

Bachelor Degree Project



AUGMENTED REALITY FRAMEWORK FOR SUPPORTING OPERATORS AND MONITORING OPERATIONS DURING MAINTENANCE ENVIRONMENTS IN INDUSTRIAL

Bachelor's Degree Project in Automation Engineering Bachelor Level
30 ECTS
Spring term 2018

Leire Amenabar Echave
Leire Carreras Orobengoa

Supervisor: Raquel Quesada Díaz
Examiner: Kanika Gandhi

Abstract

In an ever-changing and demanding world where short assembly and innovation times are indispensable, it is of paramount importance to ensure that the machinery used throughout the whole process of a product are in their best possible condition. This guarantees that the performance of each machine will be optimal, and hence, the process times will be the shortest possible, while the best quality products are obtained. Moreover, having a machine in an impeccable status permits making the necessary changes to it, in order to fulfil the requirements that a more advanced or complex product may have. Maintenance operations and their corresponding trainings have historically been time-consuming, and a vast amount of information has been transmitted from an expert to a newer operator. This means that there has been the need of working with experienced operators to secure that a good service is provided. However, different technologies like augmented reality (AR) have been shown to have a positive impact in the support and monitoring of operators in industrial maintenance operations.

The present project gathers information in regard to the framework of AR, with the aim of supporting and monitoring operators in industrial environments. The proposed method consists on the development of an artefact, which would lead to a possible improvement of the already existing solutions. It is believed that the development of an AR application could grant the necessary aid to any operator in maintenance operations. The result of this suggestion is an AR application which superimposes visual information on the physical equipment.

Keywords: augmented reality, maintenance, travelling salesman problem, expert systems, monitoring, database, QR code.

Acknowledgements

The research and work behind this project was carried out during the spring semester of the year 2018 at the Department of Automation Engineering of the University of Skövde, Sweden.

First and foremost, we gratefully acknowledge the support of our main supervisor Raquel Quesada Diaz for her encouraging, great support and patience throughout the duration of this project. Your guidance has been fruitful and rewarding.

In addition, we want to thank our examiner Kanika Gandhi for her support, advice and the opportunity to perform this project.

Furthermore, we wish to express our sincere thanks to the University of Skövde, for providing us with all the necessary facilities for the research.

Moreover, we would like to express our gratitude to Rasmus Willén, research assistant, for his help regarding AR technologies and C# programming.

This work was partially financed by Sweden's Innovation Agency (Vinnova), Sweden, through the project Virtual water and wastewater competence in northern Sweden (VVA 2). We gratefully acknowledge their provision of research funding and the support of the industrial partners Vatten- och Avfallskompetens I Norr AB (Vakin). We want to thank our industrial mentor Anna Dietrich, we felt very welcomed during our stay at the company. Special thanks go to Vakin's maintenance operators for their collaboration and insights during the testing phase of the project, we really appreciate your help and expert counselling..

We also place on record, our sense of gratitude to one and all, who directly or indirectly, have lent their hand in this venture

Last but not least, we would like to thank our family and friends for supporting us in general, and in our work and studies in particular. Thank you all for being there at all times.

Skövde, May 2018

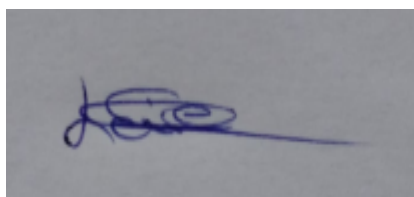
Leire Amenabar and Leire Carreras

Certify of Authenticity

Submitted by Leire Amenabar Echave and Leire Carreras Orobengoa to the University of Skövde as a Bachelor degree thesis at the School of Technology and Society. We certify that all material in this thesis project which is not our own work has been identified.

University of Skövde, 2018-06-01

X



Leire Amenabar Echave

X



Leire Carreras Orobengoa

Table of Contents

1	Introduction	1
1.1	Background.....	1
1.2	Goals.....	2
1.3	Limitations.....	3
1.4	Sustainable development	3
1.4.1	Environmental sustainability	4
1.4.2	Economic sustainability.....	5
1.4.3	Social sustainability.....	5
2	Methodology.....	6
2.1	Research Methodology	6
2.2	Process Diagram.....	7
3	Frame of Reference	9
3.1	Augmented Reality	9
3.2	QR code reader	9
3.3	Expert systems.....	11
3.4	Database	12
3.5	Travelling salesman problem.....	13
3.6	Maintenance	13
3.6.1	Corrective	14
3.6.2	Palliative	14
3.6.3	Preventive	15
4	Literature Review	16
4.1	AR Applications for inspection and maintenance	16
4.2	Expert systems.....	17
4.3	QR code reader	17
4.4	Review Summary	18
5	Reasons why AR should be chosen.....	26

5.1	Comparison between maintenance methods.....	26
5.2	Comparison between emerging technologies	27
6	Selection of the framework.....	29
6.1	Comparison of different frameworks	29
6.1.1	Language	29
6.1.2	SDK	31
6.2	Selection of framework	32
6.2.1	Language	32
6.2.2	SDK	33
7	Demonstrator development.....	34
7.1	Design and development of the demonstrator	34
7.2	Analysis of the insertion of the prototype in the industry	44
7.3	Test to validate the prototype	46
7.3.1	Test 1	46
7.3.2	Test 2	48
7.3.3	Test 3	49
7.4	Results	51
8	Conclusions	52
8.1	Summary.....	53
8.2	Discussion.....	54
8.3	Conclusions	55
8.4	Future work	56
9	References	59
Appendix A:	Collected data from the tests	65
1.	Data of participants.....	65
2.	Paper format tests	67
3.	Audio format tests	68
4.	AR in tablet format tests	70

5. AR glasses format tests	71
6. Feedback about the prototype.....	72
Appendix B: Table of all the alarms of the line	74
Appendix C: Screen captures	75
1. Login screen	75
2. QR code reader screen.....	75
3. All line State display screen	76
4. List of alarms screen.....	76
5. Preventive or palliative choice screen	77
6. All line screen.....	77
7. Machine's menu screen	77
8. Display information screen.....	78
9. Steps display screen.....	78
10. Camera screen	79
11. Comparison of the images screen.....	79
12. Pdf capture.....	80
13. Last display screen	81
Appendix D: Scripts of the program	82
1. Login of the program.....	82
2. Start point QR code reader	83
3. Start maintenance screen script	83
4. Alarms display script.....	93
5. Warning case script	96
6. All line script	97
7. Machine 1 QR read script.....	108
8. Machines menu script.....	110
9. Display information script.....	112
10. Steps display	115

11.	Camera script.....	124
12.	Compare photos.....	124
i.	Print the photo taken by the user	124
ii.	Show the image of the machine in correct state	126
13.	Creation of the pdf.....	127
14.	Last Display	131
15.	Database	131
16.	Rotate objects	134
i.	X axis.....	134
ii.	Y axis.....	135
iii.	Z axis	135

Table of Figures

Figure 1.1 Sustainability spheres (Kurrry, 2011).....	4
Figure 2.1 Process diagram of the study.....	8
Figure 3.1 QR code reader (Yanachkov, 2017).....	10
Figure 3.2 ES structure (Otair, Hamad and Jordan, 2005)	11
Figure 5.1 Comparison of LCD, HUD, AR (Henderson and Feiner, 2009).....	27
Figure 6.1 How to type in C++ (Anon., 2018)	30
Figure 6.2 How to program in Java language (Anon., 2018)	30
Figure 6.3 How to program in C# language (Anon., 2015).....	31
Figure 7.1 Machines flow analysis.....	35
Figure 7.2 Prototype machines flow.....	35
Figure 7.3 General flowchart of the program	36
Figure 7.4 Structure of login file	37
Figure 7.5 Start point QR code.....	38
Figure 7.6 Variables comparison treatment.....	38
Figure 7.7 Alarm display	39
Figure 7.8 All line display	41
Figure 7.9 QR code reader code example. Machine 1.....	41
Figure 7.10 Display information example. Machine 2.....	42
Figure 7.11 First maintenance step.....	42
Figure 7.12 Second maintenance step	43
Figure 7.13 Internal structure of created program	44
Figure 7.14 Wooden structure used for the test. Machine 1	46
Figure 7.15 Working field graphs. Test 1.....	47
Figure 7.16 Working field graphs. Test 2.....	49
Figure 7.17 Working field graphs. Test 3.....	50

Index of Tables

Table 4.1 List of AR papers and their contents. Part I	19
Table 4.2 List of AR papers and their contents. Part II	20
Table 4.3 List of AR papers and their contents. Part III.....	21
Table 4.4 List of AR papers and their contents. Part IV.....	22
Table 4.5 List of AR papers and their contents. Part V	23
Table 4.6 List of AR papers and their contents. Part VI.....	24
Table 4.7 List of AR papers and their contents. Part VII	25
Table 6.1 AR SDK comparison (Anon., 2012)	31
Table 6.2 Licenses for Unity (Anon., Last edited 2018)	32
Table 7.1 Variables state range.....	38
Table 7.2 Machine states priority levels.....	40
Table 7.3 Devices price comparison.....	45
Table 7.4 Number and age of participants. Test 1	47
Table 7.5 Results. Test 1	47
Table 7.6 Number and age of participants. Test 2.....	48
Table 7.7 Results. Test 2	49
Table 7.8 Number and age of participants. Test 3.....	50
Table 7.9 Results. Test 3	50
Table A.1 Data of participants.....	65
Table A.2 Each machine maintenance times. Paper format	67
Table A.3 Average times for each machine. Paper format.....	68
Table A.4 Average time for the total maintenance and per machine and mark. Paper format.....	68
Table A.5 Each machine maintenance times. Audio format	68
Table A.6 Average times for each machine. Audio format	69
Table A.7 Average time for the total maintenance and per machine and mark. Audio format	69
Table A.8 Each machine maintenance times. AR in tablet format.....	70
Table A.9 Average times for each machine. AR in tablet format	71

Table A.10 Average time for the total maintenance and per machine and mark. AR in tablet format	71
Table A.11 Each machine maintenance times. AR glasses format	71
Table A.12 Average times for each machine. AR glasses format	72
Table A.13 Average time for the total maintenance and per machine and mark. AR in tablet format	72
Table A.14 Usability marks given to the prototype.....	73
Table A.15 Average usability marks given to the prototype	73
Table B.1 List of alarms	74

Terminology

A

A ² R	
Adaptive Augmented Reality	17
AMIS	
Asset Management Information Services	14
AR	
Augmented Reality	i
ARES	
Augmented Reality Expert Systems	16

D

DBMS	
Data Base Management System	12

E

ER	
Entity Relationship	12
ES	
Expert Systems	1

G

GPS	
General-Purpose Problem Solver	11

H

HUD	
Head Up Display	9

I

IAR	
Intelligent Augmented Reality	22
IMS	
Information Management System	12

IT

Information Technology	6
------------------------------	---

L

LCD	
Liquid Crystal Display	27

Q

QR	
Quick Response (QR code)	2

R

RDBMS	
Relational Database Management System	12

S

SDK	
Software Development Kit	29
SEAR	
Speech-Enabled Augmented Reality	16
SQL	
Structured Query Language	12

T

TSP	
Travelling Salesman Problem	9

U

URL	
Uniform Resource Locator	9

V

VR	
Virtual Reality	9

1 Introduction

In an ever-faster world, the improvement of the efficiency of the production lines in industries is a key factor that helps to ensure the future of the company and its expansion. To achieve this goal, it is of preeminent importance that both the machinery and materials used in the production lines are in their optimal condition. In addition, it is important to find the most time-consuming parts of the production line, so that these can be modified to reduce the waste of time as much as possible during the process. In general terms, one of the most time-consuming elements in the industry is the maintenance work. This is due to the time that the operators need to learn the procedure before they perform work correctly and also because while the operators perform the job no products are produced, so the time needed to do the maintenance has not a direct positive impact in the profits of the company. The present project describes the fundamentals about the implementation of AR in industrial environments and illustrates how this technology could support the operators and optimise the maintenance operations by the enablement of the superposition of dynamic step by step information over the real world at essential locations of the production line.

1.1 Background

Industrial maintenance, referred to the conservation of a manufacturing environment in optimal conditions, is of paramount relevance nowadays. Although historically it has had its importance, with the arrival of Industry 4.0, which is based on data exchanging technologies, it is necessary to guarantee the quality of all the components. Besides, it is more sustainable to repair a machine than replacing it by a new one, whenever this option is possible, as both energy consumption and raw materials exploitation are reduced noticeably this way. In order to ensure this, an appropriate maintenance procedure has been accomplished. Enterprises need to train their employees and this practice requires a big sum of resources, from the need of expert personnel to high economic costs. The usage of AR in such operations has shown acceptable results, based on efficiency and ease of use. Thus, AR has been broadly used at industrial environments for decades. However, it has not been until recently that this technology has overcome most of the limitations it faced in the past, as the lack of precision. Furthermore, thanks to these improvements AR has evolved from being an interactive technology to an adaptive one, which enables its integration with other spotlight technologies, such as expert systems (ES).

This project studies AR and ES and analyses how the combination of both can be used for the provision of support to operators at maintenance operations in industrial environments. A demonstrator as a proof of concept will be implemented. This one will superimpose 3D models and text elements so as to provide the operators with clear instructions. Moreover, by the incorporation of ES to the system architecture, the user will not need to choose which path to follow when doing a maintenance operation, since the artificial intelligence will designate it, based on various factors, as the status of the machines or their location, which will make the procedure more efficient.

1.2 Goals

The main objective of this project is to propose a solution that could increase the efficiency of industrial maintenance operations. In order to achieve this goal, an AR framework has been developed which will allow to show real time information read from a quick response (QR) code located near a machine by the use of Android compatible AR devices, such as AR glasses, tablets and mobile phones. Besides, the AR framework will calculate the shortest possible path to be followed by the operator, so that the maintenance is done as fast as possible, once the state of all the machines have been checked. The framework will also monitor the maintenance operation by the creation of a document when the operation has been completed. The program will be suitable for any Android device which fulfils the minimum requirements and specifications needed to utilise the software, for example a camera or a compatible Android operating system (OS), an appropriate version of Android for the correct operation screen where information should be displayed, among others.

The information provided in reference to the different objects in the line will be real time data, so a connection between the AR device and internet must be implemented. In case there is an error in the line at the beginning of the maintenance, a warning will appear in the AR equipment, and a change in the normal route will proceed to prioritise the maintenance of the faulty machine.

Basically, the functions that will be performed in this project are the next ones:

1. Research of information in regard to main topics and find their application in industrial maintenance operations: AR devices implementation in maintenance, Android compatibility and QR codes.
2. Select the most suitable framework of implementation for this specific project, so that it can be properly implemented.
3. Make the design of the software and develop a demonstrator which incorporates AR

technologies. To achieve this task, a program needs to be built, which allows to create standard structures about a component of a machine. Another characteristic will be that created program has to allow to acquire information about defined points in the company, which are important when it comes to maintenance works. This means that a general template which allows to obtain information needs to be generated, in order to facilitate the traceability of the state of the component. The program should be, as mentioned before, compatible for different AR devices as mobile phones, tablets and AR glasses. At least, the program has to specify the path to be followed by the operator, this includes the steps that the operator has to perform, that will ensure a proper maintenance work.

1.3 Limitations

There are some limitations that ought to be taken into account in this project. A proof of concept in a form of a demonstrator for the AR and ES will be implemented, which will confirm its usability, by the validation of the correct choice for the framework. This prototype may be unfinished, and it will not be delivered as an early version of the final product. This way, its promising utilisation features will not be tested. Moreover, even if the project aims to propose a solution that will be compatible for AR glasses, mobile phones and tablets that are based on Android platform, some devices, which have not an updated software, may not be compatible, since the development will be pursued in one of the latest versions of this software. In addition, there will not be any production and usage costs section included in this inform, since this thesis is a preliminary study.

1.4 Sustainable development

Sustainability is the ability to be maintained in the extended future without having a negative impact on future generations. This is a term that has raised concern lately due to the population growth, which has led to a faster degradation of the environment and shortage of natural resources, in extreme cases (Willard, 2010).

In order to implement a sustainable artefact, the three fields shown in Figure 1.1, which are economic, social and environmental sustainability have been considered, as will be explained in the next sections.

In this study, which consists on the provision of support and monitoring operators during maintenance operations, the development of an artefact will be done, as a proof of concept. For this development, there is no material to be chosen, so the way of analysing the sustainability of the proof of concept is

to treat it as a service. This way, the service should respect the three sustainability types, environmental, social and economic ones, in order to provide a sustainable solution.



Figure 1.1 Sustainability spheres (Kurry, 2011)

1.4.1 Environmental sustainability

Humans need natural resources for almost everything, from nutrition to fuel for transportation. As a consequence of this dependency, a massive impact on the environment is effected (Mason, 2018). This requires actions to be taken in order to give nature time to re-generate and avoid shortage of natural resources. This will be taken into account during the development of this project.

First, by the inclusion of an internal operator; i.e. an employee in charge of doing the maintenance work, instead of hiring an external company to do so, pollution and resource exploitation related to transportation will be reduced. Even though the transportation will still be needed to obtain a replacement for the discarded or broken machine parts that should be changed, However, it is thought that shipment needs will be considerably reduced.

On the other side, machines which are in optimal condition contaminate significantly less than those which are not properly maintained. This is due to the fact that when a machine is maintained in a good condition, the probabilities of a leakage or any other polluting problems are considerably smaller.

1.4.2 Economic sustainability

As the name says it, this type of sustainability has its focus on the capital. Its aim is to ensure that the resources are conserved for future generations. The way of achieving this is by having a slower or equal rate of consumption than the production and regeneration of resources.

This project aims to contribute to the economic sustainability by the avoidance of the replacement of broken machines. One of the reasons why maintenance operations are so important is that the waste time, and hence the costs, should be minimal for a company. If whenever a machine is not broken it would be replaced by a newer one, the rate of resource consumption would be higher than the regeneration of resources, together with the high economic cost that this would imply to the company.

Moreover, as stated in paragraph 1.4.1, machines which are not in their optimal state tend to have a higher rate of pollution, which means that their energy consumption is higher. By the avoidance of this situation, companies would also save expenses in regard to energy sources.

1.4.3 Social sustainability

Societies should be maintained in the future for the preservation of human kind, together with the peace and sovereignty. With this purpose, law and order ought to be conserved (Anon., 2018).

This project does not have a special concern on the social sustainability, since it is believed that this is something that should be analysed in a company level, and not in the maintenance level. Anyhow, the artefact will be developed in a way that the basic social values are respected, by the evasion of any type of information that is believed to be harmful for ethnical, economic, social, gender related or sexual reasons.

2 Methodology

The general goal of the project is to create a framework which integrates AR devices in maintenance operations. To achieve this goal some steps have been followed.

In this chapter the reader can find the research methodology that has been used to address the result of the dilemma which has inspired the present study. The required guidelines to achieve the objectives of this project will be provided by the methodology proposed in the following paragraphs.

2.1 Research Methodology

In order to undertake the main research question of this project, a global research perspective has to be chosen. As reported by Oates (2006), there is commonly a research method for each inquiry. Nevertheless, it could be the case in which the global inquiry technique is built with more than one research procedures. The following paragraph introduces an inquiry of the chosen research technique and its appropriateness for the current thesis.

As described in the preceding paragraph, a generic objective is performed in this study, which is the explanation of an AR framework that permits the application of AR technology in maintenance operations with the goal of simplifying the tasks completed by the operators. With this objective, the target of the inquiry technique will be the development of a new information technology (IT) device through the study of several methods as mathematical proof or proof by demonstration, among others. Once that this is done, the choice of a suitable proposal for the application and creation of the AR software for maintenance operations will be made. Another important point of the project is the build-up of a demonstrator which validates the convenience of the usage of AR devices in industrial maintenance environments. As a consequence, the research methodology that will be used is design and creation strategy.

This type of research scenario shares its main target with this project, which is to develop new artefacts; that is to say, to build up IT products. In this case of study, various sources have been used as base, which means that it contains methods and theories that have been used in other studies. In any case, a new domain with the use of Unity and Vuforia has been introduced to the application of AR for maintenance operations. Thus, the application itself is the contribution to knowledge.

Furthermore, a survey system has been utilised in order to adequate the project to the operators' needs using as a base the feedback received from users with different expertise levels. Also, an extensive literature review will be included on this report, which will include previous research studies related to the field of application.

Documents have been the primary source of information through this project, which includes books, papers, research articles, etc. This data collection reports information about previous work in terms of methods or theories that have been used, how the interfaces should be done, or the data that is needed for the correct implementation, among others. Moreover, to be supported by former records might increase the reliability of this thesis.

Finally, with the purpose of validating the proposed solution, the developed demonstrator will be tested and adjusted until it is validated.

2.2 Process Diagram

Figure 2.1 shows how this project will be organised. First, a review on AR, ES and different technologies for maintenance operations will be included in order to collect knowledge about the solution that will be analysed and proposed in paragraph 7. This will be focused on an application of AR technology for maintenance operations. In the last part, an implementation will be done, by the usage of Android as a framework and finally, the results, together with the conclusions, will be presented.

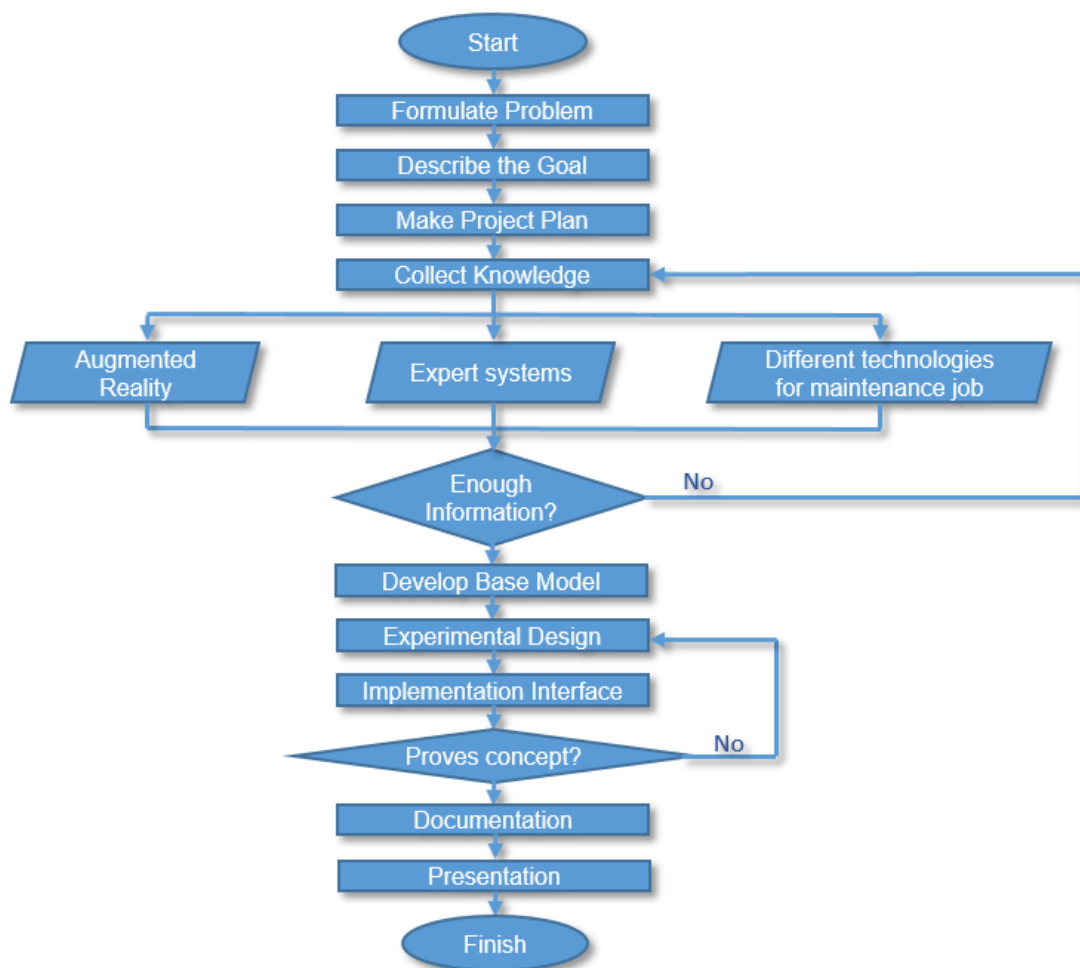


Figure 2.1 Process diagram of the study

3 Frame of Reference

The purpose of this chapter is to describe the hypothesis and theories that are related to the different fields of this project. These fields are AR, QR, ES, Database, travelling salesman problem (TSP) and maintenance.

3.1 Augmented Reality

AR is a technology which superimposes computer made information, in the form of images, video or text, among others, with real time perceptions, such as visual or auditive, of the user. The first existing reference to AR was recorded by Baum (1901) in the description of the “Character Marker”. This was described in the novel *The Master Key*, which is a fiction novel. Nevertheless, the first real application of AR was performed by Sutherland (1968), who developed the first head-up display (HUD) system. This system used computer-created graphics, to show a simple wireframe, which is a 3D model that only includes the lines and vertices that represent the skeleton of the figure to the users. In 1974, Myron Krueger built a laboratory which had projectors with video cameras that transmitted onscreen shades, called Videoplace. This laboratory got users into an interactive atmosphere. The term “Augmented Reality” was created by Tom Caudell in 1990. Since then, different applications have been developed by the usage of this term to describe their work. The first application which used this name in it was performed by Heilig (1992), when the creation of a virtual reality (VR) machine was done, which was able to display five short films created by himself. One of the first applications, which had the maintenance of a manufacturing industry as its main target, was created by Feiner, Macintyre and Seligmann (1993), who developed a knowledge based head mounted system, KARMA, which guided the operator through the maintenance process of a broken printer. Another case where AR was applied in industrial maintenance context was achieved by Nakagawa, Sano and Nakatani (1999). In their research the authors proposed a colour-video based system for the maintenance of the machinery in a production plant. They also conclude that the introduction of AR technologies in industrial maintenance operations, means an increase in production efficiency. This is caused by the loss of the need of studying the procedure steps by the operators.

3.2 QR code reader

QR codes consist of a two-dimensional square which has smaller black and white squares inside it. This is read by different devices as mobile phones, tablets, etc. with the use of their cameras, as it can be seen in the Figure 3.1, and is commonly used to store uniform resource locators (URL) or other

type of information, although other type of information, such as payment codes or website logins, can be stored in these barcodes.. Each QR code is a unique identifier, and the three distinctive squares located in the outside part of it are used for the correct location of the code for the reader device. The corner that remains is used for the normalisation of size, angle of view and orientation. The quantity of data to be stored in each QR code depends on the version, datatype and error correction level of the code. The maximum storage of one of these codes is of 7,089 numeric digits.

In the present project, QR codes will be used to link different plant machinery with their corresponding information. When the operators reach a point of the line, which has one of these markers, they will focus on the QR code with the camera of the AR device, and it will detect and read the marker, and consequently, show the corresponding data superimposed over the real world on its screen. This technology is easy to implement and has low computational work, as explained by Oliveira and Porto (2016). Consequently, its usage in a wide variety of fields has increased a lot in the past years. The first connection between QR codes and AR technology was done by Gutierrez et al. (2013), who based his selection in the versatility and variety of the QR codes.



Figure 3.1 QR code reader (Yanachkov, 2017)

Moreover, a 3D image can be displayed by the use of this code, as Ruan and Jeong (2012) demonstrated. This function will be introduced in this project to show the physical appearance of the plant machinery to the operator, so that there is no doubt when it comes to the identification and location of the part of the line that needs to be maintained.

In the case scenario of this project, the main objective is to define a framework which allows to display plant machinery real time information, present the steps which should be followed to perform maintenance operation and describe the path that should be tracked by the user.

3.3 Expert systems

ES are computer programs built with the aim of imitating the behaviour of an expert human during the decision-making process or while a solution to a problem is needed. This is achieved by the application of artificial intelligence technologies. In order to develop an ES, accurate information about the domain and the strategies for the employment of the data to problem solving is needed. The construction of an ES requires the information needs to be correctly defined, represented and employed depending to the problem-solving method used. It is not easy to find a person who has detailed expert knowledge and programming skills, which are necessary to perform an appropriate ES, since expert operators tend to be skilful in the operations they execute regularly, and programmers commonly do not have experience in all the different operations that could be performed in an industrial line. Therefore, this labour is generally divided between an expert and a computer scientist. The expert formulates his knowledge and the computer scientist programs the suitable software that includes the problem-solving methods applied. However, the most important part of the ES is to create an interface between the general problem-solving method and the specific problem which requires a solution. The unique difference between ES and conventional programs is that the former obtains its knowledge from experts while ES generate knowledge based on facts and rules, as can be seen in the Figure 3.2.

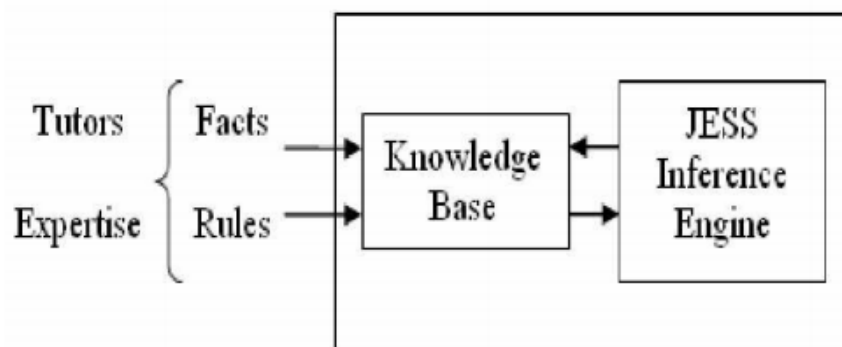


Figure 3.2 ES structure (Otair, Hamad and Jordan, 2005)

In the early mid-1960s the General-Purpose Problem Solver (GPS), considered the predecessor of ES, was created. In the mid-1970s several real ES emerged, but the efforts made to solve general knowledge-based problems were still not widely developed. Besides, several ES worked because the most important part of an ES is not its particular formalism or inference schemes, is the specific knowledge that it possesses. When the early 1980s arrived, ES technology started to be commercialised and programming tools and shells appeared, which have been very successful and are still used (Belavkin, 2017).

Since then, ES technology has improved considerably, and several AR-ES applications have appeared. For example, one of the first times that ES was used together with AR technology for maintenance duties, was when Feiner, Macintyre and Seligmann (1993) employed IBIS, a knowledge-based component for the support in maintenance operations. Another case is the research made by Jo et al. (2014), that focused on how to manage AR-interpretations concerning maintenance job.

3.4 Database

The origin of the term database date back to when databases were made by government offices, libraries, business organizations, and hospitals to compile information. Some of the principles used in that time are still used nowadays.

When it comes to a computerised database, it can be described as a structured amount of data which is stored in a computer. The data is organised into rows, columns and tables, it is also listed so that the search of information becomes easier. The data available in the database can be updated, modified and eliminated as new information is added (Anon., 2018). The first computerised data base management system (DBMS) was initiated in 1960, when computers where more available for private institutions, due to the decrease of their price. The most known data models created during this decade were CODASYL and a hierarchical model called information management system (IMS). This last DBMS was created for the Apollo program and had similar concepts to CODASYL (Chebli, 2009). Both databases, in view of the way the data was accessed, became known as navigational databases. Rullo, Sacca and Zhong (1986) demonstrated this, by the description of an approximation algorithm for a physical access path was elaborated.

In the following years, a huge investment was done in this area, which encompassed several DBMS's and different coding languages which include entity-relationship (ER), relational database management system (RDBMS) or structured query language (SQL), which became the standard query language.

Furthermore, in the early 1990s the database industry had an economic crisis and most of the remaining companies sold their complex databases for a high price. In the mid-1990s, with the advent of the internet, database industry faced an exponential growth. This accomplished the use of client-server database systems by the average desktop users, so that they could access computer systems that contained heritage data. Despite of the decline experienced by the internet industry at the beginning of the 21st century, some database applications were developed (Chebli, 2009).

Nowadays the use of DBMSs is part our everyday life with the use of internet and other types of DBMSs. As a result of the increase in the use of electronic devices, which use DBMS, there is a big investment for the increase of the capability of the data base structures, as well as for other applications.

3.5 Travelling salesman problem

TSP is an algorithmic problem focused on optimization. It is widely used in computer related fields, and it is based on finding the shortest path given a number of points.

This problem was created in the 1800s by W.R Hamilton and Thomas Kirkman, and Irish and a British mathematician, respectively. Since then, various attempts of finding a solution to the mentioned problem have been numerous.

The problem is inspired by the idea of a salesman travelling to many cities in the optimal way, which means visiting all the cities by following the shortest path. Anyway, this problem has still not got an exact solution, since more than one answer is acceptable. Hence, one of the possible solutions has been implemented in the present project.

This part of the program will be the one which acts as an ES, since it will simulate a human expert who knows the shortest path to be followed. However, it is not only the distance that will be taken into account for calculating the best path, but also the priority associated to the machine or the specific failure, as will be stated in paragraph 7.1.

3.6 Maintenance

Maintenance is the process of conserving a status or situation. In industry, maintenance operations are performed to ensure the correct functionality of the machines and to increase the lifecycle of the machines. Even if each maintenance department affronts diverse obstacles, all of them have the same objective, which is to minimise the downtime.

At the beginning of the industrialisation, there was not a need for maintenance, since the production of goods was done on small scale. The majority of the early factories, once they were mechanised, used to employ unskilled workers, so maintenance issues were fixed by the engineers and owners. However, as technology continued its development path, more types of machinery were introduced, and this led to the need of trained personnel to solve the problems. In the book Engineering Maintenance Management (W. Niebel, 1994) it is explained that in 1969 the industries maintenance

department only constituted 1-17% of the work. However, with the use of automated machinery this percentage has increased, because the machine-work replaced manual labour. Maintenance routines prolong the equipment and machinery life and this can end up as a significant cost saving for the company

In addition, maintenance departments have *“an effect on the marketability of the product and can be a factor in the change in future product demand”* (Wilson, 2002, pp. 189-192). Although the maintenance work isn't directly related to the production of the company, it has an effect on it. R. Jones did an analysis of 725 asset management information services (AMIS) and concluded that over 60% of the companies were not convinced with their systems and were not able to validate the benefits of the use of the systems, since not lasting benefits were reported (Jones, 1994). Unluckily, the fundamental agent of this disappointment is in general, a fault of planning and dedication to the employment of the system from the beginning instead of the faults of the system itself.

It is important to stress that when talking about maintenance operations, a simple concept of maintenance is thought of. However, a different type of maintenance operation should be pursued to a machine, based on its state and the priority that the operator wants to give to it. Even if the classification differs from source to source, in the present project three types of maintenance will be considered, and a description of each one is provided in the subsequent paragraphs.

3.6.1 Corrective

It is the type of maintenance that should be done whenever a machine is faulty and hence not working properly. In this type of maintenance, the aim is to identify the problem, isolate it and correct it, in order to make the machine valid again. Thus, when there is the need of executing a corrective maintenance operation, this should be prioritised over the others.

3.6.2 Palliative

Also known as run-to-fail maintenance, consists on making the machine work as long as possible, without going to the root of its problem. This means that the operator just makes changes in order to make the machine run until failure. In the present project, this type of maintenance has been considered the medium priority one.

