Universidade de Aveiro Departamento de Eletrónica, Telecomunicações e Informática
Ano 2018/2019

Renato Pereira Cabral

RobARtics: Assisting in robotics competitions using Augmented Reality

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Computadores e Telemática, realizada sob a orientação científica da Doutora Maria Beatriz Alves de Sousa Santos, Professora Associada com Agregação do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro e do Doutor Samuel Sousa Silva, Investigador do Instituto de Engenharia Electrónica e Informática de Aveiro.

o júri

Presidente:	Prof. Doutor Joaquim João Estrela Ribeiro Silvestre Madeira professor auxiliar da Universidade de Aveiro
Arguente Principal:	Prof. Doutor José Manuel Rodrigues Nunes professor auxiliar da Universidade de Aveiro
Orientador:	Doutor Samuel Sousa Silva Investigador do Instituto de Engenharia Electrónica e Informática de Aveiro

agradecimentos

Agradeço ao Bernardo Marques e ao João Alves pelo apoio e pelo conhecimento que me foram fornecendo durante a concretização da dissertação, aos professores Beatriz Santos e Samuel Silva pela orientação que disponibilizaram, ao professor Artur Pereira e aos restantes elementos que tiveram a oportunidade de testar o trabalho desenvolvido ao longo da dissertação.

palavras-chave Realidade Aumentada, Competições Robóticas

resumo

A robótica é um campo dedicado ao estudo e à engenharia de agentes, conhecidos como robôs. Os avanços tecnológicos nos últimos anos permitiram criar agentes mais avançados e inteligentes. A criação de competições robóticas, eventos onde engenheiros, ou estudantes, podem colocar os seus agentes à prova numa série de desafios, geralmente supervisionados por árbitros, com assistência de público geral. Apesar de esses eventos servirem como uma forma de entretenimento, também podem ser benéficos tanto para a equipa que traz o seu agente, como para o público que está a assistir, para aprender mais sobre o campo da robótica, bem como os próprios robôs. No entanto, há situações em que faz falta um método para fornecer informação suficiente a todas as partes, uma vez que os participantes muitas vezes não têm hipótese de se aproximar dos robôs, os investigadores precisam de interromper a equipa da competição para obter mais informação e os árbitros podem ser colocados em situações ambíguas. Esta dissertação explora um novo método de incrementar o que os utilizadores podem ver durante a competição, utilizando uma abordagem baseada em realidade aumentada (RA), com recurso a marcadores, explorando a influência que a realidade aumentada pode ter nas competições em si. Como prova de conceito, criou-se uma aplicação, com opções adequadas aos utilizadores referidos, para ser usada em dispositivos de utilização diária, como os smartphones, com o objetivo de proporcionar aos utilizadores uma maneira eficaz de obter mais informação sobre os robôs em competições robóticas. Para ser testada a sua eficiência, foi criado um questionário com uma série de tarefas, realizado por um número de participantes. A opinião de participantes em tais competições também foi incorporada no trabalho final.

keywords Augmented Reality, Robotic Competitions

abstract Robotics is field dedicated to studying and engineering agents, known as robots. The technological advancements in recent years have allowed the increase in both the robot's hardware and software complexity, allowing the creation of robotic competitions, events where engineers can put their agents to test in a series of challenges, generally supervised by referees, with public attendance. While these events act as a form of entertainment, they can also be very beneficial to both the staff and the attending public to learn more about the field of robotics, as well as the robots themselves. However, usually there's lack of a method to provide enough information to everyone involved, since attendees often don't get the chance of getting closer to the robots, researchers often need to interrupt the competition staff to learn more information and referees might face ambiguous situations. This dissertation explores a new method of amplifying what users are able to see during the competition, utilizing an Augmented Reality (AR) marker-based approach and exploring how this medium can influence the competitions themselves. As a proof-of-concept, an application was created, featuring options suitable for everyone involved that could present more information to the users, developed for mobile devices used daily, such as smartphones, aimed at providing users with an effective way of obtaining more information about the robots and robotics competitions. To test its efficiency, the application was tested among a group of users with different scientific knowledge. Also, the feedback from participants in such competitions was incorporated in the dissertation, as well.

Index

List of Figures

List of Tables

List of Abbreviations

API – Application Programming Interface

AR – Augmented Reality

CAN – Controller Area Network

DoF – Degrees of Freedom

FNR – Festival Nacional de Robótica

HMD – Head Mounted Display

HRI – Human-Robot Interaction

ICARSC - International Conference on Autonomous Robot Systems and Competitions

MR – Mixed Reality

ROS – Robot Operating System

SDK – Software Development Kit

SRP – Sociedade Robótica Portuguesa

UI – User Interface

USB – Universal Serial Bus

VR – Virtual Reality

1. Introduction

Robotic competitions are a form of entertainment that test the engineering capabilities of each participant using robots. They have come a long way since their inception, with competitions between robots dating back to the 1970s. Although the technology used in these events is constantly being upgraded, sometimes there’s uncertainty when it comes to the challenge’s outcomes. Disagreements between the competition staff or the participants may arise in ambiguous situations that can have outcomes that are hard to tell, such as deciding which robot crossed the finish line first during a race. Additionally, researchers and the attendees of such competitions are often given little information about the robots themselves, with their only option of learning more being talking to the competition staff.

Just like robotics, technologies as Augmented Reality (AR), have been constantly changing, evolving and improving, in addition to its use branching over the years through multiple areas like medical application, manufacturing, education and more recently, gaming and robotics. While nowadays there are several AR systems used mostly in interactions between humans and robots, the integration of AR interfaces with it during competitions is a concept still relatively new and unexplored, mostly in robotic competitions. This dissertation aims to create a new AR based approach to aid researchers, referees and attendees of such competitions, by introducing AR concepts to allow the visualization of more information about the robots and explore the benefits of the introduction of this medium to the robotic competitions.

1.1 Challenges

The world of robotic competitions is always evolving and there are several challenges that need to be tackled.

The huge amount of data that needs to be analysed and compared, for instance, to understand how different robot configurations influence its performance, hinders taking the best out of the researchers' insights and knowhow. To have a grasp of the differences among multiple runs might, for instance, enable spotting how the robot trajectories or speed were influenced, for particular segments of the route, and infer how to further adjust the configurations. While this kind of analysis might, eventually, be performed using unsupervised computational methods, the proposal of visual methods for exploration of the data, in place, takes advantage of user experience, fosters new insights and can, in the future, inform the proposal of such methods;

Additionally, with the rising interest of the general public in attending robot competitions, there is a pressing need to provide the audience with a contextualization of what is happening, during the competition, particularly regarding robot performance. For instance, enabling an understanding of which signal the robot just recognized, which direction it took at a track bifurcation, or where the robot went out of the track.

To address these challenges, AR might play a strategic role by providing the means to support the presentation, exploration and analysis of the relevant robot and competition data.

1.2 Objectives

This dissertation discusses the application of AR in robotics, one of the rising applications of this technology. In concrete, this work aims to:

- Understand the needs and motivations of the researchers involved in robotic competitions and the general public attending those events, particularly regarding the data and information they require to serve their intents;
- Propose AR methods to enable presenting the data and information about the robot and its performance, in context;

- Design and develop a proof-of-concept application demonstrating the proposed methods;
- Perform a preliminary evaluation of the proposed methods to assess their adequateness and inform future developments;

1.3 Structure

This work is organized by 6 chapters, each with various subchapters. The first chapter introduces the studied theme and presents the topics addressed by the dissertation.

The second chapter describes the *Festival Nacional de Robótica (FNR)*, the Portuguese robotics competition festival, alongside many details related to the competitions and the event in general. Also, it describes an overview of the current state of the art of AR, its applications, technologies used, potential advantages, current challenges and limitations, as well as presenting the work by many developers using AR, whether it was done in the field of robotics or in other areas, such as tourism and finally, it explores some of the evaluation methods that are used to test the effectiveness of certain AR systems.

The third chapter presents the personas, scenarios to contextualize the uses of the application and its system requirements like the Unity 3D engine and the ROS framework.

The fourth chapter describes the main application developed for this dissertation. It outlines the application's architecture, the prototype developed before it's development in the engine, the features available, as well as a description on how they were achieved.

The fifth chapter defines the user tests carried out to test the application's effectiveness, the tasks users had to perform during testing and the pre/pos-experience questionnaires.

The sixth chapter serves as the conclusion, presenting challenges and obstacles confronted during development and future work.

2. Background

Since their inception, AR has been constantly improving thanks to the recent technological advancements. Likewise, robotic competitions elements, such as rules and the types of robots introduced have been constantly changing, with different models being introduced as years passed. This chapter serves as an introduction to the topics and provides a state-of-the-art in AR, including its applications, devices used, how it has been evolving, as well as history about robotic competitions, some of the rules, among other relevant information. Additionally, it presents related work and applications developed using AR, many of which is integrated in robotics.

2.1. Robotic competitions

Robotic competitions can be described as events that consist of multiple tasks must be performed by the robots, usually competing to best each other during the competition. These competitions date back to 70s, using as an example the 1977 IEEE's Micromouse competition, described in the *Spectrum* magazine[1].

2.1.1. FNR

The *Festival Nacional de Robótica (FNR)* is an event designed to "encourage Science and Technology to researchers, students and general public through the use of automated robots"[2], promoted by the *Sociedade Portuguesa de Robótica (SRP)*. Alongside this, it is also where many Portuguese teams are accepted towards the RoboCup, the international robotic competition. The FNR is also the home to the International Conference

[1] https://spectrum.ieee.org/consumer-electronics/gadgets/the-amazing-micromouse-contest Visited in 7/10/2019
[2] https://web.fe.up.pt/~robotica2019/index.php/pt/ Visited in 7/10/2019

on Autonomous Robot Systems and Competitions (ICARSC), an event where researchers from many parts of the world present their latest advances in robotics.

2.1.2. Autonomous driving

The scope of the application is geared towards the autonomous driving challenge, more specifically, the driving and signal identification challenges. It consists of the deployment of a mobile and autonomous robot capable of traversing a closed circuit, with some similarities to a regular traffic road.

According to FNR, the track usually consists of an 8-pattern track, with some variations including zebra crossings, tunnels or traffic lights, none of which are previously known to the robot.

The challenge is divided in smaller tasks, that are performed in 3 consecutive days. There are driving challenges, parking challenges and vertical signal detection challenges, and robots are able to compete in two different classes, *Expert,* which includes all the challenges or *Rookie,* a more simplified approach that allows manual turn on and excludes some of the challenges[3].

2.1.3. Road sign detection criteria

During the competition, there are available twelve unique road signs, that are placed on the right side of the track. These are equally grouped in triangular warning signs, round mandatory signs and square shaped service signs (Figure 2.1).

[3] https://web.fe.up.pt/~robotica2019/index.php/pt/ Visited in 7/10/2019

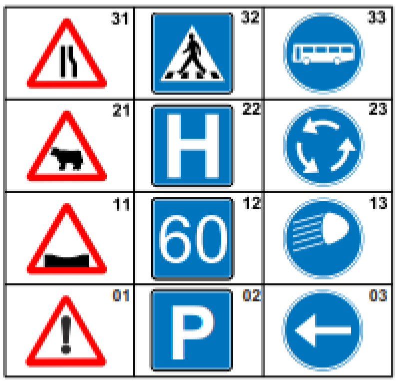


Figure 2. 1 List of the road signs available for the 19th edition of the competition[4]

Users can incorporate 2 different methods of displaying the signs according to the following criteria. The first method allows the robot to be equipped with LEDs of different colors to make the signal identification. "The red led must be used to identify a warning sign, the blue led to identify a mandatory sign and the green led to identify an information sign.[4]" Each of the LEDs must also employ a light sequence to indicate the order in which the signs are detected.

The second method allows the road signs to be displayed using a laptop, in a similar manner as the LED light sequence or with an image of the detected signal.

2.1.4. Driving challenges

The driving challenges' intention is to have robots completing two laps around the track in the shortest time possible, while accruing the least amount of penalties possible.

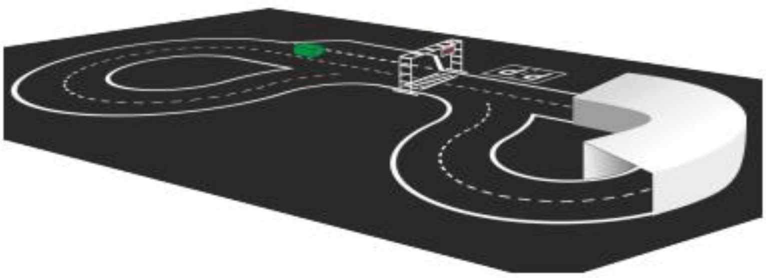


Figure 2. 2 Competition track[4]

[4] https://web.fe.up.pt/~robotica2019/images/fnr2019_Autonomous_Driving.pdf Visited in 7/10/2019

Figure 2.2 represents the track used during competitions, albeit the 19th edition of the competition removed the tunnel. There are four challenges of increasing difficulty. In the first challenge, driving at pure speed challenge, referred as D1, the robot starts before the zebra crossing and must complete 2 laps and stop right before the zebra crossing. In the second challenge, driving with signs challenge, referred as D2, the robot must complete two laps around the track while obeying the signalling panels on the ground (Figure 2.3 and 2.4).

Function	Action	Signal
1	Follow to the left	A left pointing green horizontal arrow
2	Follow to the right	A right pointing green horizontal arrow
3	Follow straight ahead	A green colored vertical arrow
4	Stop	A red colored "X"
5	End of trial	Red and green checkers flag
6	Follow to parking area	A yellow colored "P"

Figure 2. 3 Available panels and its respective actions[4]

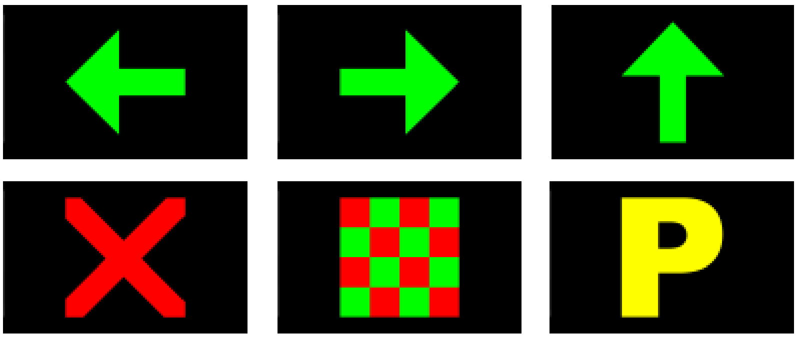


Figure 2. 4 Panel drawings[4]

In the third challenge, called driving with signs and obstacles challenge, referred to D3, not only robots must identify signs, but also must get around randomly placed obstacles during the second half of the course.

The last challenge, called driving with all challenge, referred to D4, "two obstacles are placed in the first half and a road working area is added to the second one".[4]

2.1.5. ROTA

The ROTA is a robot designed for driving challenges, created by the IRIS Lab in Universidade de Aveiro. It's a tricycle composed by two wheels with directional control, but without traction and a directionless third wheel with traction behind, adopting the Ackerman steering model (Rodrigues, 2018).

Its high-level design architecture was developed using ROS, a framework designed to establish communication between engineering components, which will be explained in further detail in the fourth chapter.

The control system for this robot is composed by a high-level layer, connected to a computer, which is responsible for coordinating the robot's movement. This layer is linked to a low-level layer through Universal Serial Bus (USB). The low-level layer itself is composed by a Controller Area Network (CAN) of microcontrollers (Rodrigues, 2018).

On top of it, it has a Kinect camera, equipped with RGB-D and a Laser Range Finder (LRF). The RGB-D component allows the signal identification, as well as capturing images while the robot is in motion. The LRF is used to detect and avoid obstacles (Rodrigues, 2018) (Figure 2.5).

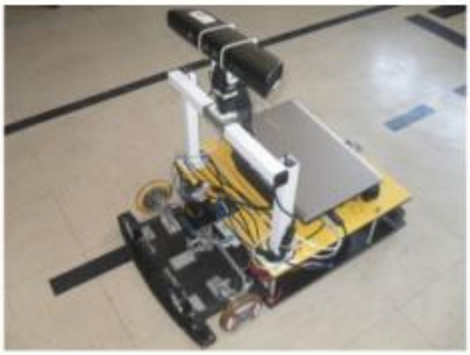


Figure 2. 5 ROTA robot

The application for this dissertation was developed with this robot model in mind. However, since it wasn't possible to use a real model of the agent, it was used and tested with the simulation version developed in the Gazebo simulator (Figure 2.6).

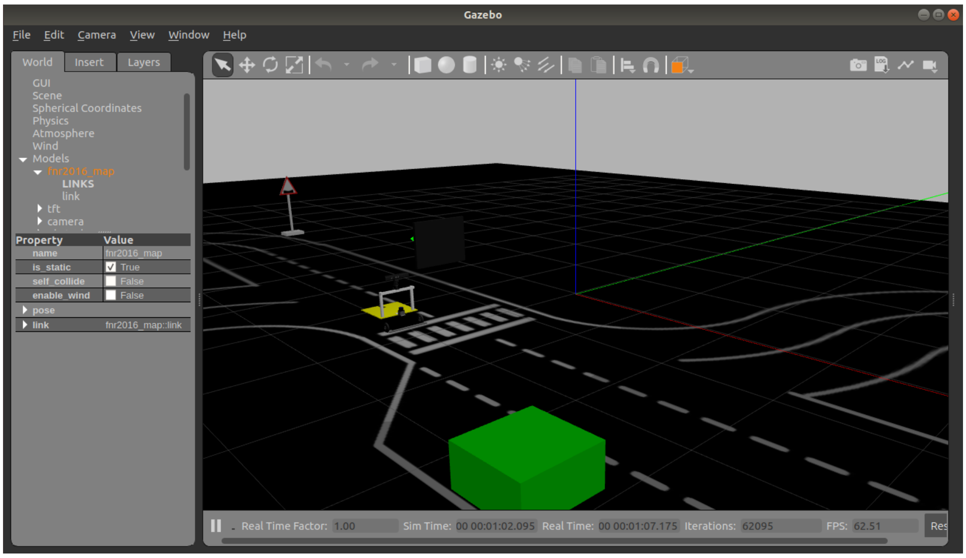


Figure 2. 6 ROTA Simulator

The simulator allows the robot to be controlled with a remote controller, such as the Playstation™ 3 controller. All the signs presented in the previous paragraphs are available to use, along the courses of previous iterations of the competitions.

