiano, Carlos F. Ospina-Trujillo, Javier Faulin, Jose M. Mozos, Javier Panadero, and Angel A. Juan. The team orienteering problem with stochastic service times and driving-range limitations: A simheuristic approach. In *2018 Winter Simulation Conference (WSC)*, pages 3025–3035, 2018.

[99] Ali Allahverdi and HM Soroush. The significance of reducing setup times/setup costs. *European journal of operational research*, 187(3):978–984, 2008.

[100] Matthijs Den Besten, Thomas Stützle, and Marco Dorigo. Ant colony optimization for the total weighted tardiness problem. In *Parallel Problem Solving from Nature PPSN VI: 6th International Conference Paris, France, September 18–20, 2000 Proceedings 6*, pages 611–620. Springer, 2000.

[101] Davide Anghinolfi and Massimo Paolucci. A new ant colony optimization approach for the single machine total weighted tardiness scheduling problem. *International Journal of Operations Research*, 5(1):1–17, 2008.

[102] Christopher M Dobson. Protein folding and misfolding. *Nature*, 426(6968):884–890, 2003.

[103] Alena Shmygelska and Holger H Hoos. An ant colony optimisation algorithm for the 2d and 3d hydrophobic polar protein folding problem. *BMC bioinformatics*, 6:1–22, 2005.

[104] Torsten Thalheim, Daniel Merkle, and Martin Middendorf. Protein folding in the hp-model solved with a hybrid population based aco algorithm. *IAENG International Journal of Computer Science*, 35(3):291–300, 2008.

[105] Stefka Fidanova and Ivan Lirkov. Ant colony system approach for protein folding. In *2008 International Multiconference on Computer Science and Information Technology*, pages 887–891, 2008.

[106] An ant colony optimization algorithm for dna sequencing by hybridization. *Computers and Operations Research*, 35(11):3620–3635, 2008. Part Special Issue: Topics in Real-time Supply Chain Management.

[107] Natnael A Wondimu, Cédric Buche, and Ubbo Visser. Interactive machine learning: A state of the art review. *arXiv preprint arXiv:2207.06196*, 2022.

[108] Seid Muhie Yimam, Chris Biemann, Ljiljana Majnaric, Šefket Šabanović, and Andreas Holzinger. An adaptive annotation approach for biomedical entity and relation recognition. *Brain Informatics*, 3(3):157–168, 2016.

[109] Carrie J Cai, Emily Reif, Narayan Hegde, Jason Hipp, Been Kim, Daniel Smilkov, Martin Wattenberg, Fernanda Viegas, Greg S Corrado, Martin C Stumpe, et al. Human-centered tools for coping with imperfect algorithms during medical decision-making. In *Proceedings of the 2019 chi conference on human factors in computing systems*, pages 1–14, 2019.

[110] Ilker Kose, Mehmet Gokturk, and Kemal Kilic. An interactive machine-learning-based electronic fraud and abuse detection system in healthcare insurance. *Applied Soft Computing*, 36:283–299, 2015.

[111] Spencer Frazier and Mark Riedl. Improving deep reinforcement learning in minecraft with action advice. In *Proceedings of the AAAI conference on artificial intelligence and interactive digital entertainment*, volume 15, pages 146–152, 2019.

[112] Robert S Gutzwiller and John Reeder. Human interactive machine learning for trust in teams of autonomous robots. In *2017 IEEE Conference on Cognitive and Computational Aspects of Situation Management (CogSIMA)*, pages 1–3. IEEE, 2017.

[113] Mu-Huan Chung, Mark Chignell, Lu Wang, Alexandra Jovicic, and Abhay Raman. Interactive machine learning for data exfiltration detection: Active learning with human expertise. In *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 280–287. IEEE, 2020.

[114] Pierre-Edouard Osche, Sylvain Castagnos, and Anne Boyer. From music to museum: applications of multi-objective ant colony systems to real world problems. In *Adaptive and Learning Agents Workshop at AAMAS (ALA 2019)*, 2019.

[115] Javier Jaen, Jose Mocholi, Alejandro Catala, and Elena Navarro. Digital ants as the best cicerones for museum visitors. *Applied Soft Computing*, 11:111–119, 01 2011.