3.6.3 Preventive

It is the type of maintenance that takes into account the aging of the machine and its special requirements and defines a period of time with a particular procedure, in order to prevent the machine from suffering failures. Hence, it is common to find preventive maintenance schedules on industrial lines, by the definition of when each machine's maintenance should be pursued.

4 Literature Review

The idea of AR was first recorded by Baum (1901) in the description of the “Character Marker”, which are a set of spectacles that indicate people’s character to the person who wears them, this reference is done in the novel called *The Master Key*. However, the person who turned this fictional idea into reality was Heilig (1992) who created the world’s first VR machine called “Sensorama Machine”. Since this creation was performed, AR technology has had a huge improvement and it is a promising technology at many fields. In the following paragraphs, a literature review about different AR applications and their affinity to industrial maintenance processes is provided.

4.1 AR Applications for inspection and maintenance

As explained in section 1.1 the main objective of industrial maintenance is to minimise line breakdowns and to keep the plant in a good working condition with the least possible impact in the company’s economy. The relevance of the plant maintenance depends on the type of plant and its production. To achieve these purposes, the implementation of AR devices in industry has been performed for several years, so that operators’ maintenance job becomes easier and the need of learning process for the performance of the activity mainly disappears. The first industrial AR application date back to when a demonstration for Boeing was performed in order to facilitate assembly operations by Caudell and Mizell (1992). Nonetheless, one of the first references with respect to the usage of AR devices in maintenance operations were Nakagawa, Sano and Nakatani (1999) in the work called “Plant maintenance support system by augmented reality” where they introduced a colour-video-based system design for the maintenance of a plant machinery. The authors suggested that the system gave visual advice of which steps should be followed by an operator for a correct maintenance to be done. After this, many maintenance applications have been developed for different backgrounds and with various purposes. For instance, Klinker et al. (2001) built up an application for AR devices in power plants, while Hincapié et al. (2011) took a different approach, since they focused on aeronautical maintenance. Together with reference to for maintenance operations, Nakajima and Itho (2003) described a support system for maintenance training. Besides, the combination of AR with other technologies has led to the creation of new applications, such as Speech-Enabled Augmented Reality (SEAR), performed by Goose et al. (2002) or AR together with ES (ARES) performed by Syberfeldt et al. (2016). All these authors, together with others, such as Henderson and Feiner (2011) and Syberfeldt et al. (2015) reach the conclusion that AR technology is intuitive and satisfying, but they all mention limitations, like the need of mechanics for changing tasks from different points of

observation (Henderson and Feiner, 2009). Nevertheless, a more recent study (Martinetti, Rajabalinejad and Van Dongen, 2017) referred that although this technology was already widely used, it still had a place in a more complex future industry, as proved by Masoni et al. (2017) where the authors made a research of AR maintenance application in industry 4.0, and reached the conclusion that past issues and limitations are almost over.

4.2 Expert systems

It refers to a computer system that acts like a human with expertise in one or various fields so as to make a decision. Feiner, Macintyre and Seligmann (1993) employed IBIS, a knowledge-based component, which was capable to decide which objects should be displayed by AR technology, based on the operator's activity, in a maintenance duty. Later, Jo et al. (2014) made a similar research in order to guide AR-renderings, also with a focus on maintenance. Apart from these applications, the association of AR with adaptive systems has also been researched in artistic fields, since Damala et al. (2012) pointed out that AR systems' next step was what they called Adaptive Augmented Reality, (A^2R), which could be useful, for example, to determine what the interest of the user in regard to one piece of art in a museum is. Also focused on this idea, Xu et al. (2012) used A^2R in a cultural application too. These findings could be applied in maintenance-workers, since bio-signals could be used for distinguishing whether an operator is tired, and thus prevent an accident to happen, as Doswell and Skinner (2014) conclude. Also focused on the next steps of AR, Stricker and Bleser (2012) explained that the change from interactive to adaptive AR systems should be done, so as to satisfy the user's needs, but it also stated that in order to achieve this, many improvements needed to be done, with reference to workflow and action recognition.

4.3 QR code reader

To be able to read information from different points of the line so that maintenance work can be performed, markers should be used whenever it is possible, due to their easy implementation, combined with the reduced computational work Oliveira and Porto (2016). With this purpose, QR codes, by reason of their versatility and variety, are a good option for AR in maintenance works, as it can be seen in an article by Gutierrez et al. (2013) which suggests an application of QR codes for the creation of AR background.

In another research QR codes were also used as indicators for Android Smartphones in an AR application, which showed up a 3D object on the marker (Ruan and Jeong, 2012). When it comes to

the design and implementation of AR systems together with QR, Wang et al. (2010) combined two technologies and got a prototype system called QRAR.

4.4 Review Summary

In accordance to the present literature review, AR is believed to be a viable technology for maintenance operations. This technology has been widely used in industry for decades and has also led to a proper training of operators, which reduces the errors and costs that traditional training imply (Nakajima and Itho, 2003). In addition, this interface has been combined with different ones, and concepts such as A²R or ARES have been introduced (Xu et al., 2012; Syberfeldt et al., 2016). Although limitations are mentioned in various previous researches, limitations from the past are believed to be almost inexistent (Masoni et al. 2017). Hence, the presented authors show a conformity with this technology, and declare that it is intuitive and easy to use.

AR technology will be used for the maintenance and training support and monitoring that will be proposed in the present study for the aforementioned reasons. As markers should be used whenever possible in AR applications (Oliveira and Porto (2016), QR codes will be placed over the plant to be maintained, and by the reading of these, information will be overlapped to the real world for the proper and effective maintenance operation.

A summary of the literature review is done in Table 4.1, Table 4.2, Table 4.3, Table 4.4, Table 4.5, Table 4.6 and Table 4.7.

Table 4.1 List of AR papers and their contents. Part I

Publication	Research Scope	Method	Limitations	Conclusions
Baum (1901)	An Electrical Fairy Tale, Founded Upon the Mysteries of Electricity and the Optimism of Its Devotees.		It is a fairy tale.	Defined glasses are capable of “reading” the personality of the people.
Caudell and Mizell (1992)	An application of HUD technology to manual manufacturing processes.	Description of the design and prototyping steps followed to do the implementation of the heads-up display	Extended tracking technology.	The use of HUD improves the efficiency and the quality of the work performed by the operator. Also, the complexity that carries on the use of AR devices.
Damala et al. (2012)	Adaptive AR for cultural heritage: ARtSENSE project.	Provision of a prototype that enables a personal experience in a guided visit to a museum.	Given information depends on the museum.	The system changes guidelines, suggests additional content or actions, based on the psychological state of the user.
Doswell and Skinner (2014)	Augmenting human cognition with A ² R.	Suggestion of a framework and description of a research methodology to achieve the main goal.		Described framework supports the creation of adaptive AR to evaluate and contextually readjust to the user’s environmental and cognitive state in real time. Additionally, its usability in medical domain is taken into account.

Table 4.2 List of AR papers and their contents. Part II

Publication	Research Scope	Method	Limitations	Conclusions
Feiner, Macintyre and Seligmann (1993)	Knowledge-based AR.	Description of the first steps to design and test a HUD which helps the user in maintenance operations.	The development of a formal model of how a user's performance will be affected by different decisions. The need of more advanced display technology.	With the help of these HUD users' job will be much easier when it comes to maintenance operations. Moreover, human errors will decrease by the use of this knowledge-based AR device.
Goose et al., (2002)	SEAR: Towards a mobile and context-sensitive speech-enabled AR.	Development of a framework, called SEAR. Also, description of how SEAR linked with a vision-based localization technique reaches a multi-modal user interface.	The SEAR prototype occupies a lot of space.	Although navigation and interaction in 3D can be difficult for the users, with the use of SEAR as its framework it becomes easier. Furthermore, some changes need to be done, so that when the user focuses on a specific sensor of the line is able to continue with the maintenance operations.
Gutierrez et al. (2013)	Application of contextual QR codes to AR technologies.	Presentation of an application which uses QR codes to generate AR environments. Moreover, an analysis of AR which support QR codes is done.	The AR device must be able to read the QR code by the use of a camera.	The system which is developed in this project demonstrates the possibility of implementing AR technologies in different contexts. Depending on the used context, the characteristics of QR codes allows to access the content from different experiences.
Heilig (1992)	World's first VR machine called "Sensorama Machine".	Description of the machine created by himself. Which displays five short films in stereoscopic 3D images.	Lack of financial backing to perform the patents, so the Sensorama machine was halted.	The Sensorama machine was a multi-sensory machine which was able to display 3D images, body inclination, provide stereo sound, and provide the user with wind and aromas which were triggered during the film.

Table 4.3 List of AR papers and their contents. Part III

Publication	Research Scope	Method	Limitations	Conclusions
Henderson and Feiner (2009)	Evaluation of the benefits of AR for task localization in maintenance of an armoured personnel carrier turret.	Design, implementation and user testing of a prototype of AR for military maintenance applications.	It is not a production ready solution. Therefore, the software does not reflect the needs of a production environment.	Difficulties to distinguish between body and head movements, when it comes to moving the image represented in the device depending users' movement. On the other hand, the prototype was able to prove that with the use of this artefacts, maintenance operations are done faster and more accurately than without them.
Henderson and Feiner (2011)	Exploration of the benefits of AR documentation for maintenance and repair.	Development of an experimental AR prototype for military area which helps with maintenance tasks.	The prototype is used as a proof of concept; it is not production-ready solution. The prototype needs a big physical space.	This AR application was able to find individual tasks in a maintenance series faster than the previous version of currently employed methods. Moreover, in this application head movements during task location were more accurate and sensitive.
Hincapié et al. (2011)	An introduction to AR with applications in aeronautical maintenance.	Presentation of examples of AR applications and demonstration of the feasibility of AR in maintenance tasks by the highlight of the advantages that this technology will introduce.	The main flaws slow down the expansion of AR in industrial environment. To overcome these limitations, better materials, faster algorithm, etc. are needed.	AR could improve human performances, this will carry out economic benefits, higher reliability, less failures and subsequent accidents, in case of car or airplane applications.

Table 4.4 List of AR papers and their contents. Part IV

Publication	Research Scope	Method	Limitations	Conclusions
Jo et al. (2014)	A unified framework for AR and knowledge-based systems in maintaining aircraft	Proposal of an intelligent augmented reality (IAR) system to minimize operation errors and time related costs and help with difficult tasks.	In case of having strong light variations, the object recognition performance starts to fail.	Used system, called IAR, involves vision-based tracking, annotation and recognition methods, which are needed to link information with images. Conjointly, provides a united resource framework.
Klinker et al. (2001)	Augmented maintenance of power plants: A prototyping case study of a mobile AR system.	Analysis of the information generation, retrieval, transmission, and visualization processes for maintenance operations in power plants. Furthermore, a little implementation work is done too.	The short time available for the implementation of the project and different problems regarding the processes to generate, access and transmit of the information.	With the implementation of the system, the focus was on four issues, which were directly related to AR and mobile aspects of the system; the linking of the information models, the definition and reuse of AR components, the multimodality of user interfaces and the mobility and unreliability of the network device.
Martinetti, Rajabalinejad and Van Dongen (2017)	Reflections on the adoptions of AR Through Problems and Opportunities.	Investigation about different possible application of AR technologies for assisting workers during maintenance operations.	As maintenance needs to be done as fast as possible, the amount of information that is going to be provided to the user has to be examined and selected.	When maintenance operators use wearable devices, the chances of blindly following given instructions are higher than of users who have learned the steps. The use of AR technology reduces human errors and increases occupational safety. In virtue of that, it can be applied in other industries.

Table 4.5 List of AR papers and their contents. Part V

Publication	Research Scope	Method	Limitations	Conclusions
Masoni et al. (2017)	Support remote maintenance in Industry 4.0 through AR.	Investigation and implementation of remote maintenance-based AR technologies.	Technology limitations, software and hardware limitations, and the bad use of AR technology.	It is the continuation of a previous research, so based on the feedback received from some end users' new features have been introduced. Description of the new version of the remote maintenance system and the principle which is behind it.
Nakagawa, Sano and Nakatani (1999)	Plant maintenance support system by AR.	Analysis, description and small implementation of AR system based in maintenance crews of plant equipment.	Time-lag between user's movement and the image on the screen. Also, because of the position of the markers, there were errors.	A function must be implemented to fix some problems in the system and to have real time plant parameters. To achieve these goals, there is a need of automatic situation recognition function which explains the situation the user finds out and what the user wants to do.
Nakajima and Itho (2003)	A support system for maintenance training by AR.	Analysis, development and evaluation of a HUD for maintenance training with object recognition.	Difficulties for the system to be used in daily maintenance operations in case of power facilities.	Developed system makes the object recognition job. To guide the user, images are displayed in HUD by Chrome-Key image.
Oliveira and Porto (2016)	AR system for maintenance of high-voltage systems.	Analysis, description and future trends supposition in case of AR for high-voltage system maintenance.	The need of using markers in many situations and the necessity of using 3D modelling in marker less situations.	Several experimental results are analysed, and the conclusion is that is important to use markers when it seems to be simpler and more comfortable to use. Another conclusion is that the use of AR in maintenance makes the work easier.

Table 4.6 List of AR papers and their contents. Part VI

Publication	Research Scope	Method	Limitations	Conclusions
Ruan and Jeong (2012)	An AR system by the use of QR code as marker in Android smartphone.	Analysis and description of different materials and methods to do QR code-based markers for AR applications.	Although QR codes have a lot of combinations, there are a limited number of possible codes.	The system has three important steps: The first one to capture the image, locate it and calculate the transform matrix, the second one to get the information from the marker and the last one to display the corresponding 3D image.
Syberfeldt et al. (2015)	Visual assembling guidance using AR.	Analysis, implementation by the development of a prototype, and a deep questioner about AR supporting assembly line workers who perform their tasks.	More than an implementation work, it is a questioner to find the points which have to be taken into account when it comes to the development of an AR system. So, a limitation is the number of tested people.	With the results they get from their questioner, they conclude that the tasks performed by the use of AR technology should be complex enough to feel that is worth using it. Also, the use of this technology must ensure the improvement in the efficiency and the system must be as perfect as possible.
Syberfeldt et al. (2016)	Dynamic operator instructions based on AR and rule-based ES.	Design and proof of concept of ARES technology, programmed in C# in Unity, with the use of Vuforia.	The developed ES do not automatically generate or modify rules. The main three AR devices (hand-held, head-worn and spatial devices) have limitations on the hardware.	It is possible to combine ES and AR in a successful way. For industrial applications, AR still has limitations that should be solved, with reference to its hardware.
Stricker and Bleser (2012)	From interactive to adaptive AR.	Analysis of existing interactive technologies and proposal of novelties for changing to A ² R.	3D scanning and modelling of scenes which are dynamic should be pursued in the future.	Advanced AR systems are presented, but in order to have a highly adaptive AR system, future work should be done in the fields of position capturing over large-scale environments.

Table 4.7 List of AR papers and their contents. Part VII

Publication	Research Scope	Method	Limitations	Conclusions
Wang et al. (2010)	Design and implementation of AR system in collaboration with QR Code.	Development of a system which uses QR code reading for AR applications to get the robustness that was not present when other types of markers were used.	Shorter effective range and tilt angle than conventional markers. Higher computational costs for the recognition of all the markers from the images.	QR has been proven to be useful and applicable in different fields. Therefore, future work will be pursued with the aim of broadening its applications and including it to handheld devices such as smartphones.
Xu et al. (2012)	An approach for using complex event processing for A ² R in cultural heritage domain.	Presentation and implementation based on iCEP framework of AR for obtaining A ² R.	If attention is divided some details are lost. Precision depends on time setting, and this depends on the person.	The results are believed to be positive. Even if there are changes that should be done, users are interested in the implementation of the technology. Both analysed parameters, delay time and accuracy rate are considered successful.

5 Reasons why AR should be chosen

As stated in paragraph 1, maintenance operations are indispensable due to the wear of machines and the fast-changing and demanding world. Same as industry, maintenance operations have also suffered changes in the last decades, and different technologies have been adapted in order to help maintenance operators in their respective duties.

In the next paragraph some of these technologies will be discussed, so as to demonstrate why AR technology has been chosen to overcome the problem in this project, over other technologies. Moreover, the performance of different AR devices will also be considered.

5.1 Comparison between maintenance methods

In the following paragraphs, traditional, audio-guided and AR technologies will be discussed, since these have been considered the three most common methods used for industrial maintenance operations in history. Together with that, these three technologies will be later analysed by the testing of them in the present project, in order to compare the effectiveness of each type of maintenance.

In the past, operators had the need of going through an intensive training period, or having the instructions physically with them, or attached to the machine to which the maintenance needed to be done, in order to know which steps should be followed to proceed with the correct maintenance of the machine. This was time consuming, and it required the presence of an expert operator for the training.

However, as the technology developed, more sophisticated methods were introduced. Audio guides, for example, were introduced in maintenance duties, which helped operators to be hands-free while pursuing maintenance operations. However, this technology still produced moments of confusion, in case the steps were not sufficiently clear, or in the case that the operator did not know where exactly the next step should be pursued.

The next big step towards a more effective way of maintaining industrial machines, considered in the present project, is the inclusion of AR technology in order to give clear information by the use of visual elements. This way, if implemented in a wearable device, the operator would be hands free, and the steps that should be followed, together with the location of the exact elements that need to be examined, would be clear (Feiner, Macintyre and Seligmann, 1993b). Moreover, it would be done in a faster way, as Henderson and Feiner (2009) concluded in their research.

5.2 Comparison between emerging technologies

AR is the technology considered and proposed as the solution to the problem in the present study, but there are other innovative technologies used for maintenance operations. In the following paragraphs, the ones that are believed to be the most common ones are presented and compared to AR, in order to conclude why choosing AR as the aid to the operators is the best choice.

One of the most usual competitors to AR technology is VR. This type of technology has been used as a support in maintenance or industrial operations (Ayala García et al., 2016). VR has principally provided help in the training stage, and not in the maintenance operations. Moreover, this technology has limitations that could be fixed by the use of AR instead (Khademi et al., 2013).

Together with this, classical HUD has also been compared to AR technology (Langlois and Soualmi, 2016), and it has been concluded that with the use of AR the user's movements are more anticipated. Henderson and Feiner (2009) have also compared AR to HUD together with liquid crystal display (LCD) and they also confirm that AR is the best solution, given the results presented in Figure 5.1.

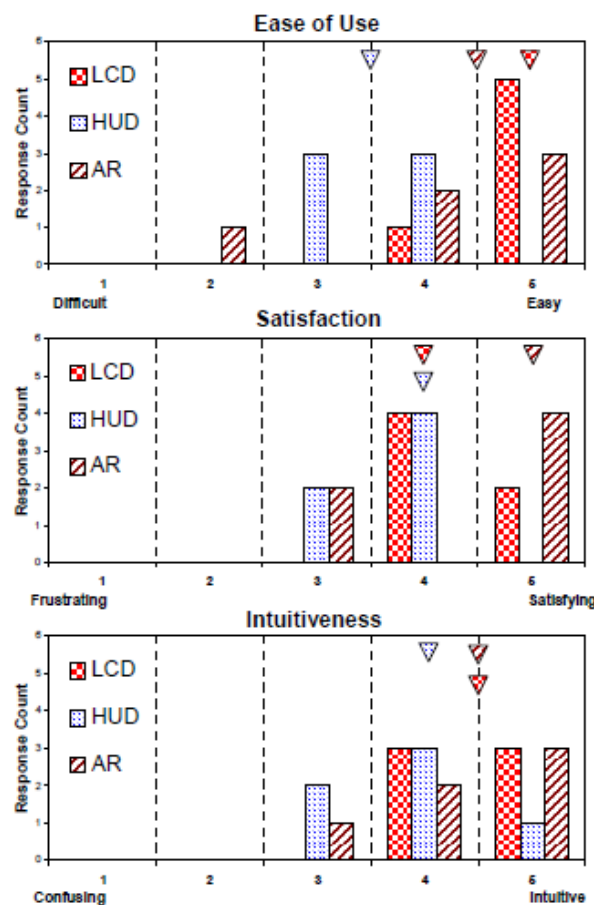


Figure 5.1 Comparison of LCD, HUD, AR (Henderson and Feiner, 2009)

With these reasons and having considered the mentioned technologies as the main competitors of AR, in regard to maintenance operations, it is believed that continuing with the suggested approach, where the main goal is supporting operators with AR technology, is the best choice.

6 Selection of the framework

In this part of the project, different frameworks are presented and compared with the aim of selecting the most suitable and easy to develop option for the implementation of this project.

6.1 Comparison of different frameworks

For the development of this project, it is necessary to investigate different frameworks to select the proper one for the application of AR in maintenance operations. To achieve this goal, the framework which will be used, must be compatible with AR glasses, mobile phones and tablets, as this is one of the requirements of this project. The AR glasses that will be used in this project, the Epson Moverio BT350, are compatible with java, C# and C++ programming languages, so one of these ones has to be chosen. Besides, when it comes to reading information of a part of the line, this will be done by the use of QR codes, so the program has to be able to use the camera of the device. Another point that has been taken into consideration is that the information printed in the screen has to be real-time information. Then, the program has to have the choice to connect to internet, so that the information can be sent and received. To perform all these tasks, there is not a unique software. Hence, more than one software's will be used, each of them for a certain job, to carry out the whole system.

There are several types of AR applications which are differently grouped based on how the information will be collected and proceeded. It can be marker-based, which is an image recognition-based system, in which when a pattern is recognized by the capture of a photo, the app is redirected to the information linked to the pattern. One of the most used systems of this type of AR is the lecture of QR codes, which is the one that will be used in the present project. Another type is location-based application, which takes the location of the device by the use of GPS and if the current position of the device is equal to the position of the destiny, runs the corresponding information.

So, to select the most suitable software development kit (SDK) for this project specification, the type of the licence must be taken into account. In this case, a free SDK will be selected. Furthermore, it must support one of the required languages (Java, C# or C++) and work in Android platform.

6.1.1 Language

In this case scenario there are three possible languages, which are Java, C# and C++. Three of them are well recognised languages and there should not be any problem to find help, if needed.

On the one hand, C++ was created in 1980, with the intention of the standardisation of the C language all over the world. This language allowed the manipulation of objects, and after some generic implementations, it became a structured program too. The way of typing this programming language can be seen in Figure 6.1.

```
1 #include <iostream>
2
3 int main() {
4     std::cout << "Hello, world!\n";
5     return 0;
6 }
```

Figure 6.1 How to type in C++ (Anon., 2018)

Moreover, Java was created in 1995, with the purpose of having as few implementation dependencies as possible. This programming language is concurrent, class-based and object oriented. The way of typing can be seen in Figure 6.2, where it can be observed that it is an intuitive way of programming. This means that even if the users are not familiar with Java programming language, they will be able to understand a large part of the code.

```
class HelloWorldApp {
    public static void main(String[] args) {
        System.out.println("Hello World!"); // Prints the string to
        the console.
    }
}
```

Figure 6.2 How to program in Java language (Anon., 2018)

Furthermore, C# was developed in 2000 by Microsoft, designed for the Common Language Infrastructure. C# is a general purpose, object-oriented programming language and the way of coding on it is a mixture between the two languages mentioned before, as can be seen in Figure 6.3, Java and C++. Accordingly, it is a very intuitive language which can be easily learnt if one of the other two languages is known.

```
C#

// A Hello World! program in C#.
using System;
namespace HelloWorld
{
    class Hello
    {
        static void Main()
        {
            Console.WriteLine("Hello World!");

            // Keep the console window open in debug mode.
            Console.WriteLine("Press any key to exit.");
            Console.ReadKey();
        }
    }
}
```

Figure 6.3 How to program in C# language (Anon., 2015)

6.1.2 SDK

A SDK is a combination of tools for the development of a software, which permits the creator of a software to develop an informatics application for a specific system. To compare different SDKs that are available currently, Table 6.1 has been created, where the most important SDKs are presented. The information given is determined by the necessities of this project (type, IOs, Unity, Marker, Natural Features, Windows Mobile, Web, 3D object tracking and GPS).

Table 6.1 AR SDK comparison (Anon., 2012)

Software development kit	Type	IOs	Unity	Marker	Natural Feature	Windows Mobile	Web	3D object tracking	GPS
Vuforia	Free + commercial SDK option	Yes	Yes	Yes Advanced + VU Mark	Yes	Yes	No	Yes (only on small size objects)	Yes
Yvision	Free + commercial SDK option	Yes		Yes		Yes			
IN2AR	Free + commercial SDK option	Yes via	Yes		Yes		Yes Flash		

		adobe ANE					- based		
AR Toolkit	Open source	Yes	Yes	Basic	Yes	Yes		No	
EasyAR	Free	Yes	Yes	Yes	Yes	Yes			
Wikitude	Free +commercial SDK option	Yes	Yes	Advanced	Yes	Yes		Beta	Yes
Kudan	Free +commercial SDK option	Yes	With SLAM	Yes	Yes	Yes	No	SLAM	Yes

6.1.2.1 Unity

Unity was created in 2005 with the purpose of creating a cross-platform game engine which allowed to develop video games. In unity there is the option to create games and simulations in three dimensions and two dimensions for computer, consoles and mobile devices. The three languages that were programmable in Unity are C#, Java, both of them explained in 6.1.1 and Boo, which was deprecated with the release of Unity 5. The different licenses that are available for Unity can be observed in the Table 6.2.

Table 6.2 Licenses for Unity (Anon., Last edited 2018)

License Name	All Engine Features and Platforms	Splash Screen	Cloud Build Queue	Multiplayer	Revenue Capacity	Performance Reporting	Premium Support	Access to Source Code	Price
Personal	Yes	Made With Unity and optional Custom Animation	Standard	20 CCUs	\$100,000	No	No	No	Free
Plus	Yes	Custom Animation and/or Made with Unity	Priority	50 CCUs	\$200,000	Yes	No	No	\$35 Monthly
Pro	Yes	Custom Animation and/or Made with Unity	Concurrent Builds	200 CCUs	Unlimited	Yes	Yes	No	\$125 Monthly
Enterprise	Yes	Custom Animation and/or Made with Unity	Dedicated Build Agents	Custom Multiplayer	Unlimited	Yes	Yes	Yes	Negotiated Pricing

6.2 Selection of framework

6.2.1 Language

As can be seen in the comparison of the languages, Java programming language is more intuitive than C++. Consequently, the expansion of this language has been very fast. Besides, Java is developed in Android platform, which is more easily programmed and implemented than C++ language -based

programs in a lot of case scenarios. One of these cases is AR applications, which, with the help of Unity program is much easier to develop and implement than with the use of C++. As aforementioned, C# is a mixture between Java and C++ which makes it a very intuitive yet easy to develop coding language.

On account of these reasons, in this project, the software will be developed in C# language, in Unity developing software for Android operating system devices.

6.2.2 SDK

To select the best SDK for this project, the minimum needs exposed in the previous section (see paragraph **¡Error! No se encuentra el origen de la referencia.**) have been taken into account, such as the differences between different SDKs. At last, Vuforia software has been selected, because is a free development software and its capability to use markers, 3D object tracking, works connected with Unity, which will be used to develop the software and due to its positive reviews when it comes to AR software development.

Therefore, the used language to develop this project will be C#, programmed in Unity environment with Vuforia SDK. Moreover, the developed program will be done for Android platform, so that it will be compatible with all the devices which accept Android platform.

7 Demonstrator development

In this section, the process followed to develop the demonstrator is explained. First, how the prototype works is explained, and which have been the ideas taken into account to design the demonstrator. Then, it is proceeded to clarify what the demonstrator testing consisted of, the results of this test and the main changes done to the application which depend on the received feedback. Finally, the conclusions achieved by these tests are revealed.

7.1 Design and development of the demonstrator

Authors are aware that the proposed solution does not fit entirely a real maintenance operation procedure, since no analysis of the whole line has been done to proceed with the maintenance duties. Due to the lack of access to real data, the line functionality has been done in base of assumptions made by the authors. With no specific data about the performance and importance of each machine in the industrial line, the priorities given to the machines, which will be used for the path calculation, and which affect the performance of the entire line, have not been considered relevant, since no production flow has been defined. However, machine 1 has been given a higher priority in case of error in this machine, since it is believed that if the first machine in the line suffers a failure there could be consequences in the proceeding machines. In the proposed solution, the states and positions of the machines have been the used information to calculate the shortest path that needs to be followed by an operator to execute the maintenance operation. Besides, in case of warning scenery, the user will have the option to decide what priority should the warning machines be awarded, as it can be seen in Table 7.2.

To do a proper analysis and path calculation of the maintenance operation of an entire line, the order in which the product or material flows from the entry to the exit of the line should be taken into account. As there has not been access to real information for the development of the prototype, the following assumptions could have been made in order to consider the maintenance of the complete line, and not the one that corresponds to individual machinery.

If a machine is connected with other two (see Figure 7.1), this first machine should be granted a higher priority than the following ones. This is because in the case of a failure in Machine 1, the rest of the line will be affected (see case b in Figure 7.1). In case that one of the two following machines has an error, the failure will only affect the machine that come after it and parallel branch, as it can be observed in the Figure 7.1 case c. At last, if one of the machines in the exit of the line is damaged, this will have

the lowest priority, because the problem will only have a negative effect on the last step of the line, as can be observed in Figure 7.1 case d.

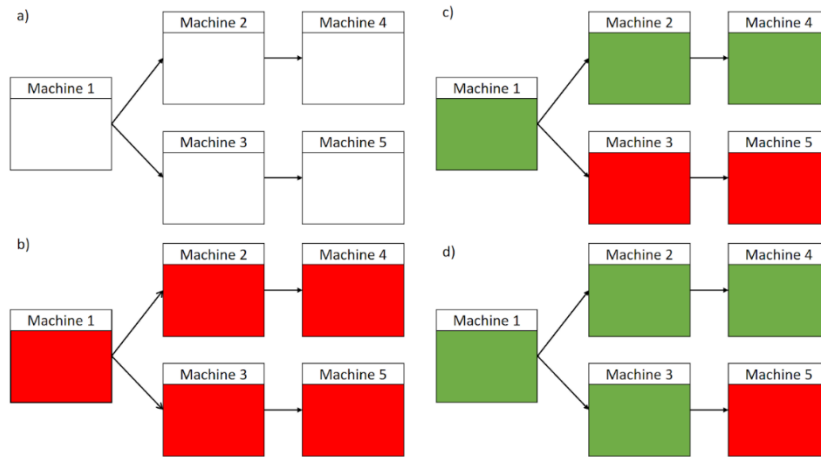


Figure 7.1 Machines flow analysis

However, in the solution which is proposed in this project, all the machines except Machine 1 have the same priority since there is not a particular material flow defined. The rest of the machines are treated as singular machines which work in parallel with no particular priority or interconnection in the line as it can be observed in Figure 7.2.

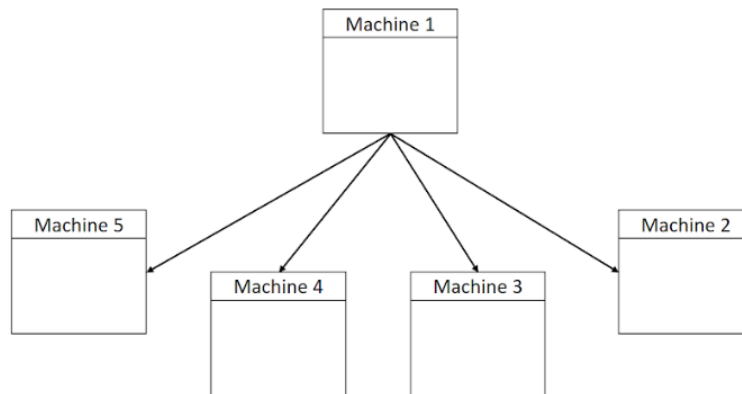


Figure 7.2 Prototype machines flow

To explain how this application works, which supports an operator during a maintenance job by the use of AR technology, the next flowchart, Figure 7.3, has been done. Here, the main steps followed by the program are represented.

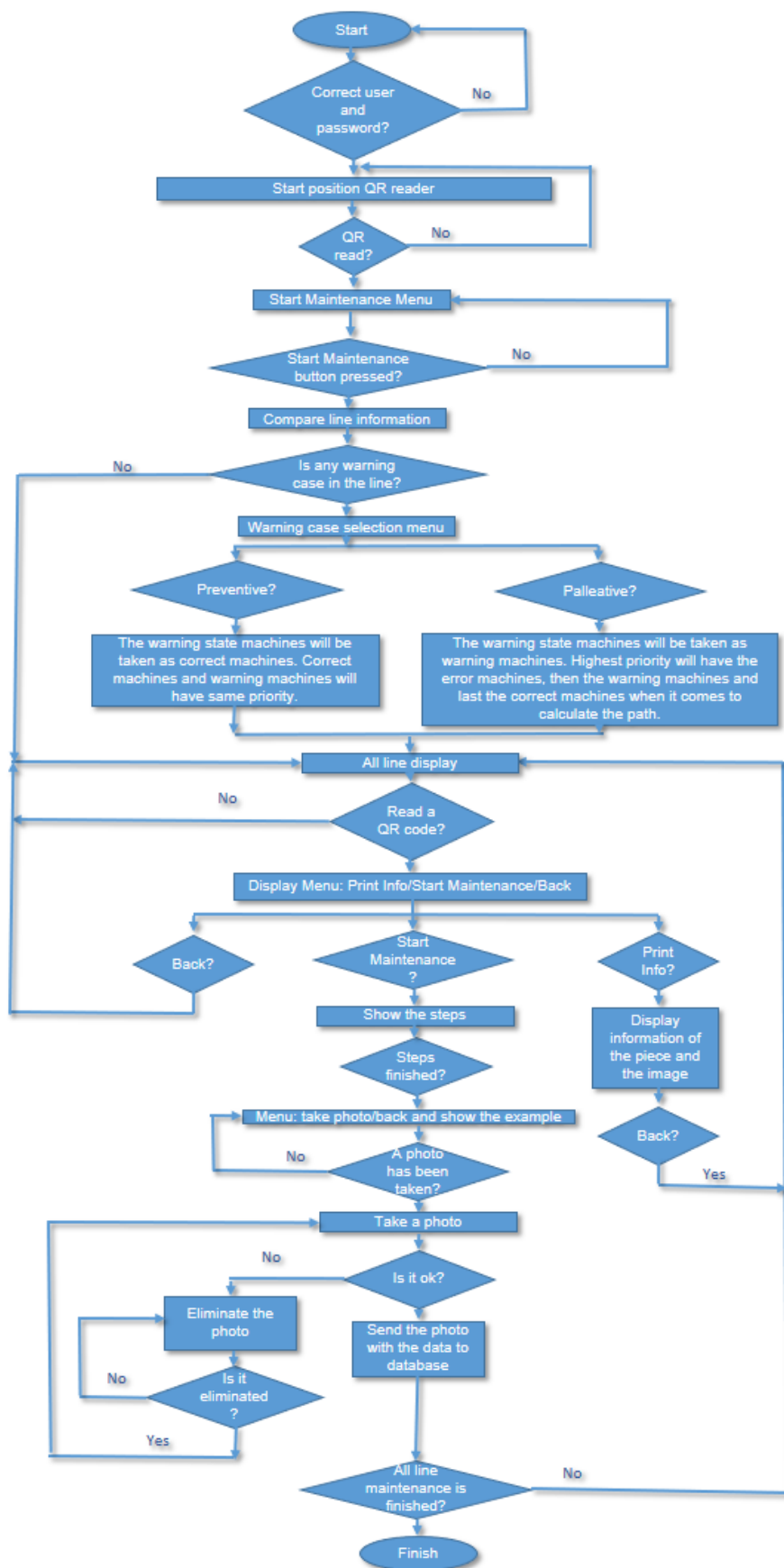


Figure 7.3 General flowchart of the program

To develop this demonstrator, it has not been possible to access to real data, so a line which has five machines, two of them electric cabinets and the other three water pumps, has been simulated.