2.2. Augmented Reality

AR can be defined as a mechanism in which information is added to the physical world in tandem with the world (Craig, 2013), or can also be defined as a variation of Virtual Reality (VR). When interacting with VR, the user is engaged with only the virtual environment and as such, the interactivity with real world is practically non-existent. On the contrary, AR complements reality, since it allows the user to see the real world with overlapping objects represented in the real world (Azuma, 1997).

Figure 2.7 shows an example of an AR marker-based application, the eDrawing. By using AR, it allows users to create 3D representation of products and projected into the world. While the table is completely free of objects, it's possible to see a virtual representation of a robot by pointing a mobile device at a marker.

Figure 2. 7: eDrawing for iOS[5]

2.2.1. Characteristics of AR

There are several specific characteristics that define AR, defined by Alan B. Craig, who built them upon Ronald T. Azuma's own definition of the AR characteristics.

. The first and foremost characteristic is "the physical world is augmented by digital information that is superimposed on a view of the physical world" (Craig, 2013), meaning that, all information is overlaid into the real world, whether is visually, auditory, touch based and may or may not have a symbolic link to the physical world. The user remains in touch with reality throughout the whole experience and is able to interact with their environment with devices that don't hinder the real world (Craig, 2013).

Next, "information in AR has always a physical space or location in the physical world like a physical counterpart to the digital information would have" (Craig, 2013). Using an example, if there's an object placed on a table in the physical world, such object will remain in the same location independently of the user's movement or viewing location, unless this object is purposely moved by interacting with it (Figure 2.8). The same is applied when visualizing augmented information in any situation, but when the viewer decides to specifically move the object. While there isn't obligatory the existence of an equivalent physical object to a certain digital object, its spatial location registration, that being how close to reality is the digital object, is placed in the physical world while its temporal location registration, the constant update of an object's position to closely match the viewer's perspective, should be both accurately maintained (Craig, 2013).

[5] https://blogs.solidworks.com/solidworksblog/2013/02/augmented-reality-in-edrawings.html Visited in 6/3/2019

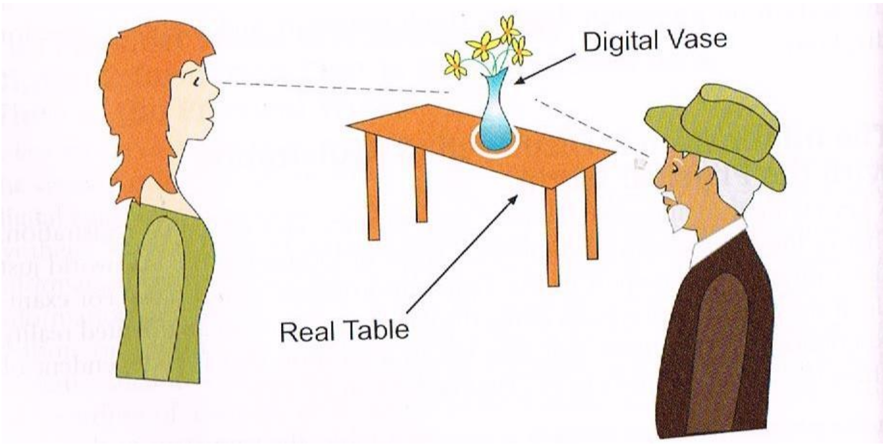


Figure 2. 8: Example of two people looking at an object from different perspectives. This aspect is maintained in AR (Craig, 2013).

Lastly, the information displayed is dependent on the geographic location of the real world and the physical perspective of the person in the physical world (Craig, 2013). When interacting with a real-world object, a person's perspective of an object will constantly change according to their location, as they experience multiple points of view of the same object. As described before, this is taken into account when creating an augmented reality experience (Craig, 2013).

Another important concept in AR is Mixed Reality (MR). (Milgram & Kishino, 1994) defined the concept of MR, a sub division of the VR technology, which focuses on combining both virtual and real worlds. This concept was further developed with the creation of the virtuality continuum (Figure 2.9).

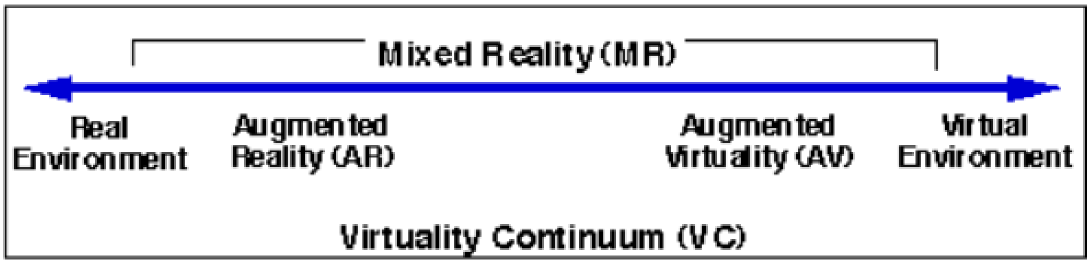


Figure 2. 9: Representation of the virtual continuum (Milgram & Kishino, 1994)

The virtual continuum serves a means to classify the different MR elements, according to the way they are perceived in a certain environment. As it can be observed, AR is part of the real aspect of the environment, between the completely real environment and the virtual environment. The ensemble of the two extremes forms the MR environment. This model is still widely used to represent the scale between virtual and real objects.

2.3. Devices used in AR

According to (Furht et al., 2010), the four main devices used in AR are displays, input devices, tracking and computers.

2.3.1. Displays

Starting with the displays, in AR systems, there are currently three types that are applied: the visual displays, aural displays and haptic displays (Krevelen & Poelman, 2010), with the latter two usage being limited. The visual displays are the most common way to represent visually augmented information. This type of display can also be classified into three different categories:

- Video see-through – Displays that work similarly to VR devices. A video feed of the real world replaces the virtual environment and the AR components are loaded into digitized images (Krevelen & Poelman, 2010).
- Optical see-through – Displays that use mirrors or lenses to display AR without interfering with the perception of reality (Krevelen & Poelman, 2010).
- Projective – Displays that extend AR directly into the real world objects without requiring any additional equipment, like eye-wear (Krevelen & Poelman, 2010).

Haptic displays function by applying force and sending it to its user information through vibrations. They are usually used as tele operable displays to control robots. Finally, aural displays are a means to represent audio in AR.

Displays are also classified according to their position relative to the interaction between the viewer and their environment. Starting with HMDs, they are a type of display worn on certain headwear or directly on the head that overlap objects with information from the real and virtual environments. Visually, they can be classified as either video see-through or optical see-through. Using video see-through displays is usually more exhausting, as the user is required to wear two cameras and each of those cameras needs to process both the virtual elements, as well as the augmentations presented (Figure 2.10).

Figure 2. 10: Example of an optical see-through model, the Meta 2 AR Glasses[6]

This configuration allows the user to have a better control over the results, since the augmentations are already transposed into digital images, as previously mentioned. In contrast, by using lenses to convey information, optical see-through devices offer a more realistic view of an augmented scene and the real world. The downsides of this configuration however, include small delays caused by the image processing, which may cause the user to see objects detached with the real objects.

Hand-held displays are typically small devices, designed to fit into the user's hands and can display information from a small screen (Figure 2.11). Using video see-through capabilities, they usually use computer vision methods, such as SLAM and sensors, like GPS, for supporting tracking sensors. The hand-held devices most commonly used for AR nowadays are Tablets and smart phones.

[6] https://www.wearvr.com/hardware/hmds/meta-2-ar-hmd Visited in April, 1 of 2019

Figure 2. 11: Example of a handheld device operating (Mekni & Lemieux, 2014)

Lastly, spatial displays “are placed statically within the environment” (Krevelen & Poelman, 2010), make use of projectors and may include screen based video see-through displays and spatial optical see-through displays. They exclude most of the technology usually required to use by a user, opting for integrating it with the environment and are most commonly used in sports television nowadays.

Video see-through approaches are more suitable when there isn’t the need to implement a mobile system. Optical see-through approaches display images that are in line with the physical environment and, similar to the previous technique, they cannot be implemented in mobile applications due “to spatially aligned optics and display technology” (Furht et al., 2010). Additionally, Spatial Augmented Reality (SAR) can also be approached using direct augmentation, thus displaying images directly on top of solid surfaces (Furht et al., 2010) (Figure 2.12).

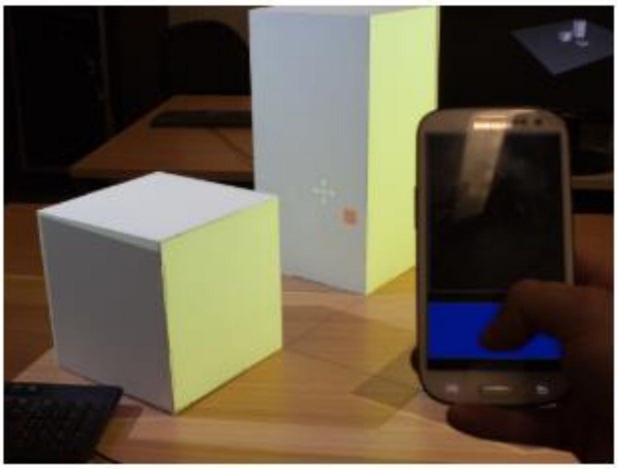


Figure 2. 12: Interaction with a SAR system via mobile (Marner, Smith, & Thomas, 2015)

2.3.2. Scene Identification

Following the classification of the displays, other important aspect of AR are the scene identification techniques. The two standard types of scene are:

- Marker-based: This technique uses markers, visual tags that represent indicators for the AR system to interact with the real world (Alkhamisi & Monowar, 2013) (Figure 2.13).
- Non-Marker-Based: Technique applied to AR systems that don't include any markers, instead opting for devices in scene identification (Alkhamisi & Monowar, 2013) (Figure 2.14).

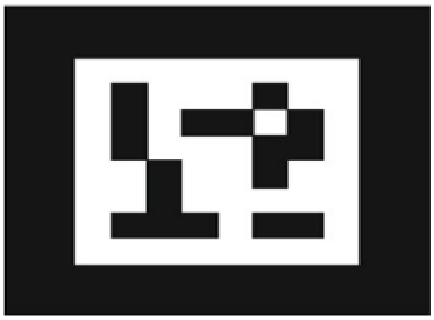


Figure 2. 13: Example of a marker (Alkhamisi & Monowar, 2013)

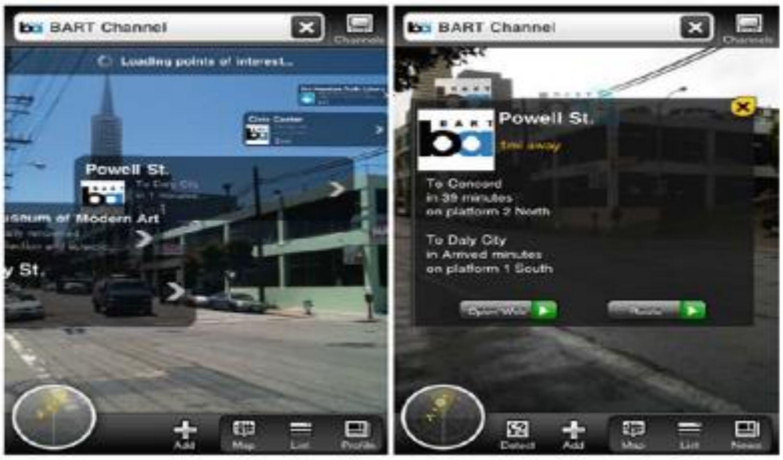


Figure 2. 14: Example of a non-marker based application (Alkhamisi & Monowar, 2013)

2.3.3. Tracking

Tracking serves the purpose of monitoring the movement of a user, as well as tracking the position and movement of a device. The generally accepted in the implementation of AR systems are mechanical, magnetic sensing, GPS, ultrasonic, inertia, acoustical (Craig, 2013) and optical (Furht et al., 2010).

2.3.4. Computers

Computers act as the core of a system. They receive signals from the tracking devices, execute the instructions from the application program according to the information received by the sensors, which in turn, create the signals to establish communication with the displays of a system (Craig, 2013).

Generally, due to the high-computing requirements of image processing, AR applications require a significant amount of RAM and a powerful Central Processing Unit (CPU). Currently, devices such as smartphones, have been having a major impact in modelling the processing units of AR systems to more practical solutions, due to their progressive evolution in terms of specifications.

2.3.5. Input devices

The input devices allow a user to interact with the virtual elements during an AR experience. There are available multiple types of input devices. For instance, (Marner et al., 2015) used a Kinect camera to track the user’s position in their Viewpoint Cursor application. On the other hand, (Valentini, 2018) integrated Leap Motion controls to develop a system that allowed users to experience a completely augmented scene through an HMD.

The selection of input devices greatly depends on the type of application developed and the type of displays chosen. For example, if the developers chose to use a SAR application, a smartphone serves as a good choice for interacting with its environment.

2.4. AR Interfaces

AR interfaces should intuitive means in which a user interacts with the virtual content of an application. The four main types implemented in AR applications are: tangible AR interfaces, collaborative AR interfaces, hybrid AR interfaces, and the multimodal interfaces. (Kim et al., 2018)

2.4.1. Tangible AR interfaces

Tangible AR interfaces utilize physical devices, such as a computer mouse, to support direct interaction with the physical world. They are generally used in HRI applications.

(Ullmer & Ishii, 2001) defined the following characteristics for general tangible interfaces:

- "Physical representations are computationally coupled to underlying digital information" (Ullmer & Ishii, 2001).
- "Physical representations embody mechanisms for interactive control" (Ullmer & Ishii, 2001).
- "Physical representations are perceptually coupled to actively mediated digital representations" (Ullmer & Ishii, 2001).
- "Physical state of tangibles embodies key aspects of the digital state of a system" (Ullmer & Ishii, 2001).

For example, (Weinstock, 2015) developed CheckMate, a tangible AR interface that combined modern AR technologies with 3D printed objects, tested in a chess game as a proof-of-concept, with 3D printed pieces and HMDs.

2.4.2. Collaborative AR Interfaces

Collaborative AR interfaces make use of multiple displays to support remote and co-located activities. Co-located sharing uses 3D interfaces to improve physical collaborative workspace. In remote sharing, AR can easily integrate multiple devices with multiple locations to enhance teleconferences or it can be applied to medical applications, such as surgeries (Furht et al., 2010). Additionally, collaborative interfaces can be face-to-face, which target to make an application suitable for use by multiple people and transitional interfaces that adopt digital animations as functionalities.

2.4.3. Hybrid AR Interfaces

Hybrid AR interfaces combine multiple distinct, but complementary interfaces as well as the possibility to interact with them through a wide range of interaction devices (Furht et al., 2010). They provide flexibility in cases where it's unknown which type of display or interactive devices will be used and can even be applied to applications to prevent unplanned issues, such as changes in input or output devices. As a result, applications that adopt this type of interface will support more common operations and can even be extended to allow users to specify new operations at run time (Zhou, Been-Lirn Duh, & Billinghurst, 2008).

2.4.4. Multimodal AR Interfaces

Multimodal AR Interfaces are a more recent type of interface compared to the previously presented types. They combine real objects with instinctive behaviors and languages, such as speech, touch and/or natural hand gestures (Furht et al., 2010). An example can be seen in the work by (Irawati, Green, Billinghurst, Duenser, & Ko, 2007), who designed an multimodal AR application that allows users to organize virtual furniture utilizing speech and paddle gestures.

2.5. Applications of AR

Throughout the years, AR technology has been applied in various domains. Table 1 contains information about several articles with contributions to AR and its many applications, cited in the section reviewing the state-of-the-art of AR applications. It was concluded that there weren't many specific studies made in the area of robotic competitions, with most of the contributions cited in this dissertation related to robotics being towards HRI.

The type of evaluation was rated based on the information displayed on this page[7].