[116] Eirini Eleni Tsiropoulou, Athina Thanou, and Symeon Papavassiliou. Quality of experience-based museum touring: A human in the loop approach. *Social Network Analysis and Mining*, 7:1–13, 2017.

[117] Shuo-Yan Chou and Shih-Wei Lin. Museum visitor routing problem with the balancing of concurrent visitors. In *Complex Systems Concurrent Engineering: Collaboration, Technology Innovation and Sustainability*, pages 345–353. Springer, 2007.

[118] THREE.js official page. `https://threejs.org`. (Accessed 12/02/2023).

[119] Tony Parisi. *WebGL: up and running.* " O'Reilly Media, Inc.", 2012.

[120] Miguel Grinberg. *Flask web development: developing web applications with python.* " O'Reilly Media, Inc.", 2018.

[121] Pallets. Werzeug Official Documentation. https://werkzeug.palletsprojects.com/en/2.2.x/. (Accessed 16/03/2023).

[122] Pallets. Jinja Official Documentation. https://jinja.palletsprojects.com/en/3.1.x/. (Accessed 16/03/2023).

[123] Python Enhancement Proposals (PEPs). PEP 3333 - Python Web Server Gateway Interface Standard. https://peps.python.org/pep-3333/. (Accessed 16/03/2023).

[124] Unbit. uWSGI official documentation. https://uwsgi-docs.readthedocs.io/en/latest/. (Accessed 18/03/2023).

[125] Benoit Chesneau. Gunicorn official documentation. https://docs.gunicorn.org/en/stable/. (Accessed 18/03/2023).

[126] Will Reese. Nginx: the high-performance web server and reverse proxy. *Linux Journal*, 2008(173):2, 2008.

[127] Jackie Kazil, David Masad, and Andrew Crooks. Utilizing python for agent-based modeling: The mesa framework. In *Social, Cultural, and Behavioral Modeling: 13th International Conference, SBP-BRiMS 2020, Washington, DC, USA, October 18–21, 2020, Proceedings 13*, pages 308–317. Springer, 2020.

[128] Andrew T Crooks and Alison J Heppenstall. Introduction to agent-based modelling. In *Agent-based models of geographical systems*, pages 85–105. Springer, 2011.

[129] Jacqueline Kazil, David Masad, and Andrew Crooks. Utilizing python for agent-based modeling: The mesa framework. 10 2020.

[130] Sean Gillies. The shapely user manual. *URL https://pypi. org/project/Shapely*, 2013.

[131] Max J Egenhofer and Robert D Franzosa. Point-set topological spatial relations. *International Journal of Geographical Information System*, 5(2):161–174, 1991.

[132] Aric Hagberg, Pieter Swart, and Daniel S Chult. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

[133] Eli Bressert. Scipy and numpy: an overview for developers. 2012.

[134] Kyle Banker, Douglas Garrett, Peter Bakkum, and Shaun Verch. *MongoDB in action: covers MongoDB version 3.0*. Simon and Schuster, 2016.

[135] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

[136] Eduardo Merino Machuca. Design and implementation of an agent-based social simulation model of energy related occupant behaviour in buildings. 2017.

[137] Farhad Ghayour and Diego Cantor. *Real-time 3D Graphics with WebGL 2: Build Interactive 3D Applications with JavaScript and WebGL 2 (OpenGL ES 3.0)*. Packt Publishing Ltd, 2018.

[138] Joshua McClymont, Dmitri Shuralyov, and Wolfgang Stuerzlinger. Comparison of 3d navigation interfaces. In *2011 IEEE International Conference on Virtual Environments, Human-Computer Interfaces and Measurement Systems Proceedings*, pages 1–6. IEEE, 2011.

[139] Paolo Burelli. Game cinematography: From camera control to player emotions. In *Emotion in Games*, pages 181–195. Springer, 2016.

[140] Gerhard Reinelt. Tsplib95. *Interdisziplinäres Zentrum für Wissenschaftliches Rechnen (IWR), Heidelberg*, 338:1–16, 1995.

[141] Barry Lord and Maria Piacente. *Manual of museum exhibitions*. Rowman & Littlefield, 2014.

[142] XP Wang, M Wang, JH Ruan, and Y Li. Multi-objective optimization for delivering perishable products with mixed time windows. *Advances in Production Engineering & Management*, 13(3):321–332, 2018.