To be able to use this demonstrator, a QR code has to be placed in the entrance of the line and in the front part of each machine, which allows to read its data.

When it comes to the program, in the first instance a login scene is displayed, where the user has to introduce his name and a password. If one of these fields is incorrect, the user will not be able to continue the application. This is done to ensure that the data of the line is restricted to those who only have the access. This way, companies' information is safe and which operator has done the maintenance is registered. To introduce a new user, the creation of a ".txt" file with the structure shown in Figure 7.4 should be done and saved in a specific directory in the used AR device. In the first line, the name of the user is written and in the second one the password, which has to have be at least eight characters long. The name of the file has to be the same of the user's name.

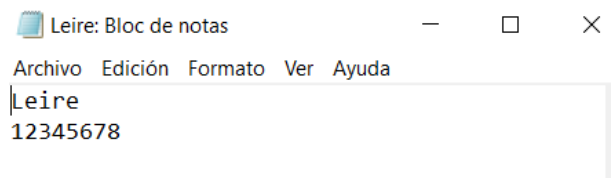


Figure 7.4 Structure of login file

Once the user name and password are accepted, the next scene is displayed, where a QR code has to be read to proceed with the program. This QR code will represent the initial point of the line, as for example, the entry of the line. This way, the initial position of the user is ensured.

When this QR code is read, which can be observed in the Figure 7.5, the next scene is displayed, where the state of the line with a short explanation appears. The state of the line can be error, warning or correct (see Figure 7.6), dependent on the values of the data of the machines, which are compared with the maximum, minimum, recommended maximum and recommended minimum values, see Table 7.1. If one of the machine is out of the maximum-minimum value range, the line will be in error state. On the other hand, if the machine data is on the maximum-minimum range, but out of the recommended values range, the line will be in warning state. At last, if the machine data is between the recommended values, the line state will be correct. If the line is in error state, it means that there is at least one machine which has an error. In this case, the first machine to which the maintenance will be done is that one, since an error state is of maximum priority.



Figure 7.5 Start point QR code

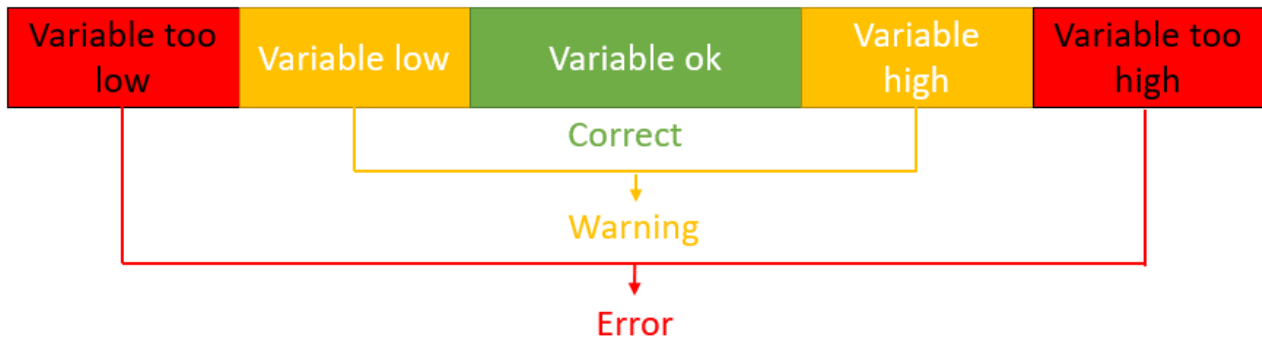


Figure 7.6 Variables comparison treatment

Table 7.1 Variables state range

Variable	Variable Ok	Variable low	Variable high	Variable too low	Variable too high
	Correct	Warning		Error	
Water temperature (°C)	15-30	10-15	30-40	<10	>40
Housing temperature (°C)	15-30	10-15	30-40	<10	>40
Pressure (kPa)	260-320	200-260	320-350	<200	>350
Water level (l)	600-800	500-600	800-1000	<500	>1000
Water flow (l/m)	70-110	50-70	110-150	<50	>150
Voltage (V)	390-410	380-390	410-420	<380	>420
Current (A)	3-4	2.5-3	4-4.5	<2.5	>4.5

In the next scene, all the error machines with their error messages together with warning machines with their warning messages are present, how can be observed in the Figure 7.7. This display allows to the operator to have a clear idea of which machines are in error and warning mode. Apart from that, gives to the operator necessary information, which is the problem of each machinery, so that the operator can decide how to react based on the priority of the problem.

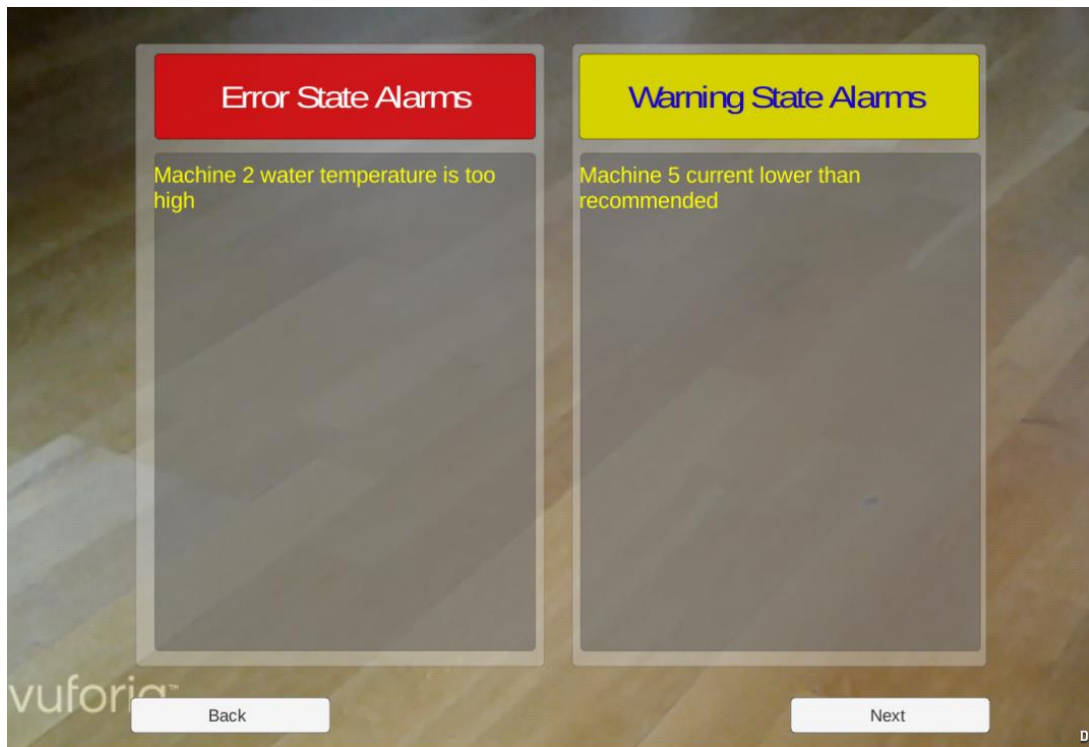


Figure 7.7 Alarm display

The next scene depends on the state of the line. In the case there is a machine with a warning, the user can select how the warning case should be treated, if there are more than one warning cases, all of them will be treated equally. In these cases, there are two options, preventive and palliative maintenance. In the case the user selects the preventive option, the warning cases will be treated as correct ones, which means that correct machines and warning machines will have the same priority, so that when it comes to the calculus of the path which has to be followed by the operator, the minimum distance between correct and warning machines will be calculated at the same time. In case the user selects the palliative option, the warning case will be treated as a warning, so its priority will be higher than the correct ones and lower than the error ones.

When it comes to the calculus of the maintenance path, the program takes care of the priorities, which can be seen in Table 7.2, and the minimum distances between different machines. In case there is an error in the line, the faulty machines will be the first one that will be repaired. As aforementioned, in case the faulty machine is Machine 1, this will always be the first one to be maintained. On the other hand, whenever there is more than one machine which contains an error, all these machines are prioritised over the others, and the path to be followed will be calculated in accordance to the total shortest possible maintenance distance. As previously stated, the warnings can be treated as warning

with the priority level of two or as correct ones. At last, the correct ones will always have the same priority level, which is the lowest one.

Table 7.2 Machine states priority levels

	Error	Warning		Correct
		Palliative	Preventive	
Priority level	1 *	2	3	3

* In case the one of the machines which contain an error is Machine 1, this one has the highest priority

On the other hand, if there is not any warning case in the line, the next scene will be directly the all line scene, which can be observed in Figure 7.8. Here, all the machines are represented with 3D objects, which correspond to each type of the machine. Each machine has its name written under it and a sphere above, which indicates the state of the machine by the use of three colours; red for error, yellow for warning and green for correct. Also, it is in this scene where the path which has to be followed by the operator is calculated. In view of the aforementioned priorities and the machines' positions, the shortest path is calculated with TSP. In this case, first, error machines positions are compared between them to find out which is the shortest way to do the maintenance, then the same is done with warning machines but with a slight variation, which is that the distance between the last error machine and the first warning machine is taken into account too, at last the same is done with correct machines. Once all the possible paths are compared and the shortest one has been selected, the program prints a pink row in movement between the current position and the next position, as can be seen in Figure 7.8, so that the user knows where to go next. Furthermore, in order to avoid misunderstanding, a text indicates what the next machine is, in the lower part of the screen. Once that the user is in the corresponding machine, the QR code corresponding to that machine should be read.

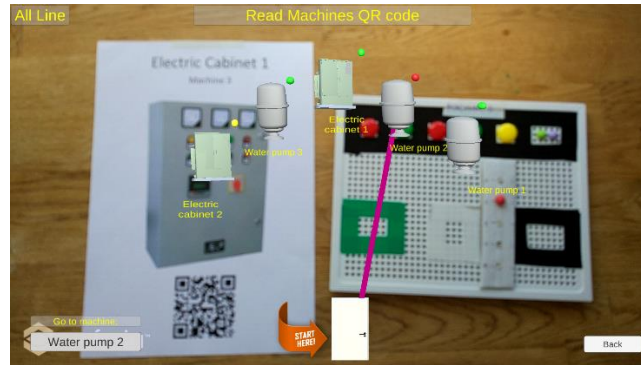


Figure 7.8 All line display

When the QR code is read, all the data of the selected machine is read from the local database, as can be seen inside the red square and the next scene is loaded automatically, as seen in the purple square in Figure 7.9. In the next scene, a 3D representation of the machine is shown and the option to display the information of the machine or start its maintenance is given.

```
private void OnTrackingFound()
{
    //Information of machine 1 is given to the variable
    Debug.Log("Trackable " + mTrackableBehaviour.TrackableName + " found");
    MachineData.Instance().MachineDataVariables[0] = MachineData.Instance().DataMachine1.WaterTemp;
    MachineData.Instance().MachineDataVariables[1] = MachineData.Instance().DataMachine1.Pressure;
    MachineData.Instance().MachineDataVariables[2] = MachineData.Instance().DataMachine1.WaterLevel;
    MachineData.Instance().MachineDataVariables[3] = MachineData.Instance().DataMachine1.HousingTemp;
    MachineData.Instance().MachineDataVariables[4] = MachineData.Instance().DataMachine1.Flow;
    MachineData.Instance().MachineName=MachineData.Instance().DataMachine1.MachineType;
    MachineData.Instance().CurrentMachineNum=MachineData.Instance().DataMachine1.MachineNum;
    MachineData.Instance().LastDispMachineNum=MachineData.Instance().MachineState[0, 2];
    SceneManager.LoadScene ("MachinesMenu");
}
```

Figure 7.9 QR code reader code example. Machine 1

If the user selects the option of displaying the information, a table with different values which correspond to the machine will appear, which can be observed in the Figure 7.10 inside the purple square. Next to them, a square which colour depends if the information value is correct (green colour) or error (red colour) are placed, these ones can be seen in the Figure 7.10 in the light blue square.

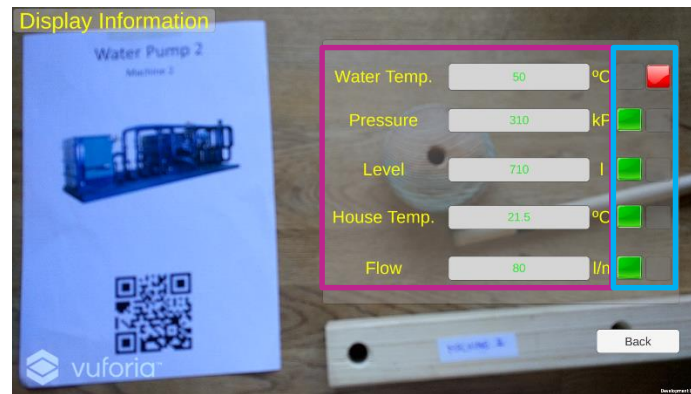


Figure 7.10 Display information example. Machine 2

On the other hand, if start maintenance is selected another scenery is loaded. Here, the steps which have to be followed by the operator are visible, as can be seen in the Figure 7.11. Steps are displayed one by one, and the CAD representation of the machine to which the maintenance operation is performed is located in the upper part of the screen. Under the written explanation of the step, 3D objects which represent needed tools to perform the maintenance are placed as it can be observed in the Figure 7.12 inside the purple square. To go to the next and previous steps, there are two buttons, which in case it is the first step, previous button disappears and if it is the last one, next button disappears. This can be observed in the difference between Figure 7.11 and Figure 7.12. Another information given in this scenery, is that based on which part of the machinery has to be maintained in that moment, the part of the machine changes its colour in the 3D model displayed in the upper left part of the screen.

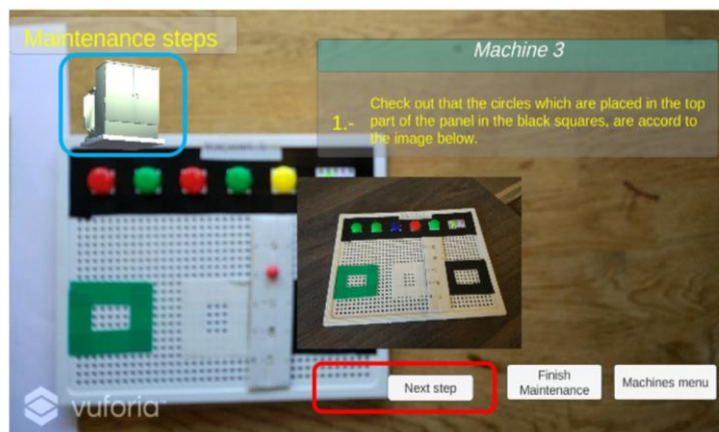


Figure 7.11 First maintenance step

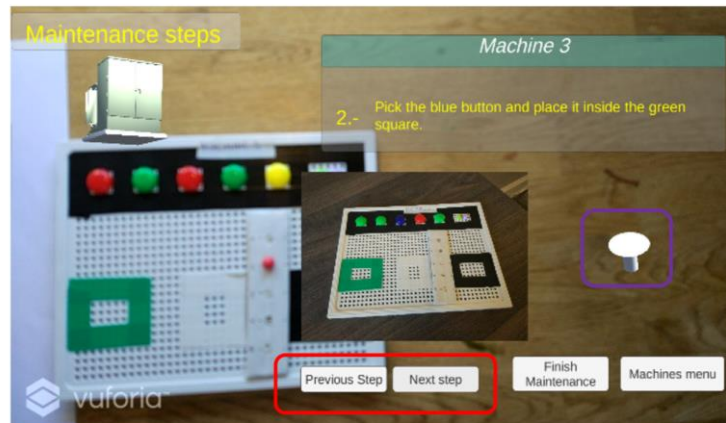


Figure 7.12 Second maintenance step

Whenever the user wants, by the push of the “Finish Maintenance” button, the next scene is loaded. In this scenery the operator can take a photo for monitoring purposes. When the photo is taken, the next scene is opened automatically, where an example of what the photo should look like is displayed. The user, has the option to delete the photo, see saved photos, go back to the camera or just continue with the maintenance by the saving of the picture together with all the information of the machine in a pdf file, in the internal memory of the AR device.

If the user decides that the photo is ok and wants to continue with the maintenance, the pdf file will be automatically created. This pdf contains all the information of the machine; in case the machine was in error or warning state which was the problem, the steps done by the operator, all the values of the parameters of the machine and the photo taken by the operator. Once the pdf is saved, the last display will be shown. In this last scene, which has been the machine to which the maintenance operation has been done is written, together with a message which grants the operator the choice of continuing with the other machine’s maintenance or finish the maintenance operation there.

If the user decides to continue the maintenance, all line scene will be loaded again, but this time the current position will be the machine which has been the last one to which has been done the maintenance and the next machine will be the next calculated by the TSP. The user will have to go to the proper machine and read the QR code to restart with the cycle explained before.

On the other hand, if the maintenance of the line is finished, the program will be closed and to restart the program, the user will have to open the application, enter the login name and password and start the maintenance from the first scene.

The internal structure of the program created in C# in Unity is the one seen in Figure 7.13.

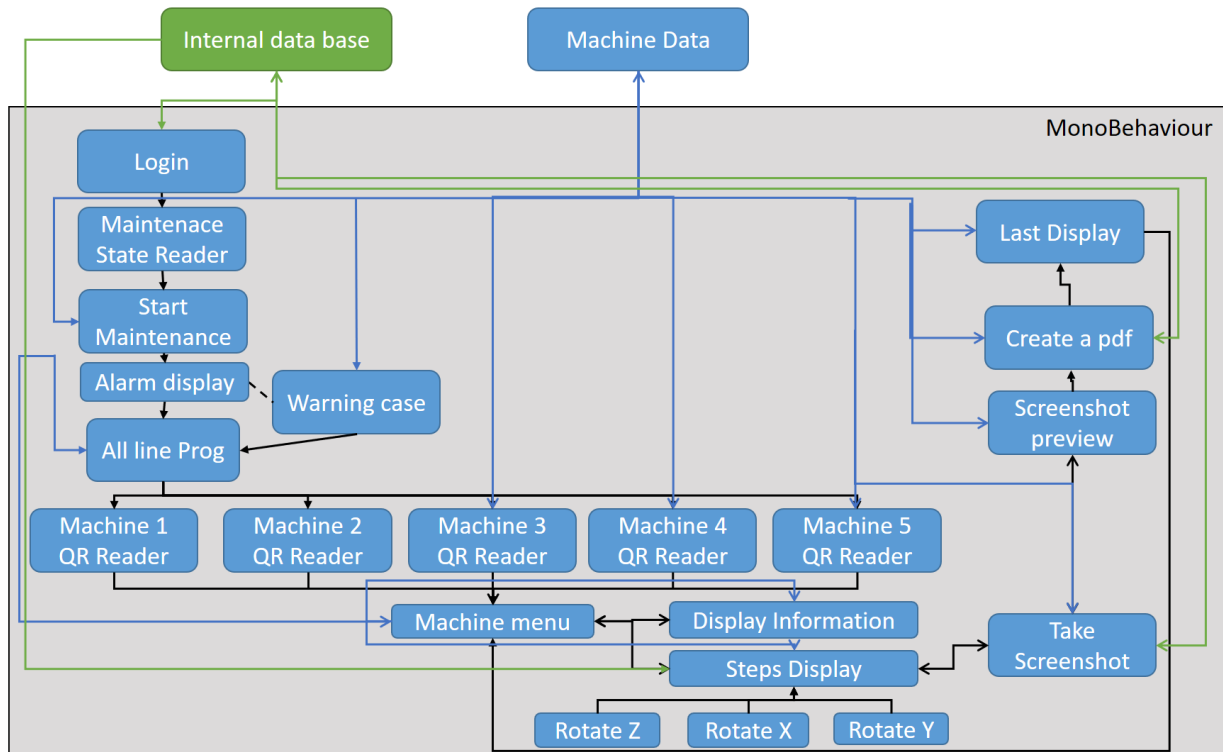


Figure 7.13 Internal structure of created program

7.2 Analysis of the insertion of the prototype in the industry

In this project, a deep cost analysis has not been conducted, since this was not one of the defined objectives, and also because the proposed solution has been implemented in form of a demonstrator and not an early stage of a final product that could be used in an industrial line. However, after the interpretation of previously performed research reports and the development of the prototype has been concluded, it has been defined that the next requirements are needed for the correct implementation of AR technology in industrial lines:

- In order to have an application that could be used with AR devices a programmer or set of programmers is needed to develop the application. This could be provided by a subcontracting enterprise or by a team of expert programmers from inside the company.
- A connexion to a database is necessary for the use of the proposed solution to monitor the maintenance operations properly. The collected data in the maintenance operation will be stored in a folder in a database, that can be internal or external.
- Sensors should be implemented in every machine. Real time data is required for the solution to check the state of each machine, and for that, sensors are needed in different parts of the line.

- In order to modify users' data, or to add a new user to the system, so that a new operator has access to the solution, an operator with expertise is needed. This operator should be able to create and save a text file in the correct folder.
- If there is a change in the line, such as a replacement or a change of position of a machine, some changes have to be done to the program. To effect these changes, there is the need of an expert in the field of programming.
- The need of proper devices for the use of this application. It is necessary to have at least one device with the minimum specifications defined in paragraphs 1.2 and 6.1. The device can be a tablet, mobile phone or AR glasses. The prices of three devices used for the implementation of the prototype are gathered as an example in Table 7.3.

Table 7.3 Devices price comparison

Device	Price (EUR)	Reference
AR glasses (Epson Moverio BT-350)	1.089	(Epson, 2018)
Tablet (Samsung GT-P5100)	290	(GSMArena, 2000-2018)
Mobile phone (Xiaomi Mi5)	240	(Anon., 2018)

In consideration of these needs, it is believed that the implementation of AR in industrial maintenance operations would not be suitable in small companies, because of the initial investment that is needed to implement the application in the company. It is also needed to contemplate that the used AR glasses model has received a negative feedback from the participants in the text, which means that for a complacent use of such technology in real applications, the investment in regard to hardware should be higher than the one shown in Table 7.3.

However, big companies would be benefited by the implementation of AR in maintenance duties, due to the fact that they mostly fulfil all the listed requirements and no great infrastructure investment needs to be done. Moreover, once the investment in proper hardware and software is done, the implementation of AR will decrease the waste time and hence the total cost of maintenance duties of the company.

7.3 Test to validate the prototype

In this part, the test that has been prepared to validate this prototype is explained. For the test, there was not any access provided to real information or machinery so some wooden pieces, see Figure 7.14, have been used to create a structure that users have to assemble following the steps given in different ways.



Figure 7.14 Wooden structure used for the test. Machine 1

The steps were explained in paper, audio and hand-held AR device (tablet) format. The user first had to do the maintenance of 4 machines, by the following of the steps written in paper, which start in the first machine and finish in the 4th one. Then, the user had to follow the same process but with the steps explained in audio recordings. At last, the developed application in the present project has been used to follow the instructions.

7.3.1 Test 1

In the first test that was performed, all the steps written in three of the formats were the same. This lead to a memorising of the steps by the end of the experiment, and hence, the user's not paying much attention to the instructions given by the third format. Moreover, when the user started the first step and finished the last one in each machine, a researcher needed to be told in order to collect data in regard to the time spent at each machine in each format.

Once the experiment was concluded, when the user had tested the three proposed formats, the user had to fulfil a short questionnaire to evaluate how user-friendly each format was. Also, in the paper given to the users there was a space to write comments. The obtained data and feedback have been used to reach some conclusions and change some design and programme related parts of the project, so that

the prototype could be improved. The data collected in the tests is represented in Table 7.4, Figure 7.15 and Table 7.5.

Table 7.4 Number and age of participants. Test 1

Average age	25.4 years old
Number of participants	8 participants

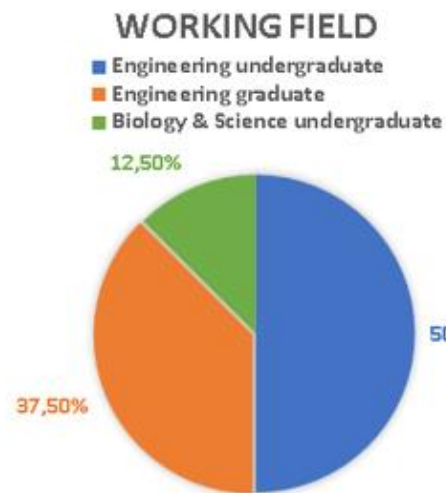


Figure 7.15 Working field graphs. Test 1

Table 7.5 Results. Test 1

	Paper format	Audio format	AR tablet
Machine 1 average (seconds)	74.97	75.77	56.96
Machine 2 average (seconds)	94.04	94.17	66.71
Machine 3 average (seconds)	54.41	65.16	46.37
Machine 4 average (seconds)	59.79	53.00	43.11
Maintenance per machine average (seconds)	70.80	72.028	53.29
User-friendly feedback (0-10)	6.5	8.12	7.5

7.3.2 Test 2

After changing some features in the test and in the prototype, another round of tests was done. The changes that have been applied for the second round are the next ones:

- The steps are different in the different formats.
- The colour of the letters in the application has been changed, so that it was easier to read in different case scenarios.
- In the camera scene, one button has been eliminated, so that when the photo is taken the next scene is loaded automatically.
- In the steps scene, there were two buttons to pass to the next or previous step and these buttons continued in the screen although all the steps were finished. This was a bit confusing for the users, so it was changed and so that when the last or the first step was reached the extra button would disappear.
- Some texts have been modified to be more understandable for the user.
- The QR code reader indicators have been deleted and, in their place, a text that indicates to read the QR code has been placed. This allows to have a clearer space in the screen.
- In the survey, apart from asking how user friendly the different formats have been and having the space for comments, some specific questions have been added to get more information about them. Besides, a Likert-type survey has been added to know users' opinion.
- The audio format has been changed, so that all the machine steps were in the same audio to make this format more user-friendly than when each step was explained in a separated audio.

Data have been collected in the same way, but in this case the participants were operators and workers from a company, which has allowed to test this prototype. This company is called Vakin and the tests have been done during two days to different profile workers. Some of the operators, did not know English, so it was not possible in those cases to perform the whole test and collect times, but they used the application and made some comments to improve it.

The data collected in the tests is represented in Table 7.6, Figure 7.16 and Table 7.7.

Table 7.6 Number and age of participants. Test 2

Average age	31.7 years old
Number of participants	3 participants

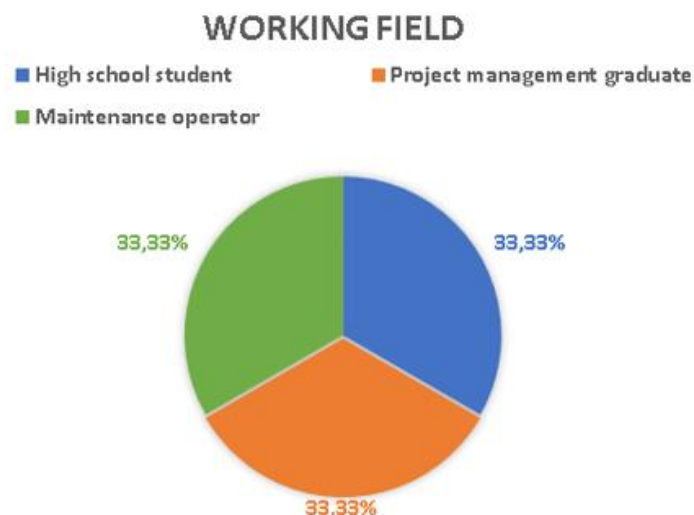


Figure 7.16 Working field graphs. Test 2

Table 7.7 Results. Test 2

	Paper format	Audio format	AR tablet
Machine 1 average (seconds)	89.33	46.77	61.73
Machine 2 average (seconds)	113.70	56.33	82.93
Machine 3 average (seconds)	86.33	46.93	74.68
Machine 4 average (seconds)	52.9	62.33	47.59
Maintenance per machine average (seconds)	85.48	53.09	66.74
User-friendly feedback (0-10)	6.33	6.66	7.0

7.3.3 Test 3

After changing some features in the test and in the prototype, another round of tests was done. The changes that have been applied for the second round are the next ones:

- The creation of a pdf format document has been added in the end of the maintenance.
- A warning and error list have been included in the beginning of the program, so that the operator knows which the problems in the line are.

- A program for the AR glasses has been implemented, so the tests have been done in four formats.
- More graphical components have been included: Coloured 3D objects, arrows and transparent objects to indicate where each piece should go.

Data have been collected in the same way, but in this case the fourth format, which is AR glasses, has been introduced.

The data collected in the tests is represented in Table 7.8, Table 7.9 and Figure 7.17.

Table 7.8 Number and age of participants. Test 3

Average age	37.25 years old
Number of participants	7 participants

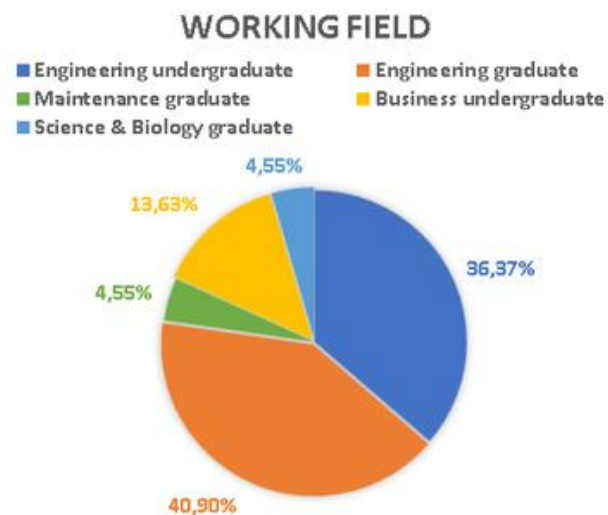


Figure 7.17 Working field graphs. Test 3

Table 7.9 Results. Test 3

	Paper format	Audio format	AR tablet	AR glasses
Machine 1 average (seconds)	94.94	50.90	49.31	66.43
Machine 2 average (seconds)	110.57	55.18	55.82	79.36

Machine 3 average (seconds)	78.81	52.30	63.12	63.03
Machine 4 average (seconds)	65.35	34.76	36.98	42.46
Maintenance per machine average (seconds)	87.42	48.28	51.31	62.82
Total maintenance average (seconds)	516.86	289.77	433.59	477.01
User-friendly feedback (0- 10)	6.59	7.05	7.86	7.14

7.4 Results

In the following paragraphs, reached conclusions from the results got from the three tests will be exposed. In sections 7.3.1, 7.3.2 and 7.3.3 some data have been disclosed. To see the tables with all the collected data, go to Appendix A:.

As explained before, in Test 1, the steps were the same for all the formats. Hence, the participants did the maintenance faster by the end of the experiment because they remembered which the upcoming step was. As a consequence, the maintenance times were reduced significantly. Apart from that, the feedback received from the participants revealed that most of them preferred AR tablet used format over paper format and audio format over the other two. This last one was reasoned by the fact that in audio format, participants had their hands free whereas by the use of the proposed AR application they had to hold the AR device with their hands, which was an obstacle to the assembly of the machines. However, participants mentioned that steps in AR application were easier to follow by means of the representation of 3D objects while the reading of the steps was done.

Another thing that can be observed is that in 3.79% of the cases, both in audio and paper formats, participants misunderstood a step. It needs to be noted that participants have not made any error when using the AR glasses, while 0.76% of error was detected when using AR in tablet. It is important to note that no errors have been detected in the AR with tablet cases.

Moreover, as can be seen in Table 7.5, the time is reduced significantly from paper format to audio format and even more from audio format to AR format. One of the main purposes of doing maintenance job by the use of AR is to reduce the time needed to do the work properly. So, it can be said that one of the key points is achieved. In Table 7.7, audio format time is shorter than the AR-hand used format, but the difference between them is slight. This can be because the improvements done in the audio format and because in hand used AR device the user needs to hold the device while doing the maintenance, what does not apply to audio format. Besides, this can be solved with the integration of AR glasses, which allows to do maintenance operations with full explanation of the steps and with the hands free. However, in Table 7.9 it can be seen that participants did not prefer AR glasses format over the rest of the formats.

In regard to the time spent at each machine, Table 7.9 shows that audio-guided format is the most time-efficient one, followed by the use of a tablet containing AR technology, AR glasses and finally paper format. Likewise, the total time spent for the maintenance of the whole line follows the same order. This could be interpreted as a lack of efficiency on the AR technology. However, the authors believe that both, the hardware and the early stage of the solution are responsible for such results.

The third experiment has also shown that the data collected is not totally reliable, since the times differ notably from a test to another one in the case of paper format, when no changes have been applied to it. Also, in regard to how user-friendly participants thought each format was, there is not a clear answer, since it has varied a lot in reference to the person and its background. However, no clear pattern could be identified for defining the preferences of the different participants.

In the Likert-type test done to the last two sets of participants, the usability of the AR application developed for both, tablet and AR glasses has been evaluated, and the results have shown that participants feel that the application's usability is of 80.97%, which compared to the time related data is encouraging.

8 Conclusions

Previous studies on AR for maintenance operations have focused on either the combination this technology with additional ones or the design and implementation of a solution for a specific device as a proof of concept, or in order to analyse its effects on the maintenance duties. For its part, the present project has successfully combined a suggestion of framework of integration for industrial

maintenance operations, together with its compatibility with diverse AR devices, and a subsequent comparison with outmoded procedures.

8.1 Summary

The main objective of the present project was to add knowledge about AR in industrial maintenance operations. The suggested method of integration consists on the development of an artefact, which is believed to extend the knowledge in the current problem. It is assumed that the implementation of AR in hands-free devices could provide with an efficient and user-friendly solution for both, experienced and inexperienced maintenance operators. Hence, the result of this approach would be an AR prototype which supports the operator with visual instructions while the maintenance of an industrial line is done.

Each industrial machine has different maintenance requirements that should be contemplated for the design of the integration framework. Together with that, it should be also considered that the development of the prototype should be done for all types of users, with no need of previous experience with AR technology nor maintenance operations.

First, a literature review has been done with the aim of collecting data in the fields related to the present project, which are AR applications for inspection and maintenance operations, ES and QR code reader.

After this, an analysis of the data has been done so as to evaluate what would be the best framework to work with, in order to get the general aims that have been defined in the beginning. When this has been accomplished, it has been proceeded to define the specific requirements of the system.

Once the definition of the framework of implementation for AR in different devices, together with the more specific requirements, has been completed, the development of a demonstrator has been built in Unity-Vuforia developing environment. The creation of a demonstrator serves as a proof of concept in order to test if the defined framework fulfils all the requirements, and hence, validate the solution. This framework has been implemented in phones and tablets with Android as their operating system, and in Epson Moverio BT350 glasses.