[7] https://cyfar.org/different-types-evaluation Visited in 28/10/19

Article	Journal	Nº citations	Publishing year	Author	Keywords	Application areas	Technologies	Advantages	Limitations	Studies	Evaluation Type
A theoretical model of mobile augmented reality acceptance in urban heritage tourism	Current Issues in Tourism	44	2018	Tom Dieck, M. Claudia Jung, Timothy	Augmented Reality, Technology Acceptance Model, Dublin AR, Urban Heritage Tourism	Tourism	AR apps using GPS, AR markers	Test successfully implemented	Differences between genders, tested only with students, should be structured more carefully	Studied using a group of young female college students	Summative, Process, Impact
Head-Mounted Augmented Reality for Explainable Robotic Wheelchair Assistance	2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)	1	2018	Zolotas, Mark Elsdon, Joshua Demiris, Yiannis	N/A	People with disabilities	AR app, Hololens, robotic wheelchair	Visual support, obstacle detection	Visually confusing to distinguish the user's inputs from the wheelchair responses	Studied on 16 able-bodied participants	Formative, Process
RoMA: Interactive Fabrication with Augmented Reality and a Robotic 3D Printer	Proc. of CHI	6	2018	Peng, Huaishu Briggs, Jimmy Wang, Cheng-Yao Guo, Kevin Kider, Joseph Mueller, Stefanie Baudish, Patrick Guimbretière, François	3D printing; Augmented Reality; Interactive Fabrication; CAD; Rapid Prototyping; Physical Prototyping.	Model design	Robotic arm, HMD	Facilitates the creation of 3D models	Limited to the use of a robotic arm, issues with dealing with large objects	Studied on designers	Impact
"Pedestrian in the Loop": An Approach Using Augmented Reality	Conference: WCX™18: SAE World Congress Experience, At Detroit, USA	2	2018	Michael Hartmann, Marco Viehweger, Wim Desmet, Michael Spitzer, Michael Stolz, Daniel Watzenig	N/A	Driving	AR App, HMD	Helps understanding the behaviour of pedestrians, allows simulation before real life tests	Variations in the environment, it's not possible to know precisely the behaviour of the pedestrians in every situation	Experimented by several users	Summative

Third Point of View Augmented Reality for Robot Intentions Visualization	Springer International Publishing Switzerland 2016	7	2018	Emanuele Ruffaldi, Filippo Brizzi, Franco Tecchia, and Sandro Bacinelli	N/A	Robotics	Tracking device, Robot equipped with VGA cameras	Retrieves information about what the robot is “thinking”, integrated in a commonly used object	Wishful thinking (it wasn’t implemented), low resolution of the robots VGA camera, AR markers	Used by industry workers	Formative
Augmented Reality Driving Using Semantic Geo-Registration	25th IEEE Conference on Virtual Reality and 3D User Interfaces, VR 2018 - Proceedings	3	2018	Han-Pang Chiu Varun Murali, Ryan Villamil. G. Drew Kessler, Supun Samarasekera, Rakesh Kumar	augmented reality; autonomous navigation; depth estimation; geo-registration; scene understanding	Military driving, video-games	HMD, Georegistratio n	Representation of realistic map design designed for military training	Map quality drastically decreases in dense areas, sometimes the system isn’t able to reproduce realistic AR augmentations	N/A	Impact
ARK: Augmented Reality for Kilobots	IEEE Robotics and Automation Letters	11	2017	Andreagiovanni Reina, Alex J. Cope, Eleftherios Nikolaidis, James A.R. Marshall and Chelsea Sabo	Multi-robot systems; swarms; virtual reality and interfaces	Swarm robotics	Kilobots	Augments the robot’s visual capacities in certain environments, easy to implement	ID assignment is scalable but it’s prone to errors	Three demos were made to test the augmentations efficiency	Process, Outcomes
AR museum: A mobile augmented reality application for interactive painting recoloring	Proc. GET	1	2017	Mattia Ryffel, Fabio Zünd ,Yağız Aksoy, Alessia Marra, Maurizio Nitti, Tunç Ozan Aydın and Bob Sumner	Augmented Reality; Color Editing; Image Processing; Soft Color Segmentation	Art	AR app	Easy to use, allows interaction with paintings in a non-intrusive way	Doesn’t unmix colors the same way as the soft color segmentation	Museum attendants	Summative
Augmented Reality (AR) Applications for Supporting Human-robot Interactive Cooperation	Procedia CIRP	34	2016	Michalos, George Karagiannis, Panagiotis Makris, Sotiris Tokçalar, Önder Chryssolouris, George	Augmented Reality; Human Aspect; Hybrid Assembly System; Robot	Human-Robot interaction	AR app	Improvements in the human robot interaction during production cycles	Requires the use of both user’s hands and can compromise the assembly process, low quality hardware may lead to uncomfortable ness	Studied on humans working at a factory	Summative, Impact

Augmented Reality for Feedback in a Shared Control Spraying Task	2018 International Conference on Robotics and Automation (ICRA)	1	2018	Joshua Elsdon and Yiannis Demiris	Virtual Reality and Interfaces;Human Factors and H	Spraying	AR app, HMD, Spray robot	New and improved painting system	The results vary with each user, so the test results might not be very precise	Various experiments to test the accuracy in different modes: automatic, semi-auto, manual	Formative,Impact
The Robot Engine - Making The Unity 3D Game Engine Work For HRI	Proceedings of the 24th IEEE International Symposium on Robot and Human Interactive Communication	23	2015	Christoph Bartneck, Marius Soucy, Kevin Fleuret, Eduardo B. Sandoval	Innovative Robot Designs; Robots in art and entertainment; Anthropomorphic Robots and Virtual Humans	Human-Robot Interaction engine	Unity 3D engine,	Allows users with little to no programming knowledge to animate a robot, customizable	Only able to communicate with Arduino micro-controllers, doesn't support equipment such as joysticks, the complexity of the Unity engine for beginners, competition with apps with similar purpose, like Choregraphe.	Two case studies: LEGO Fireman and InMoov robot	Impact, formative
Augmented Reality for Industrial Robot Programmers: Workload Analysis for Task-based , Augmented Reality-supported Robot Control	25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)	12	2016	Susanne Stadler, Kevin Kain, Manuel Giuliani, Nicole Mirnig, Gerald Stollnberger, and Manfred Tscheligi	N/A	Augmented reality in industrial robot programmers	Sphero robot, AR,	The study conducted proved that, with the help of an AR interface, robot programmers comfortably completed tasks	There's the need to know the target audience before creating an AR interface	A study with 19 participants to demonstrate the effects of an AR interface and comparing this results with a non-AR approach	Summative, Outcome
Communicating Robot Motion Intent with Augmented Reality	Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot	9	2018	Michael Walker, Hooman Hedayati, Jennifer Lee, Daniel Szafir	ARHMD; Aerial robots; acm reference format; aerial robots; arhmd; augmented reality; drones; interface design; mixed; mixed reality; reality; robot intent;	Robotics	AscTec Hummingbird, Microsoft HoloLens, Unity	Study conducted to evaluate the efficiency of the fam work	The authors were able to prove that augmentations helped the participants completing the	The solution relies on a motion tracking system for precise robot localization and navigation, which means	Process, Outcomes

	Interaction - HRI '18				robots; virtuality continuum				tasks imposed in the study	that designs must be translated into the motion tracking coordinate space.	
Virtual Borders: Accurate Definition of a Mobile Robot's Workspace Using a RGB-D Google Tango Tablet	2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)	1	2017	Dennis Sprute, Klaus Tönnies and Matthias König	N/A	Mobile Robotics	RGB-D Google Tango,SLAM, ROS	Study divided in two experiments, to test 3 factors: accuracy, teaching time and correctness	Allows users to control the robot's navigation through the environment	Requires hardware that may not be accessible to some users.	Summative, Outcomes
An Augmented Interface to Display Industrial Robot Faults	Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)	No citations	2018	De Pace, Francesco Manuri, Federico Sanna, Andrea Zappia, Davide	Augmented reality; Human-machines interfaces; Industrial robots; Industry 4.0	Human-Robot Interaction	ROS, Ubuntu, Vuforia	Two studies were conducted. The first had user's evaluating a virtual robot's faults in different scenarios using an HMD. The second study followed the same objectives, but with a real robot.	Implements technology that allows workers to carry out tasks, while keeping an eye on eventual faults encountered on robots.	Issues emerged from the representation of angular speed by systems such as RVIZ	Impact
Augmented Reality (AR) Applications for Supporting Human-robot Interactive Cooperation	Procedia CIRP	41	2016	Michalos, George Karagiannis, Panagiotis Makris, Sotiris Tokçalar, Önder Chryssolouris, George	Augmented Reality; Human Aspect; Hybrid Assembly System; Robot	Human-Robot interactive cooperation	ROS, Vuforia, Unity 3D engine	A case study was performed to test the AR support tools	Successfully integrated AR technology in an automotive environment	Requires the use of the operator's hands, which leads to disruption in the assemblage process. Poor hardware quality. he use of legacy systems to retrieve data proved to be difficult to understand.	Outcomes, Summative

Reality Editor: Enabling Connection Between Humans and Machines	N/A	N/A	2019	Nancy White	Augmented Reality	Human-Machine interaction	AR	Small demo performed showcasing the application's abilities.	Works with any machine, as long as they are compatible and registered in the application	Doesn't work with certain types of machines	Summative, Impact
Augmented reality technologies, systems and applications	Multimedia Tools and Applications	702	2010	Furht, Borko Ivkovic, Misa Anisetti, Marco Damiani, Ernesto Ceravolo, Paolo Carmigniani, Julie	ar; augmented; augmented reality; augmented reality applications; augmented reality iphone4; augmented reality technologies; b; carmigniani; furht; j; reality on mobile devices; systems	Survey	AR	Survey containing the state-of-the-art of AR technologies at the time it was written	N/A	N/A	Summative
How to Safely Augment Reality : Challenges and Directions	Proceedings of the 17th International Workshop on Mobile Computing Systems and Applications	24	2016	Lebeck, Kiron Kohno, Tadayoshi Roesner, Franziska	N/A	Challenges in AR	Microsoft's Hololens, OSs	Study on different environments to view the challenges of AR applications	The authors provided their results on how different output modules perform on AR applications	N/A	Summative
Making Augmented Reality a Reality	Applied Industrial Optics: Spectroscopy, Imaging and Metrology 2017	22	2017	Azuma, Ronald T.	N/A	Challenges in AR display systems	N/A	Description of the challenges encountered in AR display methods, prominently imaging and optics	N/A	N/A	Summative
Revisiting Trends in Augmented Reality Research: A Review of the 2nd Decade of ISMAR (2008-2017)	Transactions on Visualization and Computer Graphics	23	2018	Kim, Kangsoo Member, Student Billinghurst, Mark Member, Senior Bruder, Gerd Been-Lirn Duh, Henry Welch, Gregory F	N/A	Survey	Various devices	Overview of the developments in AR technology for the last 10 years	N/A	N/A	Summative

Evaluating Augmented Reality Systems	Handbook of Augmented Reality	29	2011	Dünser, Andreas Billinghurst, Mark	N/A	HCI, AR systems	N/A	Analysis of the issues related to evaluation in AR systems	N/A	N/A	Summative
Toward Standard Usability Questionnaires for Handheld Augmented Reality	IEEE Computer Graphics and Applications	27	2015	Santos, Marc Ericson C. Polvi, Jarkko Taketomi, Takafumi Yamamoto, Goshiro Sandor, Christian Kato, Hirokazu	augmented reality; computer graphics; handheld augmented reality; information interfaces; usability engineering; virtual reality	HARUS, Point Cloud SDK	N/A	Development of HARUS, a method of evaluation of AR systems in mobile devices	N/A	N/A	Summative
Usability Principles for Augmented Reality Applications in a Smartphone Environment	International Journal of Human-Computer Interaction	57	2013	Ko, Sang Min Chang, Won Suk Ji, Yong Gu	N/A	AR application prototype	Android based smartphone	Development of new AR prototypes for mobile devices using the results obtained from heuristic evaluation and usability principles	The authors found that the prototypes gave better and more accurate results than the old methods of evaluation, which were archaic and couldn't be currently used	N/A	Summative
Evaluating the augmented reality human-robot collaboration system	International Journal of Intelligent Systems Technologies and Applications	52	2008	Green, Scott A. Chase, J. Geoffrey Chen, Xiao Qi Billinghurst, Mark	AR; HRI; augmented reality; collaboration; communication; human-robot collaboration; human-robot interaction; intelligent systems	AR application	N/A	Evaluation of an AR system applied to Human-Robot Interaction	They found the condition related to teleoperating a robot using a camera feed to be most effective method, surpassing both the other AR methods proposed	N/A	Summative

Table 1: Details about every article/website used to describe the state of the art

2.5.1. General Applications

The following examples of AR applications are more intended to give a general description of more relative unknown AR integrations and aren't related to robotics.

(Tom Dieck & Jung, 2018) aimed to present an acceptable AR model in the context of tourism through a thorough study, taking place in Dublin, as they believed that the latest mobile technologies have revolutionized the way people interact with their environment and since AR technology models in tourism never received much attention in the pasts years, with only theoretical sketches on implementing this model developed in the past.

The study was conducted by five focus groups with 44 participants, composed of young British female students each. The students were given two mobile AR apps, one was GPS based while the other used augmented reality markers. After using and trying these apps, they were asked to provide feedback on the app's usefulness and ergonomic capabilities, such as its price, justifiability in the context of tourism, quality of the interface, as well as its risks.

Although considered a successful study, there were some limitations. This test was only applied to a small minority of people, in this case, female British students, which may lead to some biased opinions, as well as not representing the general population.

AR can also be used as tool to produce enhancements in arts field. (Ryffel et al., 2017) presented a mobile augmented reality application that allowed its users to modify the colors of a painting in a non-intrusive way. The motivation to develop such app came from the interest in attracting the attention of young visitors to art museums, as well as delivering an innovative way of interacting with the paintings.

The application worked by using the soft color segmentation algorithm, which allowed each painting to be divided in different layers. The user pointed their mobile device to the desired painting and from there, they could modify the painting at will (Figure 2.15).

Figure 2. 15: Application of augmentations on a picture using a tablet (Ryffel et al., 2017)

The application was developed using the Unity 5.5 engine, with the tracking of the painting done in Vuforia 6.2.6.

Its main advantages were the easiness of use and the fact that allowed interaction with fragile paintings without touching them. Some limitations included the de mix of certain layers was not as fluid as the soft segmentation algorithm.

AR can also be a key component in the development of new features for vehicle driving. (Hartmann et al., 2018) developed “Pedestrian in the Loop”, a new AR based method of testing procedures designed to reinforce a vehicle's safety in many scenarios, mainly centred around pedestrian behavior.

Its conception came from the need of a new method of autonomous vehicle testing, given that previous and more conventional ways of testing aren’t capable of handling unexpected and realistic scenarios, thus providing insufficient data. This method provided information about the vehicle, the decisions made by a pedestrian, as well as the incorporation of real environments.

This was achieved by creating a virtual simulation of a collision avoidance scenario. The car model was made using SketchUp and rendered through the Unity 3D engine. The simulation is visualized using the Microsoft’s Hololens (Figure 2.16).

One of the main issues during development was related to the fact that pedestrian behavior is difficult to predict in most scenarios, despite the amount of data gathered from different sources (smartphones, onboard sensors).

Figure 2. 16: Demonstration of an augmented vehicle indoors (Hartmann et al., 2018)

(Chiu et al., 2018) presented a new, low-cost, AR based driving approach, compared to other techniques. Although AR is an approach being used in the simulation of driving environments for activities such as military training and video gaming, current solutions use expensive devices, like light detection and ranging sensors (LIDAR), with GPS trackers on the vehicle.

This solution resorted to Geo-Registration, the process of assigning 3D-world coordinates to the pixels of an image[8]. By replacing the costlier 3D sensors with a monocular camera, the authors were able to capture 2D video frames within the 3D geo-referenced data, which corresponds to data obtained through coordinate marking.

The AR application was developed using the Unity 3D engine, to allow the inclusion of animated models for enhancing the sense of realism. The geo-referenced data was collected using an airborne LIDAR sensor. For the military applications, the results were displayed in a monitor located inside the vehicle, while in the automotive industry, the augmented videos were displayed through an integrated heads up display. A depth quality module that allowed users to perceive augmented information when moving at high-speed was also integrated (Figure 2.17).

[8] http://www.gti.ssr.upm.es/geo-registration Visited in 8/3/2019

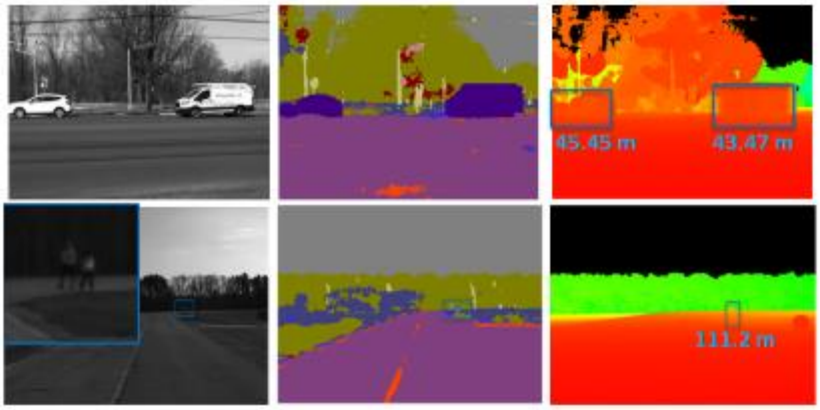


Figure 2. 17: Example of depth map usage with the system (Chiu et al., 2018)

The advantages of this method were its inexpensive approach to augmented reality driving, with the geo-registration process providing a more precise estimation of pose and the scene generating module capable of creating more realistic environments, thanks to the frames obtained by the monocular camera.

However, this system was developed specifically for the type of activities described earlier. Some limitations to more generalized scenarios include the decrease in quality of the generated map in areas with high density, which leads to a subsequent decrease in the realism presented by the produced map.

Finally, the article written by Nancy White[9] describes an interview with Ben Reynolds, a researcher at the PTC Lab, to know more about the Reality Editor. This system allowed users to "reprogram and interact with machines" using AR. The driving force behind its creation was the possibility of creating a Human-Machine Interface (HMI) that can be used by any kind of user, while setting new state-of-the-art technology to allow a smoother interaction with digital technologies".

The way it worked was by connecting a machine to the Reality Editor platform and navigating through the menu displayed by the application for the given machine. This could be done to multiple devices, as long as it was all registered in the application. Using this method, it made this kind of interactions more consistent, as opposed to more conventional methods, which include too many devices and often felt disjointed.

[9]https://www.ptc.com/en/product-lifecycle-report/augmented-reality-connecting-humans-machines?utm_source=linkedin_free&utm_medium=social_corp&utm_campaign=blog_realitylab4&fbclid=IwAR37Truh4ANKfqmgurJZ3LDI8zNvu0vi3tmRqnKcqccGM6C2ze_ZLX9f_cY Visited in 1/5/2019

2.5.2. AR in Robotics

AR applications in technology have considerably risen in the last years and they are widely used when it comes to the robotics field, whether it's in Human-Robot Interaction (HRI) or enhancements to the robots themselves. The following articles show multiple integrations of AR into robotic systems.

The work presented in the paper by (Zolotas, Elsdon, & Demiris, 2018), described an effective way of providing aid via a head-mounted device, the Microsoft's Hololens, implemented on a robotic wheel chair, to enhance the perception of users with disabilities.

The authors believed that, despite several advances in robotic wheelchair engineering, it was too soon to move away from controlled laboratory environments for modern healthcare, with one of the many challenges being overcoming the obstacle of aiding users lacking in sensor motor capabilities to navigate through environments using a standard joystick controller.

The focus was the implementation of a shared control system, a system capable of combining motor commands generated by both the user and an intelligent system, without hindering the operator's ability to navigate the wheelchair. To this end, they created a system composed by the shared control system mentioned previously, a grid map processing and visualization for assistive feedback.

Through the shared control method, the robotic system could project its forward state while being validated against an obstacle map. The grid map sent a reconstructed image visually to the user, with a map containing harmful obstacle in the environment. Finally, the assistive feedback was formed by the Microsoft's Hololens, which provided visual information on the robot's dynamics, such as grey and green arrows, representing the robot's target direction and the user input, respectively.

The main hardware component of the project, the robotic wheelchair, was developed internally. It is controlled using a joystick with a circuit board that enables an Arduino UNO to translate the user issued commands into motor signals. An on-board laptop was used as the main driver to control the wheelchair and all the processes included in the ROS framework. The Unity game engine was used to develop the AR application and to deploy it on the Microsoft's Hololens.

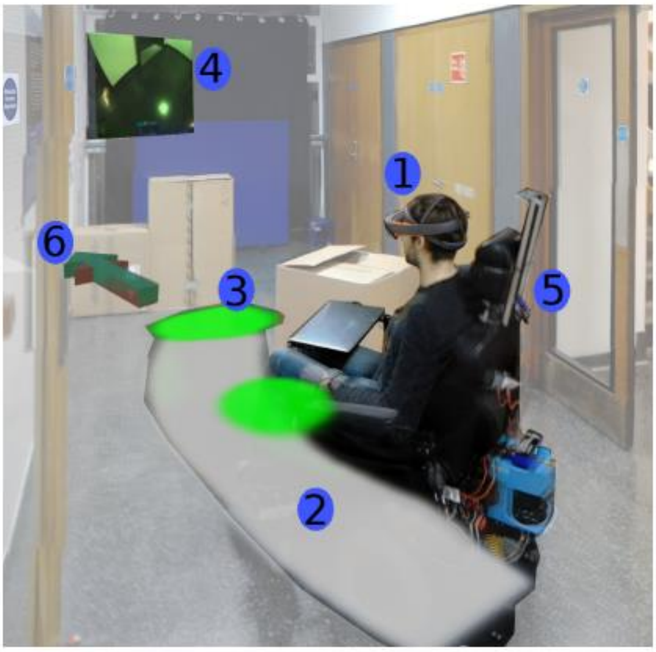


Figure 2. 18: Overview of the augmentations experienced by the user while operating with the system. 1) HMD module; 2) Trajectory generated by user input; 3) Potential collisions; 4) Rear view display; 5) Rear camera; 6) Green arrow: user raw input; Red arrow: Corrected input (Zolotas et al., 2018)

To explore the effectiveness of the visual aids, as well as checking the benefits of an AR interface on learning the wheelchair's patterns, a test was conducted. Users were aged between 20 and 31 years old. Two groups were formed. One had access to the information through a Hololens, while the other had no means of visualization. All subjects were requested to complete a "navigation route (Figure 2.18) four times in sequence" (Zolotas et al., 2018). The test's main purpose was to observe the improvements AR provided to the user experience while learning the wheelchair's management. The group of subjects with access to the HMD wasn't given any explanation on how to depict the visual representations during the course and were asked to perform one last lap without the HMD to study the task-learnt skills independency from visual cues (Zolotas et al., 2018).

According to the results obtained, the performance during the trials of the group without any visual aids was overall significantly better than the other group, reaching their skill ceiling by the third trial, while the other group took longer to achieve the same level of skill. Also, by the end of the test, the participants were asked to fill a questionnaire related to the advantages of each of the visual representations provided to them during the test. Most of the participants disliked the obstacle markers and the grey path outline due to being barely noticeable from the user's perspective. The most positively rated feature was the rear-view display augmentations.

This project constituted an early attempt at implementing an AR interface into a robotic wheelchair system, in which the authors concluded that there's still room for improvement by solving one of the main limitations of this system, such as the placement of certain AR elements to make the environment less busy.

(Peng et al., 2018) proposed RoMA, an interactive AR modelling experience. While there were other interactive fabrication models, each with its advantages and drawbacks, the driving force behind its conception was creating a fabrication device capable of integrating design and fabrication at the same time.

This was achieved with the integration of a robotic 3D printing arm capable of rotating 360º (Figure 2.19). The designer wears an HDM, an ORVision stereoscopic camera connected to an Oculus Rift VR headset, to see the augmented design and control the robotic arm inputs through modified versions of the Oculus Rift controllers. To hold the model, a rotating platform was used. OpenGL and OpenVR were used to render the scene displayed on the HMD used by the designer.

To test this system, a designer performed a series of actions, such as designing a teapot from scratch using the available primitives, adding new components to existing objects and creating new layers from previously used primitives.

There were several limitations encountered when testing the RoMA technology. The main constraint found was the use of a robotic arm, which led to several downgrades, such as the printing speed, in order to ensure the users safety. This also hindered the integration of RoMA as a future 3D digital modeling method. The other limitation included issues in dealing with large objects, due to the small size of the rotating placeholder use to create models.

(Brizzi et al., 2018) proposed a new way of visualizing robot intentions using AR, as its non-intrusive impact on a worker compared to the other conventional methods used, such as display panels on the robot or HMDs, coupled with its intuitive nature could severely decrease the complexity of certain tasks.

The foundations of this work were the several advances in the robotic technology that have made possible the cooperation between human workers and robots in the industry, like factory maintenance. This kind of tasks involving cooperation cover the necessity of the operator understand the state of the robot and its intentions, as well as the robot need to understand the operator decisions, to ensure both the safety of a worker and the supervision of the machine.

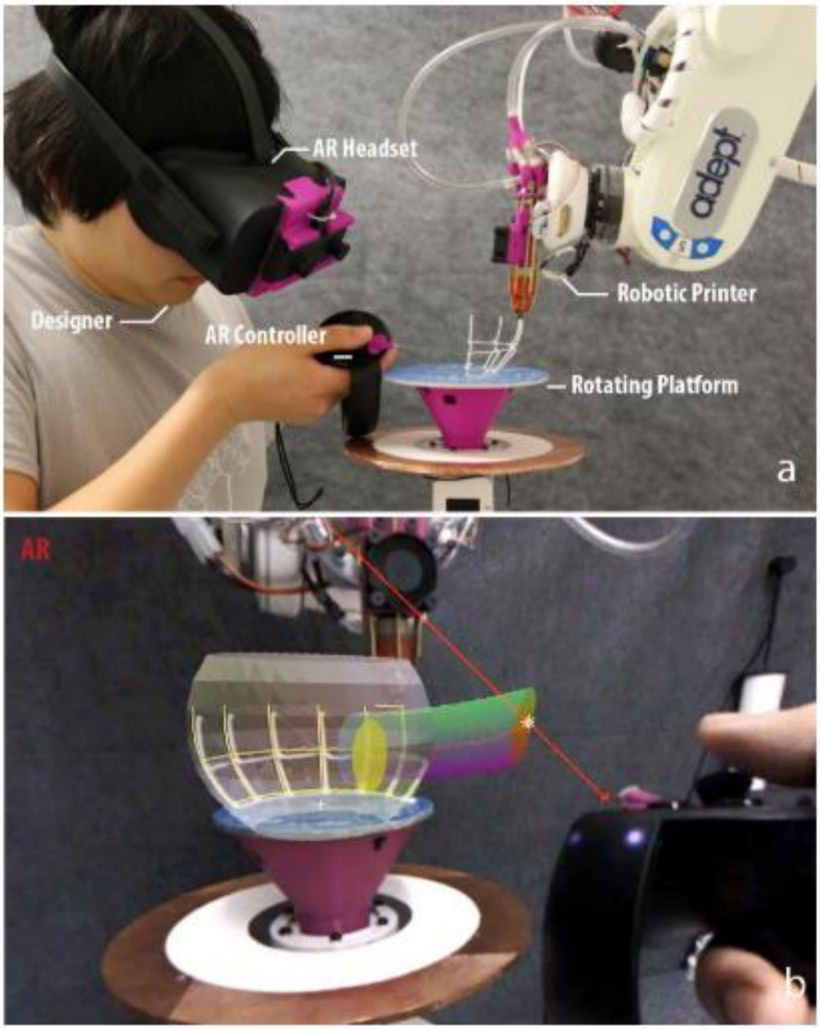


Figure 2. 19: The RoMA system (Peng et al., 2018)

To this end, it was created a device specifically built to fit the standard worker helmet. This device's display module was composed by two LCD displays with LED retro-illumination and its computer vision is made using full-HD cameras, scaled down VGA resolution. It was also equipped with a 6-DOF motion sensor.

The robot, named Baxter (Figure 2.20), included two VGA cameras, one at each tip of its arms, augmented with functionalities similar to a Kinect 360. It operated through an internal computer running Robot Operating System (ROS)[10] and the connection between the device and the robot was made using the CoCo framework for Mixed Reality (Ruffaldi & Brizzi, 2016).

[10] https://www.ros.org/ Visited in 27/10/19

However, this idea was presented as a wishful thinking, meaning that the only information available were sketches for future implementation, with no AR technology implementations.

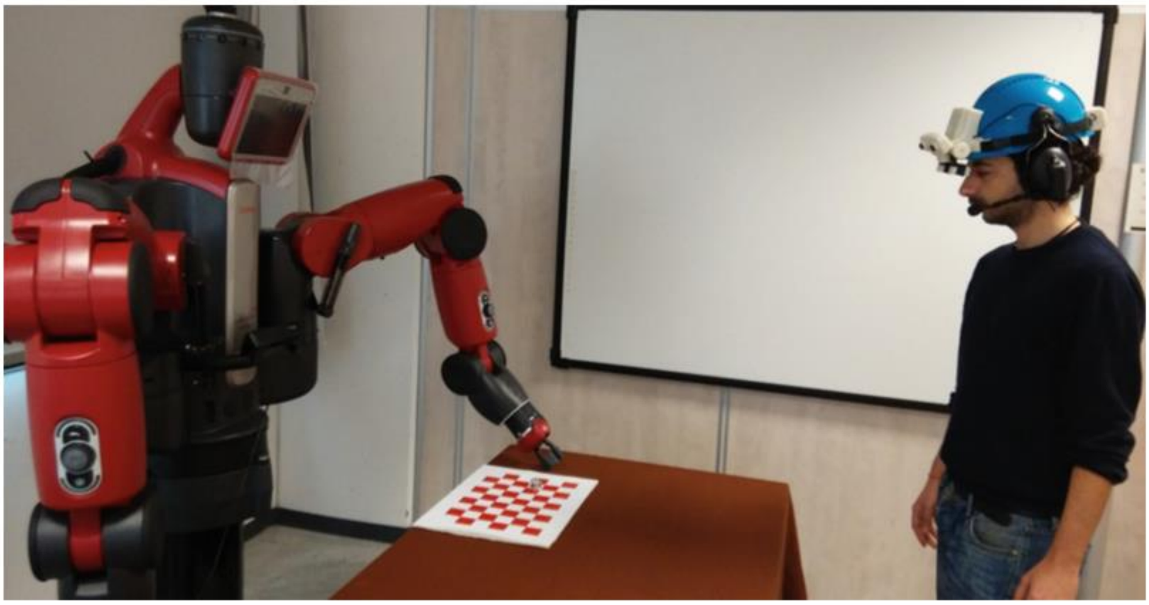


Figure 2. 20: The Baxter robot, alongside a worker wearing the developed HMD (Brizzi et al., 2018)

(Nikolaidis, Cope, Reina, Sabo, & Marshall, 2017) proposed an AR system with the goal of increasing the usability and flexibility of Kilobots, model of swarm robots.

These were robots designed to study environments, through calibration, sensing and tracking processes. While Kilobots presented a relatively cheap solution to eventual issues with such processes, these however, were limited by the range of their capabilities due to its inexpensiveness.

The system developed was called ARK (AR for Kilobots) and was composed by three main components. The base control software structure organized the communication, tracking methods and interactions with virtual environments. The communication between the Kilobots and the base control software was made using infrared signals and finally, the tracking of Kilobots was achieved using four overhead cameras, which sent the retrieved information to the base station. The image visualized by the cameras was merged into a single one using OpenCV, as well as locating each one of the Kilobots.

To demonstrate this approach results, several demonstrations were made. The first demonstration was created with the purpose of showing the ARK's capabilities of retrieving information from the Kilobots wirelessly and in parallel communication, by using the swarm robots to display different combinations of numbers from a random interval. The second demonstration included a more artistic performance by the Kilobots to organized themselves to create a word, with the help of several messages sent by the ARK system and a demonstration of the capabilities of sensing a virtual environment. The final demonstration served to demonstrate the robots “thinking” using AR (Figure 2.21).

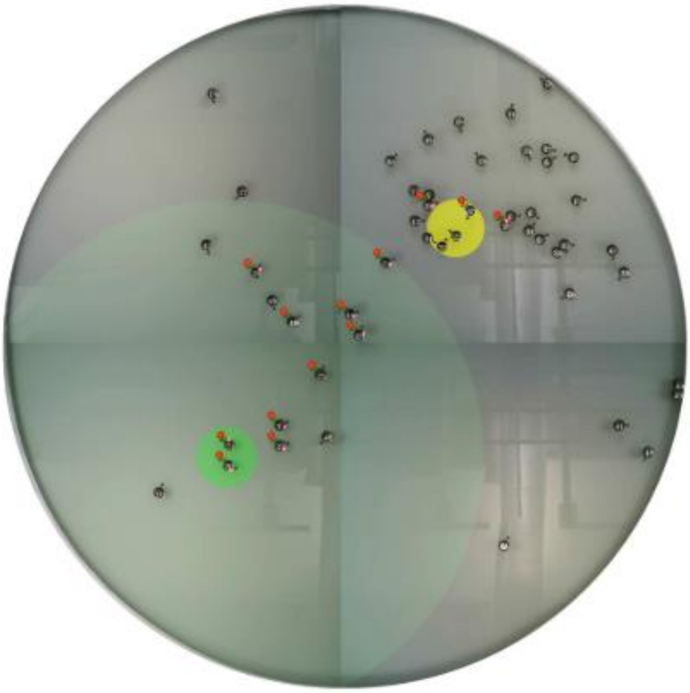


Figure 2. 21: Demonstration of the third experiment (Nikolaidis et al., 2017)

It was established a starting area, in which the Kilobots picked up virtual load units, and a destination area, the drop-off point of the virtual load. Each time a robot picked up a unit, it would signal by lighting its LEDs.

Although this approach effectively increased the swarm robots’ capabilities, some of its functionalities, such as the ID assignment, proved to be scalable at the cost of small errors in implementation.

(Sprute, Tönnies, & König, 2017) introduced a way of limiting a robot’s movement to a certain area, while giving the user feedback using a mobile device.

They saw the increase on robot usage at people’s homes, mostly cleaning robots, which usually perform tasks around the house and felt that sometimes there’s the need to restrict the areas which the robot has access to, mostly due to privacy. Previous work in this area allowed to effectively constraining the robot to certain areas, but they often didn't provide a feedback system.

This solution utilized a Google Tango device, an RGB-D tablet, due to its high accuracy and 6 Degrees of Freedom (DoF) supported, as well as an unspecified AR app (Figure 2.22). To use the application, the user was required to move the mobile device and mark points that when united, formed a virtual border. The table could also act as a mean to visualize the information through an on-board camera.

To test this method efficiency, it was created an evaluation process based on three metrics: correctness, accuracy and teaching time. All the methods were implemented as a ROS package and a map was previously developed using a SLAM algorithm based on a particle filter.

The authors organized a study, to compare the proposed teaching methods with methods that allow the integration of virtual borders with a map previously known, using markers or pointers.

The first experiment was performed by fifteen people and their goal was to define the area the robot is not supposed to go utilizing the app. The second experiment consisted on testing the application on self-developed datasets of maps. Both tests evaluated how accurately the user defined virtual borders are transferred to the system, how long does the process of teaching a user how to create a virtual border last and, ultimately, how effectively is this method on changing the robot's trajectory. The outcomes showed that the accuracy was similar to the guideline methods, although the teaching time saw a significant reduction.

Figure 2. 22: The Virtual Borders application working. The area which the robot can traverse is delimited by the coloured lines (Sprute et al., 2017)

One of the main limitations the authors found about this method was the fact that it requires expensive external hardware, such as the Google Tango, which might not be accessible to certain users.

(Elsdon & Demiris, 2018) took a new direction in the usability of spray robots, to increase the user's awareness when employing a task with such robots, by developing an AR application with a shared control method to allow important information to be transferred to the agent.

They found that the use of this type of handheld robots has been advantageous for the fact that it allowed the user to move freely, while allowing the robot to complete its task without repercussions. This was usually the case in spray painting structures, where both user and the robot needed to collaborate. However, the motivation came from the need of having the robot understanding the status of a task for this collaboration to be successful.

The authors didn't use defined markers, or any calibration related to the user component. The Microsoft's Hololens were the main component of the system, which allowed the user to visualize silhouettes of the area that needs to be painted (Figure 2.23). The spraying region also displays three colors, each one indicating the state of the painting. Blue, if the area was under sprayed, green if it had the correct amount of spraying and finally red, if the user had "oversprayed" the area.

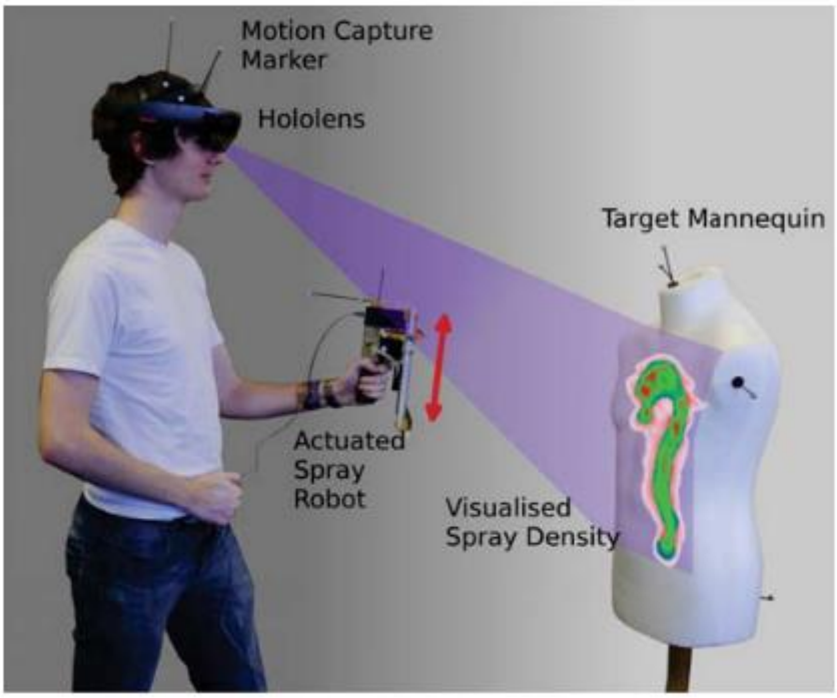


Figure 2. 23: Experimental design of the work (Elsdon & Demiris, 2018)

Several tests, with 18 participants, were run to ensure the application efficiency. The first test had the purpose of studying the ability of the system to find the correspondence between the frame generated by the Microsoft's Hololens and the world frame. The results found that the error of the correspondence depends on each user. The other test focused on the accuracy, completion time of a task and the task load, using different types of spraying: manual, semi-auto and auto. Results of this experiment showed that, for the task load, the increasing in automation helped to reduce the task load. For the completion time, manual achieved the better results, with little difference between the other two modes. Finally, for the accuracy, the automatic configuration showed the best results while the manual configuration displayed the worst.