Finally, tests have been performed in a practical scenario so as to receive feedback from potential operators with different backgrounds, and to determine if the work done has been successful.

8.2 Discussion

The present project has had its main focus on the support and monitoring of maintenance operators. Nonetheless, the development of the solution and the creation of the prototype for validation and testing has proven that AR is applicable in a wider spectrum of fields inside the industry. For example, as demonstrated with the prototype, the verified usability could be beneficial for assembly operations, inspection, disassembly or learning, among other fields.

When it comes to the efficiency of this project, a test to level the efficiency of the prototype in a real industrial environment, a spaghetti diagram or a complete cost analysis have not been done. Hence, the efficiency of the proposed prototype could not be completely analysed and solid results regarding the efficiency of the prototype have not been obtained. On the other hand, with respect to the data collected from the experiments, together with the feedback received from the participants, the authors believe that the usage of AR in maintenance operations has been proven to be efficient. Even if the results do not present the proposed solution as the optimal maintenance support, the evaluation of the developed application has been positive. The lack of efficiency of the use of the proposed solution is attributed to the early stage of the application, together with its running time.

It should be noted that the physical environment in which each test has been performed has altered the obtained results. There has not been any standardisation of the physical environment, since the resources used, such as rooms or tables, have not been the same in all the tests. Therefore, the time elapsed in between machines varies significantly when changing the experiment setting.

It has been recognized that in the tests in which the participants read the instructions for the different steps from a document in paper, the times obtained have been considerably higher than in the rest of the tests. However, the authors do not consider that these measurements reflect the reality. It should be taken into consideration that these results have been notably affected by the way the tests have been pursued. The tests began with the instructions given in paper format, which means that the participants were not familiar with the test yet and required more time on this part of the other parts. Moreover, the paper instructions did not include a graphical description in each step, which caused a confusion in some participants. In order to have a better comparison between the four proposed formats in the tests, a first round of familiarisation with the pieces should have been done and instructions in paper should have included more graphical elements.

With regard to the delimitations that could have been faced in the testing phase, it has not been considered what would have happened if there was a colour-blind person. The tests used colours to differentiate pieces, and a conflict could have occurred if a person with sight disorders would have attempted to complete the proposed experiment. Also, all the instructions were written only in English, and this has been a problem for some participants which did not have a good knowledge of the language. These issues should have been considered prior to the execution of the tests, with not only colour differentiation but also unique patterns associated with the colour, should have been included with the aim of reducing the mentioned limitations. With respect to the audio-guide, an online translator with a robotic voice could have been used, instead of a recorded person, to guarantee a neutral pronunciation in each language.

In reference to the proposed solution for AR glasses, the results have been addressed as hands-free solutions. However, a mousepad was needed in the tests in order to navigate through the possibilities offered by the program. This also needs to be considered when analysing the usability of each technology.

8.3 Conclusions

The implementation of the prototype was successful. Thus, it has been proven that the adding of AR technology to industrial maintenance operations increases efficiency and therefore maintenance time and costs decrease. By the reading of a QR code the state of the line is displayed, and the optimal path, given priorities and distance between machines, is calculated and indicated. Once in the correct machine, another QR will be read and the option of displaying information or doing the maintenance procedure provides the operator a visual, easy to understand solution.

However, it is needed to stress that the demonstrator is not perfect and that a number of improvements are proposed in paragraph 8.4 for future work.

Firstly, it is important to note that the developed software has been developed on a demonstrator level, which means that it is an initial solution. Accordingly, it cannot be considered a final solution, and so its promising utilisation features cannot be tested.

Secondly, as foreseen in paragraph 1.3 some devices are not compatible with the developed software, because the development environment used for creating the demonstrator does not offer support to older Android devices.

However, it is needed to stress that thanks to the proper choice of the framework of development, the software has been implemented in tablets and mobile phones that work in Android operating system, as initially aimed, and also in EPSON Moverio 350T glasses.

Moreover, with the purpose of monitoring each operator's work together with the machines states, a standard template has been created. This template will be filled with the corresponding information in the end of each machine's maintenance procedure automatically, which creates a register that can be later checked, in case this is needed.

With reference to where this information is stored, even if initially a database was mentioned, it is locally saved. This means that the files which contain the information about the machine that has gone through a maintenance process are stored in the device that the operator used at that moment, and not in a public remote database, due to the lack of connection with any external database in developed demonstrator.

Moreover, even if ES have been implemented in the form of TSP, which simulates an expert human that knows which the shortest path is, in view of priorities, the solution is not able to modify or create rules by itself.

Despite the following proposed improvements, it is believed that industrial maintenance operations could enhance from the proposed solution. To prove this, some tests have been done with people of diverse ages and expertise fields. The main objective of these tests was to mount a structure with the help of indications given in different formats, which are paper, audio and hand-held AR device. The results got in these tests, have been helpful for concluding that by the use of the developed application in AR devices, the time needed to perform the maintenance of a machine is reduced significantly (see paragraphs 7.3.1, 7.3.2 and 7.3.3).

8.4 Future work

Even though, it has been proved that the use of AR in industrial maintenance operations could be beneficial for the operators, there are still some improvements, which are explained in the paragraphs below, that could be done in future work.

All the data in the AR program is displayed in English, and it has been seen that this may be an issue for not English-speakers, so by the adding of different language options to the program this should help in the comprehension of it. However, even if CAD files have been used for giving the instructions

in a graphical way, it should be considered to create more realistic figures, so that the operators do not have the need of reading any instruction, since it would be sufficiently clear by the displayed figures

Moreover, as stated before, there are some compatibility issues with older devices. Therefore, it would be helpful if the program was developed in various versions of the developing software, for a major reachability.

In addition, in order to do a better monitoring of the maintenance operation, a way of communicating with an expert operator should be implemented. This way, in case there is a problem or a misunderstanding that the inexperienced operator cannot solve by itself, the experienced operator could have access to its camera and could guide the operator towards the correct procedure.

Furthermore, it would be beneficial to include a piece of code that enables giving different amount of information defined by the user's expertise. This way, the system could calculate how expert an operator is, based on the quantity of times it has gone through the maintenance procedure by the use of the AR solution, and hence, adapt itself to the different needs.

It is also recommendable to do a proper analysis of the ergonomics in regard to the use of wearable AR devices, since it could have effects on the operators' sight. Also, an entire-body examination could be beneficial, in order to compare the ergonomics with and without the use of such devices.

Also, in order to obtain a fully hands-free support, it would be advisable to add technology to the proposed application, so as to choose different options without the need of a mousepad. Voice recognition is not recommendable, since it is believed that in an industrial environment, where the noise of the machines could be loud, the human voice could not be perfectly recognisable. On the other hand, a solution based on eye-tracking could be a good option. This way, operators could be able to select options with the movement of their eyes. Additionally, object recognition could also be an appropriate supplement. Thus, whenever each step is successfully completed, the application will notice, and it will automatically proceed to the posterior steps.

It would also be desirable to have access to real time data in order to read the state of each machine at every moment. For that, access to a database or a connection to a server should be implemented. Moreover, if this task was accomplished, it would mean that remote control of the industrial line could be pursued. Additionally, the maintenance report created in the end of each maintenance operation could be saved in the proposed database and could be accessed remotely, from outside the production line (or the factory, in the case of a remote monitoring from a different location).

With respect to how the information is provided in the proposed application, it would be suggestable if more meaningful dynamic figures were added. AR grants the possibility of having dynamic figures, over static ones, which can be found in paper instructions. Therefore, this opportunity should be used, and more complex graphical guidance should be offered.

In regard to security, it would be optimal if the login to the program was done by iris recognition. This way, operators would not need to enter their usernames, which would also require a mousepad or keyboard, this way the data and the system will be protected against non-authorised access to the program.

Finally, in relation to the picture taken as a proof of the maintenance job that has been completed, it would be desirable to include zooming options to the camera. It should be taken into consideration that it is not comparable to take a photo of a whole industrial machine or to a smaller dimension screw. Then, by the inclusion of these feature, operators could easily adjust which object in the image to capture. Moreover, in the developed prototype, there is an option to navigate through the different photos captured by the device. This has been reported as confusing, so, it would be better to just show the last taken image on the screen.

9 References

Anon., 2012. *Augmented Reality SDK comparison.* [Online]
Available at: <https://socialcompare.com/en/comparison/augmented-reality-sdks>
[Accessed 01 03 2018].

Anon., 2015. *Microsoft.* [Online]
Available at: <https://docs.microsoft.com/en-us/dotnet/csharp/programm8-guide/inside-a-program/hello-world-your-first-program>
[Accessed 09 05 2018].

Anon., 2018. *C++.* [Online]
Available at: <https://es.wikipedia.org/wiki/C%2B%2B>
[Accessed 02 03 2018].

Anon., 2018. *Encyclopaedia Britannica.* [Online]
Available at: <https://www.britannica.com/technology/database>
[Accessed 28 01 2018].

Anon., 2018. *Java (programming language).* [Online]
Available at: [https://en.wikipedia.org/wiki/Java_\(programming_language\)](https://en.wikipedia.org/wiki/Java_(programming_language))
[Accessed 02 03 2018].

Anon., 2018. *United Nations Global Compact.* [Online]
Available at: <https://www.unglobalcompact.org/what-is-gc/our-work/social>
[Accessed 06 02 2018].

Anon., 2018. *URL.* [Online]
Available at: <https://en.wikipedia.org/wiki/URL>
[Accessed 21 2 2018].

Anon., Last edited 2018. *Wikipedia.* [Online]
Available at: [https://en.wikipedia.org/wiki/Unity_\(game_engine\)](https://en.wikipedia.org/wiki/Unity_(game_engine))
[Accessed 09 05 2018].

Anon., n.d. *Introduction to Environmental Auditing in Public Sector*. [Online]
Available at: <https://sisu.ut.ee/env-intro/book/1-1-sustainable-development>
[Accessed 26 02 2018].

Ayala García, A., Galván Bobadilla, I., Arroyo Figueroa, G., Pérez Ramírez, M. and Muñoz Román, J., 2016. Virtual reality training system for maintenance and operation of high-voltage overhead power lines. *Virtual Reality*, 20(1), pp.27–40.

Baum, L.F., 1901. *The Master Key: An Electrical Fairy Tale, Founded Upon the Mysteries of Electricity and the Optimism of Its Devotees*. *Statewide Agricultural Land Use Baseline 2015* .

Belavkin, R. V., 2017. *Middlesex University*. [Online]
Available at: <http://www.eis.mdx.ac.uk/staffpages/rvb/teaching/BIS4435/06-Expert-Systems.pdf>
[Accessed 28 01 2018].

Bloom, Z., n.d. *The History of the URL: Path, Fragment, Query, and Auth*. [Online]
Available at: <https://eager.io/blog/the-history-of-the-url-path-fragment-query-auth/>
[Accessed 22 2 2018].

Caudell, T.P. and Mizell, D.W., 1992. Augmented reality: an application of heads-up display technology to manual manufacturing processes. *Proceedings of the Twenty-Fifth Hawaii International Conference on System Sciences*, [online] pp.659–669 vol.2. Available at: <<http://ieeexplore.ieee.org/document/183317/>>.

Chebli, C., 2009. *DBMS*. [Online]
Available at: <http://4840895.blogspot.se/2009/04/history-of-dbms.html>
[Accessed 28 01 2018].

Damala, A., Stojanovic, N., Schuchert, T., Moragues, J., Cabrera, A. and Gilleade, K., 2012. Adaptive augmented reality for cultural heritage: ARtSENSE project. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. pp.746–755.

Doswell, J.T. and Skinner, A., 2014. Augmenting human cognition with adaptive augmented reality. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. pp.104–113.

European Union, 2012. Directive 2012/19/EU of the European Parliament and of the Council of 4 July 2012 on waste electrical and electronic equipment (WEEE). *Official Journal of the European Union*, pp. 88-110.

Feiner, S., Macintyre, B. and Seligmann, D., 1993a. Knowledge-based augmented reality. *Communications of the ACM*.

Feiner, S., Macintyre, B. and Seligmann, D., 1993b. Knowledge-based augmented reality. *Communications of the ACM*, [online] 36(7), pp.53–62. Available at: <<http://portal.acm.org/citation.cfm?doid=159544.159587>>.

Goose, S., Sudarsky, S., Zhang, X. and Navab, N., 2002. SEAR: Towards a mobile and context-sensitive speech-enabled augmented reality. *Proceedings - 2002 IEEE International Conference on Multimedia and Expo, ICME 2002*, 1, pp.849–852.

Gutierrez, F., Abud, M.A., Vera, F. and Sanchez, J.A., 2013. Application of contextual QR codes to augmented reality technologies. *23rd International Conference on Electronics, Communications and Computing, CONIELECOMP 2013*, pp.264–269.

Heilig, M.L., 1992. EL Cine del Futuro: The Cinema of the Future. *Presence: Teleoperators and Virtual Environments*, [online] 1(3), pp.279–294. Available at: <<https://doi.org/10.1162/pres.1992.1.3.279>>.

Henderson, S. and Feiner, S., 2011. Exploring the benefits of augmented reality documentation for maintenance and repair. *IEEE Transactions on Visualization and Computer Graphics*, 17(10), pp.1355–1368.

Henderson, S.J. and Feiner, S., 2009. Evaluating the benefits of augmented reality for task localization in maintenance of an armored personnel carrier turret. *Science and Technology Proceedings - IEEE 2009 International Symposium on Mixed and Augmented Reality, ISMAR 2009*, pp.135–144.

Hincapié, M., Caponio, A., Rios, H. and González Mendívil, E., 2011. An introduction to Augmented Reality with applications in aeronautical maintenance. *International Conference on Transparent Optical Networks*, pp.1–4.

Jo, G.S., Oh, K.J., Ha, I., Lee, K.S., Hong, M.D., Neumann, U. and You, S., 2014. A unified framework for augmented reality and knowledge-based systems in maintaining aircraft. *Proceedings of the*

National Conference on Artificial Intelligence, [online] 4, pp.2990–2997. Available at: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-84908208676&partnerID=tZOtx3y1>>.

Jones, R., 1994. Computer-aided maintenance management systems. *Computing & Control Engineering Journal*, [online] 5(4), pp.189–192. Available at: <http://digital-library.theiet.org/content/journals/10.1049/cce_19940405>.

Klinker, G., Creighton, O., Dutoit, A.H., Kobylinski, R., Vilsmeier, C. and Brgge, B., 2001. Augmented maintenace of powerplants: A prototyping case study of a mobile {AR} system. *Proc. {ISAR} 2001*, pp.124–133.

Kurry, A., 2011. *Sustainable Development*. [Online] Available at: <https://macaulay.cuny.edu/eportfolios/akurry/2011/12/21/sustainable-development/> [Accessed 26 02 2018].

Langlois, S. and Soualmi, B., 2016. Augmented reality versus classical HUD to take over from automated driving: An aid to smooth reactions and to anticipate maneuvers1. In: *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*. pp.1571–1578.

Macaulay, n.d. *Introduction to Environmental Auditing in Public Sector*. [Online] Available at: <https://sisu.ut.ee/env-intro/book/1-1-sustainable-development> [Accessed 26 02 2018].

Martinetti, A., Rajabalinejad, M. and Van Dongen, L., 2017. Shaping the Future Maintenance Operations: Reflections on the Adoptions of Augmented Reality Through Problems and Opportunities. In: *Procedia CIRP*. pp.14–17.

Mason, M., 2018. *Environmental Science*. [Online] Available at: <https://www.environmentalscience.org/sustainability> [Accessed 05 02 2018].

Masoni, R., Ferrise, F., Bordegoni, M., Gattullo, M., Uva, A.E., Fiorentino, M., Carrabba, E. and Di Donato, M., 2017. Supporting Remote Maintenance in Industry 4.0 through Augmented Reality. *Procedia Manufacturing*, 11, pp.1296–1302.

Nakagawa, T., Sano, T. and Nakatani, Y., 1999. Plant maintenance support system by augmented reality. *IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on Systems,*

Man, and Cybernetics (Cat. No.99CH37028), 1, pp.768–773.

Nakajima, C. and Itho, N., 2003. A support system for maintenance training by augmented reality. *Proceedings - 12th International Conference on Image Analysis and Processing, ICIAP 2003*, pp.158–163.

Oates, B.J., 2006. *Researching Information Systems and Computing*. Sage Publications.

Oliveira, R. and Porto, I., 2016. Augmented Reality System for Maintenance of High-Voltage Systems.

Otaïr, M.A., Hamad, A.Q.A.L. and Jordan, M.-, 2005. Expert Personalized E-Learning Recommender System. *Knowledge Creation Diffusion Utilization*, (May 2005).

Ruan, K. and Jeong, H., 2012. An augmented reality system using QR code as marker in android smartphone. *2012 Spring World Congress on Engineering and Technology, SCET 2012 - Proceedings*, pp.1–3.

Rullo, P., Sacca, D. and Zhong, Q., 1986. An approximation algorithm for the physical access path selection in the CODASYL environment. In: *1986 IEEE Second International Conference on Data Engineering*. pp.200–207.

Stricker, D. and Bleser, G., 2012. From interactive to adaptive augmented reality. *Proceedings - 2012 International Symposium on Ubiquitous Virtual Reality, ISUVR 2012*, pp.18–21.

Sutherland, I.E., 1968. A Head-Mounted, Three-Dimensional Display. *AFIPS Proceedings of the Fall Joint Computer Conference*, Part I(16), pp.757–764.

Syberfeldt, A., Danielsson, O., Holm, M. and Wang, L., 2015. Visual Assembling Guidance Using Augmented Reality. *Procedia Manufacturing*, [online] 1, pp.98–109. Available at: <<http://linkinghub.elsevier.com/retrieve/pii/S2351978915010689>>.

Syberfeldt, A., Danielsson, O., Holm, M. and Wang, L., 2016. Dynamic Operator Instructions Based on Augmented Reality and Rule-based Expert Systems. In: *Procedia CIRP*. pp.346–351.

W. Niebel, B., 1994. *Engineering Maintenance Management*. 2 ed. United States: CRC Press.

Wang, J.T., Shyi, C.N., Hou, T.W. and Fong, C.P., 2010. Design and implementation of augmented reality system collaborating with QR code. In: *ICS 2010 - International Computer Symposium*. pp.414–418.

Willard, B., 2010. *Sustainability Advantage*. [Online]
Available at: <https://sustainabilityadvantage.com/2010/07/20/3-sustainability-models/>
[Accessed 05 02 2018].

Wilson, A., 2002. *Asset maintenance management: A guide to developing strategy and*. 1 ed. Nueva York: Industrial Press.

Xu, Y., Stojanovic, N., Stojanovic, L., Cabrera, A. and Schuchert, T., 2012. An approach for using complex event processing for adaptive augmented reality in cultural heritage domain. In: *Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems - DEBS '12*. [online] pp.139–148. Available at: <<http://dl.acm.org/citation.cfm?doid=2335484.2335500>>.

Yanachkov, M., 2017. *How to scan QR codes and regular barcodes using just your camera app and Now on Tap*. [Online]
Available at: https://www.phonearena.com/news/How-to-scan-QR-codes-and-regular-barcodes-using-just-your-camera-app-Android_id91329
[Accessed 20 02 2018].

Appendix A: Collected data from the tests

In the following paragraphs all the data collected from the tests is gathered in tables.

Each empty blue row in the table separates the three sets of tests that have been done. This is the reason why in the section that corresponds to AR glasses data there is no such a line, since these tests have only been performed in the last set of tests. Likewise, the entire line maintenance times have only been collected in the last set of tests, and so the rest of the spaces related to this parameter are empty.

The boxes in red represent that the participant has committed a mistake in that task.

For the proper comparison of the data, it has been decided that the collected data in reference to the time spent at each machine will include tenths of seconds, while the time elapsed for the maintenance of all the line will be represented only in seconds. Moreover, the averages have been rounded to two tenths of seconds.

In section 6, the Likert-type questionnaire results have been translated to marks from 0 to 4, where 0 is the worst mark and 4 the best possible one.

1. Data of participants

In Table A.1 the data of the participants is gathered. This data has been used to see if there is any tendency regarding age or nationality, and to check if being familiar with the field of engineering affects the results.

Table A.1 Data of participants

Participant number	Age (years-old)	Nationality	Expertise field	Expertise level
1	22	Spain	Engineering	Undergraduate
2	29	Spain	Neuroscience	Graduate
3	22	Spain	Biology	Undergraduate
4	24	Spain	Engineering	Undergraduate
5	26	Turkey	Engineering	Graduate
6	22	Spain	Nursing	Undergraduate
7	26	Turkey	Engineering	Graduate

8	32	Sweden	Engineering	Graduate
9	18	Sweden	Technology	Undergraduate
10	38	Sweden	Maintenance	Graduate
11	39	Sweden	Project manager	Graduate
12	41	Sweden	Engineering	Graduate
13	50	Sweden	Engineering	Graduate
14	35	Spain	Engineering	Graduate
15	23	Sweden	Engineering	Undergraduate
16	26	U.S.A.	Engineering	Undergraduate
17	24	Sweden	Engineering	Graduate
18	24	Spain	Engineering	Graduate
19	25	Spain	Engineering	Graduate
20	21	France	Business	Undergraduate
21	22	Germany	Business	Undergraduate
22	34	India	Maintenance	Graduate
23	22	Spain	Engineering	Undergraduate
24	32	Spain	Engineering	Graduate
25	36	Sweden	Engineering	Graduate
26	21	Spain	Engineering	Undergraduate
27	34	Sweden	Engineering	Graduate
28	24	Italy	Business	Undergraduate
29	25	Italy	Engineering	Undergraduate
30	23	Spain	Engineering	Undergraduate
31	28	Spain	Pharmacy	Graduate
32	25	Greece	Engineering	Undergraduate
33	22	Spain	Engineering	Undergraduate

2. Paper format tests

In Table A.2 the time needed by each participant for completing the maintenance of each of the 4 machines is represented. Also, the total time elapsed for the maintenance of the entire line is shown in the section of test number 3. Moreover, the mark given to how user-friendly participants thought this format was is listed.

Table A.2 Each machine maintenance times. Paper format

Participant number	Machine 1 (seconds)	Machine2 (seconds)	Machine3 (seconds)	Machine4 (seconds)	Entire line maintenance (seconds)	Given mark (0-10)
1	76.8	69.8	70.7	70.3		8
2	85.3	85.7	58.3	73.6		7
3	48.7	85.2	34.2	31.9		7
4	110.9	148.3	70.8	78		0
5	87.7	143	55.5	57.8		6
6	68.6	87.9	54.4	74.4		8
7	79.8	87.6	55.4	67.5		8
8	42.0	44.8	36	24.8		8
9	116.1	146.1	117.2	73.1		6
10	100.2	102.6	63.9	47.7		7
11	51.7	91.4	7.9	37.9		6
12	93.1	118.2	105.3	69.7	485	8
13	77.0	86.1	80.3	46.9	431	7
14	58.2	97.3	60.7	108.8	487	7
15	108.4	106.1	57.1	56.3	472	0
16	72.1	92.3	77.3	56.8	421	6
17	169.0	158.4	120.0	85	718	8
18	125.0	191.0	74.2	78.5	681	8
19	89.2	83.2	74.8	58.4	395	8
20	55.4	59.9	59.4	46.6	419	10
21	55.4	59.9	59.4	46.6	419	10
22	128.7	246.7	87.8	88.0	660	8
23	90.2	85.1	74.2	55.1	405	6
24	111.3	97.9	75.9	76	564	4
25	113.5	104.7	90.2	64.3	520	6
26	110.8	102.5	61.3	52.6	513	7
27	63.4	70.9	77.5	42.9	396	4
28	70.3	97.4	75.8	61.8	496	9
29	89.0	87.9	75.6	79.9	491	7
30	130.8	192.9	82.2	81.5	667	4
31	95.2	90.8	86.6	63.5	531	5
32	70.2	85.2	71.1	48.1	527	10
33	112.5	118.1	107.2	70.4	673	3

In Table A.3 the number of participants in each test is shown, together with the average times needed for executing the maintenance operation of each machine. In Table A.4 the total maintenance average times are represented, together with the average given user-friendliness mark.

Table A.3 Average times for each machine. Paper format

Number of participants	Machine 1 average (seconds)	Machine2 average (seconds)	Machine3 average (seconds)	Machine4 average (seconds)
8	74.97	94.04	54.41	59.79
3	89.33	113.37	86.33	52.90
22	94.94	110.57	78.81	65.35

Table A.4 Average time for the total maintenance and per machine and mark. Paper format

Number of participants	Average time per machine (seconds)	Entire line maintenance average (seconds)	Average given mark (0-10)
8	70.80		6.50
3	85.48		6.33
22	87.42	516.86	6.59

3. Audio format tests

In Table A.5 the time needed by each participant for completing the maintenance of each of the 4 machines is represented. Also, the total time elapsed for the maintenance of the entire line is shown in the section of test number 3. Moreover, the mark given to how user-friendly participants thought this format was is listed.

Table A.5 Each machine maintenance times. Audio format

Participant number	Machine 1 (seconds)	Machine2 (seconds)	Machine3 (seconds)	Machine4 (seconds)	Entire line maintenance (seconds)	Given mark (0-10)
1	55.0	67.7	46.7	46.8		6
2	64.9	118.9	41.7	61.3		7
3	67.1	79.9	50.9	55.4		6
4	73.1	92.0	64.3	52.9		10
5	85.6	112	81.5	53.6		10
6	72.8	79.5	51.1	49.5		9
7	67.2	118.1	114.5	52.8		9
8	120.5	85.3	70.6	51.7		8
9	62.5	73.8	50.8	88.8		7
10	44.6	48.1	50.2	52.3		5

11	33.2	47.1	39.8	45.9		8
12	43.8	47.1	76.6	43.9	342	8
13	50.3	62.0	80.9	40.7	392	2
14	34.5	47.9	81.0	60.0	350	2
15	41.0	47.8	53.0	41.0	249	7
16	84.0	70.0	41.9	31.9	301	6
17	59.4	76.9	73.0	46.4	337	6
18	47.5	72.4	44.4	33.2	328	10
19	57.7	63.4	60.0	42.5	343	6
20	45.4	44.9	34.8	27.4	218	7
21	45.4	44.9	34.8	27.4	218	7
22	39.5	46.8	61.9	27.6	258	10
23	40.1	48.9	55.1	29.1	247	9
24	38.7	46.8	34.5	27.9	232	9
25	38.5	47.1	26.8	39.6	257	10
26	39.1	50.3	59.0	27.9	252	4
27	38.8	46.9	44.8	38.2	256	5
28	43.1	68.9	35.9	27.7	311	10
29	42.8	49.4	36.7	28.6	249	8
30	39.2	78.9	45.9	30.2	336	7
31	45.1	58.2	78.4	35.1	308	8
32	42.7	47.5	46.3	30.2	293	9
33	39.7	46.7	44.8	28.2	298	5

In Table A.6 the number of participants in each test is shown, together with the average times needed for executing the maintenance operation of each machine. In Table A.7, the total maintenance average times are represented, together with the average given user-friendliness mark.

Table A.6 Average times for each machine. Audio format

Number of participants	Machine 1 average (seconds)	Machine2 average (seconds)	Machine3 average (seconds)	Machine4 average (seconds)
8	75.77	94.17	65.16	53.00
3	46.77	56.33	46.93	62.33
22	50.90	55.18	52.30	34.76

Table A.7 Average time for the total maintenance and per machine and mark. Audio format

Number of participants	Average time per machine (seconds)	Entire line maintenance average (seconds)	Average given mark (0-10)
8	72.03		8.12
3	53.09		6.66
22	48.28	289.77	7.05

4. AR in tablet format tests

In Table A.8 the time needed by each participant for completing the maintenance of each of the 4 machines is represented. Also, the total time elapsed for the maintenance of the entire line is shown in the section of test number 3. Moreover, the mark given to how user-friendly participants thought this format was is listed.

Table A.8 Each machine maintenance times. AR in tablet format

Participant number	Machine 1 (seconds)	Machine2 (seconds)	Machine3 (seconds)	Machine4 (seconds)	Entire line maintenance (seconds)	Given mark (0-10)
1	54.1	63.7	39.2	23.4		6
2	44.2	58.4	22.2	34.3		8
3	35.5	40.6	28.3	26.1		9
4	72.3	64.0	48.7	32.3		7
5	57.4	89.3	70	50.1		6
6	50.2	75.3	54.9	42		7
7	83.7	79.7	60.2	87.9		9
8	58.3	62.7	47.5	48.8		8
9	73.0	65.0	84.1	60.0		8
10	65.1	133.3	62.7	42.4		7
11	47.1	50.5	77.2	40.4		6
12	62.8	78.0	104.1	41.9	490	8
13	55.1	63.1	61.6	40.7	443	6
14	59.0	59.0	51.6	37.0	433	9
15	53.4	58.0	62.0	43.8	546	9
16	73.0	53.9	80.0	38.0	452	7
17	53.2	79.3	98.2	42.1	512	8
18	46.1	77.5	73.1	30.3	492	3
19	57.7	63.4	60.0	42.5	459	5
20	32.2	35.1	36.8	44.6	336	9
21	32.2	35.1	36.8	44.6	336	9
22	46.7	89.8	77.0	20.0	480	9
23	48.3	51.2	61.5	32.6	441	9
24	56.8	54.4	62.0	29.3	450	6
25	61.5	53.2	80.4	40.8	440	8
26	58.1	58.7	69.1	41.2	429	9
27	31.1	34.8	52.0	33.3	316	7
28	46.8	50.3	68.5	35.4	431	9
29	38.8	48.8	57.7	40.2	431	8
30	52.3	55.7	72.5	40.1	452	9
31	50.7	53.1	72.4	34.6	461	9
32	33.6	35.9	44.7	30.7	352	9
33	35.4	39.7	47.6	29.9	357	8

In Table A.9 the number of participants in each test is shown, together with the average times needed for executing the maintenance operation of each machine. In Table A.10, the total maintenance average times are represented, together with the average given user-friendliness mark.

Table A.9 Average times for each machine. AR in tablet format

Number of participants	Machine 1 average (seconds)	Machine2 average (seconds)	Machine3 average (seconds)	Machine4 average (seconds)
8	56.96	66.71	46.37	43.11
3	61.73	82.93	4.68	47.59
22	49.31	55.82	63.12	36.98

Table A.10 Average time for the total maintenance and per machine and mark. AR in tablet format

Number of participants	Average time per machine (seconds)	Entire line maintenance average (seconds)	Average given mark (0-10)
8	53.29		7.5
3	66.73		7
22	51.31	433.59	7.86

5. AR glasses format tests

In Table A.11 the time needed by each participant for completing the maintenance of each of the 4 machines is represented. It is needed to note that only the third test results are shown, since in the previous ones the implementation of the solution in AR glasses was still undone. Also, the total time elapsed for the maintenance of the entire line is shown. Moreover, the mark given to how user-friendly participants thought this format was is listed.

Table A.11 Each machine maintenance times. AR glasses format

Participant number	Machine 1 (seconds)	Machine2 (seconds)	Machine3 (seconds)	Machine4 (seconds)	Entire line maintenance (seconds)	Given mark (0-10)
12	79.3	99.3	76.9	48.8	556	6
13	77.0	86.0	74.0	49.8	771.3	5
14	68.1	108.9	67.0	32.0	454	5
15	55.2	78.0	60.1	44.9	463	8
16	64.1	70.5	60.6	51.0	410	9
17	84.8	96.3	71.7	55.6	557	7
18	56.9	60.0	62.0	28.7	449	7
19	60.1	56.2	50.0	44.0	416	7
20	87.9	91.1	64.8	47.7	508	7

21	87.9	91.1	64.8	47.7	508	7
22	49.1	76.6	60.4	42.6	464	8
23	50.2	70.3	52.6	32.0	461	6
24	65.8	57.4	56.2	45.0	420	8
25	76.4	73.1	68.4	57.4	512	7
26	80.5	80.8	69.1	45.9	451	8
27	32.0	58.4	63.6	28.7	429	7
28	45.8	63.5	46.0	32.1	442	8
29	48.9	74.2	52.3	38.9	460	7
30	83.6	93.2	73.2	41.7	472	8
31	86.1	96.4	69.9	46.2	479	8
32	59.7	80.1	60.5	36.8	431	8
33	62.1	84.8	62.3	36.5	381	6

In Table A.12 the number of participants in each test is shown, together with the average times needed for executing the maintenance operation of each machine. In Table A.13, the total maintenance average times are represented, together with the average given user-friendliness mark.

Table A.12 Average times for each machine. AR glasses format

Number of participants	Machine 1 average (seconds)	Machine2 average (seconds)	Machine3 average (seconds)	Machine4 average (seconds)
22	66.43	79.36	63.03	42.46

Table A.13 Average time for the total maintenance and per machine and mark. AR in tablet format

Number of participants	Average time per machine (seconds)	Entire line maintenance average (seconds)	Average given mark (0-10)
22	62.82	477.01	7.14

6. Feedback about the prototype

When the tests have been completed, a survey is handed to each participant to calculate how usable they think the solution is and in order to improve the solution by the received feedback. This survey has only been executed in the last two sets of tests, and that is why the first set of participants is not mirrored in the results, shown in Table A.14 and Table A.15. Due to the wide variety of answers in questions 5 and 6, they will not be listed in the report. However, the authors have considered the answers for the improvement of the prototype. The asked questions are listed below, together with the obtained results.

- Question 1: I found the application easy to use.
- Question 2: I found the structure clear in order to find the information.
- Question 3: I found the steps easy to understand.

- Question 4: All the information required is clearly explained.
- Question 5: If applies, what information is missing?
- Question 6: How can the application be improved?

Table A.14 Usability marks given to the prototype

Participant number	Question 1 (0-4)	Question 2 (0-4)	Question 3 (0-4)	Question 4 (0-4)
9	3	3	3	3
10	3	3	4	3
11	2	3	2	3
12	2	1	1	1
13	3	2	1	0
14	4	4	4	4
15	4	3	3	3
16	4	4	3	3
17	4	4	3	3
18	3	4	4	4
19	4	4	4	3
20	4	4	4	4
21	1	2	3	4
22	4	4	4	3
23	3	4	3	3
24	4	4	3	2
25	4	4	3	4
26	3	2	4	4
27	3	4	4	4
28	4	2	3	3
29	4	4	3	4
30	4	3	4	3
31	4	2	3	3
32	2	3	2	3
33	4	3	4	3

Table A.15 Average usability marks given to the prototype

Number of participants	Question 1 (0-4)	Question 2 (0-4)	Question 3 (0-4)	Question 4 (0-4)	Overall usability (%)
25	3.45	3.23	3.18	3.09	80.97

Appendix B: Table of all the alarms of the line

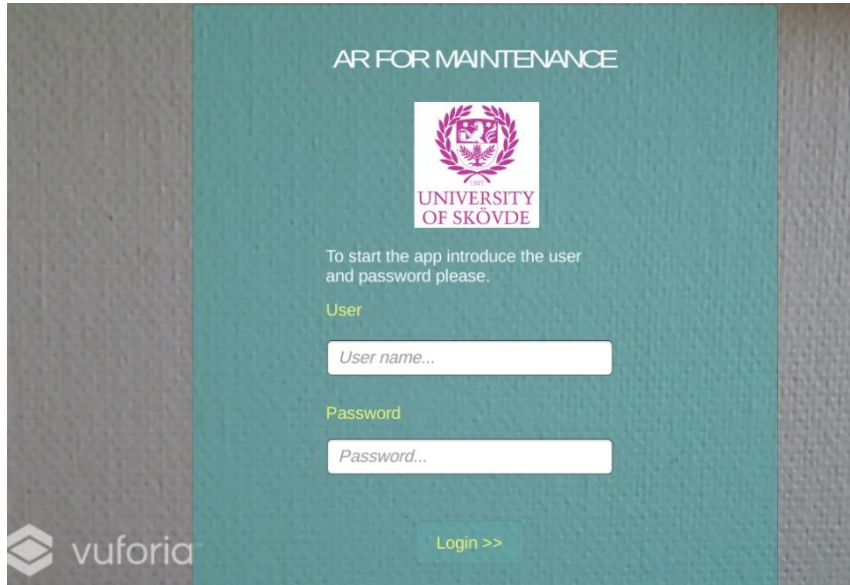
Table B.16 List of alarms

Variable	State	Alarm
Water temperature	Ok	-
	High	"Machine " + Machine number+ " water temperature is high"
	Low	"Machine " + Machine number+ " water temperature is low"
	Too high	"Machine " + Machine number+ " water temperature is too high"
	Too low	"Machine " + Machine number+ " water temperature is too low"
Housing temperature	Ok	-
	High	"Machine " + Machine number+ " housing temperature is high"
	Low	"Machine " + Machine number+ " housing temperature is low"
	Too high	"Machine " + Machine number+ " housing temperature is too high"
	Too low	"Machine " + Machine number+ " housing temperature is too low"
Pressure	Ok	-
	High	"Machine " + Machine number+ " pressure is high"
	Low	"Machine " + Machine number+ " pressure is low"
	Too high	"Machine " + Machine number+ " pressure is too high"
	Too low	"Machine " + Machine number+ "pressure is too low"
Water level	Ok	-
	High	"Machine " + Machine number+ " water level is high"
	Low	"Machine " + Machine number+ " water level is low"
	Too high	"Machine " + Machine number+ " water level is too high"
	Too low	"Machine " + Machine number+ "water level is too low"
Water flow	Ok	-
	High	"Machine " + Machine number+ " water flow is high"
	Low	"Machine " + Machine number+ " water flow is low"
	Too high	"Machine " + Machine number+ " water flow is too high"
	Too low	"Machine " + Machine number+ "water flow is too low"
Voltage	Ok	-
	High	"Machine " + Machine number+ " Voltage is high"
	Low	"Machine " + Machine number+ " Voltage is low"
	Too high	"Machine " + Machine number+ " Voltage is too high"
	Too low	"Machine " + Machine number+ "Voltage is too low"
Current	Ok	-
	High	"Machine " + Machine number+ " Current is high"
	Low	"Machine " + Machine number+ " Current is low"
	Too high	"Machine " + Machine number+ "Current is too high"
	Too low	"Machine " + Machine number+ "Current is too low"

Appendix C: Screen captures

In this appendix, the different screens that will be shown to the operator are represented, in form of screen captures.

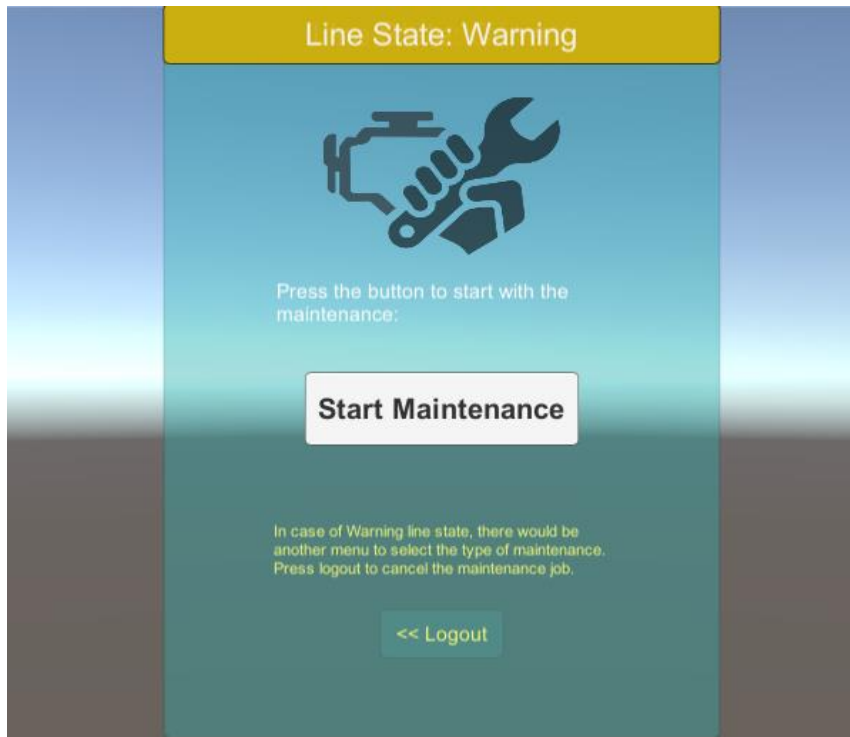
1. Login screen



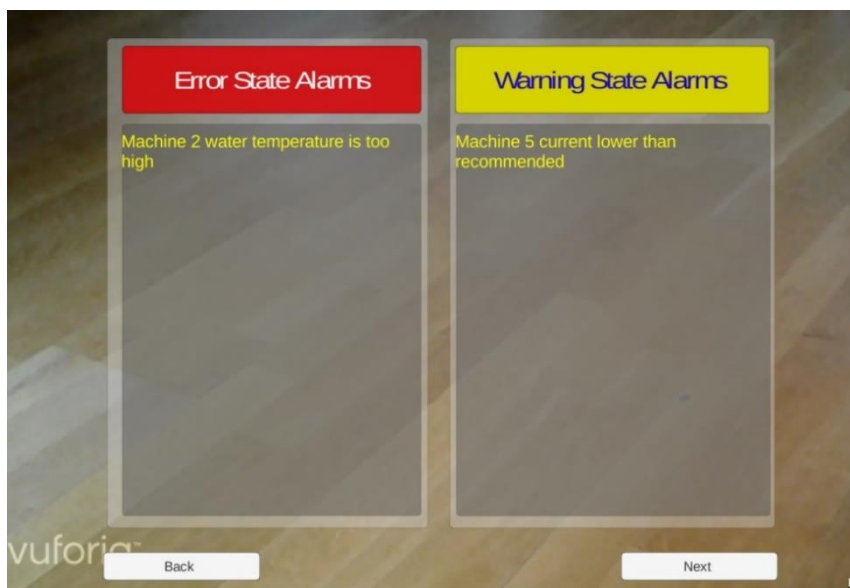
2. QR code reader screen



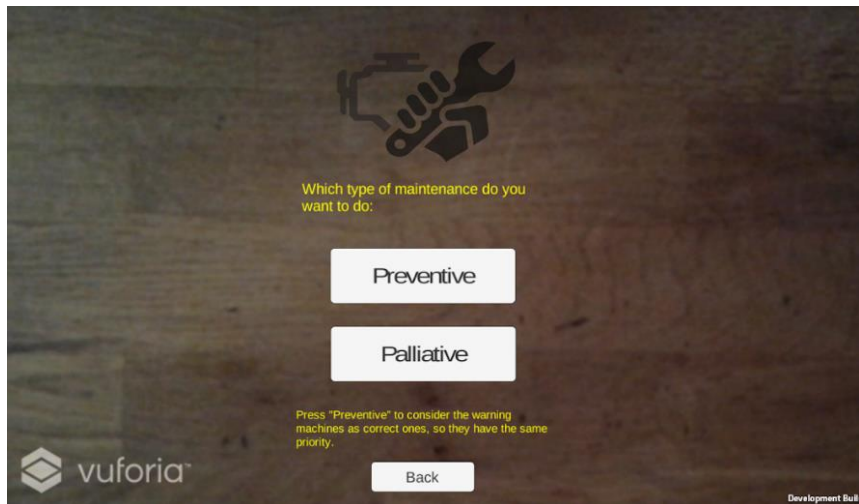
3. All line State display screen



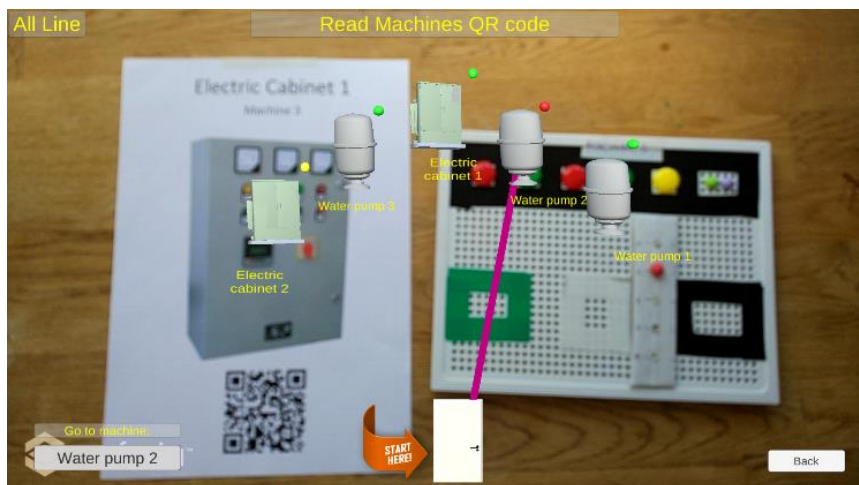
4. List of alarms screen



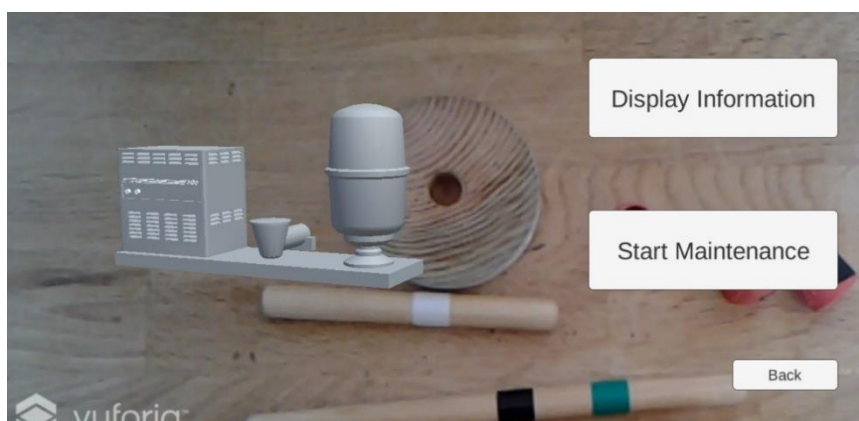
5. Preventive or palliative choice screen



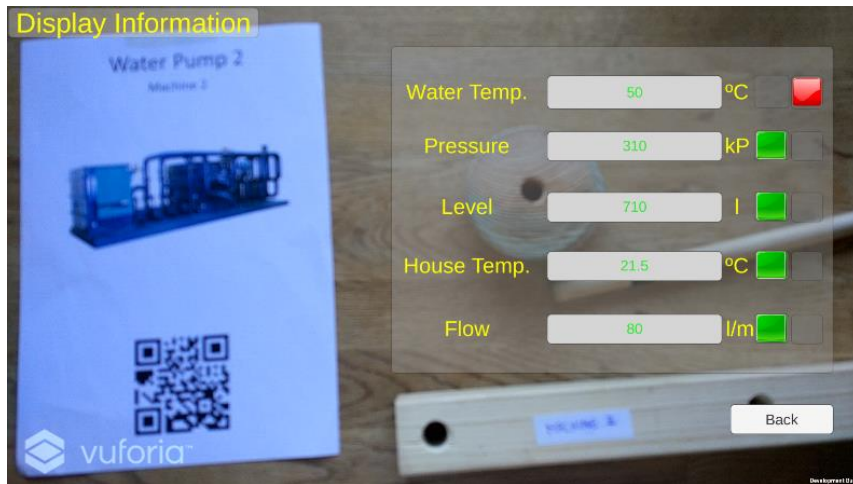
6. All line screen



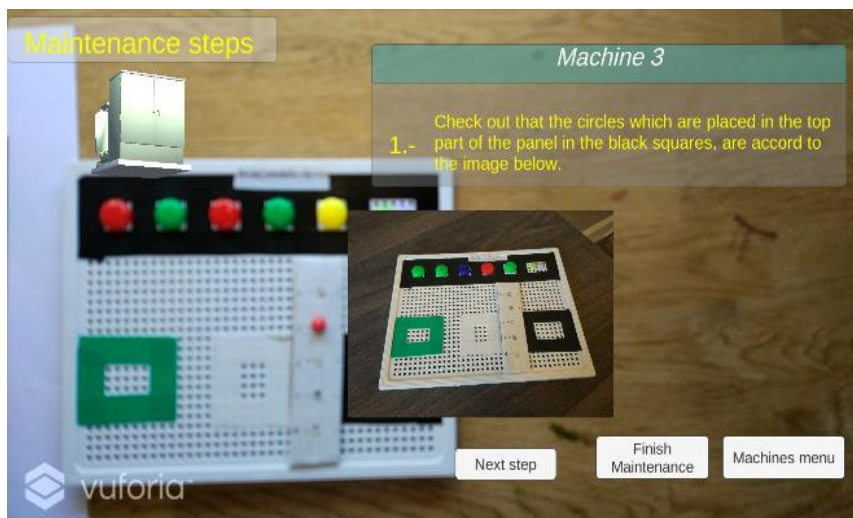
7. Machine's menu screen



8. Display information screen



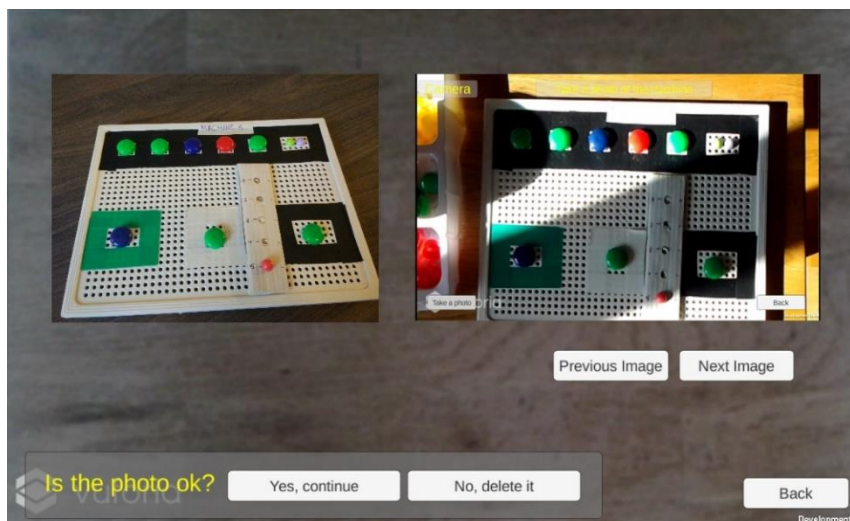
9. Steps display screen



10. Camera screen



11. Comparison of the images screen



12. Pdf capture

Water pump, Machine2

Maintenance operation information:

Data: 18-05-2018 11-29-40

Operator name: Leire

Machines ID: ASD20573

Machines part number: AD1250

Machine values:

-Water temperature: 50°C

-Pressure: 310kPa

-Level of the water: 710l

-Housing temperature: 21.5°C

-Flow of the water: 80l/m

Machine Alarms:

-There has been a water temperature error!

Followed steps:

-Step 0: completed

-Step 1: completed

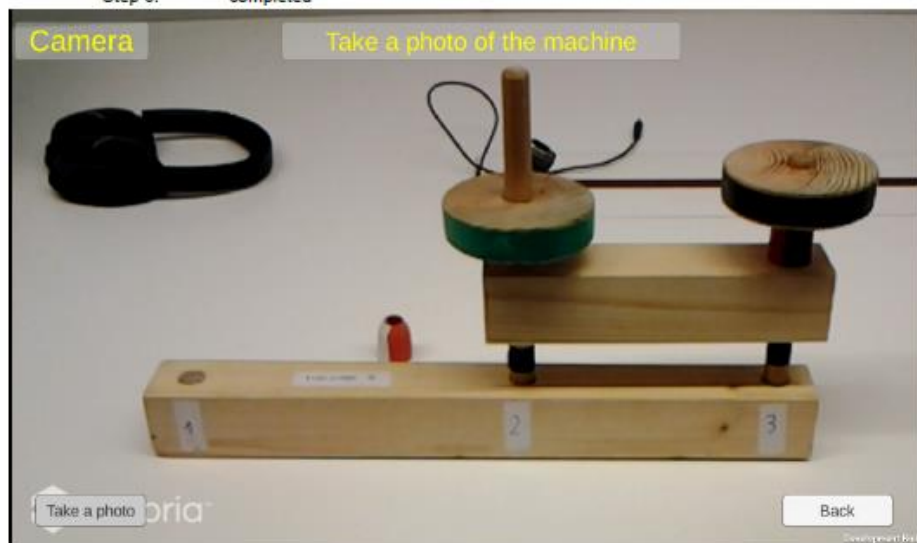
-Step 2: completed

-Step 3: completed

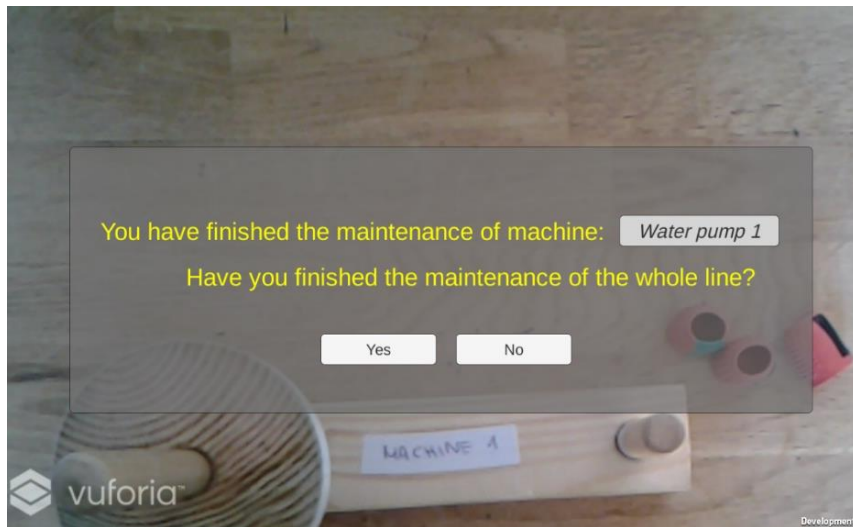
-Step 4: completed

-Step 5: completed

-Step 6: completed



13. Last display screen



Appendix D: Scripts of the program

1. Login of the program

```
/*=====
Copyright (c) 2013-2015 MrBuFF 1
URL: https://www.youtube.com/watch?v=vFs0_skd0E4&t=254s
All Rights Reserved.
Confidential and Proprietary - Protected under copyright and other laws.
Modified by: Leire Amenabar and Leire Carreras
=====*/
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using System;
using System.IO;
using System.Text.RegularExpressions;
using UnityEngine.SceneManagement;

public class Login: MonoBehaviour {
    public GameObject username;
    public GameObject password;
    private String[] Lines;
    string path = @"/storage/emulated/0/Maintenance/Login/"

    public void LoginButton(){
        bool UN = false;
        bool PW = false;

        if (MachineData.Instance().Username != "") {
            if (File.Exists(path+ MachineData.Instance().Username + ".txt")) {
                Lines = File.ReadAllLines (path+ MachineData.Instance().Username + ".txt");
                if (MachineData.Instance().Username == Lines [0]) {
                    MachineData.Instance().OpUsername = MachineData.Instance().Username;
                    UN = true;
                }
            }else {
                Debug.LogWarning ("Username invalid");
            }
        }else {
            Debug.LogWarning ("Username field empty");
        }

        if (MachineData.Instance().Password != "") {
            if (MachineData.Instance().Password.Length == 8) {
                if (System.IO.File.Exists (path+ MachineData.Instance().Username + ".txt"))
                    if (MachineData.Instance().Password == Lines [1]) {
                        PW = true;
                    }
            }else {
                Debug.LogWarning ("Invalid password");
            }
        }else {
            Debug.LogWarning ("The password must contain 8 characters");
        }
        } else {
            Debug.LogWarning ("Password field empty");
        }

        if (UN == true && PW == true) {
            username.GetComponent<InputField> ().text = "";
        }
    }
}
```

```
password.GetComponent<InputField> ().text = "";
print ("Login Sucessful");
SceneManager.LoadScene ("MaintenanceStateReader", LoadSceneMode.Single);
}
}

//update is called once per frame
void Update (){
    if (Input.GetKeyDown (KeyCode.Tab)) {
        if (username.GetComponent<InputField> ().isFocused) {
            password.GetComponent<InputField> ().Select ();
        }
    }
    if (Input.GetKeyDown (KeyCode.Return)) {
        if (MachineData.Instance().Password != " " && MachineData.Instance().Password != " ") {
            LoginButton();
        }
    }
    MachineData.Instance().Username = username.GetComponent<InputField> ().text;
    MachineData.Instance().Password = password.GetComponent<InputField> ().text;
}
}
```

2. Start point QR code reader

```
/*=====
Programmers: Leire Amenabar and Leire Carreras
Date: May 09, 2018
Revision: 1.0
Copyright 2018
Description: In this script when logout button is pressed, the login scene is loaded so that
it can be entered again in the program.
All Rights Reserved.
Confidential and Proprietary - Protected under copyright and other laws.
=====*/

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class MaintenanceStateReader : MonoBehaviour {

    public void Logout(){
        SceneManager.LoadScene ("LoginMenu");
    }

}
```

3. Start maintenance screen script

```
/*=====
Programmers: Leire Amenabar and Leire Carreras
Date: May 09, 2018
Revision: 1.0
Copyright 2018
Description: This script displays the state of the line and allows to start with the
maintenance.
=====*/

//Libraries
using System.Collections;
using System.Collections.Generic;
```

```

using UnityEngine;
using UnityEngine.UI;
using System;
using System.IO;
using System.Text.RegularExpressions;
using UnityEngine.SceneManagement;

public class StartMaintenanceMenu : MonoBehaviour {

    //Static variables
    static int MachineQty1 = 5;
    //Public variables
    //Text
    public Text errorText;
    public Text warningText;
    public Text correctText;
    //GameObjects
    public GameObject errorStateCambas;
    public GameObject correctStateCambas;
    public GameObject warningStateCambas;

    private MachineData.WaterpumpData[] WaterPumpArray = new
MachineData.WaterpumpData[MachineQty1];
    private MachineData.ElecCabinetData[] ElectricCabinetArray = new
MachineData.ElecCabinetData[MachineQty1];
    private int WaterPumpQty;
    private int ElectricCabQty;

    void Start () {

        MachineData.Instance().InitiateNow = true;
        Debug.LogWarning (MachineData.Instance().InitiateNow + "balioa du initiatenowk");
        MachineData.Instance().error = false;
        MachineData.Instance().warning = false;
        MachineData.Instance().correct = false;
        MachineData.Instance().GeneralError = false;
        MachineData.Instance().GeneralWarning = false;
        MachineData.Instance().GeneralCorrect = false;

        WaterPumpArray [0] = MachineData.Instance().DataMachine1;
        WaterPumpArray [1] = MachineData.Instance().DataMachine2;
        WaterPumpArray [2] = MachineData.Instance().DataMachine4;
        WaterPumpQty = 3;

        ElectricCabinetArray [0] = MachineData.Instance().DataMachine3;
        ElectricCabinetArray [1] = MachineData.Instance().DataMachine5;
        ElectricCabQty =2;
        MachineData.Instance().s = 0;
        MachineData.Instance().preventive = false;
        MachineData.Instance().corrective = false;
        MachineData.Instance().palliative = false;
    }

    public void StartMaintenance(){
        SceneManager.LoadScene("AlarmDisplay");
    }

    public void Logout(){
        SceneManager.LoadScene ("LoginMenu");
    }
}

```

```

}

/***** COMPARISON *****/
void CompareData ()
{
    for (int i = 0; i < WaterPumpQty; i++) {
        // Water temp check
        if (WaterPumpArray [i].WaterTemp > MachineData.Instance().MaxWaterTemp){
            // MachineData.Instance().error: Machine i water temperature too high
            errorStateCambas.SetActive (true);
            errorText.text = "Machine " + WaterPumpArray[i].MachineNum + " water temperature too
high";
            correctStateCambas.SetActive (false);
            warningStateCambas.SetActive (false);
            MachineData.Instance().error = true;
            MachineData.Instance().GeneralError = true;
            MachineData.Instance ().MachineState [i , 0] = WaterPumpArray [i].MachineNum;
            MachineData.Instance ().MachineState [i , 1] = 0;
        }
        if (WaterPumpArray [i].WaterTemp < MachineData.Instance().MinWaterTemp) {
            // MachineData.Instance().error: Machine i water temperature too low
            errorStateCambas.SetActive (true);
            errorText.text = "Machine " + WaterPumpArray[i].MachineNum + " water temperature too
low";
            correctStateCambas.SetActive (false);
            warningStateCambas.SetActive (false);
            MachineData.Instance().error = true;
            MachineData.Instance().GeneralError = true;
            MachineData.Instance ().MachineState [i , 0] = WaterPumpArray [i].MachineNum;
            MachineData.Instance ().MachineState [i , 1] = 0;
        }
        if ((WaterPumpArray [i].WaterTemp > MachineData.Instance().RecMaxWaterTemp) &&
(WaterPumpArray[i].WaterTemp < MachineData.Instance().MaxWaterTemp)){
            // MachineData.Instance().warning: Machine i water temperature high
            if (MachineData.Instance ().error == false) {
                MachineData.Instance ().MachineState [i , 0] = WaterPumpArray [i].MachineNum;
                MachineData.Instance ().MachineState [i , 1] = 1;
            }
            if (MachineData.Instance ().GeneralError == false) {
                errorStateCambas.SetActive (false);
                correctStateCambas.SetActive (false);
                warningStateCambas.SetActive (true);
                warningText.text = "Machine " + WaterPumpArray[i].MachineNum + " water temperature
high";
            }
            MachineData.Instance().GeneralWarning = true;
            MachineData.Instance().warning = true;
        }
        if ((WaterPumpArray [i].WaterTemp < MachineData.Instance().RecMinWaterTemp) &&
(WaterPumpArray[i].WaterTemp > MachineData.Instance().MinWaterTemp)){
            // MachineData.Instance().warning: Machine i water temperature low
            if (MachineData.Instance ().error == false) {
                MachineData.Instance ().MachineState [i , 0] = WaterPumpArray [i].MachineNum;
                MachineData.Instance ().MachineState [i , 1] = 1;
            }
            if (MachineData.Instance ().GeneralError == false) {
                errorStateCambas.SetActive (false);
                correctStateCambas.SetActive (false);
                warningStateCambas.SetActive (true);
                warningText.text = "Machine " + WaterPumpArray[i].MachineNum + " water temperature
high";
            }
        }
    }
}

```

```

MachineData.Instance().GeneralWarning = true;
MachineData.Instance().warning = true;
}
if ((WaterPumpArray [i].WaterTemp <= MachineData.Instance().RecMaxWaterTemp) &&
(WaterPumpArray[i].WaterTemp >= MachineData.Instance().RecMinWaterTemp)){
    // MachineData.Instance().correct: Machine i water temperature is correct
    if ((MachineData.Instance ().error == false)&&(MachineData.Instance ().warning == false))
    {
        MachineData.Instance ().MachineState [i , 0] = WaterPumpArray [i].MachineNum;
        MachineData.Instance ().MachineState [i , 1] = 2;
    }
    if ((MachineData.Instance ().GeneralError == false)&&(MachineData.Instance
()).GeneralWarning == false)) {
        errorStateCambas.SetActive (false);
        correctStateCambas.SetActive (true);
        warningStateCambas.SetActive (false);
        warningText.text = "Machine " + WaterPumpArray[i].MachineNum + " water temperature is
correct";
    }
    MachineData.Instance().correct = true;
    MachineData.Instance().GeneralCorrect = true;
}
//Housing temp check
if (WaterPumpArray [i].HousingTemp > MachineData.Instance().MaxHousingTemp){
    // MachineData.Instance().error: Machine i housing temperature too high
    errorStateCambas.SetActive (true);
    errorText.text = "Machine " + WaterPumpArray[i].MachineNum + " housing temperature too
high";
    correctStateCambas.SetActive (false);
    warningStateCambas.SetActive (false);
    MachineData.Instance().error = true;
    MachineData.Instance().GeneralError = true;
    MachineData.Instance ().MachineState [i , 0] = WaterPumpArray [i].MachineNum;
    MachineData.Instance ().MachineState [i , 1] = 0;
}
if (WaterPumpArray [i].HousingTemp < MachineData.Instance().MinHousingTemp) {
    // MachineData.Instance().error: Machine i housing temperature too low
    errorStateCambas.SetActive (true);
    errorText.text = "Machine " + WaterPumpArray[i].MachineNum + " housing temperature too
low";
    correctStateCambas.SetActive (false);
    warningStateCambas.SetActive (false);
    MachineData.Instance().error = true;
    MachineData.Instance().GeneralError = true;
    MachineData.Instance ().MachineState [i , 0] = WaterPumpArray [i].MachineNum;
    MachineData.Instance ().MachineState [i , 1] = 0;
}
if ((WaterPumpArray [i].HousingTemp > MachineData.Instance().RecMaxHousingTemp) &&
(WaterPumpArray[i].HousingTemp < MachineData.Instance().MaxHousingTemp)){
    // MachineData.Instance().warning: Machine i housing temperature high
    if (MachineData.Instance ().error == false) {
        MachineData.Instance ().MachineState [i , 0] = WaterPumpArray [i].MachineNum;
        MachineData.Instance ().MachineState [i , 1] = 1;
    }
    if (MachineData.Instance ().GeneralError == false) {
        errorStateCambas.SetActive (false);
        correctStateCambas.SetActive (false);
        warningStateCambas.SetActive (true);
        warningText.text = "Machine " + WaterPumpArray[i].MachineNum + " housing temperature
high;
    }
    MachineData.Instance().GeneralWarning = true;

```

```

MachineData.Instance().warning = true;
}
if ((WaterPumpArray [i].HousingTemp < MachineData.Instance().RecMinHousingTemp) &&
(WaterPumpArray[i].HousingTemp > MachineData.Instance().MinHousingTemp)){
    // MachineData.Instance().warning: Machine i housing temperature low
    if (MachineData.Instance ().error == false) {
        MachineData.Instance ().MachineState [i , 0] = WaterPumpArray [i].MachineNum;
        MachineData.Instance ().MachineState [i , 1] = 1;
    }
    if (MachineData.Instance ().GeneralError == false) {
        errorStateCambas.SetActive (false);
        correctStateCambas.SetActive (false);
        warningStateCambas.SetActive (true);
        warningText.text = "Machine " + WaterPumpArray[i].MachineNum + " housing temperature
low";
    }
    MachineData.Instance().GeneralWarning = true;
    MachineData.Instance().warning = true;
}
if ((WaterPumpArray [i].HousingTemp <= MachineData.Instance().RecMaxHousingTemp) &&
(WaterPumpArray[i].WaterTemp >= MachineData.Instance().RecMinHousingTemp)){
    // MachineData.Instance().correct: Machine i housing temperature is correct
    if ((MachineData.Instance ().error == false)&&(MachineData.Instance ().warning == false))
    {
        MachineData.Instance ().MachineState [i , 0] = WaterPumpArray [i].MachineNum;
        MachineData.Instance ().MachineState [i , 1] = 2;
    }
    if ((MachineData.Instance ().GeneralError == false)&&(MachineData.Instance
().GeneralWarning == false)) {
        errorStateCambas.SetActive (false);
        correctStateCambas.SetActive (true);
        warningStateCambas.SetActive (false);
        warningText.text = "Machine " + WaterPumpArray[i].MachineNum + " housing temperature is
correct";
    }
    MachineData.Instance().correct = true;
    MachineData.Instance().GeneralCorrect = true;
}
//Pressure check
if (WaterPumpArray [i].Pressure > MachineData.Instance().MaxPressure){
    // MachineData.Instance().error: Machine i pressure too high
    errorStateCambas.SetActive (true);
    errorText.text = "Machine " + WaterPumpArray[i].MachineNum + " pressure too high";
    correctStateCambas.SetActive (false);
    warningStateCambas.SetActive (false);
    MachineData.Instance().error = true;
    MachineData.Instance().GeneralError = true;
    MachineData.Instance ().MachineState [i , 0] = WaterPumpArray [i].MachineNum;
    MachineData.Instance ().MachineState [i , 1] = 0;
}
if (WaterPumpArray [i].Pressure < MachineData.Instance().MinPressure) {
    // MachineData.Instance().error: Machine i pressure too low
    errorStateCambas.SetActive (true);
    errorText.text = "Machine " + WaterPumpArray[i].MachineNum + " pressure too low";
    correctStateCambas.SetActive (false);
    warningStateCambas.SetActive (false);
    MachineData.Instance().error = true;
    MachineData.Instance().GeneralError = true;
    MachineData.Instance ().MachineState [i , 0] = WaterPumpArray [i].MachineNum;
    MachineData.Instance ().MachineState [i , 1] = 0;
}

```

```

if ((WaterPumpArray [i].Pressure > MachineData.Instance().RecMaxPressure) &&
(WaterPumpArray[i].Pressure < MachineData.Instance().MaxPressure)){
    // MachineData.Instance().warning: Machine i pressure high
    if (MachineData.Instance ().error == false) {
        MachineData.Instance ().MachineState [i , 0] = WaterPumpArray [i].MachineNum;
        MachineData.Instance ().MachineState [i , 1] = 1;
    }
    if (MachineData.Instance ().GeneralError == false) {
        errorStateCambas.SetActive (false);
        correctStateCambas.SetActive (false);
        warningStateCambas.SetActive (true);
        warningText.text = "Machine " + WaterPumpArray[i].MachineNum + " pressure high";
    }
    MachineData.Instance().GeneralWarning = true;
    MachineData.Instance().warning = true;
}

if ((WaterPumpArray [i].Pressure < MachineData.Instance().RecMinPressure) &&
(WaterPumpArray[i].Pressure > MachineData.Instance().MinPressure)){
    // MachineData.Instance().warning: Machine i pressure low
    if (MachineData.Instance ().error == false) {
        MachineData.Instance ().MachineState [i , 0] = WaterPumpArray [i].MachineNum;
        MachineData.Instance ().MachineState [i , 1] = 1;
    }
    if (MachineData.Instance ().GeneralError == false) {
        errorStateCambas.SetActive (false);
        correctStateCambas.SetActive (false);
        warningStateCambas.SetActive (true);
        warningText.text = "Machine " + WaterPumpArray[i].MachineNum + " pressure low";
    }
    MachineData.Instance().GeneralWarning = true;
    MachineData.Instance().warning = true;
}

if ((WaterPumpArray [i].Pressure <= MachineData.Instance().RecMaxPressure) &&
(WaterPumpArray [i].Pressure >= MachineData.Instance().RecMinPressure)){
    // MachineData.Instance().correct: Machine i pressure is correct
    if ((MachineData.Instance ().error == false)&&(MachineData.Instance ().warning == false)) {
        MachineData.Instance ().MachineState [i , 0] = WaterPumpArray [i].MachineNum;
        MachineData.Instance ().MachineState [i , 1] = 2;
    }
    if ((MachineData.Instance ().GeneralError == false)&&(MachineData.Instance
().GeneralWarning == false)) {
        errorStateCambas.SetActive (false);
        correctStateCambas.SetActive (true);
        warningStateCambas.SetActive (false);
        warningText.text = "Machine " + WaterPumpArray[i].MachineNum + " pressure is correct";
    }
    MachineData.Instance().correct = true;
    MachineData.Instance().GeneralCorrect = true;
}

//Water level check
if (WaterPumpArray [i].WaterLevel > MachineData.Instance().MaxWaterLevel){
    // MachineData.Instance().error: Machine i.machineNum water level too high
    errorStateCambas.SetActive (true);
    errorText.text = "Machine " + WaterPumpArray[i].MachineNum + " water level too high";
    correctStateCambas.SetActive (false);
    warningStateCambas.SetActive (false);
    MachineData.Instance().error = true;
    MachineData.Instance().GeneralError = true;
    MachineData.Instance ().MachineState [i , 0] = WaterPumpArray [i].MachineNum;
    MachineData.Instance ().MachineState [i , 1] = 0;
}