The main limitations encountered during the development of the application were calibrating the Microsoft's Hololens and the motion capture without the use of markers or precise alignment.

(Walker, Hedayati, Lee, & Szafir, 2018) provided a new alternative to help Human-Robot Interaction (HRI) by taking an approach based on AR.

The author's motivation for this project was to provide a hands-free solution to help workers who work simultaneously with robots feel more secure and improve their task performance by allowing them to see what the robot's intentions are when it comes to motion.

They stated that there were several robots which, due to their design, it was difficult for them to convey their intentions. Although there were already many methods used to distinguish robot's decisions, such as the use of projectors to display its decisions, these approaches often came with major downsides. In this case, projections might obstruct the view of the workers in certain scenarios, or in some situations, the projection might be difficult to deliver on irregular surfaces, which could lead to workers question a robot's usability and decrease safety at work.

A framework was created, which characterized different approaches to integrate AR with HRI. The first, augmenting the environment, aimed to use the shared working environment as the point of reference for the augmentations. Next, augmenting the robot, referred to connecting visual information directly to the robot's platform, which made it a very cheap alternative to adapting a robot to a certain environment. Lastly, augmenting the user interface, consisted on giving a more realistic interface to the user, which might include features such as mini maps for knowing the robot's pose.

In terms of technology, the robot platform used was the AscTec Hummingbird. For the ARHMD, the Microsoft Hololens were used, with multiple designs prototyped using the Unity 3D game engine.

An experiment was conducted, in order to evaluate how users cooperate with a flying robot through the framework. Sixty participants were hired for the study. The aim was for each participant to make beaded strings within eight minutes. These were in containers put in different workstations and participants would walk up to the workstation and collect a bead. This environment was shared with another flying robot, which checked each of the stations frequently (Figure 2.24).

This study successfully proved that augmentations, such as arrows, improved the performance of the users during the task. They were able to predict more easily the robot's movement and plan their own actions accordingly to the robot's intentions.

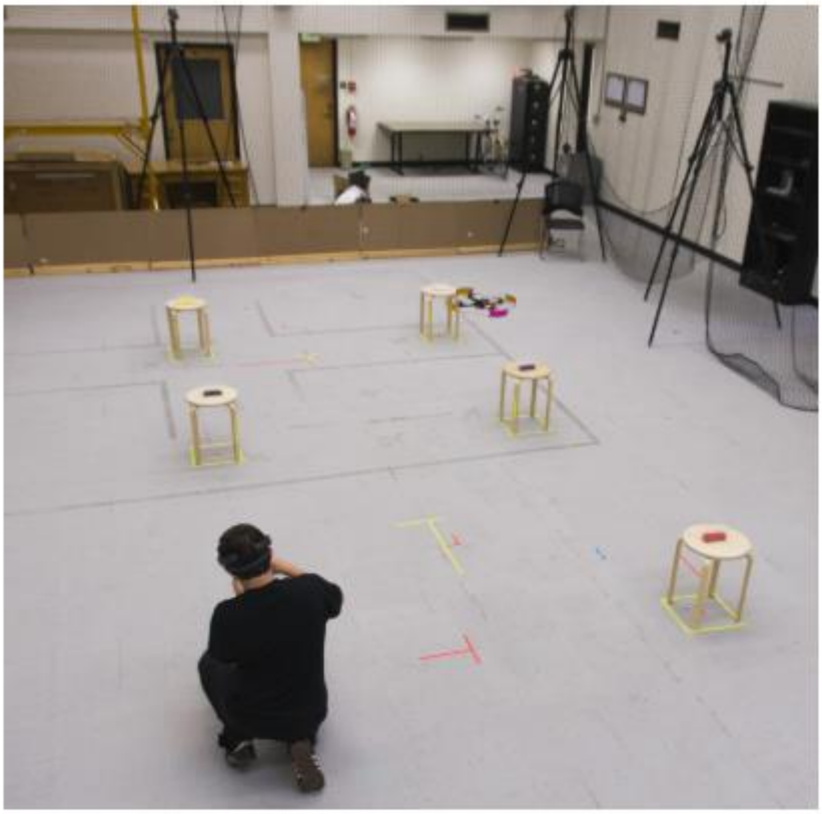


Figure 2. 24: Overview of the environment in which the users performed the test (Walker et al., 2018)

Although the study proved that the framework was well implemented, it didn't come without some limitations. This solution relied on a motion tracking system for precise robot localization and navigation, which meant that designs must be translated into the motion tracking coordinate space. Also, the design, task and measure choices used in the experiment might not be applicable to other scenarios.

(Bartneck, Soucy, Fleuret, & Sandoval, 2015) took a new approach in HRI and created a plugin named TRE (The Robotic Engine), a plugin developed and based on the Unity 3D Engine, designed to facilitate animating and controlling robots visually.

The authors asserted that, while HRI had always been a topic pushed by the combination of multiple professional specializations, one of the main issues encountered in HRI was the lack of technical knowledge from the less specialized members of a multidisciplinary team to program more advanced robots. There were already several tools to help the development of a robot by complementing the absence of programming skills with easy-to-use tools, such as the LEGO Mindstorms and the NAO platform, although they proved to be too limited or complex for a non-programmer to learn.

As previously mentioned, the TRE was developed using the assets of an already existing Game Engine. The Unity 3D was used, since it integrated fundamental features of HRI, namely an optimal interface for animation purposes and the possibility of communication with external devices, as well as the engine’s surrounding community.

Its architecture consisted of four scripts. The *MainScript,* as its name implies, was the main component and served two purposes: controlled some of the other script’s functions (e.g. activating and deactivating services such as the computer vision module) and established connection between the modules and Unity’s GameObjects.

Two case studies were conducted for testing purposes. In the first case (Figure 2.25), the goal was to control a LEGO fireman robot through TRE. First, most of its internal components were removed and replaced by servo motors and the robot was confined to an Arduino board and speaker. Next, a virtual representation of the robot was created using Blender.

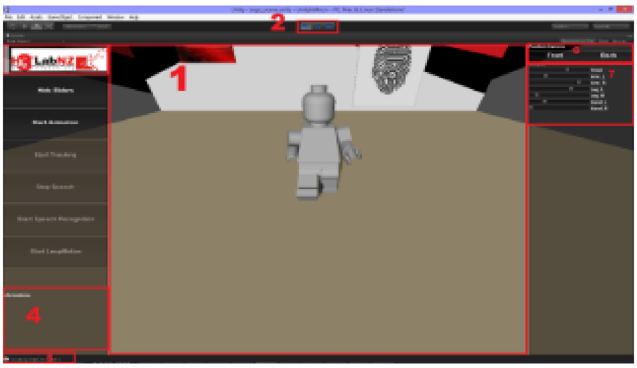


Figure 2. 25: The LEGO Fireman project in Unity 3D using TRE (Bartneck et al., 2015)

The second case utilized the components of an InMoov robot, an open source hardware robot. The authors printed all the robot parts, since they were available to download for free, and built their version of the robot. Similarly the previous case, the TRE was programmed to control the virtual representation of the robot and finally, it was possible to control the InMoov using Unity (Figure 2.26).

Figure 2. 26: The InMoov Robot (Bartneck et al., 2015)

Several limitations of the project included the fact that TRE was only able to communicate with an Arduino micro-controller, didn't support equipment such as joysticks, the complexity of the Unity engine for beginners to learn and the competition with apps with similar purpose, like Choregraphe.

Continuing analysis on HRI, (Stadler et al., 2016) aimed to explore an AR approach to improve industrial robot programming and how this medium influences a worker's performance while completing tasks.

They acknowledge that robot programming interfaces were going through several changes, in terms of technological advancements, such as the emerging of new types of robot, as well as economical changes. As such, programming robots became a rather complex task, where the code needed to be altered or updated whenever new robot configurations come out, something that wasn't done immediately, as well as being perceived by many as an increase in the workload.

By conducting a study, the authors carefully selected an AR interface for task-based programming and supervised it to observe the impact of AR on the robot programmers during the guidance process and measured the total workload of the user.

This study was performed by 19 male participants, within AR scenarios and Non-AR scenarios. All participants had previous background knowledge within the robot programming field. The participants were divided in two groups: one was composed by the participants that had never used AR technology in practice while the other was composed by the participants that had experienced AR in first-hand.

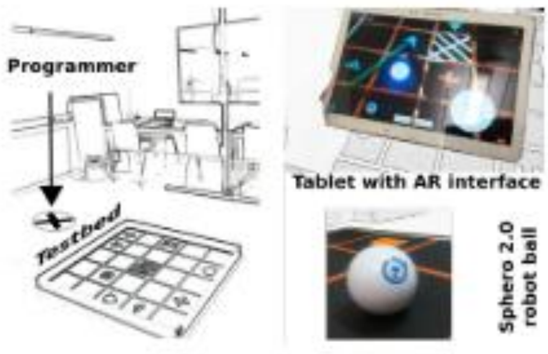


Figure 2. 27: The architecture of the experiment (Stadler et al., 2016)

The robot used was called Sphero, a spherical robot able to move in every direction, which was found to be suitable for the study (Figure 2.27).

The AR interface was developed in the Unity 3D engine, using the Sphero command library and the Vuforia AR library and the mobile device presented to the users was Samsung Galaxy Tab 4. The environment used for testing was the environment used to control the robot.

During the study, the users were asked to complete several tasks (trajectory teaching, overlapping) which they were taught the basics before completing them. The AR sessions were arbitrary and at the end of the study, the users were asked to comment on the use of the AR technologies.

The results of this study showed that AR support helped to improve the workload perception on the participants. Nonetheless, it was concluded that the target audience must be carefully selected, and the design process goals must be clear in order to create a successful AR interface.

The work by (De Pace, Manuri, Sanna, & Zappia, 2018) described the creation of an AR application devised to help workers to visualize and detect errors or faults that might occur to robots while performing task in an industrial environment.

Due to the industrial environment having become very automated, where many robots have replaced human workers in repetitive and automated tasks, workers were able to handle more thorough tasks. As previously described, it was found that these conditions usually created an HRI scenario and in such environments, there was the need to create a trusty relationship between the workers and the automatons. AR became a useful tool in this type of environments, as it allowed workers to see the robot's trajectory in context with the real world. However, they found that HRI "was a concept not fully explored yet and faults in the robots might lead to workers having to analyse other information, an area that nevertheless lacked research" (De Pace et al., 2018).

The application proposed replaced the slower and more archaic method of fault detection previously used, with each of the faults categorized and represented through different 3D icons. It was developed using Unity 3D engine and the Vuforia library, with the Kinetic version of ROS (Robot Operating System) serving as a server to send data between the robot and an Android device.

The authors performed some studies to test the application usability. The participants were asked to analyse different scenes and identify the origin of the faults (Figure 2.28). Two were visualized with Moverio AR glasses, while the other two made use of mobile devices. The participants were also asked to fill a questionnaire regarding information, ranging from the user's awareness regarding robotics and AR to the expressiveness of the augmented information conveyed during the experiment. The results indicated that, although it was experimented by a limited selection of participants, generally the users were able to understand the cause of the faults.

Further, an additional test was decided to be performed using a real robot, with the intention of studying the variances between the analysis of a real and a virtual robot. The robot used was an InMoov robot (Figure 2.29). The Android devices were kept the same and ten new participants were called. The results showed that it made little difference when it came to use the application to interact with a real robot and users generally preferred to visualize through the Moverio glasses.

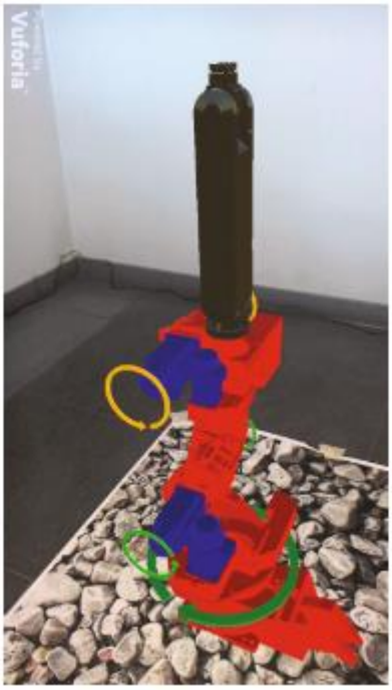


Figure 2. 28: Scene 1 test scenario (De Pace et al., 2018)

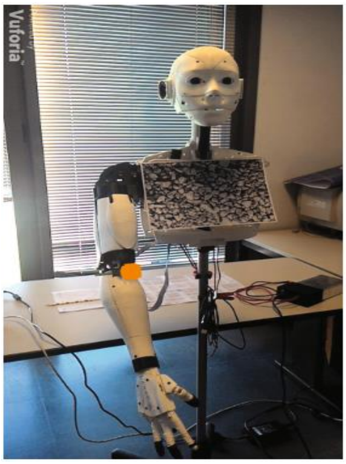


Figure 2. 29: The InMoov robot model used for testing (De Pace et al., 2018)

Finally, (Michalos, Karagiannis, Makris, Tokçalar, & Chryssolouris, 2016) developed an AR tool designed to help workers in an industrial environment with robots.

Their analysis showed the significant improvements in both AR and VR technology over the last few decades made possible the creation of tools applicable to areas such as manufacturing. These activities were usually made by industrial robots and usually, humans only take part during the production process. This made the basis behind the motivation of this paper, as it was believed that cooperation between humans and robots could greatly improve small scale production environments where flexibility plays an important role.

The AR tool was created with the goal being the increase of the operator's safety when collaborating with robots. Such tool was also designed to fit the standard industrial requirements.

The first main functionality was designed to give the operator visualization of all components and their respectively order of assemblage. This was done with the aid of 3D models. The second functionality allowed the operator to visualize through a headset the safe areas of a component when it was cooperating with a robot. Interference areas and the trajectory of the robot were all the information that the operator had access to (Figure 2.30). Through the headset, the user could see warning messages, which indicated various status of the robot, such as robot motion, emergency stops during production among others. Additionally, the robot was also capable of sending information about the production status and completion date.

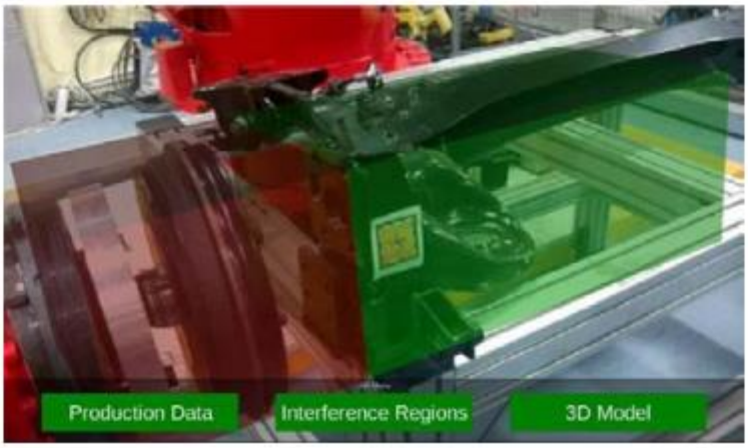


Figure 2. 30: Example of usage of the application. The green area indicates the safe zone, while the red area indicates the robot is operating in that zone (Michalos et al., 2016)

To create the application, the Unity 3D engine and the Vuforia AR library were used. For higher adaptability, it was decided that communications between the modules would be done through ROS. Therefore, to handle the ROS topics, it was used the JSON Library. The hardware technologies include an Android tablet, a MySQL server and a laptop running Ubuntu OS.

They concluded that the application was successfully implemented and through experiments, it proved to be not very complex to use. However, several limitations were found. This application required the use of the operator's hands, which could lead to disruption in the assemblage process. The quality of the hardware also hindered the application performance, since it was a marker-based solution. Also, the use of legacy systems to retrieve data proved to be difficult to understand.

2.6. Evaluation in AR

Like every other product on the market, AR systems need to be evaluated and tested before launch. There are several evaluation methods commonly used for usability evaluation of interactive systems. Such a method is Heuristic Evaluation based on the use of Nielsen's usability heuristics[11], which are 10 general usability principles. Usability tests and Controlled experiments are also common methods and have been used to evaluate AR systems. However, more exhaustive methods of evaluation may be needed to evaluate this type of systems, given their subtle nature.

2.6.1. Challenges and Proposed Methods

According to (Dünser & Billinghurst, 2011), most of the usability tests applied to most products, such as the heuristic previously described, weren't often considered when evaluating AR applications. These might use other techniques besides the more conventional usability tests, such as the "think aloud method and heuristic evaluations" (Dünser & Billinghurst, 2011). Being a relatively new addition to the research in AR systems, these test results might be influenced by user's lack of knowledge or by the many different hardware and software implementations.

Starting with evaluation approach of traditional methods, the consensus among developers was that, AR evaluation couldn't solely rely on them, due to the limitations of usability tests, like point and click interactions, which didn't fit in virtual environments. Another important aspect referred to was the lack of support by the "Nielsen's heuristic related to localization and manipulation of objects in a 3D environment" (Dünser & Billinghurst, 2011).

Other methods utilized in AR evaluation were prototypes build for a one-time usage. These were created by researchers to provide developers with a proof of concept implementation of a feature and to test issues related to them. Although they provided

[11] https://www.nngroup.com/articles/ten-usability-heuristics/ Visited in 10th July of 2019

multiple ideas on how they should move forward with the implementation, these were often unrelated and therefore, couldn't be put together in an evaluation test.

The clear definition of the users which would use the product, as well as knowing their needs was also an important factor. By deciding which were the users a certain system targets, developers could get more scope on which paths to take during development, so that when user tests are required, they know who they should experiment with, to get qualitative feedback.