```

```

if (WaterPumpArray [i].WaterLevel < MachineData.Instance().MinWaterLevel) {
    // MachineData.Instance().error: Machine i water level too low
    errorStateCambas.SetActive (true);
    errorText.text = "Machine " + WaterPumpArray[i].MachineNum + " water level too low";
    correctStateCambas.SetActive (false);
    warningStateCambas.SetActive (false);
    MachineData.Instance().error = true;
    MachineData.Instance().GeneralError = true;
    MachineData.Instance ().MachineState [i , 0] = WaterPumpArray [i].MachineNum;
    MachineData.Instance ().MachineState [i , 1] = 0;
}
if ((WaterPumpArray [i].WaterLevel > MachineData.Instance().RecMaxWaterLevel) &&
(WaterPumpArray[i].Pressure < MachineData.Instance().MaxWaterLevel)){
    // MachineData.Instance().warning: Machine i water level high
    if (MachineData.Instance ().error == false) {
        MachineData.Instance ().MachineState [i , 0] = WaterPumpArray [i].MachineNum;
        MachineData.Instance ().MachineState [i , 1] = 1;
    }
    if (MachineData.Instance ().GeneralError == false) {
        errorStateCambas.SetActive (false);
        correctStateCambas.SetActive (false);
        warningStateCambas.SetActive (true);
        warningText.text = "Machine " + WaterPumpArray[i].MachineNum + " water level high";
    }
    MachineData.Instance().GeneralWarning = true;
    MachineData.Instance().warning = true;
}
if ((WaterPumpArray [i].WaterLevel < MachineData.Instance().RecMinWaterLevel) &&
(WaterPumpArray[i].WaterLevel > MachineData.Instance().MinWaterLevel)){
    // MachineData.Instance().warning: Machine i water level low
    if (MachineData.Instance ().error == false) {
        MachineData.Instance ().MachineState [i , 0] = WaterPumpArray [i].MachineNum;
        MachineData.Instance ().MachineState [i , 1] = 1;
    }
    if (MachineData.Instance ().GeneralError == false) {
        errorStateCambas.SetActive (false);
        correctStateCambas.SetActive (false);
        warningStateCambas.SetActive (true);
        warningText.text = "Machine " + WaterPumpArray[i].MachineNum + " water level low";
    }
    MachineData.Instance().GeneralWarning = true;
    MachineData.Instance().warning = true;
}
if ((WaterPumpArray [i].WaterLevel <= MachineData.Instance().RecMaxWaterLevel) &&
(WaterPumpArray [i].WaterLevel >= MachineData.Instance().RecMinWaterLevel)){
    // MachineData.Instance().correct: Machine i water level is correct
    if ((MachineData.Instance ().error == false)&&(MachineData.Instance ().warning == false)) {
        MachineData.Instance ().MachineState [i , 0] = WaterPumpArray [i].MachineNum;
        MachineData.Instance ().MachineState [i , 1] = 2;
    }
    if ((MachineData.Instance ().GeneralError == false)&&(MachineData.Instance
().GeneralWarning == false)) {
        errorStateCambas.SetActive (false);
        correctStateCambas.SetActive (true);
        warningStateCambas.SetActive (false);
        warningText.text = "Machine " + WaterPumpArray[i].MachineNum + " water level is correct";
    }
    MachineData.Instance().correct = true;
    MachineData.Instance().GeneralCorrect = true;
}
//Flow check
if (WaterPumpArray [i].Flow > MachineData.Instance().MaxFlow){

```

```

// MachineData.Instance().error: Machine i.machineNum flow too high
errorStateCambas.SetActive (true);
errorText.text = "Machine " + WaterPumpArray[i].MachineNum + " flow too high";
correctStateCambas.SetActive (false);
warningStateCambas.SetActive (false);
MachineData.Instance().error = true;
MachineData.Instance().GeneralError = true;
MachineData.Instance ().MachineState [i , 0] = WaterPumpArray [i].MachineNum;
MachineData.Instance ().MachineState [i , 1] = 0;
}
if (WaterPumpArray [i].Flow < MachineData.Instance().MinFlow) {
// MachineData.Instance().error: Machine i flow too low
errorStateCambas.SetActive (true);
errorText.text = "Machine " + WaterPumpArray[i].MachineNum + " flow too low";
correctStateCambas.SetActive (false);
warningStateCambas.SetActive (false);
MachineData.Instance().error = true;
MachineData.Instance().GeneralError = true;
MachineData.Instance ().MachineState [i , 0] = WaterPumpArray [i].MachineNum;
MachineData.Instance ().MachineState [i , 1] = 0;
}
if ((WaterPumpArray [i].Flow > MachineData.Instance().RecMaxFlow) && (WaterPumpArray[i].Flow
< MachineData.Instance().MaxFlow)){
// MachineData.Instance().warning: Machine i flow high
if (MachineData.Instance ().error == false) {
MachineData.Instance ().MachineState [i , 0] = WaterPumpArray [i].MachineNum;
MachineData.Instance ().MachineState [i , 1] = 1;
}
if (MachineData.Instance ().GeneralError == false) {
errorStateCambas.SetActive (false);
correctStateCambas.SetActive (false);
warningStateCambas.SetActive (true);
warningText.text = "Machine " + WaterPumpArray[i].MachineNum + " flow high";
}
MachineData.Instance().GeneralWarning = true;
MachineData.Instance().warning = true;
}
if ((WaterPumpArray [i].Flow < MachineData.Instance().RecMinFlow) && (WaterPumpArray[i].Flow
> MachineData.Instance().RecMinFlow)){
// MachineData.Instance().warning: Machine i flow low
if (MachineData.Instance ().error == false) {
MachineData.Instance ().MachineState [i , 0] = WaterPumpArray [i].MachineNum;
MachineData.Instance ().MachineState [i , 1] = 1;
}
if (MachineData.Instance ().GeneralError == false) {
errorStateCambas.SetActive (false);
correctStateCambas.SetActive (false);
warningStateCambas.SetActive (true);
warningText.text = "Machine " + WaterPumpArray[i].MachineNum + " flow low";
}
MachineData.Instance().GeneralWarning = true;
MachineData.Instance().warning = true;
}
if ((WaterPumpArray [i].Flow <= MachineData.Instance().RecMaxFlow) && (WaterPumpArray
[i].Flow >= MachineData.Instance().RecMinFlow)){
// MachineData.Instance().correct: Machine i flow is correct
if ((MachineData.Instance ().error == false)&&(MachineData.Instance ().warning == false)) {
MachineData.Instance ().MachineState [i , 0] = WaterPumpArray [i].MachineNum;
MachineData.Instance ().MachineState [i , 1] = 2;
}
if ((MachineData.Instance ().GeneralError == false)&&(MachineData.Instance
().GeneralWarning == false)) {

```

```

errorStateCambas.SetActive (false);
correctStateCambas.SetActive (true);
warningStateCambas.SetActive (false);
warningText.text = "Machine " + WaterPumpArray[i].MachineNum + " flow is correct";
}
MachineData.Instance().correct = true;
MachineData.Instance().GeneralCorrect = true;
}
MachineData.Instance ().MachineState [i , 2] = i+1;
MachineData.Instance ().error = false;
MachineData.Instance ().correct = false;
MachineData.Instance ().warning = false;
}
for (int n = 0; n < ElectricCabQty; n++) {
//Voltage check
if (ElectricCabinetArray [n].Voltage > MachineData.Instance().MaxVoltage){
// MachineData.Instance().error: Machine i.machineNum voltage too high
errorStateCambas.SetActive (true);;
errorText.text = "Machine " + ElectricCabinetArray[n].MachineNum + " voltage too high";
correctStateCambas.SetActive (false);
warningStateCambas.SetActive (false);
MachineData.Instance().error = true;
MachineData.Instance().GeneralError = true;
MachineData.Instance ().MachineState [n+3 , 0] = ElectricCabinetArray[n].MachineNum;
MachineData.Instance ().MachineState [n+3 , 1] = 0;
}
if (ElectricCabinetArray [n].Voltage < MachineData.Instance().MinVoltage) {
// MachineData.Instance().error: Machine i voltage too low
errorStateCambas.SetActive (true);;
errorText.text = "Machine " + ElectricCabinetArray[n].MachineNum + " voltage too low";
correctStateCambas.SetActive (false);
warningStateCambas.SetActive (false);
MachineData.Instance().error = true;
MachineData.Instance().GeneralError = true;
MachineData.Instance ().MachineState [n+3 , 0] = ElectricCabinetArray[n].MachineNum;
MachineData.Instance ().MachineState [n+3 , 1] = 0;
}
if ((ElectricCabinetArray [n].Voltage > MachineData.Instance().RecMaxVoltage) &&
(ElectricCabinetArray[n].Voltage < MachineData.Instance().MaxVoltage)){
// MachineData.Instance().warning: Machine i voltage high
if (MachineData.Instance ().error == false) {
MachineData.Instance ().MachineState [n+3 , 0] = ElectricCabinetArray[n].MachineNum;
MachineData.Instance ().MachineState [n+3 , 1] = 1;
}
if (MachineData.Instance ().GeneralError == false) {
errorStateCambas.SetActive (false);
correctStateCambas.SetActive (false);
warningStateCambas.SetActive (true);
warningText.text = "Machine " + ElectricCabinetArray[n].MachineNum + " voltage high";
}
MachineData.Instance().GeneralWarning = true;
MachineData.Instance().warning = true;
}
if ((ElectricCabinetArray [n].Voltage < MachineData.Instance ().RecMinVoltage) &&
(ElectricCabinetArray [n].Voltage > MachineData.Instance ().MinVoltage)) {
// MachineData.Instance().warning: Machine i voltage low
if (MachineData.Instance ().error == false) {
MachineData.Instance ().MachineState [n+3 , 0] = ElectricCabinetArray [n].MachineNum;
MachineData.Instance ().MachineState [n+3 , 1] = 1;
}
if (MachineData.Instance ().GeneralError == false) {
errorStateCambas.SetActive (false);

```

```

correctStateCambas.SetActive (false);
warningStateCambas.SetActive (true);
warningText.text = ElectricCabinetArray[n].MachineNum + " voltage low";
}
MachineData.Instance().GeneralWarning = true;
MachineData.Instance().warning = true;
}
if ((ElectricCabinetArray [n].Voltage <= MachineData.Instance().RecMaxVoltage) &&
(ElectricCabinetArray [n].Voltage >= MachineData.Instance ().RecMinVoltage)){
// MachineData.Instance().correct: Machine i voltage is correct
if ((MachineData.Instance ().error == false)&&(MachineData.Instance ().warning == false)) {
MachineData.Instance ().MachineState [n+3 , 0] = ElectricCabinetArray [n].MachineNum;
MachineData.Instance ().MachineState [n+3 , 1] = 2;
}
if ((MachineData.Instance ().GeneralError == false)&&(MachineData.Instance
().GeneralWarning == false)) {
errorStateCambas.SetActive (false);
correctStateCambas.SetActive (true);
warningStateCambas.SetActive (false);
warningText.text = "Machine " +ElectricCabinetArray[n].MachineNum + " voltage is correct";
}
MachineData.Instance().correct = true;
MachineData.Instance().GeneralCorrect = true;
}
//Current check
if (ElectricCabinetArray [n].Current > MachineData.Instance().MaxCurrent){
// MachineData.Instance().error: Machine i.machineNum current too high
errorStateCambas.SetActive (true);;
errorText.text = "Machine " + ElectricCabinetArray[n].MachineNum + " current too high";
correctStateCambas.SetActive (false);
warningStateCambas.SetActive (false);
MachineData.Instance().error = true;
MachineData.Instance().GeneralError = true;
MachineData.Instance ().MachineState [n+3 , 0] =ElectricCabinetArray[n].MachineNum;
MachineData.Instance ().MachineState [n+3 , 1] = 0;
}
if (ElectricCabinetArray [n].Current < MachineData.Instance().MinCurrent) {
// MachineData.Instance().error: Machine i current too low
errorStateCambas.SetActive (true);;
errorText.text = "Machine " + ElectricCabinetArray[n].MachineNum + " voltage too low";
correctStateCambas.SetActive (false);
warningStateCambas.SetActive (false);
MachineData.Instance().error = true;
MachineData.Instance().GeneralError = true;
MachineData.Instance ().MachineState [n+3 , 0] = ElectricCabinetArray[n].MachineNum;
MachineData.Instance ().MachineState [n+3 , 1] = 0;
}
if ((ElectricCabinetArray [n].Current > MachineData.Instance().RecMaxCurrent) &&
(ElectricCabinetArray[n].Current < MachineData.Instance().MaxCurrent)){
// MachineData.Instance().warning: Machine i current high
if (MachineData.Instance ().error == false) {
MachineData.Instance ().MachineState [n+3 , 0] = ElectricCabinetArray[n].MachineNum;
MachineData.Instance ().MachineState [n+3 , 1] = 1;
}
if (MachineData.Instance ().GeneralError == false) {
errorStateCambas.SetActive (false);
correctStateCambas.SetActive (false);
warningStateCambas.SetActive (true);
warningText.text = "Machine " + ElectricCabinetArray[n].MachineNum + " current high";
}
MachineData.Instance().GeneralWarning = true;
MachineData.Instance().warning = true;
}

```

```

    }
    if ((ElectricCabinetArray [n].Current < MachineData.Instance().RecMinCurrent) &&
(ElectricCabinetArray[n].Current > MachineData.Instance().MinCurrent)){
        // MachineData.Instance().warning: Machine i current low
        if (MachineData.Instance ().error == false) {
            MachineData.Instance ().MachineState [n+3 , 0] =ElectricCabinetArray[n].MachineNum;
            MachineData.Instance ().MachineState [n+3 , 1] = 1;
        }
        if (MachineData.Instance ().GeneralError == false) {
            errorStateCambas.SetActive (false);
            correctStateCambas.SetActive (false);
            warningStateCambas.SetActive (true);
            warningText.text = "Machine " + ElectricCabinetArray[n].MachineNum + " current low";
        }
        MachineData.Instance().GeneralWarning = true;
        MachineData.Instance().warning = true;
    }
    if ((ElectricCabinetArray [n].Current <= MachineData.Instance().RecMaxCurrent) &&
(ElectricCabinetArray [n].Current >= MachineData.Instance().RecMinCurrent)){
        // MachineData.Instance().correct: Machine i current is correct
        if ((MachineData.Instance ().error == false)&&(MachineData.Instance ().warning == false))
        {
            MachineData.Instance ().MachineState [n+3 , 0] = ElectricCabinetArray [n].MachineNum;
            MachineData.Instance ().MachineState [n+3 , 1] = 2;
        }
        if ((MachineData.Instance ().GeneralError == false)&&(MachineData.Instance
().GeneralWarning == false)) {
            errorStateCambas.SetActive (false);
            correctStateCambas.SetActive (true);
            warningStateCambas.SetActive (false);
            warningText.text = "Machine " + ElectricCabinetArray[n].MachineNum + " current is
correct";
        }
        MachineData.Instance().correct = true;
        MachineData.Instance().GeneralCorrect = true;
    }
    MachineData.Instance ().MachineState [n+3 , 2] = n+1;
    MachineData.Instance ().error = false;
    MachineData.Instance ().correct = false;
    MachineData.Instance ().warning = false;
}
}

void Update (){
    CompareData ();
}
}

```

4. Alarms display script

```

/*=====
Programmers: Leire Amenabar and Leire Carreras
Date: May 09, 2018
Revision: 1.0
Copyright 2018
Description: This script displays all the error and warning alarms that are present in the
line.
Confidential and Proprietary - Protected under copyright and other laws.
=====*/
//Libraries
using System.Collections;

```

```

using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using System.IO;
using UnityEngine.SceneManagement;

public class AlarmDisplay : MonoBehaviour {
    static int MachineQty2 = 5;
    public Text errorText;
    public Text warningText;
    private MachineData.WaterpumpData[] WaterPumpArray = new
MachineData.WaterpumpData[MachineQty2];
    private MachineData.ElecCabinetData[] ElectricCabinetArray = new
MachineData.ElecCabinetData[MachineQty2];

    private int WaterPumpQty;
    private int ElectricCabQty;
    public void NextButton()
    {
        // When the next button is pressed depending if there is any warning case in the line,
        displays different scenes
        if ((MachineData.Instance().GeneralCorrect) && (!MachineData.Instance().GeneralWarning) &&
(!MachineData.Instance().GeneralError))
        {
            MachineData.Instance().preventive = true;
            MachineData.Instance().corrective = false;
            MachineData.Instance().palliative = false;
            SceneManager.LoadScene("AllLineDisplay");
        }
        if (MachineData.Instance().GeneralWarning)
        {
            SceneManager.LoadScene("WarningCaseScenario");
        }
        if ((MachineData.Instance().GeneralError) && (!MachineData.Instance().GeneralWarning))
        {
            MachineData.Instance().corrective = true;
            MachineData.Instance().preventive = false;
            MachineData.Instance().palliative = false;
            SceneManager.LoadScene("AllLineDisplay");
        }
    }
    // Use this for initialization
    void Start () {
        WaterPumpArray[0] = MachineData.Instance().DataMachine1;
        WaterPumpArray[1] = MachineData.Instance().DataMachine2;
        WaterPumpArray[2] = MachineData.Instance().DataMachine4;
        WaterPumpQty = 3;
        ElectricCabinetArray[0] = MachineData.Instance().DataMachine3;
        ElectricCabinetArray[1] = MachineData.Instance().DataMachine5;
        ElectricCabQty = 2;
    }

    // Update is called once per frame
    void Update()
    {
        for (int i = 0; i < WaterPumpQty; i++)
        {
            // Water temp check
            if (WaterPumpArray[i].WaterTemp > MachineData.Instance().MaxWaterTemp)
            {
                errorText.text = "Machine " + WaterPumpArray[i].MachineNum + " water temperature is too
high\n";
            }
        }
    }
}

```

```

    }
    if (WaterPumpArray[i].WaterTemp < MachineData.Instance().MinWaterTemp)
    {
        errorText.text = "Machine " + WaterPumpArray[i].MachineNum + " water temperature too
low\n";
    }
    if ((WaterPumpArray[i].WaterTemp > MachineData.Instance().RecMaxWaterTemp) &&
(WaterPumpArray[i].WaterTemp < MachineData.Instance().MaxWaterTemp))
    {
        warningText.text = "Machine " + WaterPumpArray[i].MachineNum + " water temperature higher
than recommended\n";
    }
    if ((WaterPumpArray[i].WaterTemp < MachineData.Instance().RecMinWaterTemp) &&
(WaterPumpArray[i].WaterTemp > MachineData.Instance().MinWaterTemp))
    {
        warningText.text = "Machine " + WaterPumpArray[i].MachineNum + " water temperature lower
than recommended\n";
    }

    //Housing temp check
    if (WaterPumpArray[i].HousingTemp > MachineData.Instance().MaxHousingTemp)
    {
        errorText.text = "Machine " + WaterPumpArray[i].MachineNum + " housing temperature too
high\n";
    }
    if (WaterPumpArray[i].HousingTemp < MachineData.Instance().MinHousingTemp)
    {
        errorText.text = "Machine " + WaterPumpArray[i].MachineNum + " housing temperature too
low\n";
    }
    if ((WaterPumpArray[i].HousingTemp > MachineData.Instance().RecMaxHousingTemp) &&
(WaterPumpArray[i].HousingTemp < MachineData.Instance().MaxHousingTemp))
    {
        warningText.text = "Machine " + WaterPumpArray[i].MachineNum + " housing temperature
higher than recommended\n";
    }
    if ((WaterPumpArray[i].HousingTemp < MachineData.Instance().RecMinHousingTemp) &&
(WaterPumpArray[i].HousingTemp > MachineData.Instance().MinHousingTemp))
    {
        warningText.text = "Machine " + WaterPumpArray[i].MachineNum + " housing temperature lower
than recommended\n";
    }

    //Pressure check
    if (WaterPumpArray[i].Pressure > MachineData.Instance().MaxPressure)
    {
        errorText.text = "Machine " + WaterPumpArray[i].MachineNum + " pressure too high\n";
    }
    if (WaterPumpArray[i].Pressure < MachineData.Instance().MinPressure)
    {
        errorText.text = "Machine " + WaterPumpArray[i].MachineNum + " pressure too low\n";
    }
    if ((WaterPumpArray[i].Pressure > MachineData.Instance().RecMaxPressure) &&
(WaterPumpArray[i].Pressure < MachineData.Instance().MaxPressure))
    {
        warningText.text = "Machine " + WaterPumpArray[i].MachineNum + " pressure higher than
recommended\n";
    }
    if ((WaterPumpArray[i].Pressure < MachineData.Instance().RecMinPressure) &&
(WaterPumpArray[i].Pressure > MachineData.Instance().MinPressure))
    {
        warningText.text = "Machine " + WaterPumpArray[i].MachineNum + " pressure lower than
recommended\n";
    }

```

```

    }
    //Water level check
    if (WaterPumpArray[i].WaterLevel > MachineData.Instance().MaxWaterLevel)
    {
        errorText.text = "Machine " + WaterPumpArray[i].MachineNum + " water level too high\n";
    }
    if (WaterPumpArray[i].WaterLevel < MachineData.Instance().MinWaterLevel)
    {
        errorText.text = "Machine " + WaterPumpArray[i].MachineNum + " water level too low\n";
    }
    if ((WaterPumpArray[i].WaterLevel > MachineData.Instance().RecMaxWaterLevel) &&
    (WaterPumpArray[i].Pressure < MachineData.Instance().MaxWaterLevel))
    {
        warningText.text = "Machine " + WaterPumpArray[i].MachineNum + " water level higher than
recommended\n";
    }
    if ((WaterPumpArray[i].WaterLevel < MachineData.Instance().RecMinWaterLevel) &&
    (WaterPumpArray[i].WaterLevel > MachineData.Instance().MinWaterLevel))
    {
        warningText.text = "Machine " + WaterPumpArray[i].MachineNum + " water level lower than
recommended\n";
    }
    //Flow check
    if (WaterPumpArray[i].Flow > MachineData.Instance().MaxFlow)
    {
        errorText.text = "Machine " + WaterPumpArray[i].MachineNum + " flow too high\n";
    }
    if (WaterPumpArray[i].Flow < MachineData.Instance().MinFlow)
    {
        errorText.text = "Machine " + WaterPumpArray[i].MachineNum + " flow too low\n";
    }
    if ((WaterPumpArray[i].Flow > MachineData.Instance().RecMaxFlow) && (WaterPumpArray[i].Flow
< MachineData.Instance().MaxFlow))
    {
        warningText.text = "Machine " + WaterPumpArray[i].MachineNum + " flow higher than
recommended\n";
    }
    if ((WaterPumpArray[i].Flow < MachineData.Instance().RecMinFlow) && (WaterPumpArray[i].Flow
> MachineData.Instance().RecMinFlow))
    {
        warningText.text = "Machine " + WaterPumpArray[i].MachineNum + " flow lower than
recommended\n";
    }
}
}

```

5. Warning case script

```

/*=====
Programmers: Leire Amenabar and Leire Carreras
Date: May 09, 2018
Revision: 1.0
Copyright 2018
Description: This script displays two ways to treat warning cases, which are preventive or
palliative. Depending which is selected by the user,
a different priority will be given to the warning case machineries.
Confidential and Proprietary - Protected under copyright and other laws.
=====*/
//Libraries
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

```

```
using UnityEngine.SceneManagement;

public class WarningCaseProg : MonoBehaviour {

    // Use this for initialization
    void Start () {
        MachineData.Instance().palliative = false;
        MachineData.Instance().preventive = false;
    }

    //Preventive button action
    public void PreventiveButton(){
        MachineData.Instance().preventive = true;
        MachineData.Instance().palliative = false;
        MachineData.Instance().corrective = false;
        SceneManager.LoadScene ("AllLineDisplay");
    }

    //Palliative button action
    public void PalliativeButton(){
        MachineData.Instance().preventive = false;
        MachineData.Instance().palliative = true;
        MachineData.Instance().corrective = false;
        SceneManager.LoadScene ("AllLineDisplay");
    }

    //Back button action
    public void WarningBack(){
        SceneManager.LoadScene ("AlarmDisplay");
    }

}
```

6. All line script

```
/*=====
Programmers: Leire Amenabar and Leire Carreras
Date: May 09, 2018
Revision: 1.0
Copyright 2018
Description: In this program, all line is represented by 3D objects with their respective
names and states (error, warning and correct).
        Apart from that, the path which has to be followed by the user is calculated, and
displayed by a moving arrow
All Rights Reserved.
Confidential and Proprietary - Protected under copyright and other laws.
=====*/

//Library
using System;
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;
using UnityEngine.UI;

public class AllLineProgram : MonoBehaviour
{
```

```
// Public variables
public Material[] materials;
//Machines GameObjects
public GameObject Machine1;
public GameObject Machine2;
public GameObject Machine3;
public GameObject Machine4;
public GameObject Machine5;
//Machine positions
public Transform InitialPos;
public Transform Machine1data;
public Transform Machine2data;
public Transform Machine3data;
public Transform Machine4data;
public Transform Machine5data;
//Bulbs variables
public MeshRenderer BulbMachine1;
public MeshRenderer BulbMachine2;
public MeshRenderer BulbMachine3;
public MeshRenderer BulbMachine4;
public MeshRenderer BulbMachine5;
//Machines names
public Text Machine1Name;
public Text Machine2Name;
public Text Machine3Name;
public Text Machine4Name;
public Text Machine5Name;
//Next machine name text
public Text GoToMachineName;

//Private variables
//Arrows variables
private LineRenderer lineRenderer;
private float counter;
private float dist;
private Transform origin; //takes the position of the object
private Transform destination; //takes the position of the object
private float lineDrawSpeed = 30f; //how quickly the arrow moves
//Program variables
private int NextMachineNumber;
private int CurrentMachineNumber;
private int LastUsedMachine;
private int PreviousUsedMachine;
private bool PriorityMachine;
static int MachineQty = 5;
private static int MaxRowQty = 100;
private int GoodQty = 0;
private int WarningQty = 0;
private int ErrorQty = 0;
private MachineData.MachinePosition[] GoodArray = new
MachineData.MachinePosition[MachineQty];
private MachineData.MachinePosition[] WarningArray = new
MachineData.MachinePosition[MachineQty];
private MachineData.MachinePosition[] ErrorArray = new
MachineData.MachinePosition[MachineQty];

//Machine Positions variables
MachineData.MachinePosition StartPosition = new MachineData.MachinePosition (0,0, 0);
MachineData.MachinePosition PosMachine1 = new MachineData.MachinePosition (1, 5, 10);
MachineData.MachinePosition PosMachine2 = new MachineData.MachinePosition (2, 1, 2);
MachineData.MachinePosition PosMachine3 = new MachineData.MachinePosition (3, 6, 7);
MachineData.MachinePosition PosMachine4 = new MachineData.MachinePosition (4, 12, 4);
```

```

MachineData.MachinePosition PosMachine5 = new MachineData.MachinePosition (5, 21, 5);

private MachineData.MachinePosition[,] GoodMatrix = new
MachineData.MachinePosition[MaxRowQty, MachineQty];
private MachineData.MachinePosition[,] WarningMatrix = new
MachineData.MachinePosition[MaxRowQty, MachineQty];
private MachineData.MachinePosition[,] ErrorMatrix = new
MachineData.MachinePosition[MaxRowQty, MachineQty];

private bool CreateGoodMatrix = false;
private bool CreateWarningMatrix = false;
private bool CreateErrorMatrix = false;

//Distance variables
private double TotalDistance = 0;
private double ShortestDistance = 100000000;
private MachineData.MachinePosition[] ShortestPath = new
MachineData.MachinePosition[MachineQty];

//Counter variables
private int w;

//PROGRAM

//Back button function
public void BackButton ()
{
    SceneManager.LoadScene ("MaintenanceMenu");
}

//Initialize
void Start ()
{
    CurrentMachineNumber = 0;
    NextMachineNumber = 0;
    PriorityMachine = false;

    //Set visible the machines
    Machine1.SetActive (true);
    Machine2.SetActive (true);
    Machine3.SetActive (true);
    Machine4.SetActive (true);
    Machine5.SetActive (true);

    //Take the names of the machines from the database and writes them
    Machine1Name.text = MachineData.Instance().DataMachine1.MachineType + " " +
MachineData.Instance().MachineState[0,2];
    Machine2Name.text = MachineData.Instance().DataMachine2.MachineType + " " +
MachineData.Instance().MachineState[1,2];
    Machine3Name.text = MachineData.Instance().DataMachine3.MachineType + " " +
MachineData.Instance().MachineState[3,2];
    Machine4Name.text = MachineData.Instance ().DataMachine4.MachineType + " " +
MachineData.Instance().MachineState[2,2];
    Machine5Name.text = MachineData.Instance().DataMachine5.MachineType + " " +
MachineData.Instance().MachineState[4,2];

    //Arrow Start
    lineRenderer = GetComponent<LineRenderer> ();

```

}

```
//Update is called once per frame (the main function)
void Update ()
{
    int i = 0;
    int j = 0;
    int k = 0;
    //Bulbs state
    if (materials.Length != 0) {
        //Machine 1
        if ((MachineData.Instance().MachineState[0,0]==1) &&
(MachineData.Instance().MachineState[0,1] == 2)) { //When the machine is in correct production
state sets the bulb to green
            BulbMachine1.sharedMaterial = materials [0]; //This sets the material color
            GoodArray [i] = PosMachine1;
            i++;
        }
        if ((MachineData.Instance().MachineState[0,0]==1) &&
(MachineData.Instance().MachineState[0,1] == 0)) { //When the machine is in wrong production
state sets the bulb to red
            BulbMachine1.sharedMaterial = materials [1];
            ShortestPath [0] = PosMachine1;
            PriorityMachine = true;
        }
        if ((MachineData.Instance().MachineState[0,0]==1) &&
(MachineData.Instance().MachineState[0,1] == 1)) { //When the machine is in warning production
state sets the bulb to yellow
            BulbMachine1.sharedMaterial = materials [2];
            WarningArray [j] = PosMachine1;
            j++;
        }
        //Machine 2
        if ((MachineData.Instance().MachineState[1,0]==2) &&
(MachineData.Instance().MachineState[1,1]==2)) { //When the machine is in correct production
state sets the bulb to green
            BulbMachine2.sharedMaterial = materials [0];
            GoodArray [i] = PosMachine2;
            i++;
        }
        if ((MachineData.Instance().MachineState[1,0]==2) &&
(MachineData.Instance().MachineState[1,1]==0)) { //When the machine is in wrong production
state sets the bulb to red
            BulbMachine2.sharedMaterial = materials [1];
            ErrorArray [k] = PosMachine2;
            k++;
        }
        if ((MachineData.Instance().MachineState[1,0]==2) &&
(MachineData.Instance().MachineState[1,1]==1)) { //When the machine is in warning production
state sets the bulb to yellow
            BulbMachine2.sharedMaterial = materials [2];
            WarningArray [j] = PosMachine2;
            j++;
        }
        //Machine 3
        if ((MachineData.Instance().MachineState[2,0]==4) &&
(MachineData.Instance().MachineState[2,1]==2)) { //When the machine is in correct production
state sets the bulb to green
            BulbMachine3.sharedMaterial = materials [0];
            GoodArray [i] = PosMachine3;
            i++;
        }
    }
}
```

```

    }
    if ((MachineData.Instance().MachineState[2,0]==4) &&
(MachineData.Instance().MachineState[2,1]==0)) { //When the machine is in wrong production
state sets the bulb to red
        BulbMachine3.sharedMaterial = materials [1];
        ErrorArray [k] = PosMachine3;
        k++;
    }
    if ((MachineData.Instance().MachineState[2,0]==4) &&
(MachineData.Instance().MachineState[2,1]==1)) { //When the machine is in warning production
state sets the bulb to yellow
        BulbMachine3.sharedMaterial = materials [2];
        WarningArray [j] = PosMachine3;
        j++;
    }
    //Machine 4
    if ((MachineData.Instance().MachineState[3,0]==3) &&
(MachineData.Instance().MachineState[3,1]==2)) { //When the machine is in correct production
state sets the bulb to green
        BulbMachine4.sharedMaterial = materials [0];
        GoodArray [i] = PosMachine4;
        i++;
    }
    if ((MachineData.Instance().MachineState[3,0]==3) &&
(MachineData.Instance().MachineState[3,1]==0)) { //When the machine is in wrong production
state sets the bulb to red
        BulbMachine4.sharedMaterial = materials [1];
        ErrorArray [k] = PosMachine4;
        k++;
    }
    if ((MachineData.Instance().MachineState[3,0]==3) &&
(MachineData.Instance().MachineState[3,1]==1)) { //When the machine is in warning production
state sets the bulb to yellow
        BulbMachine4.sharedMaterial = materials [2];
        WarningArray [j] = PosMachine4;
        j++;
    }
    //Machine 5
    if ((MachineData.Instance().MachineState[4,0]==5) &&
(MachineData.Instance().MachineState[4,1]==2)) { //When the machine is in correct production
state sets the bulb to green
        BulbMachine5.sharedMaterial = materials [0];
        GoodArray [i] = PosMachine5;
        i++;
    }
    if ((MachineData.Instance().MachineState[4,0]==5) &&
(MachineData.Instance().MachineState[4,1]==0)) { //When the machine is in wrong production
state sets the bulb to red
        BulbMachine5.sharedMaterial = materials [1];
        ErrorArray [k] = PosMachine5;
        k++;
    }
    if ((MachineData.Instance().MachineState[4,0]==5) &&
(MachineData.Instance().MachineState[4,1]==1)) { //When the machine is in warning production
state sets the bulb to yellow
        BulbMachine5.sharedMaterial = materials [2];
        WarningArray [j] = PosMachine5;
        j++;
    }
}

// Total quantity of good bad and warning

```

```

GoodQty = i;
WarningQty = j;
ErrorQty = k;
if (MachineData.Instance ().palliative == true) {
    permute (ErrorArray, 0, GoodQty - 1);
    permute (WarningArray, 0, GoodQty - 1);
    permute (GoodArray, 0, GoodQty - 1);
    CalculatePath ();
}
if (MachineData.Instance ().preventive == true) {
    for (int u = 0; u < WarningQty; u++) {
        GoodArray [GoodQty + u] = WarningArray [u];
    }
    GoodQty = GoodQty + WarningQty;
    WarningQty = 0;
    permute (ErrorArray, 0, GoodQty - 1);
    permute (GoodArray, 0, GoodQty - 1);
    CalculatePath ();
}

if (MachineData.Instance ().InitiateNow == true) {
    origin = InitialPos;
    MachineData.Instance ().InitiateNow = false;
    Debug.LogWarning (MachineData.Instance ().InitiateNow + "balioa du initiatenowk");
}
if ((origin == InitialPos)&&(MachineData.Instance ().MachineFinished ==
false)&&(MachineData.Instance().s==0)) {
    Debug.LogWarning (MachineData.Instance().s);
    Debug.LogWarning (ShortestPath [MachineData.Instance().s].MachineNumber);
    if (ShortestPath[MachineData.Instance().s].MachineNumber == PosMachine1.MachineNumber)
    {
        NextMachineNumber = 1;
    }
    if (ShortestPath[MachineData.Instance().s].MachineNumber == PosMachine2.MachineNumber)
    {
        NextMachineNumber = 2;
    }
    if (ShortestPath [MachineData.Instance().s].MachineNumber == PosMachine3.MachineNumber)
    {
        NextMachineNumber = 3;
    }
    if (ShortestPath [MachineData.Instance().s].MachineNumber == PosMachine4.MachineNumber)
    {
        NextMachineNumber = 4;
    }
    if (ShortestPath [MachineData.Instance().s].MachineNumber == PosMachine5.MachineNumber)
    {
        NextMachineNumber = 5;
    }
    MachineData.Instance().s = 1;
    MachineData.Instance ().MachineFinished = false;
}

if (MachineData.Instance ().MachineFinished==true) {
    Debug.LogWarning (MachineData.Instance ().MachineFinished);

    if (ShortestPath[MachineData.Instance().s-1].MachineNumber ==
PosMachine1.MachineNumber) {
        CurrentMachineNumber = 1;
    }
    if (ShortestPath [MachineData.Instance().s-1].MachineNumber ==
PosMachine2.MachineNumber) {

```

```

        CurrentMachineNumber = 2;
    }
    if (ShortestPath [MachineData.Instance().s-1].MachineNumber ==
PosMachine3.MachineNumber) {
        CurrentMachineNumber = 3;
    }
    if (ShortestPath [MachineData.Instance().s-1].MachineNumber ==
PosMachine4.MachineNumber) {
        CurrentMachineNumber = 4;
    }
    if (ShortestPath [MachineData.Instance().s-1].MachineNumber ==
PosMachine5.MachineNumber) {
        CurrentMachineNumber = 5;
    }
    if (ShortestPath [MachineData.Instance().s].MachineNumber == PosMachine1.MachineNumber)
{
        NextMachineNumber = 1;
    }
    if (ShortestPath [MachineData.Instance().s].MachineNumber == PosMachine2.MachineNumber)
{
        NextMachineNumber = 2;
    }
    if (ShortestPath [MachineData.Instance().s].MachineNumber == PosMachine3.MachineNumber)
{
        NextMachineNumber = 3;
    }
    if (ShortestPath [MachineData.Instance().s].MachineNumber == PosMachine4.MachineNumber)
{
        NextMachineNumber = 4;
    }
    if (ShortestPath [MachineData.Instance().s].MachineNumber == PosMachine5.MachineNumber)
{
        NextMachineNumber = 5;
    }

        MachineData.Instance().s = MachineData.Instance().s+1;
    MachineData.Instance ().MachineFinished = false;
}

//Set the origin and destination
if ((NextMachineNumber == 1) && (destination != Machine1data)) {
    counter = 0;
    destination = Machine1data;
    GoToMachineName.text = Machine1Name.text;
}
if ((NextMachineNumber == 2) && (destination != Machine2data)) {
    counter = 0;
    destination = Machine2data;
    GoToMachineName.text = Machine2Name.text;
}
if ((NextMachineNumber == 3) && (destination != Machine3data)) {
    counter = 0;
    destination = Machine3data;
    GoToMachineName.text = Machine3Name.text;
}
if ((NextMachineNumber == 4) && (destination != Machine4data)) {
    counter = 0;
    destination = Machine4data;
    GoToMachineName.text = Machine4Name.text;
}
if ((NextMachineNumber == 5) && (destination != Machine5data)) {
    counter = 0;

```

```

destination = Machine5data;
GoToMachineName.text = Machine5Name.text;
}
//Current machines
if (CurrentMachineNumber == 1) {
    origin = Machine1data;
}
if (CurrentMachineNumber == 2) {
    origin = Machine2data;
}
if (CurrentMachineNumber == 3) {
    origin = Machine3data;
}
if (CurrentMachineNumber == 4) {
    origin = Machine4data;
}
if (CurrentMachineNumber == 5) {
    origin = Machine5data;
}
// Draw the row between machines (origin and destination)
if ((0 < NextMachineNumber)) {
    lineRenderer.SetPosition (0, origin.position);
    dist = Vector3.Distance (origin.position, destination.position);
    if (counter < dist)
    {
        counter += 1f / lineDrawSpeed;
        float x = Mathf.Lerp(0, dist, counter);
        Vector3 pointAlongLine = x * Vector3.Normalize(destination.position -
origin.position) + origin.position;
        lineRenderer.SetPosition(1, pointAlongLine);
    }
}
}

//*****Swap function*****//
void swap (MachineData.MachinePosition[] array1, int j, int i)
{
    MachineData.MachinePosition temp = array1 [j];
    array1 [j] = array1 [i];
    array1 [i] = temp;
}

//*****PERMUTATION *****//
/* Function to print permutations of string
This function takes three parameters:
1. String
2. Starting index of the string
3. Ending index of the string. */

void permute (MachineData.MachinePosition[] a, int l, int r)
{
    int i;
    int j;

    if (l == r) {

        if (CreateGoodMatrix == true) {
            for (j = 0; j <= r; j++) {
                GoodMatrix [w, j] = a [j];
            }
        }
    }
}

```

```

    }

    if (CreateWarningMatrix == true) {
        for (j = 0; j <= r; j++) {
            WarningMatrix [w, j] = a [j];
        }
    }

    if (CreateErrorMatrix == true) {
        for (j = 0; j <= r; j++) {
            ErrorMatrix[w, j] = a[j];
        }
    }
} else {
    for (i = 1; i <= r; i++) {
        swap (a, l, i);
        permute (a, l + 1, r);
        swap (a, l, i); //backtrack
    }
}
}

/***** DISTANCE CALCULATOR *****/
double GetDistance (double x1, double y1, double x2, double y2)
{
    double distance = Math.Sqrt (Math.Pow ((x2 - x1), 2) + Math.Pow ((y2 - y1), 2));
    return distance;
}

/***** FACTORIAL *****/
// The factorial is calculated to know how many combinations there are for each array
public int factorial (int num)
{
    int result;

    if (num == 0) {
        result = 1;
    } else {
        result = num;
        for (int i = num - 1; i >= 1; i--) {
            result = result * i;
        }
    }

    return result;
}

/***** CALCULATE PATH *****/

void CalculatePath ()
{
    double GoodDist = 0;
    double WarningDist = 0;
    double ErrorDist = 0;
    double ErrorWarningDist = 0;
    double WarningGoodDist = 0;
    double ErrorGoodDist = 0;
    double StartErrorDist = 0;
    double StartPriorDist = 0;
    double PriorErrorDist = 0;
    double PriorWarningDist = 0;
    double PriorGoodDist = 0;
    double StartWarningDist = 0;
    double StartGoodDist = 0;

```

```

int GoodPermQty = factorial (GoodQty);
int WarningPermQty = factorial (WarningQty);
int ErrorPermQty = factorial (ErrorQty);

for (int i = 0; i <= ErrorPermQty - 1; i++) {
if ((ErrorQty == 1) || (ErrorQty == 0)) {
    ErrorDist = 0;
} else {
    for (int j = 0; j <= ErrorQty - 1; j++) {
        ErrorDist = GetDistance (ErrorMatrix [i, j].x, ErrorMatrix [i, j].y, ErrorMatrix
[i, j + 1].x, ErrorMatrix [i, j + 1].y);
    }
}

    for (int k = 0; k <= WarningPermQty - 1; k++) {
        if ((WarningQty == 1) || (WarningQty == 0)) {
            WarningDist = 0;
        } else {
            for (int l = 0; l <= WarningQty - 1; l++) {
                WarningDist = GetDistance (WarningMatrix [k, l].x, WarningMatrix [k, l].y,
WarningMatrix [k, l + 1].x, WarningMatrix [k, l + 1].y);
            }
        }

        for (int m = 0; m <= GoodPermQty - 1; m++) {
            if ((GoodQty == 1) || (GoodQty == 0)) {
                GoodDist = 0;
            } else {
                for (int n = 0; n <= GoodQty - 1; n++) {
                    GoodDist = GetDistance(GoodMatrix[m, n].x, GoodMatrix[m, n].y,
GoodMatrix[m, n + 1].x, GoodMatrix[m, n + 1].y);
                }
            }

            if (ErrorQty != 0 && WarningQty != 0) {
                ErrorWarningDist = GetDistance(WarningMatrix[k, 0].x, WarningMatrix[k, 0].y,
ErrorMatrix[i, ErrorQty - 1].x, ErrorMatrix[i, ErrorQty - 1].y);
            } else {
                ErrorWarningDist = 0;
            }

            if (WarningQty != 0 && GoodQty != 0) {
                WarningGoodDist = GetDistance(GoodMatrix[m, 0].x, GoodMatrix[m, 0].y,
WarningMatrix[k, WarningQty - 1].x, WarningMatrix[k, WarningQty - 1].y);
            } else {
                WarningGoodDist = 0;
            }

            if (WarningQty == 0 && (ErrorQty != 0 && GoodQty != 0)) {
                ErrorGoodDist = GetDistance(GoodMatrix[m, 0].x, GoodMatrix[m, 0].y,
ErrorMatrix[i, ErrorQty - 1].x, ErrorMatrix[i, ErrorQty - 1].y);
            } else {
                ErrorGoodDist = 0;
            }

            if (PriorityMachine != false) {
                StartErrorDist = 0;
                StartWarningDist = 0;
                StartGoodDist = 0;
                StartPriorDist = GetDistance(StartPosition.x, StartPosition.y,
ShortestPath[0].x, ShortestPath[0].y);
                if (ErrorQty != 0)

```

```

        {
            PriorGoodDist = 0;
            PriorWarningDist = 0;
            PriorErrorDist = GetDistance(ShortestPath[0].x, ShortestPath[0].y,
ErrorMatrix[i, 0].x, ErrorMatrix[i, 0].y);
        }
        else if (WarningQty != 0)
        {
            PriorGoodDist = 0;
            PriorErrorDist = 0;
            PriorWarningDist = GetDistance(ShortestPath[0].x, ShortestPath[0].y,
WarningMatrix[k, 0].x, WarningMatrix[k, 0].y);
        }
        else if (GoodQty != 0)
        {
            PriorWarningDist = 0;
            PriorErrorDist = 0;
            PriorGoodDist = GetDistance(ShortestPath[0].x, ShortestPath[0].y,
GoodMatrix[m, 0].x, GoodMatrix[m, 0].y);
        }
        else if (ErrorQty != 0) {
            StartErrorDist = GetDistance (StartPosition.x, StartPosition.y, ErrorMatrix [i,
0].x, ErrorMatrix [i, 0].y);
            StartWarningDist = 0;
            StartGoodDist = 0;
        } else if (WarningQty != 0) {
            StartWarningDist = GetDistance (StartPosition.x, StartPosition.y, WarningMatrix [k,
0].x, WarningMatrix [k, 0].y);
            StartErrorDist = 0;
            StartGoodDist = 0;
        }
        else {
            StartGoodDist = GetDistance (StartPosition.x, StartPosition.y, GoodMatrix [m, 0].x,
GoodMatrix [m, 0].y);
            StartErrorDist = 0;
            StartWarningDist = 0;
        }

        TotalDistance = ErrorDist + WarningDist + GoodDist + ErrorWarningDist + WarningGoodDist +
ErrorGoodDist + StartGoodDist + StartWarningDist + StartErrorDist + StartPriorDist +
PriorGoodDist + PriorWarningDist + PriorErrorDist;

        if (TotalDistance < ShortestDistance) {
            ShortestDistance = TotalDistance;
            if ((ShortestPath[0].x == PosMachine1.x) && (ShortestPath[0].y ==
PosMachine1.y))
            {
                for (int o = 1; o <= ErrorQty - 1; o++)
                {
                    ShortestPath[o] = ErrorArray[o];
                }
            }
            else
            {
                for (int o = 0; o <= ErrorQty - 1; o++)
                {
                    ShortestPath[o] = ErrorArray[o];
                }
            }
        }
    }

```

```

/*=====
Copyright (c) 2010-2014 Qualcomm Connected Experiences, Inc.
All Rights Reserved.
Confidential and Proprietary - Protected under copyright and other laws.
Modified by: Leire Amenabar and Leire Carreras
=====*/

using UnityEngine;
using UnityEngine.SceneManagement;

namespace Vuforia
{
    public class Machine1QrRead : MonoBehaviour,
    ITrackableEventHandler
    {
        #region PRIVATE_MEMBER_VARIABLES
        private TrackableBehaviour mTrackableBehaviour;
        #endregion // PRIVATE_MEMBER_VARIABLES

        #region UNITY_MONOBEHAVIOUR_METHODS

        void Start()
        {
            mTrackableBehaviour = GetComponent<TrackableBehaviour>();
            if (mTrackableBehaviour)
            {
                mTrackableBehaviour.RegisterTrackableEventHandler(this);
            }
        }
        #endregion // UNITY_MONOBEHAVIOUR_METHODS
        #region PUBLIC_METHODS

        /// <summary>
        /// Implementation of the ITrackableEventHandler function called when the
        /// tracking state changes.
        /// </summary>
        public void OnTrackableStateChanged(
            TrackableBehaviour.Status previousStatus,
            TrackableBehaviour.Status newStatus)
        {
            if (newStatus == TrackableBehaviour.Status.DETECTED ||
                newStatus == TrackableBehaviour.Status.TRACKED ||

```

```

        newStatus == TrackableBehaviour.Status.EXTENDED_TRACKED)
    {
        OnTrackingFound();
    }
    else
    {
        OnTrackingLost();
    }
}
#endregion // PUBLIC_METHODS
#region PRIVATE_METHODS
private void OnTrackingFound()
{
    //Information of machine 1 is given to the variable
    Debug.Log("Trackable " + mTrackableBehaviour.TrackableName + " found");
    MachineData.Instance().MachineDataVariables[0] =
MachineData.Instance().DataMachine1.WaterTemp;
    MachineData.Instance().MachineDataVariables[1] =
MachineData.Instance().DataMachine1.Pressure;
    MachineData.Instance().MachineDataVariables[2] =
MachineData.Instance().DataMachine1.WaterLevel;
    MachineData.Instance().MachineDataVariables[3] =
MachineData.Instance().DataMachine1.HousingTemp;
    MachineData.Instance().MachineDataVariables[4] = MachineData.Instance().DataMachine1.Flow;
    MachineData.Instance().MachineName=MachineData.Instance().DataMachine1.MachineType;
    MachineData.Instance().CurrentMachineNum=MachineData.Instance().DataMachine1.MachineNum;
    MachineData.Instance().LastDispMachiNum=MachineData.Instance().MachineState[0,2];
    MachineData.Instance().MachineID = MachineData.Instance().DataMachine1.IDNum;
    MachineData.Instance().MachinePartNumber = MachineData.Instance().DataMachine1.PartNum;
    SceneManager.LoadScene ("MachinesMenu");
}

private void OnTrackingLost()
{
    Renderer[] renderComponents = GetComponentsInChildren<Renderer>(true);
    Collider[] colliderComponents = GetComponentsInChildren<Collider>(true);
    // Disable rendering:
    foreach (Renderer component in renderComponents)
    {
        component.enabled = false;
    }
    // Disable colliders:
    foreach (Collider component in colliderComponents)
    {
        component.enabled = false;
    }
    Debug.Log("Trackable " + mTrackableBehaviour.TrackableName + " lost");
}

#endregion // PRIVATE_METHODS
}
}

```

*For all the other machines is the same code but changes OnTrackingFound() function. Example, machine 2:

```

private void OnTrackingFound()
{
    //Information of machine 2 is given to the variable
    Debug.Log("Trackable " + mTrackableBehaviour.TrackableName + " found");
}

```

```
MachineData.Instance().MachineDataVariables[0] =
MachineData.Instance().DataMachine2.WaterTemp;
MachineData.Instance().MachineDataVariables[1] =
MachineData.Instance().DataMachine2.Pressure;
MachineData.Instance().MachineDataVariables[2] =
MachineData.Instance().DataMachine2.WaterLevel;
MachineData.Instance().MachineDataVariables[3] =
MachineData.Instance().DataMachine2.HousingTemp;
MachineData.Instance().MachineDataVariables[4] = MachineData.Instance().DataMachine2.Flow;
MachineData.Instance().LastDispMachiNum=MachineData.Instance().MachineState[1,2];
MachineData.Instance().MachineName=MachineData.Instance().DataMachine2.MachineType;
MachineData.Instance().CurrentMachineNum=MachineData.Instance().DataMachine2.MachineNum;
MachineData.Instance().MachineID = MachineData.Instance().DataMachine2.IDNum;
MachineData.Instance().MachinePartNumber = MachineData.Instance().DataMachine2.PartNum;
SceneManager.LoadScene ("MachinesMenu");
}
```

8. Machines menu script

```
/*=====
Programmers: Leire Amenabar and Leire Carreras
Date: May 09, 2018
Revision: 1.0
Copyright 2018
Description: In this script selected machines menu is displayed. Besides, the values are
compared so that errors are selected.
All Rights Reserved.
Confidential and Proprietary - Protected under copyright and other laws.
=====*/

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;
using UnityEngine.UI;

public class MachinesMenuProg : MonoBehaviour {

    public GameObject WaterPumpObject;
    public GameObject ElectricCabinetObject;

    public void Back(){
        SceneManager.LoadScene ("MaintenanceMenu");
    }

    public void DisplayInformation(){
        SceneManager.LoadScene ("DisplayInformation");
    }

    public void StartMaintenance(){
        SceneManager.LoadScene ("TextStepsDisplay");
    }

    void start () {
        MachineData.Instance().WaterTemperatureIssue = false;
        MachineData.Instance().PressureIssue = false;
        MachineData.Instance().FixFilters = false;
        MachineData.Instance().WaterTemperatureIssue = false;
        MachineData.Instance().FixFilters = false;
        MachineData.Instance().VoltageIssue = false;
        MachineData.Instance().CurrentIssue = false;
    }
}
```

```

WaterPumpObject.SetActive (false);
ElectricCabinetObject.SetActive (false);
}

void Update () {
    if (MachineData.Instance().MachineName == "Water pump") {
        WaterPumpObject.SetActive (true);
        ElectricCabinetObject.SetActive (false);
        //Water Temperature issue
        if ((MachineData.Instance().MachineDataVariables[0] >
MachineData.Instance().MaxWaterTemp) || (MachineData.Instance().MachineDataVariables[0] <
MachineData.Instance().MinWaterTemp)) {
            MachineData.Instance().WaterTemperatureIssue = true;
            MachineData.Instance().WaterTemperatureError = true;
        } else {
            MachineData.Instance().WaterTemperatureIssue = false;
            MachineData.Instance().WaterTemperatureError = false;
        }
        //Pressure Issue
        if ((MachineData.Instance().MachineDataVariables[1] >
MachineData.Instance().MaxPressure) || (MachineData.Instance().MachineDataVariables[1] <
MachineData.Instance().MinPressure)) {
            MachineData.Instance().PressureIssue = true;
            MachineData.Instance().PressureError = true;
        } else {
            MachineData.Instance().PressureIssue = false;
            MachineData.Instance().PressureError = false;
        }
        //Water level
        if ((MachineData.Instance().MachineDataVariables[2] >
MachineData.Instance().MaxWaterLevel) || (MachineData.Instance().MachineDataVariables[2] <
MachineData.Instance().MinWaterLevel)) {
            MachineData.Instance().FixFilters = true;
            MachineData.Instance().LevelError = true;
        } else {
            MachineData.Instance().FixFilters = false;
            MachineData.Instance().LevelError = false;
        }
        //Housing temperature
        if ((MachineData.Instance().MachineDataVariables[3] >
MachineData.Instance().MaxHousingTemp) || (MachineData.Instance().MachineDataVariables[3] <
MachineData.Instance().MinHousingTemp)) {
            MachineData.Instance().WaterTemperatureIssue = true;
            MachineData.Instance().HousingTempError = true;
        } else {
            MachineData.Instance().WaterTemperatureIssue = false;
            MachineData.Instance().HousingTempError = false;
        }
        //Flow
        if ((MachineData.Instance().MachineDataVariables[4] > MachineData.Instance().MaxFlow) ||
(MachineData.Instance().MachineDataVariables[4] < MachineData.Instance().MinFlow)) {
            MachineData.Instance().FixFilters = true;
            MachineData.Instance().FlowError = true;
        } else {
            MachineData.Instance().FixFilters = false;
            MachineData.Instance().FlowError = false;
        }
    }
}

if (MachineData.Instance().MachineName == "Electric cabinet") {
    WaterPumpObject.SetActive (false);
}

```

```
ElectricCabinetObject.SetActive (true);
//Voltage issue
if ((MachineData.Instance().MachineDataVariables[0] >
MachineData.Instance().MaxVoltage)|| (MachineData.Instance().MachineDataVariables[0] <
MachineData.Instance().MinVoltage)) {
    MachineData.Instance().VoltageIssue = true;
} else {
    MachineData.Instance().VoltageIssue = false;
}
//Current Issue
if ((MachineData.Instance().MachineDataVariables[1] >
MachineData.Instance().MaxCurrent)|| (MachineData.Instance().MachineDataVariables[1] <
MachineData.Instance().MinCurrent)) {
    MachineData.Instance().CurrentIssue = true;
} else {
    MachineData.Instance().CurrentIssue = false;
}
}
}
}
```

9. Display information script

```
/*=====
Programmers: Leire Amenabar and Leire Carreras
Date: May 09, 2018
Revision: 1.0
Copyright 2018
Description: In this script all the information of the selected machine is displayed.
Depending the value is in correct or error state, a
            red or green square will appear next to the value.
All Rights Reserved.
Confidential and Proprietary - Protected under copyright and other laws.
=====*/
//Libraries
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using System;
using System.IO;
using System.Text.RegularExpressions;
using UnityEngine.SceneManagement;

public class DisplayInformationProg : MonoBehaviour {

//Public variables
//Text
public Text InfoName1;
public Text InfoName2;
public Text InfoName3;
public Text InfoName4;
public Text InfoName5;
public Text DataInfo1;
public Text DataInfo2;
public Text DataInfo3;
public Text DataInfo4;
public Text DataInfo5;
public Text Unit1;
public Text Unit2;
public Text Unit3;
```

```
public Text Unit4;
public Text Unit5;

//Lights
public GameObject RedLight1;
public GameObject GreenLight1;
public GameObject RedLight2;
public GameObject GreenLight2;
public GameObject RedLight3;
public GameObject GreenLight3;
public GameObject RedLight4;
public GameObject GreenLight4;
public GameObject RedLight5;
public GameObject GreenLight5;

//Back function
public void Back(){
SceneManager.LoadScene ("MachinesMenu");
}

// Use this for initialization
void Start () {
//Diactivate all gameobjects
RedLight1.SetActive (false);
GreenLight1.SetActive (false);
RedLight2.SetActive (false);
GreenLight2.SetActive (false);
RedLight3.SetActive (false);
GreenLight3.SetActive (false);
RedLight4.SetActive (false);
GreenLight4.SetActive (false);
RedLight5.SetActive (false);
GreenLight5.SetActive (false);
}

// Update is called once per frame
void Update () {

if (MachineData.Instance().MachineName == "Water pump") {
//Name
InfoName1.text = "Water Temp.";
InfoName2.text = "Pressure";
InfoName3.text = "Level";
InfoName4.text = "House Temp.";
InfoName5.text = "Flow";
//Units
Unit1.text = "°C";
Unit2.text = "kPa";
Unit3.text = "l";
Unit4.text = "°C";
Unit5.text = "l/m";
//Take the values
DataInfo1.text = MachineData.Instance().MachineDataVariables[0].ToString();
DataInfo2.text = MachineData.Instance().MachineDataVariables[1].ToString();
DataInfo3.text = MachineData.Instance().MachineDataVariables[2].ToString();
DataInfo4.text = MachineData.Instance().MachineDataVariables[3].ToString();
DataInfo5.text = MachineData.Instance().MachineDataVariables[4].ToString();

//Info 1 lights
if (MachineData.Instance().WaterTemperatureError == true) {
RedLight1.SetActive (true);
}
```

```

    GreenLight1.SetActive (false);
} if(MachineData.Instance().WaterTemperatureError == false) {
    RedLight1.SetActive (false);
    GreenLight1.SetActive (true);
}
//Info 2 lights
if (MachineData.Instance().PressureError == true) {
    RedLight2.SetActive (true);
    GreenLight2.SetActive (false);
} if(MachineData.Instance().PressureError == false) {
    RedLight2.SetActive (false);
    GreenLight2.SetActive (true);
}
//Info 3 lights
if (MachineData.Instance().LevelError== true) {
    RedLight3.SetActive (true);
    GreenLight3.SetActive (false);
} if (MachineData.Instance().LevelError== false) {
    RedLight3.SetActive (false);
    GreenLight3.SetActive (true);
}
//Info 4 lights
if (MachineData.Instance().HousingTempError == true) {
    RedLight4.SetActive (true);
    GreenLight4.SetActive (false);
} if (MachineData.Instance().HousingTempError == false) {
    RedLight4.SetActive (false);
    GreenLight4.SetActive (true);
}
//Info 5 lights
if (MachineData.Instance().FlowError == true) {
    RedLight5.SetActive (true);
    GreenLight5.SetActive (false);
} if(MachineData.Instance().FlowError == false) {
    RedLight5.SetActive (false);
    GreenLight5.SetActive (true);
}
}

if (MachineData.Instance().MachineName == "Electric cabinet") {
//Names
    InfoName1.text = "Voltage";
    InfoName2.text = "Intensity";
    InfoName3.text = " ";
    InfoName4.text = " ";
    InfoName5.text = " ";
//Units
    Unit1.text = "V";
    Unit2.text = "A";
    Unit3.text = " ";
    Unit4.text = " ";
    Unit5.text = " ";
    DataInfo1.text = MachineData.Instance().MachineDataVariables[0].ToString();
    DataInfo2.text = MachineData.Instance().MachineDataVariables[1].ToString();
    DataInfo3.text = " ";
    DataInfo4.text = " ";
    DataInfo5.text = " ";
//Info 1 lights
    if (MachineData.Instance().VoltageIssue == true) {
        RedLight1.SetActive (true);
        GreenLight1.SetActive (false);
    } if(MachineData.Instance().VoltageIssue == false) {

```

```

    RedLight1.SetActive (false);
    GreenLight1.SetActive (true);
}
//Info 2 lights
if (MachineData.Instance().CurrentIssue == true) {
    RedLight2.SetActive (true);
    GreenLight2.SetActive (false);
} if (MachineData.Instance().CurrentIssue == false) {
    RedLight2.SetActive (false);
    GreenLight2.SetActive (true);
}
RedLight3.SetActive (false);
GreenLight3.SetActive (false);
RedLight4.SetActive (false);
GreenLight4.SetActive (false);
RedLight5.SetActive (false);
GreenLight5.SetActive (false);
}
}
}

```

10. Steps display

```

/*=====
Programmers: Leire Amenabar and Leire Carreras
Date: May 09, 2018
Revision: 1.0
Copyright 2018
Description: This script displays the steps which has to be followed by the user with 3d
objects.
Confidential and Proprietary - Protected under copyright and other laws.
=====*/
//Libraries
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using System;
using System.IO;
using System.Text.RegularExpressions;
using UnityEngine.SceneManagement;

public class StepsDisplayProg : MonoBehaviour {

    //Public Variables
    public Text Title;
    //Steps Number text
    public Text StepNumber;
    //Steps text
    public Text Step;
    //GameObjects->3Dobjects
    public GameObject Screwdriver;
    public GameObject CircuitBreaker;
    public GameObject FbxWrench;
    public GameObject Bearing;
    public GameObject Spray;
    public GameObject Filter;
    public GameObject WaterPumpObject;
    public GameObject ElectricCabinetObject;
    //GameObjects->2Dobjects
    public GameObject ElectricalProtection;

```

```

public GameObject Terminals;
public GameObject Pilots;
public GameObject MotorTerminals;
public GameObject Cabinet;
public GameObject GroundTerminal;
public GameObject CleanIndustry;
public GameObject ScreenPic;
public GameObject LubricateBearing;
    //Number of machines
public int FileNum=5;
double[,] Results = new double[5, 50];

//WaterPump variables
public Material[] materials;//Allows input of material colors in a set size of array;
public MeshRenderer Base; //What are we rendering? Input object(Sphere,Cylinder,...) to
render.
public MeshRenderer Level;
public MeshRenderer ElectricalCabinet;
public MeshRenderer Buttons;
public MeshRenderer PumpFilter;

//Private Variables
private int LineCounter=0;
private int FileCounter=0;

private Text title;
string PalliativePath = @"/storage/emulated/0/Maintenance/DocumentationSteps/Corrective/";
string PreventivePath = @"/storage/emulated/0/Maintenance/DocumentationSteps/Preventive/";
private String[] Lines;
private int LineQty;
private int FilesQty;
string [] fileEntries;
private bool AllFiles;

//Back button action
public void Back(){
    SceneManager.LoadScene ("MachinesMenu");
}
//Done button action
public void Done(){
    SceneManager.LoadScene ("CameraDisplay");
}

//Next and previous buttons action
public void next(){
if (AllFiles == true) {
    if (FileCounter <= FilesQty - 1) {
        LineCounter++;
        if (LineCounter <= LineQty - 1) {
            StepNumber.text = LineCounter + "-";
            Step.text = Lines [LineCounter];
        }
        if ((LineCounter == LineQty) && (FileCounter != FilesQty - 1)) {
            LineCounter = 0;
            FileCounter++;
            Lines = File.ReadAllLines (fileEntries [FileCounter]);
            LineQty = File.ReadAllLines (fileEntries [FileCounter]).Length;
            Title.text = Lines [LineCounter];
            LineCounter = 1;
            StepNumber.text = LineCounter + "-";
            Step.text = Lines [LineCounter];
        }
    }
}

```

```

        if ((LineCounter == LineQty) && (FileCounter == FilesQty - 1)) {
            LineCounter = LineQty - 1;
            StepNumber.text = LineCounter + ".-";
            Step.text = Lines [LineCounter];
        }
    } else {
        FileCounter = FilesQty - 1;
    }
}
}
if(AllFiles == false){
    LineCounter++;
    if (LineCounter <= LineQty - 1) {
        StepNumber.text = LineCounter + ".-";
        Step.text = Lines [LineCounter];
    }

    if (LineCounter == LineQty) {
        if ((MachineData.Instance().PressureIssue == false) && (MachineData.Instance
).WaterTemperatureIssue == false)) {
            fileEntries = Directory.GetFiles (PreventivePath);
            FilesQty = Directory.GetFiles (PreventivePath).Length;
            AllFiles = true;
            LineCounter = 0;
            Lines = File.ReadAllLines (fileEntries [FileCounter]);
            LineQty = File.ReadAllLines (fileEntries [FileCounter]).Length;
            Title.text = Lines [LineCounter];
            LineCounter = 1;
            StepNumber.text = LineCounter + ".-";
            Step.text = Lines [LineCounter];
        } else {
            if (MachineData.Instance().PressureIssue) {
                FileCounter = 1;
                LineCounter = 0;
                Lines = File.ReadAllLines (fileEntries [FileCounter]);
                LineQty = File.ReadAllLines (fileEntries [FileCounter]).Length;
                Title.text = Lines [LineCounter];
                LineCounter = 1;
                StepNumber.text = LineCounter + ".-";
                Step.text = Lines [LineCounter];
            }
            else if (MachineData.Instance().WaterTemperatureIssue) {
                FileCounter = 2;
                LineCounter = 0;
                Lines = File.ReadAllLines (fileEntries [FileCounter]);
                LineQty = File.ReadAllLines (fileEntries [FileCounter]).Length;
                Title.text = Lines [LineCounter];
                LineCounter = 1;
                StepNumber.text = LineCounter + ".-";
                Step.text = Lines [LineCounter];
            }
        }
    }
}
}
}

public void previous(){
    if (AllFiles==true) {
        if (FileCounter >= 0) {
            LineCounter = LineCounter - 1;
            if (LineCounter > 0) {
                StepNumber.text = LineCounter + ".-";
                Step.text = Lines [LineCounter];
            }
        }
    }
}

```

```

    }
    if ((LineCounter == 0) && (FileCounter > 0)) {
        FileCounter = FileCounter - 1;
        Lines = File.ReadAllLines (fileEntries [FileCounter]);
        LineQty = File.ReadAllLines (fileEntries [FileCounter]).Length;
        Title.text = Lines [LineCounter];
        LineCounter = LineQty - 1;
        StepNumber.text = LineCounter + "-";
        Step.text = Lines [LineCounter];
    }
    if ((LineCounter == 0) && (FileCounter == 0)) {
        Title.text = Lines [LineCounter];
        LineCounter = 1;
        StepNumber.text = LineCounter + "-";
        Step.text = Lines [LineCounter];
    }

} else {
    FileCounter = 0;
}
}
if(AllFiles==false){
    LineCounter = LineCounter - 1;
    if (LineCounter > 0) {
        StepNumber.text = LineCounter + "-";
        Step.text = Lines [LineCounter];
    }
    if (LineCounter == 0) {
        Title.text = Lines [LineCounter];
        LineCounter = 1;
        StepNumber.text = LineCounter + "-";
        Step.text = Lines [LineCounter];
    }
}
}
}
//Main program
// Use this for initialization
void Start () {
    //Diactivate all the gameobjects
    Screwdriver.SetActive (false);
    Pilots.SetActive (false);
    ElectricalProtection.SetActive (false);
    Terminals.SetActive (false);
    CircuitBreaker.SetActive (false);
    MotorTerminals.SetActive (false);
    FbxWrench.SetActive (false);
    Cabinet.SetActive (false);
    GroundTerminal.SetActive (false);
    CleanIndustry.SetActive (false);
    ScreenPic.SetActive (false);
    Bearing.SetActive (false);
    Spray.SetActive (false);
    LubricateBearing.SetActive (false);
    Filter.SetActive (false);
    WaterPumpObject.SetActive (false);
    ElectricCabinetObject.SetActive (false);

    // Put the materials to the water Pump
    Base.sharedMaterial = materials [1];
    Level.sharedMaterial = materials [2];
    ElectricalCabinet.sharedMaterial = materials [2];
    Buttons.sharedMaterial = materials [3];