Finally, the various components in an AR system were still too distant from each other, making it hard to create a framework, abstract enough to allow it to be adapted and modified according to each project, while still allowing the development of standard procedures.

(Santos et al., 2015) created the HAR Usability Scale (HARUS), a method of evaluation for HAR (Handheld Augmented Reality) systems in response to the lack of consistent and valid tools for the evaluation of HAR. It consisted of two questionnaires with each focusing on different evaluation aspects.

Its objective was to merge the different types of issues that usually arise when interacting with these systems, such as depth perception or tracking issues. The development was divided in two phases: The first phase dealt with the questionnaires design. The surveys were composed by 16 statements, where users had to give feedback on how much they agreed with them and the total score obtained by the questionnaires would give the result of HARUS. The second phase was used to validate how HARUS performed during several experiments.

The first task consisted in users writing down annotations on real objects using an HAR tool. The second task allowed users to view already written virtual notes on real life objects. In the third task, it was evaluated a "HAR system for memorizing Filipino vocabulary words in a real environment" (Santos et al., 2015). The final tasks was manipulating the "position of a virtual object in the real environment" (Santos et al., 2015). For each of the experiments, the scores of both the System Usability Scale (SUS)[12] and HAR Usability Scale (HARUS) were generally positive, although the last task showed that users with more difficulties in completing the task were more likely to give a lower usability score at the end.

[12] https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html Visited in 27/10/19

To conclude, HARUS proved to be a consistent method of evaluation, according to the value of the Cronbach Alpha.

2.6.2. Usability principles of AR

To face most of the challenges imposed during the evaluation of an AR system, researcher created a set of foundations to be used to study how their system interacts with users. This is an important analysis, since it gives developers feedback to improve their systems and establish a scope for the project. Next, two works on this topic are presented.

(Ko, Chang, & Ji, 2013) developed new usability principles to aid the evaluation of AR applications in mobile devices. After creating usability tests and defining a heuristic evaluation, they created a prototype to see the results of their usability tests. The process was divided by phases. In the first phase, they collected usability principles from already exiting studies to help them form new usability tests. An experiment was conducted to make an agreement on how a group of user guidelines should be classified, by picking many usability principles, and then pick a certain number of them to differentiate the relation between each one of them. They were classified into categories. The first one encapsulated the principles used to give users information like visual information. The second one referred to the principles used to measure the way the user interacts with the application, to reduce the overall learning curve of its usage. The third category covered the principles that provided the user information about errors, management and personalization (Ko et al., 2013). The fourth englobed any interactions between the user and the application. The final category was related to the principles surrounding the usage of the application.

These principles were tested on 3 AR applications that use the user's location, to determine the direction of smartphone AR application and to see how the applications would perform. The heuristic evaluation focused on a set of problems that were gathered from it. The most common issues were found when users dealt with devices handed or "too much information was displayed on a small screen" (Ko et al., 2013).

Also, a prototype was developed with the results obtained from the heuristic in mind. Then, they conducted a test involving the completion of two tasks, with participants that had previously tried AR applications, not only involving the prototype developed, but also 2 more applications, one in iOS and the other in Android (Figure 2.31).

Tasks for the Usability Testing

Task 1	Search the nearest () from here. Then input text of the name and distance.
Task 2	Find () and make a phone call after confirming the location on the map by using the detailed information.

Figure 2. 31: Tasks involved in the usability test (Ko et al., 2013)

From the results, it was concluded that the prototype had the lowest completion time for a task, with users preferring it over the iOS platform, which took almost double the time to complete a task.

Although these usability principles proved to be useful, there were some limitations when applying them. The heuristic data was obtained from apps that didn’t offer English language support, the prototype was based on Android OS in which usability tests may not transition well into other OSs, like iOS. Also, the screen size and the type of applications that the principles were tested didn’t reflect every possible mobile device or AR system available.

(Grandi, Debarba, Nedel, & Maciel, 2017) created a new AR based 3D user interface designed for collaborative object manipulation using mobile devices. To test their application, they created several tasks to see how much effect grouping has in object crossing. One of the tasks, which consisted in crossing objects of different dimensions and orientation within a tunnel, was used to for evaluation, where they measured the time that took each user to complete the task, how accurate they were and the number of transformations (rotations) made during the task (Figure 2.32).

Figure 2. 32: Demonstration of the task and the checkpoints used to test the accuracy of the users (Grandi et al., 2017)

The evaluation was made using a characterization form and, after giving each user a mobile device, which had to be recalibrated, they started the test. The authors created several hypotheses that they were able to confirm as the experiment went on. At the end of the task, each user answered to a post-experiment questionnaire.

Additionally, the authors created 4 different hypotheses to see if the implemented interface would provide any significant advantages in collaborative work (Grandi et al., 2017).

They were:

- “Groups with more than one member complete the tasks faster”
- “Groups with more than one member complete the tasks with more accuracy”
- “For the tested group size range, if groups increase in members, the time to complete tasks drops proportionally”
- “For the tested group size range, if groups increase in members, the accuracy to complete tasks increase proportionally”

To check the accuracy of these hypothesis, the authors oversaw a Kruskal-Wallis, which could “determine if a statistically significant difference existed between the levels of group size without assuming residuals to follow the normal distribution.” (Grandi et al., 2017)

When analysing the impact that group size had on completion time, hypothesis H1 and H3 were scrapped, due to the negative outcome of the previously mentioned test. On the other hand, the increase in group size had a positive outcome in the Kruskal-Wallis test, which confirmed the hypothesis H2 and H4.

2.7. Challenges & Opportunities of AR applications

Although AR applications have been created for several years, AR itself is still considered a relatively new technology. Some improvements are aimed towards the use of HMDs and other visualization tools, such as virtual retinal displays, and a focus on constructing controlled environments using sensors or actuators (Furht et al., 2010).

(Furht et al., 2010) discussed many possible AR applications for near future implementation, since they believe that current mobile devices underutilize this technology.

AR's blend of reality and virtual objects characteristics can provide a tool that might be used as a means to enhance missing senses for certain users. For example, hard-of-hearing users could receive visual cues informing them of lacking audio signals and sightless users could receive audio cues to alert them of unknown visual events.

A key factor that allowed the rise of AR applications was the social and ethical acceptance criteria. For example, AR applications need to be non-intrusive and subtle, while providing a learning curve to its users. The cost of this kind of application was also a factor that comes into consideration in areas, like medicine.

Dangers and more ethical concerns raised when dealing with applications that could detect and recognize people (Furht et al., 2010), since it could be done against their will and there's the risk of this type of information being leaked into the internet. Additionally, there needed to exist a sense of realism when it comes to exploring potential upgrades in such applications, meaning that, AR applications were tools designed to aid or ease a user's task and shouldn't in any way disrupt their daily activities.

(Lebeck, Kohno, & Roesner, 2016) commented on the risks of AR, stating that AR applications might be faulty or designed with mischievous intents. Most applications of this kind had access to sensor inputs, which, in most cases, weren't controlled by neither the user nor the OS, which might lead to security and privacy concerns. Other commonly used devices such as HMDs, might show more susceptible information without being properly regulated. In terms of output, there were multiple concerns arising from the various devices referred before. For example, it was possible for an outside source to override the content displayed to the user, possibly distracting or endangering the user. For their new output module, they considered several fronts where it might occur the two assumption referred previously (Lebeck et al., 2016):

- "Who displayed certain content"
- "What kind of content can be displayed using the application"
- "The situations when content is displayed"
- "Devices which the application can interact with and display content"

They concluded that current AR systems available hadn't still fully explored security and privacy measures and while it's an important step during the development of such systems, it mustn't hinder the application's capabilities.

(Azuma, 2017) made an examination on the technical obstacles in imaging and optics confronted in near-eye optical see-through AR display systems. Regarding imaging, there was the need to establish a semantic relation between the AR systems and its surroundings to maximize the effectiveness of such systems. This was still a feature that the majority of the current AR systems didn't implement, despite the recent advancements in computer vision on those systems, since they "only recovered the geometry and not the meaning of the objects in the surrounding environment" (Azuma, 2017). Using smart imaging through platforms such as Google Tango, it was possible to incorporate depth sensing used to improve smart imaging.

Concerning the optics, the more recent contributions to this area focused on optical see-through head-worn approaches, due to being little intrusive in the user's view of the real world and optical displays allow the user to keep in touch with reality, despite eventual power shortages. However, several challenges elevated when developing this kind of device. For example, finding a universally acceptable eyewear format, such as sunglasses, and fit the capabilities of these displays on such a small form factor came off as a difficult task. Interactions with other people while using these devices are also important. Currently, "many optical combiners route some of the displayed light away from the wearer's eyes, out toward other people in the environment" (Azuma, 2017), which in return concealed the user's eyes. Other discussed issues included eyestrain caused by the displays in situations where the user was required to switch looking between close and ranged objects, due to their "fixed focal distance" (Azuma, 2017), and the small field-of-view, which might hindered the type of usage intended by some AR applications.

In terms of evolution of AR, as reported by (Kim et al., 2018) , the results of the study that spans over a decade of showcases and evolution in AR presented at the renowned International Symposium on Mixed and Augmented Reality (ISMAR) conference, showed a decrease in the calibration and display areas of AR in the past 10 years, while rendering and evaluation areas showed an increase research compared to the previous decade.

Speaking of evaluation of AR systems in robotics, (Green, Chase, Chen, & Billinghurst, 2008) compared the performance of three different types of interface of an AR system, designed to aid users during interactions between robots and humans. The authors conducted a test with 10 participants, aged from 28-80, with some participants being engineers and others having no knowledge on robotics whatsoever.

The test consisted in escorting a virtual robot through a maze and for each of the interfaces, it was attributed three conditions. The first, called *Immersive Test*, was a "typical teleoperation mode with a single ego-centric view from the robot's onboard camera” (Green et al., 2008). The second, called *Speech and Gesture no Planning (SGnoP),* was an early version of an Augmented Reality Human Robot Collaboration *(*AR-HRC*)* system that allowed its users to interact with the robot in an AR environment by using gestures or speech*.* The third condition, called *Speech and Gesture with Planning, Review and Modification (SGwPRM),* was the complete version of the system previously used in the second condition and in addition to the previous functionalities, it allowed users to create a trajectory for the robot to follow before giving the execution order.

The results obtained were evaluated using an ANOVA test and Bonferroni correction values. The values from the ANOVA test showed how meaningful were the comparisons between the performance of multiple interfaces[13], while the correction value helped to keep the results consistent.

Additionally, users were asked to rate several post-questionnaire statements. The statements related to the individual testing of each interface were built using a Likert scale from 1 to 7, while the statements to evaluate every interface were ranked according to the user’s preference.

They concluded that the best results came from *Immersive* interface, due to its straightforward design, while both the other interfaces were found to have a steep learning curve for some users. Despite this fact, SGwPRM managed to get the most accuracy out of the three interfaces, while *Immersive* got the worst results in terms of collisions with walls and accuracy.

[13]https://www.statisticshowto.datasciencecentral.com/probability-and-statistics/hypothesis-testing/anova/#targetText=An%20ANOVA%20test%20is%20a,there's%20a%20difference%20between%20them. Visited in 28/10/19

3. Requirements for Assisting Robotics Competitions

In this chapter, the scenarios and personas that define the goals of this dissertation are introduced, to help to keep the features implemented in the application within the scope of the project. Additionally, functional and non-functional requirements for the proposed application are presented.

3.1. Overall method

A Human-centred approach was followed and as a first step to better understand the target scenario and audience, this work started by informal interviews to potential users followed by brainstorming sessions involving software engineers, and augmented reality and robotics experts. Several stakeholders were identified, along with their main motivations, and an analysis of the current practices around robotic competitions lead to multiple scenarios, which were modified to encompass the use of a novel set of supporting features. Following on the recommendations of (Cooper, Reimann, & Cronin, 2007), these were materialized in user profiles, adopting some of the concepts related to Personas (e.g., an explicit consideration of motivations) and context scenarios, as presented in what follows.

3.2. Personas

To act within the scenarios described on the next section, 3 personas, suited to represent the types of users that might interact with the system were created. They are presented in the following pages.

Henry (General Public)

Age: 12
Work: Student at Escola Secundária Dr. Jaime Magalhães Lima, Esgueira
Family: Lives with his parents and his older brother
Location: Aveiro, Portugal
Nationality: British

Bio: Henry is a twelve-year-old boy and the youngest son of Alvin. He is at the sixth grade in school and has good grades. During his free time, he likes to play video games, play football with his friends, help his parents at home. Henry is a little bit of nerd, as since an early age, he has always shown a passion for technology and trying the latest technological experiences. His father is an engineer working at Husqvarna and Henry wishes to be like him and pursue a similar career when he grows up. At young age, it's still difficult for him to understand some concepts around the area's his father works with.

Goals:

- Have fun
- Discover the world of robotics and augmented reality
- Experience in first hand a robot competition

Frustrations:

- Issues during the competition, such as technical breakdowns that might occur on the robots
- Not understanding how to use the application that controls the robot's camera
- At young age, it may be difficult for him to understand many concepts in this matter

Motivation: It's his first time at robot competitions and he has never interacted with a racing robot before. Henry wants to understand what happens during a competition.

Alvin (Researcher)

Age: 32
Work: Engineer at Bosch in the robotics department
Family: Lives with his wife and children
Location: Aveiro, Portugal
Nationality: British

Bio: Alvin is a robotics engineer working at Husqvarna. He is 32 years old and recently got his PhD in robotics engineering. His typical day consists on waking up at 7h, starts working at 8h30, leaves at 16h30, goes to the gym after work, arrives home at 19h and spends the rest of the day with his family. Alvin considers himself a nerd and isn't a very outgoing person, usually preferring to stay at home doing investigation. Since he works with robots, Alvin loves programming and always had a knack for challenges imposed by his work. He shares the same interests in technology as his son does and he has fun teaching his son about what he does at work and even demonstrating him.

Goals:

- Increase his knowledge in robotic path planning, locomotion and obstacle detecting techniques
- Discover new ways of combining deliberative and reactive solutions on robot architectures
- Experience in first hand a robot competition
- Learn the applications of AR in robotics and how it can improve the robot's performance

Frustrations:

- Issues during the competition, such as technical breakdowns that might occur on the robots and this can prevent Alvin from gathering information
- Alvin might not find adequate information or the kind of data that can be used to help him and his team on their project

- AR concepts are complex and since Alvin has little to no experience in this field, it might take some time to comprehend them

Motivation: Alvin is currently developing the new version of Automowers, the company's line of automatic lawn mowing robots, alongside a small team. Their task is to improve and stabilize the robot's capabilities of obstacle detection, path planning and find new ways of locomotion that can be implemented in the robot, with technologies such as AR, since previous versions had issues when detecting and work around small obstacles, such as flowerbeds or, occasionally, would get stuck while traversing the garden.

Marvin (Referee)

Age: 21
Work: Finishing graduation at Universidade de Aveiro
Family: Lives with his girlfriend in Aveiro
Location: Aveiro, Portugal
Nationality: American

Bio: Marvin is a 21-year-old student at the University in the last year of his graduation in Computer Science. He is a reasonably good student, though he sometimes has trouble learning understanding some subjects. He typically wakes up early since his classes start early and sometimes, he stays at the University working on projects or finishing studying some subjects. During his free time, he likes to read and play team sports. He is very sociable, goes well with practically everyone around him and usually takes lead in his projects. When Marvin completes his graduation, he wishes to pursue a master's degree on Computer Science.

Goals:

- Gain experience to boost his professional skills when he finds a job
- Experience a new environment which he is not used to
- Occupy his free time

- Learn information about the world of robotics and AR

Frustrations:

- The schedule of work given to Marvin might not suit his university schedule
- The workload can become stressful if time is not well managed

Motivation: Marvin is currently trying to find a part time job, which he can use to complement his free time and earn some experience. Although his interests are more geared towards subjects such as computer architecture, robotics is an area that he would like to know more about, besides having the adequate qualifications, like being responsible and having strong communication skills, which are needed to referee a competition. Marvin would like to have something to help him make decisions in more ambiguous situations.

3.3. Scenarios

To give context on how this work can be proved useful in its desired environment, several scenarios were created, displaying possible uses for the system, being the actors, the personas presented previously.

1st Scenario - Henry goes to a competition with his father - Henry sees his first race:

It's Saturday morning and Henry goes with his father to attend his very first robotic competition. Henry is young and is still not familiarized with the basic concepts in robotics, such as how a machine can operate in an autonomous way. As the competition starts, spectators are able to see additional information about the robots competing, with the help of the RobARtics app, which displays the information in a projected large screen, thus providing the needed explanations to Henry, about what's happening during the competition, such as what the robots are doing or are supposed to do.

Since there are many robots competing, Henry picked his favourite robot and, with the help of RobARtics, he checks how it is performing, compared to the others. Henry's father then instructs him to look at screen, located to the right of the competition track. He sees the track, from above and the robot moving, but there is more information, just as if he was watching a race in their home TV.

At some point, the robot reaches a bifurcation, in the track, and Henry gets excited because the robot did not follow a blue line that he can see over the track, in the screen. His father explains that the blue line shows where the robot should have passed.

Sub-scenario:

Henry sees how his favorite robot performs, when compared to the competitors: Rubicon is Henry's favorite robot. It reminds him of a cartoon hero he is very fond of. When Rubicon enters the track, to compete, Henry gets anxious. He would really like Rubicon to win. Rubycon starts and Henry watches its performance on the big screen. It is following the correct path, even in the first track bifurcation, but, as a shadow, Henry can also see the position of Hamlet, the best performing robot, until now, just as if both robots were in the track, at this time. Rubicon is just a little bit behind. Henry sees that Hamlet failed to abide to the last sign and is hopeful that Rubicon will do better. And it does, getting the highest score of the competition.

2nd Scenario - Alvin validates new navigation algorithms and learns how robots behave in competitions:

Alvin has been invited by a friend to attend to his robotic competition, as part of an agreement between them to help Alvin find new technologies at the competition, like AR, path planning techniques and to learn more on how robots react when facing obstacles, which he could implement in his company's latest lawn-mowing robots to improve their performance. While his son is watching the details of his favorite robot in the big display, Alvin takes the opportunity to use the RobARtics app himself to check more detailed information, such as the signal identification capabilities of the robot, it's orientation while traversing the track, as well as a heat map to check the accuracy of the robot's path planning methods.

3rd Scenario - Marvin accesses a complex competition situation:

Usually, Marvin is a competent referee during competitions and he is generally capable of handling many difficult situations. However, he came across an unclear case in whether a certain robot was able or not to stay consistently on track the whole race, to decide if is disqualified from the race. Using the RobARtics app, Marvin checks the previous trajectories of this robot and compares each lap the robot completed. With this information, he can conclude if this particular robot did too many infringements to continue to participate in the event.

3.4. Requirements

The RobARtics application was developed with the goal to provide a useful tool that would help the described users in similar scenarios to the ones presented previously. In what follows the main functional and non-functional requirement identified based on the personas and scenarios are presented.

4. RobARtics – System Development

4.1. First Prototype

Before developing the application, a small prototype was developed to test and assert if the planned features to be implemented in the final version of the application fitted the scope of the dissertation. This prototype was developed using Balsamiq Mokups[14], a tool used to sketch low fidelity prototypes.

The features in the prototype were split into two different users, the public and the researcher/referee, representative of the personas described previously.

Public features:

- View Robot Details
- View Trajectory
- Enable Signal Identification
- Enable Ghost Racer

Researcher/Referee features:

- View Robot Details
- View Trajectory
- Enable Signal Identification
- Enable Ghost Racer
- View Heat Map

[14] https://balsamiq.com/ Visited in 27/10/19

- View Odometry
- Logbook

This prototype was tested with 6 users. Half had previous experience with robotics and AR, while the other half only had tested AR applications. They were instructed to perform a series of given tasks, covering every functionality intended for the application. At the end, the participants were asked for feedback and their opinions about the features presented. Only one user didn't feel capable to give proper feedback, due to their lack of knowledge in the area.

Although the general public profile had less features in comparison with the other profile, the tasks allowed users to interact with every feature. Thus, the Figures 4.1 to 4.6 are all taken from the researcher/referee profile.

The first task users had to complete was going to the "Robot Details" page, switch between robots and observe their characteristics. This page was supposed to allow users to check specific information about the registered robots, such as the type of sensors they use and localization techniques, as well as allowing them to see how one robot compared to the others (Figure 4.1).

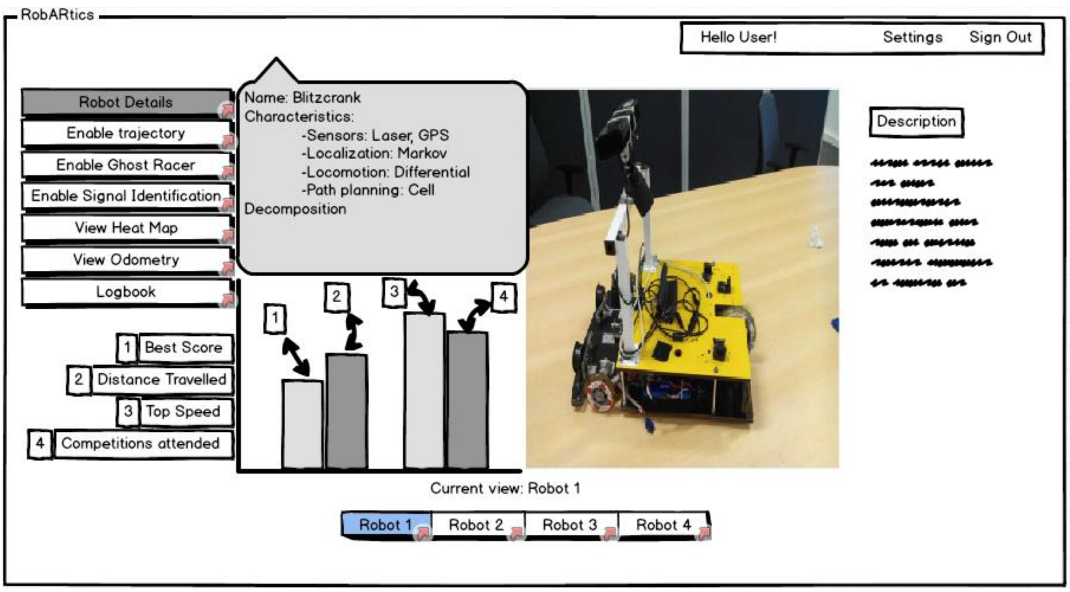


Figure 4. 1: The prototype's robot details page

The results for this task were all positive. The users found the buttons at the bottom of the page to be an intuitive way of checking they characteristics and thought that the initial presentation was clean. A suggestion made by a user included displaying the robot information's on the top right of the page, to give more space to the bar graphs.

The second task was to analyze the "Enable Trajectories", which intended to simulate the visualization the robot's trajectory of a previous race or the trajectory it's currently planning in a real time race. From there, the users were asked to view both trajectory modes (Figure 4.2).

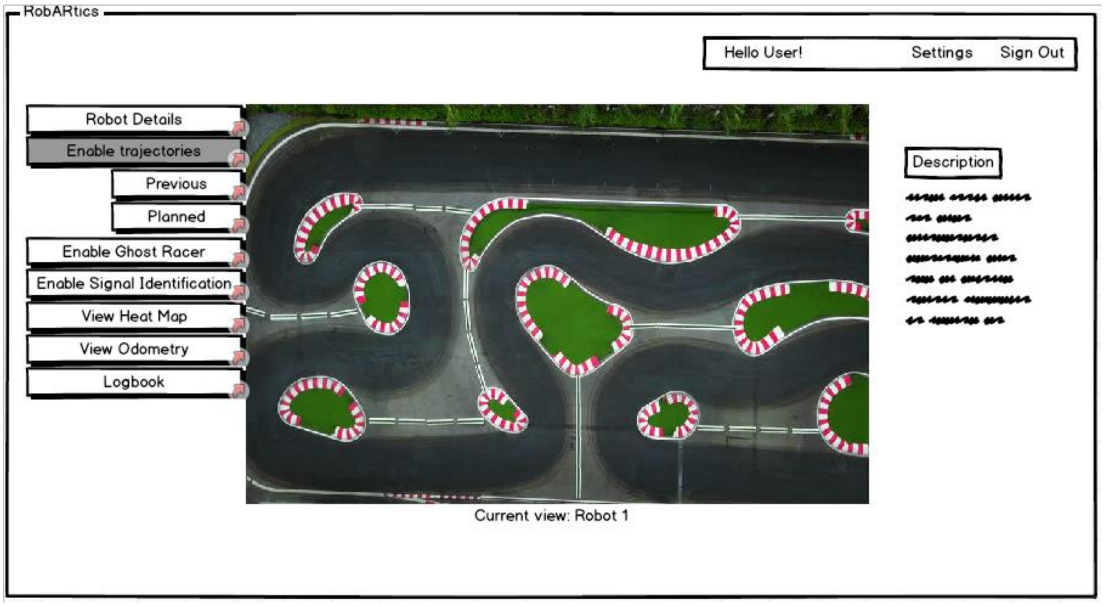


Figure 4. 2: The "Enable Trajectory" page

General feedback from the users with little knowledge on the robotics area was positive, although users with more knowledge found the name "Enable trajectories" to be a bit derivative from its actual function.

Next, users had to visualize the Ghost Racer by clicking in the "Enable Ghost Racer". This simulated a feature in which a virtual representation of the robot from a previous lap would be projected in the track alongside the real one (Figure 4.3), useful in comparing its performance between laps.

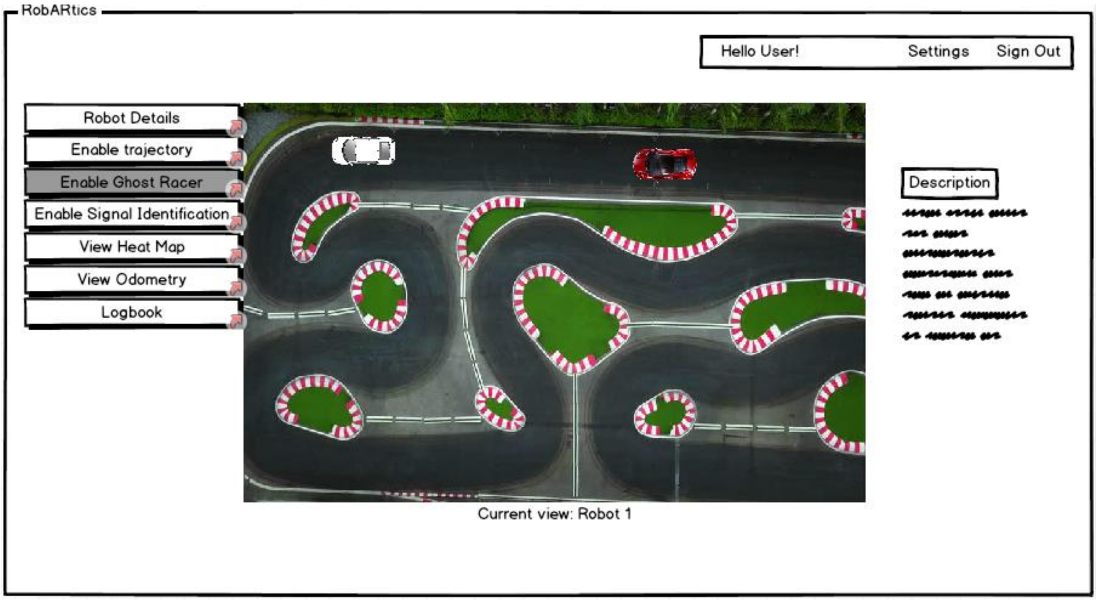


Figure 4. 3: The "Enable Ghost Racer" page

Some users were already familiarized with this concept, since it has always been a popular feature in racing videogames. They found this feature to be somewhat difficult to simulate and therefore, gave it a positive comment, although slightly unclear. Some of the more inexperienced users said that this feature should remain exclusive to the researcher/referee.

The following task had users simulating the signal identification capabilities of a robot. This means that the agent, whenever saw a traffic signal, such as turning on the low beams in a tunnel, or a cattle alert sign, it would display on the top right of the camera the signal it detected (Figure 4.4).

This feature was positively received, as users found very clear how the prototype demonstrated the robot detecting the signal and understood the concept well.

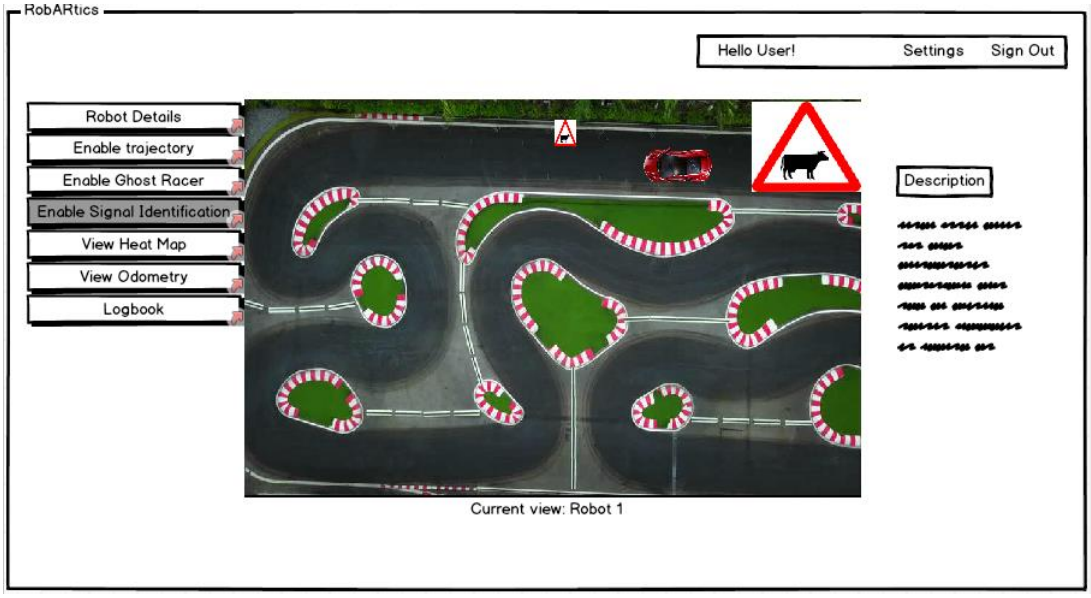


Figure 4. 4: The "Enable Signal Identification" page

Next, the task was to observe the heat map generated by the robot. A heat map consists of an analysis software that uses color to represent data instead of height and width, commonly used by bar graphs (Figure 5.5). In this case, it displays the areas where the robot spent more time during a certain number of laps. Red represents where it stayed the most, followed by yellow, green and finally blue.

Figure 4. 5: The "View Heat Map" page

Like the Ghost Racer feature, both experienced and novice users found that this feature should stay exclusive to the researcher/referee profile. Users generally understood the heat map concept well, despite causing some confusion in some users about the number of laps required to generate the heat map.

Following this task was the visualization of the robot's odometry. Odometry is a method that uses data read from the robot's sensors to estimate changes in its position over time. In this prototype, this change is demonstrated using a column besides the track view, updating the agent's position as it moves (Figure 4.6).

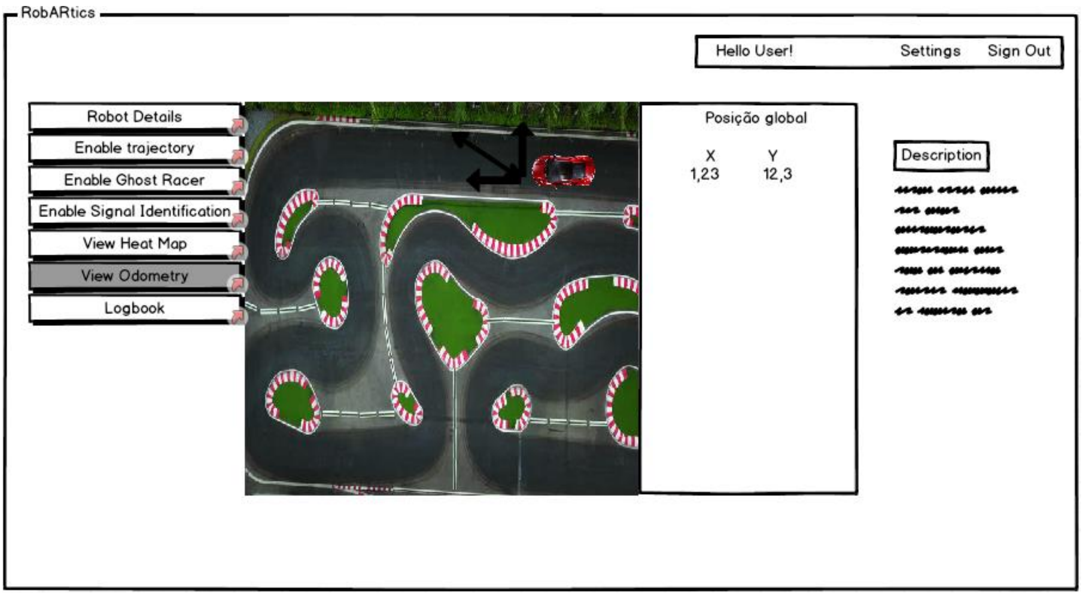


Figure 4. 6: The "View Odometry" page

Once again, experienced users agreed that this feature shouldn't be included in the public's profile. Some users found odometry to be a complex concept to understand and due to this, they couldn't give proper feedback on this feature. The more experienced said the column should still be display on the right but overlapping with the map.

Finally, the last task was to check the logbook, which contains a historic of the recorded values made by the robot, such as road signs detected and the heat map (Figure 4.7).

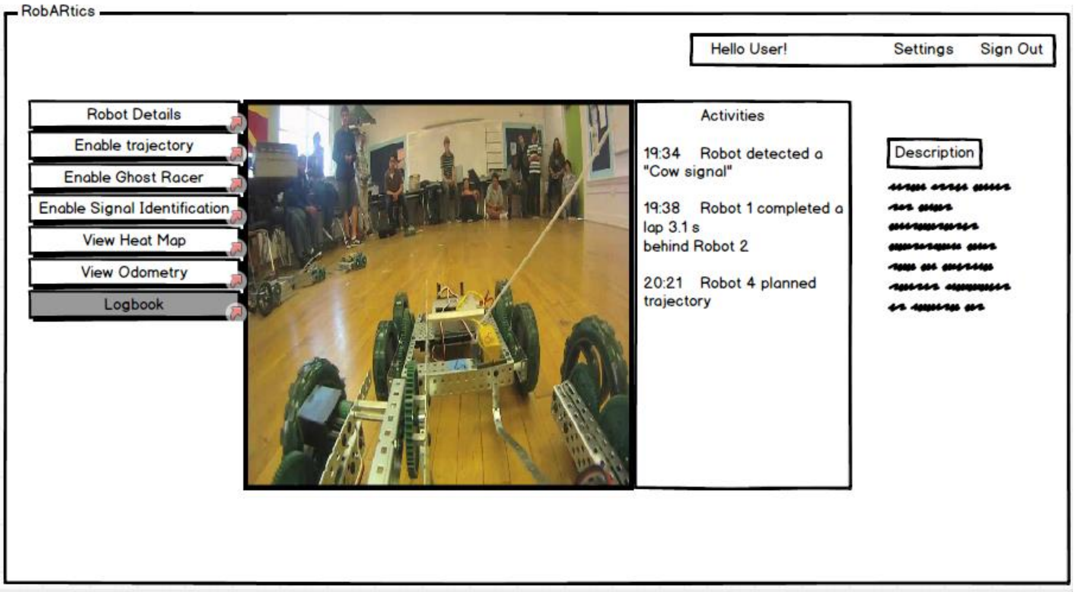


Figure 4. 7: The "Logbook" page

Generally, users found this feature to be a nice touch, since it allowed them to view all the information and actions performed in a single page.

Some of final comments made by some users include allowing overlap between some features, such as the Ghost Racer and the Trajectories, the integration of a “Back” button and create a default page for the application (resembling a home menu).