```

```
PumpFilter.sharedMaterial = materials [2];

if (MachineData.Instance ().MachineName == "Water pump") {
    WaterPumpObject.SetActive (true);
    ElectricCabinetObject.SetActive (false);
}
if (MachineData.Instance ().MachineName == "Electrical cabinet") {
    WaterPumpObject.SetActive (false);
    ElectricCabinetObject.SetActive (true);
}
if (MachineData.Instance().preventive) {
    fileEntries = Directory.GetFiles(PreventivePath);
    FilesQty = Directory.GetFiles(PreventivePath).Length;
    AllFiles = true;
}
if (MachineData.Instance().palliative) {
    AllFiles = false;
    fileEntries = Directory.GetFiles (PalliativePath);
    if (MachineData.Instance().FixFilters) {
        FileCounter = 0;
    }
    else if (MachineData.Instance().PressureIssue) {
        FileCounter = 1;
    }
    else if (MachineData.Instance().WaterTemperatureIssue) {
        FileCounter = 2;
    }
} else {
    fileEntries = Directory.GetFiles(PreventivePath);
    FilesQty = Directory.GetFiles(PreventivePath).Length;
    AllFiles = true;
}
Lines = File.ReadAllLines (fileEntries[FileCounter]);
LineQty = File.ReadAllLines(fileEntries[FileCounter]).Length;
Title.text = Lines [LineCounter];
LineCounter = 1;
StepNumber.text = LineCounter + "-";
Step.text = Lines [LineCounter];
}

// Update is called once per frame
void Update () {

//First File Images
if ((Lines [0] == "Electric System")) {
    if (LineCounter == 1) {
        Screwdriver.SetActive (false);
        ElectricalProtection.SetActive (true);
        Terminals.SetActive (false);
        Pilots.SetActive (false);
        CircuitBreaker.SetActive (true);
        MotorTerminals.SetActive (false);
        FbxWrench.SetActive (false);
        Cabinet.SetActive (false);
        GroundTerminal.SetActive (false);
        CleanIndustry.SetActive (false);
        ScreenPic.SetActive (false);
        Bearing.SetActive (false);
        Spray.SetActive (false);
        LubricateBearing.SetActive (false);
        Filter.SetActive (false);
    }
}
```

```

ElectricalCabinet.sharedMaterial = materials [0];
Base.sharedMaterial = materials [1];
Level.sharedMaterial = materials [2];
Buttons.sharedMaterial = materials [3];
PumpFilter.sharedMaterial = materials [2];
}
if (LineCounter == 2) {
    Screwdriver.SetActive (true);
    ElectricalProtection.SetActive (false);
    Terminals.SetActive (true);
    Pilots.SetActive (false);
    CircuitBreaker.SetActive (false);
    MotorTerminals.SetActive (false);
    FbxWrench.SetActive (false);
    Cabinet.SetActive (false);
    GroundTerminal.SetActive (false);
    CleanIndustry.SetActive (false);
    ScreenPic.SetActive (false);
    Bearing.SetActive (false);
    Spray.SetActive (false);
    LubricateBearing.SetActive (false);
    Filter.SetActive (false);
    ElectricalCabinet.sharedMaterial = materials [0];
    Base.sharedMaterial = materials [1];
    Level.sharedMaterial = materials [2];
    Buttons.sharedMaterial = materials [3];
    PumpFilter.sharedMaterial = materials [2];
}
if (LineCounter == 3) {
    Screwdriver.SetActive (false);
    ElectricalProtection.SetActive (false);
    Terminals.SetActive (false);
    Pilots.SetActive (true);
    CircuitBreaker.SetActive (false);
    MotorTerminals.SetActive (false);
    FbxWrench.SetActive (false);
    Cabinet.SetActive (false);
    GroundTerminal.SetActive (false);
    CleanIndustry.SetActive (false);
    ScreenPic.SetActive (false);
    Bearing.SetActive (false);
    Spray.SetActive (false);
    LubricateBearing.SetActive (false);
    Filter.SetActive (false);
    ElectricalCabinet.sharedMaterial = materials [2];
    Base.sharedMaterial = materials [1];
    Level.sharedMaterial = materials [2];
    Buttons.sharedMaterial = materials [0];
    PumpFilter.sharedMaterial = materials [2];
}
if (LineCounter == 4) {
    Screwdriver.SetActive (false);
    ElectricalProtection.SetActive (false);
    Terminals.SetActive (false);
    Pilots.SetActive (false);
    CircuitBreaker.SetActive (false);
    MotorTerminals.SetActive (true);
    FbxWrench.SetActive (true);
    Cabinet.SetActive (false);
    GroundTerminal.SetActive (false);
    CleanIndustry.SetActive (false);
    ScreenPic.SetActive (false);

```

```

    Bearing.SetActive (false);
    Spray.SetActive (false);
    LubricateBearing.SetActive (false);
    Filter.SetActive (false);
    ElectricalCabinet.sharedMaterial = materials [0];
    Base.sharedMaterial = materials [1];
    Level.sharedMaterial = materials [2];
    Buttons.sharedMaterial = materials [3];
    PumpFilter.sharedMaterial = materials [2];
}
if (LineCounter == 5) {
    Screwdriver.SetActive (false);
    ElectricalProtection.SetActive (false);
    Terminals.SetActive (false);
    Pilots.SetActive (false);
    CircuitBreaker.SetActive (false);
    MotorTerminals.SetActive (false);
    FbxWrench.SetActive (false);
    Cabinet.SetActive (true);
    GroundTerminal.SetActive (true);
    CleanIndustry.SetActive (false);
    ScreenPic.SetActive (false);
    Bearing.SetActive (false);
    Spray.SetActive (false);
    LubricateBearing.SetActive (false);
    Filter.SetActive (false);
    ElectricalCabinet.sharedMaterial = materials [0];
    Base.sharedMaterial = materials [1];
    Level.sharedMaterial = materials [2];
    Buttons.sharedMaterial = materials [3];
    PumpFilter.sharedMaterial = materials [2];
}
}
}
/Second file images
if ((Lines [0] == "General")) {
    if (LineCounter == 1) {
        Screwdriver.SetActive (false);
        ElectricalProtection.SetActive (false);
        Terminals.SetActive (false);
        Pilots.SetActive (false);
        CircuitBreaker.SetActive (false);
        MotorTerminals.SetActive (false);
        FbxWrench.SetActive (false);
        Cabinet.SetActive (false);
        GroundTerminal.SetActive (false);
        CleanIndustry.SetActive (false);
        ScreenPic.SetActive (false);
        Bearing.SetActive (false);
        Spray.SetActive (false);
        LubricateBearing.SetActive (false);
        Filter.SetActive (false);
        ElectricalCabinet.sharedMaterial = materials [2];
        Base.sharedMaterial = materials [1];
        Level.sharedMaterial = materials [0];
        Buttons.sharedMaterial = materials [3];
        PumpFilter.sharedMaterial = materials [2];
    }
    if (LineCounter == 2) {
        Screwdriver.SetActive (false);
        ElectricalProtection.SetActive (false);
        Terminals.SetActive (false);
        Pilots.SetActive (false);
    }
}

```

```

CircuitBreaker.SetActive (false);
MotorTerminals.SetActive (false);
FbxWrench.SetActive (false);
Cabinet.SetActive (false);
GroundTerminal.SetActive (false);
CleanIndustry.SetActive (false);
ScreenPic.SetActive (true);
Bearing.SetActive (false);
Spray.SetActive (false);
LubricateBearing.SetActive (false);
Filter.SetActive (false);
ElectricalCabinet.sharedMaterial = materials [0];
Base.sharedMaterial = materials [1];
Level.sharedMaterial = materials [2];
Buttons.sharedMaterial = materials [3];
PumpFilter.sharedMaterial = materials [2];
}
if (LineCounter == 3) {
    Screwdriver.SetActive (false);
    ElectricalProtection.SetActive (false);
    Terminals.SetActive (false);
    Pilots.SetActive (false);
    CircuitBreaker.SetActive (false);
    MotorTerminals.SetActive (false);
    FbxWrench.SetActive (false);
    Cabinet.SetActive (false);
    GroundTerminal.SetActive (false);
    CleanIndustry.SetActive (true);
    ScreenPic.SetActive (false);
    Bearing.SetActive (false);
    Spray.SetActive (false);
    LubricateBearing.SetActive (false);
    Filter.SetActive (false);
    ElectricalCabinet.sharedMaterial = materials [2];
    Base.sharedMaterial = materials [0];
    Level.sharedMaterial = materials [2];
    Buttons.sharedMaterial = materials [3];
    PumpFilter.sharedMaterial = materials [2];
}
}
//Third file images
if ((Lines [0] == "Mechanic System")) {
    if (LineCounter == 1) {
        Screwdriver.SetActive (false);
        ElectricalProtection.SetActive (false);
        Terminals.SetActive (false);
        Pilots.SetActive (false);
        CircuitBreaker.SetActive (false);
        MotorTerminals.SetActive (false);
        FbxWrench.SetActive (false);
        Cabinet.SetActive (false);
        GroundTerminal.SetActive (false);
        CleanIndustry.SetActive (false);
        ScreenPic.SetActive (true);
        Bearing.SetActive (false);
        Spray.SetActive (false);
        LubricateBearing.SetActive (false);
        Filter.SetActive (false);
        ElectricalCabinet.sharedMaterial = materials [0];
        Base.sharedMaterial = materials [1];
        Level.sharedMaterial = materials [2];
        Buttons.sharedMaterial = materials [3];
    }
}

```

```
PumpFilter.sharedMaterial = materials [2];
}
if (LineCounter == 2) {
    Screwdriver.SetActive (false);
    ElectricalProtection.SetActive (false);
    Terminals.SetActive (false);
    Pilots.SetActive (false);
    CircuitBreaker.SetActive (false);
    MotorTerminals.SetActive (false);
    FbxWrench.SetActive (false);
    Cabinet.SetActive (false);
    GroundTerminal.SetActive (false);
    CleanIndustry.SetActive (false);
    ScreenPic.SetActive (false);
    Bearing.SetActive (true);
    Spray.SetActive (false);
    LubricateBearing.SetActive (false);
    Filter.SetActive (false);
    ElectricalCabinet.sharedMaterial = materials [2];
    Base.sharedMaterial = materials [1];
    Level.sharedMaterial = materials [2];
    Buttons.sharedMaterial = materials [3];
    PumpFilter.sharedMaterial = materials [0];
}
if (LineCounter == 3) {
    Screwdriver.SetActive (false);
    ElectricalProtection.SetActive (false);
    Terminals.SetActive (false);
    Pilots.SetActive (false);
    CircuitBreaker.SetActive (false);
    MotorTerminals.SetActive (false);
    FbxWrench.SetActive (false);
    Cabinet.SetActive (false);
    GroundTerminal.SetActive (false);
    CleanIndustry.SetActive (false);
    ScreenPic.SetActive (false);
    Bearing.SetActive (false);
    Spray.SetActive (true);
    LubricateBearing.SetActive (true);
    Filter.SetActive (false);
    ElectricalCabinet.sharedMaterial = materials [2];
    Base.sharedMaterial = materials [1];
    Level.sharedMaterial = materials [2];
    Buttons.sharedMaterial = materials [3];
    PumpFilter.sharedMaterial = materials [0];
}
if (LineCounter == 4) {
    Screwdriver.SetActive (false);
    ElectricalProtection.SetActive (false);
    Terminals.SetActive (false);
    Pilots.SetActive (false);
    CircuitBreaker.SetActive (false);
    MotorTerminals.SetActive (false);
    FbxWrench.SetActive (false);
    Cabinet.SetActive (false);
    GroundTerminal.SetActive (false);
    CleanIndustry.SetActive (false);
    ScreenPic.SetActive (false);
    Bearing.SetActive (false);
    Spray.SetActive (false);
    LubricateBearing.SetActive (false);
    Filter.SetActive (true);
}
```

```

    ElectricalCabinet.sharedMaterial = materials [2];
    Base.sharedMaterial = materials [1];
    Level.sharedMaterial = materials [2];
    Buttons.sharedMaterial = materials [3];
    PumpFilter.sharedMaterial = materials [0];
}
}
}

```

11. Camera script

```

/*=====
Copyright (c) 29/01/2018 code by Alexander Zotov
URL: https://www.youtube.com/watch?v=DQeylS0l4S4
All Rights Reserved.
Confidential and Proprietary - Protected under copyright and other laws.
Modified by: Leire Amenabar and Leire Carreras
=====*/
//Libraries
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

//Main program
public class TakeScreenshot : MonoBehaviour {

    [SerializeField]

    // Main function
    public void TakeAShot()
    {
        StartCoroutine ("CaptureIt");//calles capture photo function
        SceneManager.LoadScene ("PopUpTakePhoto");
    }

    IEnumerator CaptureIt()
    {
        //Captures the screen display and gives a name
        string timeStamp = System.DateTime.Now.ToString("dd-MM-yyyy-HH-mm-ss");
        string fileName = "Screenshot" + timeStamp + ".png";
        string pathToSave = fileName;

        ScreenCapture.CaptureScreenshot(pathToSave);
        yield return new WaitForEndOfFrame();
    }
}

```

12. Compare photos

i. Print the photo taken by the user

```

/*=====
Copyright (c) 29/01/2018 code by Alexander Zotov
URL: https://www.youtube.com/watch?v=DQeylS0l4S4
All Rights Reserved.
Confidential and Proprietary - Protected under copyright and other laws.
Modified by: Leire Amenabar and Leire Carreras
=====*/
//Libraries

```

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using System.IO;

//Main program
public class ScreenshotPreview : MonoBehaviour {
    //Variables
    [SerializeField]
    GameObject canvas;
    string[] files = null;

    // Use this for initialization
    void Start () {
        files = Directory.GetFiles(Application.persistentDataPath + "/", "*.png");
        if (files.Length > 0) {
            MachineData.Instance().whichScreenShotIsShown =files.Length-1;
            GetPictureAndShowIt ();
        }
    }

    void GetPictureAndShowIt()
    {
        string pathToFile = files [MachineData.Instance().whichScreenShotIsShown];
        Texture2D texture = GetScreenshotImage (pathToFile);
        Sprite sp = Sprite.Create (texture, new Rect (0, 0, texture.width, texture.height),
            new Vector2 (0.5f, 0.5f));
        canvas.GetComponent<Image> ().sprite = sp;
    }

    Texture2D GetScreenshotImage(string filePath)
    {
        Texture2D texture = null;
        byte[] fileBytes;
        if (File.Exists (filePath)) {
            fileBytes = File.ReadAllBytes (filePath);
            texture = new Texture2D (2, 2, TextureFormat.RGB24, false);
            texture.LoadImage (fileBytes);
        }
        return texture;
    }

    public void NextPicture()
    {
        if (files.Length > 0) {
            MachineData.Instance().whichScreenShotIsShown += 1;
            if (MachineData.Instance().whichScreenShotIsShown > files.Length - 1)
                MachineData.Instance().whichScreenShotIsShown = 0;
            GetPictureAndShowIt ();
        }
    }

    public void PreviousPicture()
    {
        if (files.Length > 0) {
            MachineData.Instance().whichScreenShotIsShown -= 1;
            if (MachineData.Instance().whichScreenShotIsShown < 0)
                MachineData.Instance().whichScreenShotIsShown = files.Length - 1;
            GetPictureAndShowIt ();
        }
    }
}
```

```

}
//Eliminates actual picture
public void EliminatePicture()
{
    string pathToFile = files [MachineData.Instance().whichScreenShotIsShown];
    File.Delete (pathToFile);
    if (files.Length > 0) {
        MachineData.Instance().whichScreenShotIsShown -= 1;
        if (MachineData.Instance().whichScreenShotIsShown < 0)
            MachineData.Instance().whichScreenShotIsShown = files.Length - 1;
        GetPictureAndShowIt ();
    }
}
}

```

ii. Show the image of the machine in correct state

```

//*****Image
Appear*****//
//Programmers: Leire Amenabar and Leire Carreras
//Date: May 09, 2018
//Revision: 1.0
//Copyright 2018
//Description: In this script depending the current machine, its photo is displayed.
//*****
*****//
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class ImageAppear : MonoBehaviour {
    // Public variables
    public GameObject Machine1Photo;
    public GameObject Machine2Photo;
    public GameObject Machine3Photo;
    public GameObject Machine4Photo;

    // Use this for initialization
    void Start () {
        Machine1Photo.SetActive (false);
        Machine2Photo.SetActive (false);
        Machine3Photo.SetActive (false);
        Machine4Photo.SetActive (false);
    }

    // Update is called once per frame
    void Update () {
        if (MachineData.Instance ().CurrentMachineNum == 1) {
            Machine1Photo.SetActive (true);
            Machine2Photo.SetActive (false);
            Machine3Photo.SetActive (false);
            Machine4Photo.SetActive (false);
        }
        if (MachineData.Instance ().CurrentMachineNum == 2) {
            Machine1Photo.SetActive (false);
            Machine2Photo.SetActive (true);
            Machine3Photo.SetActive (false);
            Machine4Photo.SetActive (false);
        }
        if ((MachineData.Instance ().CurrentMachineNum == 3) || (MachineData.Instance
().CurrentMachineNum == 5)){
            Machine1Photo.SetActive (false);

```

```

Machine2Photo.SetActive (false);
Machine3Photo.SetActive (true);
Machine4Photo.SetActive (false);
}
if (MachineData.Instance ().CurrentMachineNum == 4) {
Machine1Photo.SetActive (false);
Machine2Photo.SetActive (false);
Machine3Photo.SetActive (false);
Machine4Photo.SetActive (true);
}
}
}

```

13. Creation of the pdf

```

/*=====
Copyright (c) 03/08/2017 code by Sammy Haddou - AsteroGames
All Rights Reserved.
Confidential and Proprietary - Protected under copyright and other laws.
Modified by: Leire Amenabar and Leire Carreras
=====*/
//Libraries
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using iTextSharp;
using iTextSharp.text;
using iTextSharp.text.pdf;
using UnityEngine.SceneManagement;
using System.IO;
using System;
using UnityEngine.UI;
using System.Text.RegularExpressions;

// Main program
public class PdfCreator : MonoBehaviour {

    //Variables
    string[] files = null;
    private string pdfName = "My Generated PDF (" + System.DateTime.Now.ToString("yyy-MM-dd_HH-mm-ss") + ")";
    private string path = @"/storage/emulated/0/Maintenance/";
    private string PdfPath;

    byte[] imageBytes;
    RenderTexture rt;

    //The function that creates the pdf
    public void GeneratePDF () {
        string timeStamp = System.DateTime.Now.ToString("dd-MM-yyyy-HH-mm-ss");
        PdfPath = path + "Pdf" + timeStamp + ".pdf"; //Takes defined path to create the pdf file on
it
        createPDF(pdfName); //call to the function
        print("pdf is saved !");
        SceneManager.LoadScene ("LastDisplay");//once the pdf is created loads the last display
    }

    public void createPDF (string fileName) {

```

```

MemoryStream stream = new MemoryStream();
Document doc = new Document(PageSize.A4);
PdfWriter pdfWriter = PdfWriter.GetInstance(doc, stream);

PdfWriter.GetInstance(doc, new FileStream(PdfPath, FileMode.OpenOrCreate,
FileAccess.ReadWrite, FileShare.None));

BaseFont bfHelv = BaseFont.CreateFont(BaseFont.HELVETICA, BaseFont.CP1252, false);

iTextSharp.text.Font fontNormal = new iTextSharp.text.Font(bfHelv, 10,
iTextSharp.text.Font.NORMAL, iTextSharp.text.BaseColor.BLACK);
iTextSharp.text.Font fontBold = new iTextSharp.text.Font(bfHelv, 10,
iTextSharp.text.Font.BOLD, iTextSharp.text.BaseColor.BLACK);

doc.Open();
doc.NewPage();
PdfPTable mainTable = new PdfPTable(1); //the table of the document
mainTable.HorizontalAlignment = Element.ALIGN_CENTER;

PdfPCell tmpCell = new PdfPCell(); // a cell for the title
tmpCell.Border = iTextSharp.text.Rectangle.NO_BORDER;
tmpCell.BorderWidth = 0;
tmpCell.AddElement(new Phrase( MachineData.Instance().MachineName + ", Machine"+
MachineData.Instance().CurrentMachineNum, fontBold));
mainTable.AddCell(tmpCell);

PdfPCell tmpCell12 = new PdfPCell(); // a cell for the normal text
tmpCell12.Border = iTextSharp.text.Rectangle.NO_BORDER;
tmpCell12.BorderWidth = 0;
tmpCell12.AddElement(new Phrase("\n Maintenance operation information: ", fontNormal));
tmpCell12.AddElement(new Phrase("Data: " + System.DateTime.Now.ToString("dd-MM-yyyy HH-
mm-ss") + "\n", fontNormal));
tmpCell12.AddElement(new Phrase("Operator name: " + MachineData.Instance().OpUsername +
"\n", fontNormal));
tmpCell12.AddElement(new Phrase("Machines ID: " + MachineData.Instance().MachineID + "\n",
fontNormal));
tmpCell12.AddElement(new Phrase("Machines part number: " +
MachineData.Instance().MachinePartNumber + "\n\n", fontNormal));
if (MachineData.Instance().MachineName == "Water pump")
{
    tmpCell12.AddElement(new Phrase("Machine values: ", fontNormal));
    tmpCell12.AddElement(new Phrase("-Water temperature: " +
MachineData.Instance().MachineDataVariables[0] + "°C\n", fontNormal));
    tmpCell12.AddElement(new Phrase("-Pressure: " +
MachineData.Instance().MachineDataVariables[1] + "kPa\n", fontNormal));
    tmpCell12.AddElement(new Phrase("-Level of the water: " +
MachineData.Instance().MachineDataVariables[2] + "l\n", fontNormal));
    tmpCell12.AddElement(new Phrase("-Housing temperature: " +
MachineData.Instance().MachineDataVariables[3] + "°C\n", fontNormal));
    tmpCell12.AddElement(new Phrase("-Flow of the water: " +
MachineData.Instance().MachineDataVariables[4] + "l/m\n\n", fontNormal));
    tmpCell12.AddElement(new Phrase("Machine Alarms: ", fontNormal));
    if (MachineData.Instance().WaterTemperatureError == true)
    {
        tmpCell12.AddElement(new Phrase("-There has been a water temperature error!\n ",
fontNormal));
    }
    if (MachineData.Instance().PressureError == true)
    {
        tmpCell12.AddElement(new Phrase("-There has been a pressure error!\n", fontNormal));
    }
}

```

```

//Depending the error that has been generated in the line, its error message is
displayed.
if (MachineData.Instance().LevelError == true)
{
    tmpCell12.AddElement(new Phrase("-There has been a water level error!\n ", fontNormal));
}
if (MachineData.Instance().HousingTempError == true)
{
    tmpCell12.AddElement(new Phrase("-There has been an error with the housing
temperature!\n", fontNormal));
}
if (MachineData.Instance().FlowError == true)
{
    tmpCell12.AddElement(new Phrase("-There has been a water flow error!\n ", fontNormal));
}
if((MachineData.Instance().WaterTemperatureError == false)&&
(MachineData.Instance().PressureError == false)&& (MachineData.Instance().LevelError ==
false)&& (MachineData.Instance().HousingTempError == false)&&
(MachineData.Instance().FlowError == false))
{
    tmpCell12.AddElement(new Phrase("-There was not any error!\n ", fontNormal));
}

}

if (MachineData.Instance().MachineName == "Electric cabinet")
{
    //variables of the electric cabinet are written in the pdf
    tmpCell12.AddElement(new Phrase("Machine values:\n ", fontNormal));
    tmpCell12.AddElement(new Phrase("Voltage: " + MachineData.Instance().MachineDataVariables[0]
+ "V\n", fontNormal));
    tmpCell12.AddElement(new Phrase("Intensity: " +
MachineData.Instance().MachineDataVariables[1] + "A\n\n", fontNormal));
    tmpCell12.AddElement(new Phrase("-Machine Alarms:\n ", fontNormal));
    //errors of the electric cabinet are written in the pdf
    if (MachineData.Instance().VoltageIssue == true)
    {
        tmpCell12.AddElement(new Phrase("-There has been a voltage error!\n\n ", fontNormal));
    }
    if (MachineData.Instance().CurrentIssue == true)
    {
        tmpCell12.AddElement(new Phrase("-There has been a current error!\n\n ", fontNormal));
    }
    if((MachineData.Instance().VoltageIssue == false)&& (MachineData.Instance().CurrentIssue ==
false))
    {
        tmpCell12.AddElement(new Phrase("-There was not any error!\n\n ", fontNormal));
    }
}

// Which steps have been completed are written in the pdf
tmpCell12.AddElement(new Phrase("Followed steps:\n", fontNormal));
for (int j=0; j < MachineData.Instance().TotalNumLines; j++)
{
    if ((j <= MachineData.Instance().LastStepNum))
    {
        tmpCell12.AddElement(new Phrase("-Step " + j + ":                completed\n", fontNormal));
    }
    if ((MachineData.Instance().LastStepNum != MachineData.Instance().TotalNumLines - 1)&&(j >
MachineData.Instance().LastStepNum)&& (j < MachineData.Instance().TotalNumLines-1))
    {
        tmpCell12.AddElement(new Phrase("-Step " + j + ":                not executed\n",
fontNormal));
    }
}

```

```

    }
    if ((MachineData.Instance().LastStepNum != MachineData.Instance().TotalNumLines - 1) && (j
> MachineData.Instance().LastStepNum) && (j == MachineData.Instance().TotalNumLines - 1))
    {
        tmpCell12.AddElement(new Phrase("-Step " + j + ":                not executed\n\n",
fontNormal));
    }
}

mainTable.AddCell(tmpCell12);

PdfPCell tmpCell13 = new PdfPCell(); // a cell for the picture
tmpCell13.Border = iTextSharp.text.Rectangle.NO_BORDER;
tmpCell13.BorderWidth = 0;
tmpCell13.Colspan = 2;

files = Directory.GetFiles(Application.persistentDataPath + "/", "*.png");
MachineData.Instance().whichScreenShotIsShown = files.Length - 1;
string pathToFile = files[MachineData.Instance().whichScreenShotIsShown]; // takes the last
photo taken by the user
Texture2D texture = GetScreenshotImage(pathToFile); //goes to the function get screen shot
AddImageToPDFCell(tmpCell13, GetScreenshotImage(pathToFile), 600, 300); // the image is add
to the pdf
mainTable.AddCell(tmpCell13);
doc.Add(mainTable);
doc.Close();
pdfWriter.Close();
stream.Close();
//Done editing the document, we close it and save it.
}

// Gets the screenshot from the file that has been saved in the previous script.
Texture2D GetScreenshotImage(string filePath)
{
    Texture2D texture = null;
    byte[] fileBytes;
    if (File.Exists (filePath)) {
        fileBytes = File.ReadAllBytes (filePath);
        texture = new Texture2D (2, 2, TextureFormat.RGB24, false);
        texture.LoadImage (fileBytes);
    }
    return texture;
}

//will draw the generated screenshot directly inside the PDF.
public void AddImageToPDFCell (PdfPCell cell, Texture2D img, float scaleX, float scaleY) {
    imageBytes = img.EncodeToPNG();
    iTextSharp.text.Image finalImage = iTextSharp.text.Image.GetInstance(imageBytes);
    finalImage.Alignment = iTextSharp.text.Image.ALIGN_CENTER;
    finalImage.Border = iTextSharp.text.Rectangle.NO_BORDER;
    finalImage.BorderColor = iTextSharp.text.BaseColor.WHITE;
    finalImage.ScaleToFit(scaleX, scaleY);
    cell.AddElement(finalImage);
}

}

```

14. Last Display

```

/*****Last display
prog*****/
//Programmers: Leire Amenabar and Leire Carreras
//Date: May 09, 2018
//Revision: 1.0
//Copyright 2018
//Description: In this script the current machines name is displayed. Besides, asks if whole
the line maintenance has finished or not
//              depending on that different scenes are load.
/*****
*****/
//Libraries
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;
using UnityEngine.UI;

public class LastDisplayProg : MonoBehaviour {
    //Variables
    public Text MachineNameText;

    //if whole line maintenance is finished, the program is closed
    public void YesButton(){
        Application.Quit ();
    }

    //if the whole line maintenance is not finished, displays all line display again
    public void NoButton(){
        MachineData.Instance ().MachineFinished = true;
        SceneManager.LoadScene ("AllLineDisplay");
    }
    // Use this for initialization
    void Start () {
        MachineNameText.text = MachineData.Instance().MachineName + " " +
MachineData.Instance().LastDispMachiNum;
    }

}

```

15. Database

```

/*****Display information program*****/
//Programmers: Leire Amenabar and Leire Carreras
//Date: May 09, 2018
//Revision: 1.0
//Copyright 2018
//Description: This script is the database of the program, where all the information and
structures are saved.
/*****
*****/
//Libraries
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using System;
using System.IO;

```

```
using System.Text.RegularExpressions;

public class MachineData
{
    public bool error;
    public bool warning;
    public bool correct;
    public bool GeneralError;
    public bool GeneralWarning;
    public bool GeneralCorrect;
    public bool FixFilters;
    public bool MachineFinished=false;
    public bool PressureIssue;
    public bool WaterTemperatureIssue;
    public bool WaterTemperatureError;
    public bool HousingTempError;
    public bool PressureError;
    public bool LevelError;
    public bool FlowError;
    public bool VoltageIssue;
    public bool CurrentIssue;
    public string MachineName;
    public int CurrentMachineNum;
    public bool InitiateNow;
    public int whichScreenShotIsShown= 0;
    public double[] MachineDataVariables = new double[5];

    public bool preventive;
    public bool palliative;
    public bool corrective;
    public string MachineID;
    public string MachinePartNumber;
    public Text NextMachineName;
    public int s;
    public int LastDispMachiNum;
    public string Username;
    public string OpUsername;
    public string Password;
    public int LastStepNum;
    public int TotalNumLines;

    public int[,] MachineState = new int[5, 3];

    // Variables to compare data with maximum and minimum limits
    public double MaxWaterTemp=40;
    public double MinWaterTemp=10;
    public double RecMaxWaterTemp=30;
    public double RecMinWaterTemp=15;
    public double MaxHousingTemp=40;
    public double MinHousingTemp=10;
    public double RecMaxHousingTemp=30;
    public double RecMinHousingTemp=15;
    public double MaxPressure=350;
    public double MinPressure=200;
    public double RecMaxPressure=320;
    public double RecMinPressure=260;
    public double MaxWaterLevel=1000;
    public double MinWaterLevel=500;
    public double RecMaxWaterLevel=800;
    public double RecMinWaterLevel=600;
    public double MaxFlow=150;
    public double MinFlow=50;
```

```
public double RecMaxFlow=110;
public double RecMinFlow=70;
public double MaxVoltage=420;
public double MinVoltage=380;
public double RecMaxVoltage=410;
public double RecMinVoltage=390;
public double MaxCurrent=4.5;
public double MinCurrent=2.5;
public double RecMaxCurrent=4;
public double RecMinCurrent=3;

//Struct
public struct MachinePosition
{
    public double x;
    public double y;
    public int MachineNumber;

    public MachinePosition (int numberMachine, double positionX, double positionY)
    {
        x = positionX;
        y = positionY;
        MachineNumber = numberMachine;
    }
}

public struct WaterpumpData
{
    public double WaterTemp;
    public double HousingTemp;
    public double Pressure;
    public double WaterLevel;
    public double Flow;
    public string PartNum;
    public string IDNum;
    public string MachineType;
    public int MachineNum;

    public WaterpumpData (int numMachine, double tempWater, double tempHousing, double
press, double levelWater, double flowWater, string partNumber, string ID, string typeMachine)
    {
        WaterTemp = tempWater;
        HousingTemp = tempHousing;
        Pressure = press;
        WaterLevel = levelWater;
        Flow = flowWater;
        PartNum = partNumber;
        IDNum = ID;
        MachineType = typeMachine;
        MachineNum = numMachine;
    }
}

public struct ElecCabinetData
{
    public double Voltage;
```

```

public double Current;
public string PartNum;
public string IDNum;
public string MachineType;
public int MachineNum;

public ElecCabinetData (int numMachine, double volts, double amperes, string
partNumber, string ID, string typeMachine)
{
    Voltage = volts;
    Current = amperes;
    PartNum = partNumber;
    IDNum = ID;
    MachineType = typeMachine;
    MachineNum = numMachine;
}

// Machines information
public WaterpumpData DataMachine1 = new WaterpumpData (1, 20, 20, 300, 700, 80, "AD1250",
"ASD20563", "Water pump");
public WaterpumpData DataMachine2 = new WaterpumpData (2, 50, 21.5, 310, 710, 80,
"AD1250", "ASD20573", "Water pump");
public ElecCabinetData DataMachine3 = new ElecCabinetData (3, 400, 3.5, "PIK56",
"AKJ98630", "Electric cabinet");
public WaterpumpData DataMachine4 = new WaterpumpData (4, 20, 20, 300, 700, 80, "AD1250",
"ASD21563", "Water pump");
public ElecCabinetData DataMachine5 = new ElecCabinetData (5, 400, 2.6, "PIK56",
"AK99852", "Electric cabinet");

private static MachineData instance = null;

public static MachineData Instance()
{
    if (instance == null)
    {
        instance = new MachineData();
    }
    return instance;
}
}

```

16. Rotate objects

i. X axis

```

/*=====
Programmers: Leire Amenabar and Leire Carreras
Date: May 09, 2018
Revision: 1.0
Copyright 2018
Description: This script rotates selected object in x axis with the speed defined in the
variables.
Confidential and Proprietary - Protected under copyright and other laws.
=====*/
//Libraries
using System.Collections;

```

```
using System.Collections.Generic;
using UnityEngine;

//Main program
public class RotateX : MonoBehaviour {
    private float speed=1f;

    // Update is called once per frame
    void Update () {
        transform.Rotate ( speed,  0,  0);
    }
}
```

ii. Y axis

```
/*=====
Programmers: Leire Amenabar and Leire Carreras
Date: May 09, 2018
Revision: 1.0
Copyright 2018
Description: This script rotates selected object in y axis with the speed defined in the
variables.
Confidential and Proprietary - Protected under copyright and other laws.
=====*/
//Libraries
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

//Main program
public class RotateY : MonoBehaviour {
    private float speed=1f;

    // Update is called once per frame
    void Update () {
        transform.Rotate ( 0,  speed,  0);
    }
}
```

iii. Z axis

```
/*=====
Programmers: Leire Amenabar and Leire Carreras
Date: May 09, 2018
Revision: 1.0
Copyright 2018
Description: This script rotates selected object in z axis with the speed defined in the
variables.
Confidential and Proprietary - Protected under copyright and other laws.
=====*/
//Libraries
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

//Main program
public class RotateZ : MonoBehaviour {
    private float speed=1f;

    // Update is called once per frame
```

```
void Update () {  
    transform.Rotate ( 0,    0,  speed);  
}  
}
```