4.2. Architecture and Implementation

RobARtics was made with the Unity 3D game engine developed by Unity Technologies[15]. This engine was chosen for the development due to its versatility and primarily, the ability to develop AR applications and communicate with Robotic Operating System (ROS). The engine itself is a game engine, supported by platforms like mobile, video game consoles and desktop and designed to allow a more accessible approach to game development. It allows the development of applications in both 2D or 3D, the inclusion of plugins, such as the one used in this work, the ROS# and numerous post-processing and scene rendering features according to each available platform, such as render-to-texture and texture compression, respectively.

[15] https://unity.com/ Visited in 27/10/19

The programming language used to code the RobARtics application was C#, the engine's scripting Application Programming Interface (API), while the AR components were handled by the Vuforia Software Development Kit (SDK), which permits the creation of AR applications on mobile devices. As previously mentioned, the communication between the engine and ROS, used by the robots was established through the ROS# scripts, prominently the Odometry Subscriber. ROS# was created by Dr. Martin Bischoff and was available for free to use.

To receive and display data from the simulator where ROTA was deployed, it was integrated alongside Unity, ROS, an open source, agglomerate of software frameworks used in robot software development, which usually integrates hardware abstraction and message passing between processes methods. ROS is primarily programmed in C++, Python and Lisp on the Linux operative system (OS) and it is characterized by its scaling and flexibility, allowing software to adapted to larger development cycles and the multiple programming languages that might be used to implement it. Its architecture follows the publisher-subscriber model. It represents its processes as nodes, connected to each other forming a graph, with a single ROS process called Master, allowing the establishment of connections between the other nodes and handling the server updates. Topics are the means of transport of the messages between the nodes, in which a node must publish to the topic of another node to send messages and subscribe to a topic when it wants to receive them. In the case of RobARtics, the nodes were .bag files and the ROS connector component, present in each ROS# script. From there, topic information was parsed and, with values such as the position, they needed to be converted to Unity world coordinates.

The ROTA robot, the main agent tested and used in the application, was created and controlled by using the ROTA simulator, developed in the Gazebo robot simulator.

After testing the prototype, some of the feedback was taken into consideration and the development of the application started with the design of its layout in the Unity 3D engine.

The application was originally intended to be used in desktop computers, with later being added the support for mobile devices, although as the development of the application went through, it was ultimately engineered as a mobile tool, more specifically, for Android devices. The presented architecture (Figure 4.8) shows how the different components are interconnected within the application. The multiple functions are available after entering the main menu. Some of its functions, particularly the *View trajectory* and *View Global Position* functions, are both connected to ROS through the ROS# plugin through an odometry subscriber script. The plugin then requests the data from the simulator, mainly the position and rotation, which is sent from the simulator as a topic. This information is displayed in the *Logbook* and in the *View Global Position* pages.

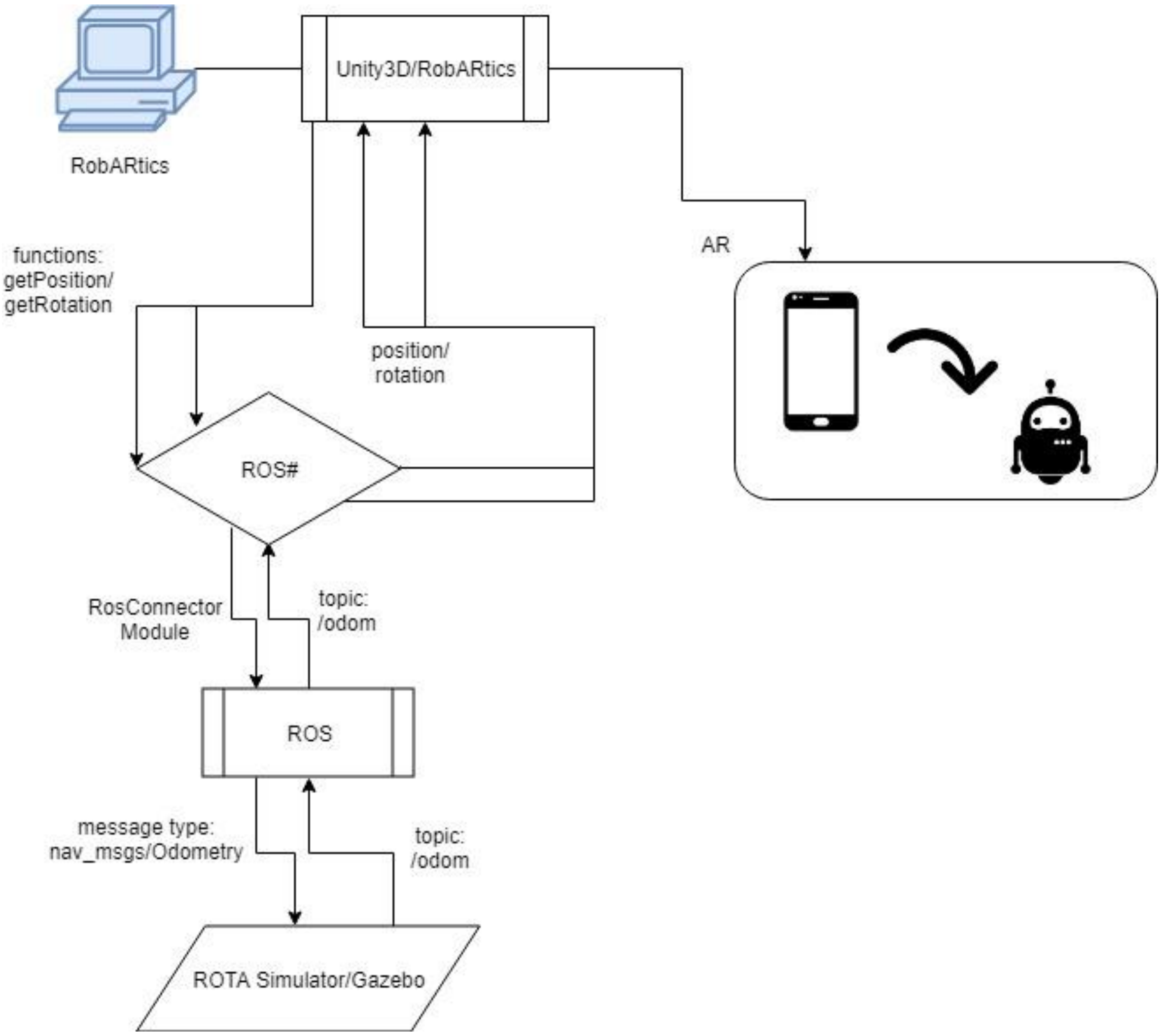


Figure 4. 8: Architecture of RobARtics

4.3. RobARtics – Functionality

Since the development of the first prototype, some of the features were excluded from the final version of the application. The *View Heatmap* feature proved to be complex and was removed due to time constrains. Originally, there was also the possibility to choose from the public and researcher/referee profiles, similar to the first prototype, but ultimately, the profile selection was removed, because the public profile was identical to the researcher/referee profile, albeit supporting less technical features and the main persona which this work was intended to was the researchers and the referees, which are the people involved in robotics and in the competition itself. While the distinction between profiles was removed, the features implemented for the public are still present, making RobARtics still perfectly usable in that context.

The final list of available features is the following:

- Add and View Robot Details
- View Trajectory (Planned and Previous)
- Enable Signal Identification
- Enable Ghost Racer
- View Robot's Speed
- View Robot's Direction
- View Global Position
- Lap Counter and Lap Time
- Logbook

On the prototype version, the *View Global Position* function was incorrectly named as *View Odometry*, since odometry is the usage of data obtained from the robot's sensors to estimate its position and global position indicates the robot's coordinates, which was the intended function.

More new and useful features were introduced. Users are now able to see the robot's movement speed as it drives around the track, which direction is the robot facing and a timer indicating the time the robot is taking to complete the course, next to a lap counter to show how many laps around the track has the robot completed.

4.3.1. Login

Much of the prototype's feedback was taken in account and several changes were made to the applications final design. As explained previously, the profile selection was removed. Instead, now both users start at the same login page.

The user is asked to enter his/her participant number, to be later included in the results statistic (Figure 4.9).

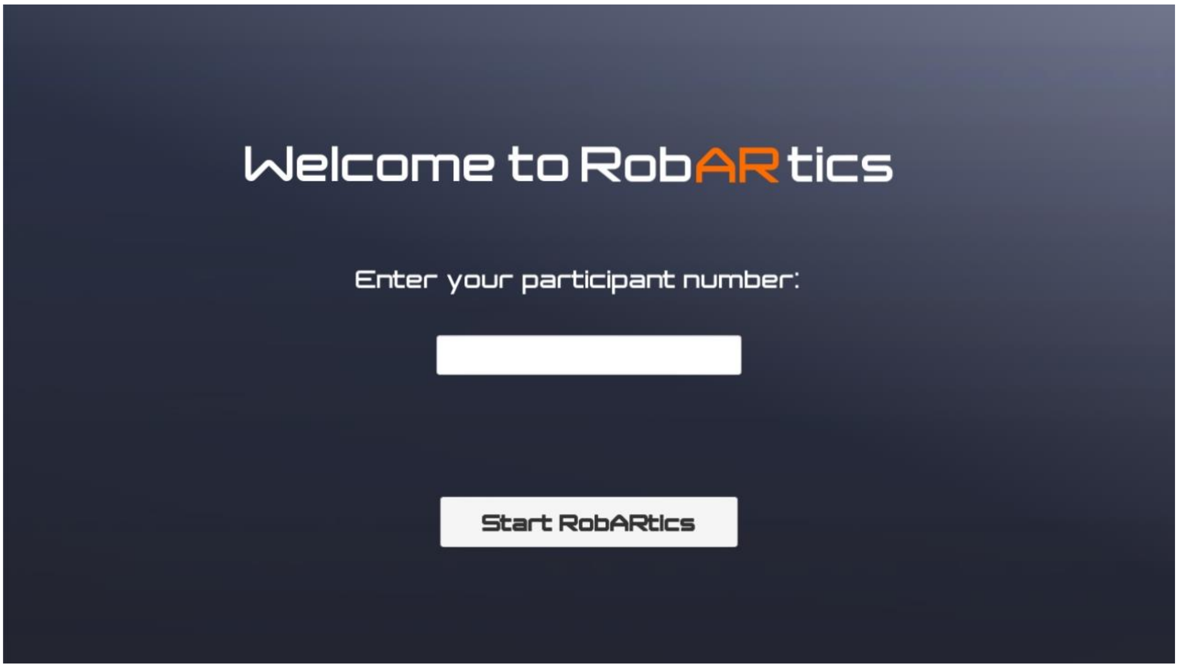


Figure 4. 9: Login Page

After pressing "Start RobARtics", users are taken to the app's main screen (Figure 4.10).

4.3.2. Main Screen

The major changes between the prototype and the final application were the multiple scenes that were selectable through buttons were scrapped, favoring a single main scene where every possible feature that could be selected by those buttons was contained in a single screen. The features are controlled by check boxes, letting users manage which information they would like to see on screen. This way, multiple bits of information can be viewed at the same time, unlike the previous design where each feature was contained in an individual screen.

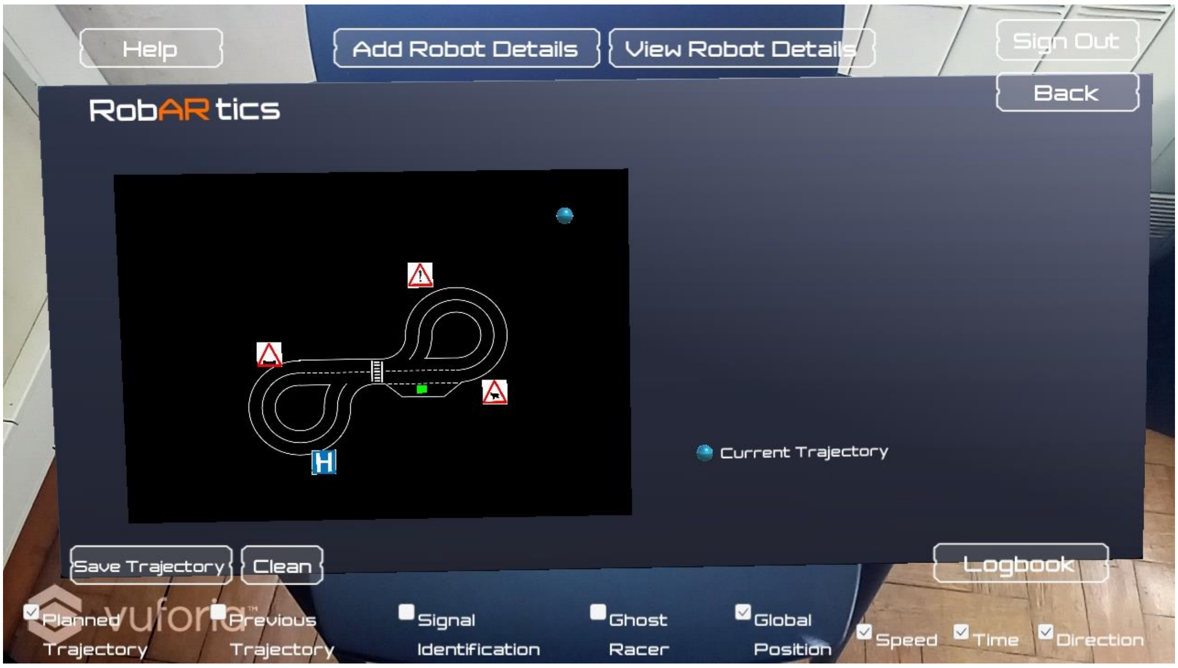


Figure 4. 10: RobARtics main screen, presented in AR

The application's main screen is composed by two different layers (Figure 4.10). The overlay canvas, which contains all the User Interface (UI) buttons, is fixed on the screen and acts as a tangible interface for users to select their options and navigate through the other menus. The blue canvas behind it is totally viewed in AR, using a marker-based approach. The idea was to make such marker be placed somewhere around the agent, thus allowing users to point at it and see the details focused around that robot.

Right away, the only details that are presented before the application receives data is the track, based on the FNR 16th edition, the caption indicating which will be the color of a trajectory to be displayed and the UI elements.

4.3.3. Add & View Robot Details

Clicking on the *Add Robot* button will bring the user to the following page (Figure 4.11), where various information about the robot can be introduced such as:

- Name: The name given to the robot
- Sensors: The type of sensors the robot uses
- Localization: The methods used by the robot to find its position in an environment
- Locomotion: The means used by a robot to move within an environment
- Path Planning: The algorithms used by the robot to travel across an environment

This information is saved across the application using PlayerPrefs. These are variables that allow the storage of values and maintain them between scenes, as well as sessions. In this case, the information is stored as a *string*. The details are intended to be small indications inputted by the competition staff to give the public and researchers an opportunity for learning a general easy to understand description of the robot they are currently pointing at. Clicking the save button will store the information written.

Figure 4. 11: Add Robot's Details page

These details can be view later by pressing the *View Robot Details* button in the main screen or right on the *Add Details* page (Figure 4.12). This page is presented in AR.

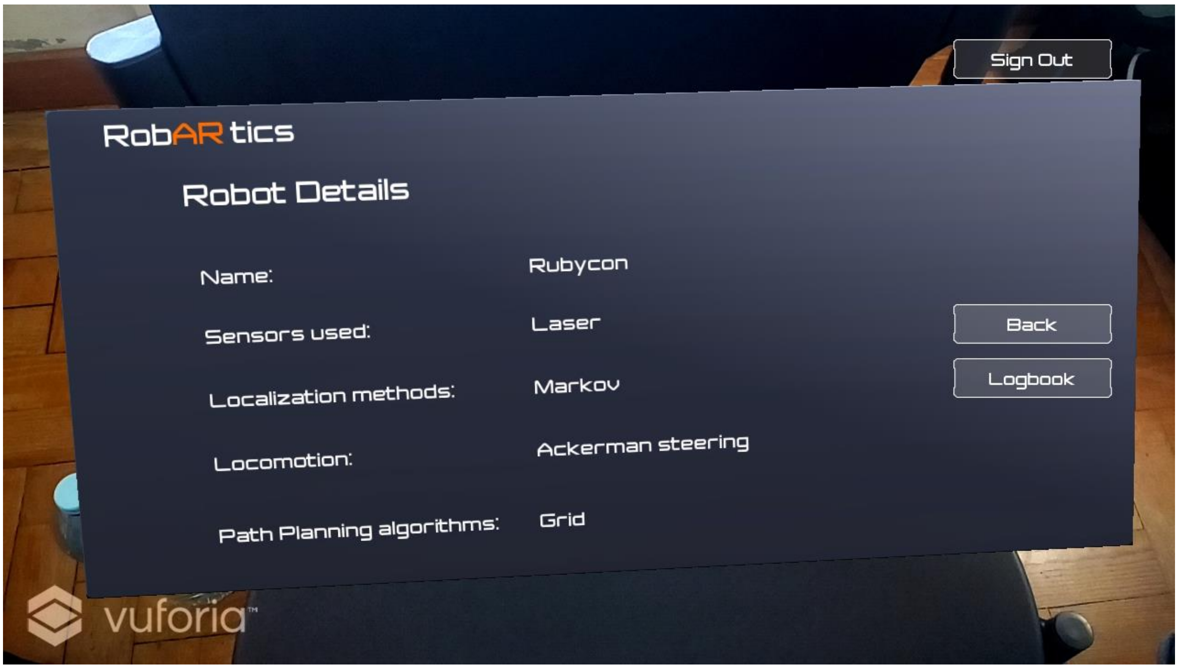


Figure 4. 12: View Robot's Details page

4.3.4. Planned Trajectory

Going back to the main screen, the *Planned Trajectory* option is enabled by default, since it's the tool that measures and represents on the track the trajectory that currently the robot is going for. The user can choose between two modes of visualization: *Previous* or *Planned* (Figure 4.13).

After running a *Planned Trajectory,* users may use the *Clean* button to clear the track. To develop this feature, the first step was to set up a camera in the ROTA Simulator, recording a top-down perspective of the simulator's track. This camera would be later used to record frames of the robot traversing the track (Figure 4.14). To save every information published by the ROTA robots' topics, it was used the *rosbag* tool that allowed recording and playing back active ROS topics. The frames, alongside the .bag file, were brought to the Unity 3D environment. From there, the user may start a *rosbridge* connection to allow interactivity between ROS and the engine. The IP connection was set up in the *Scene* inspector.

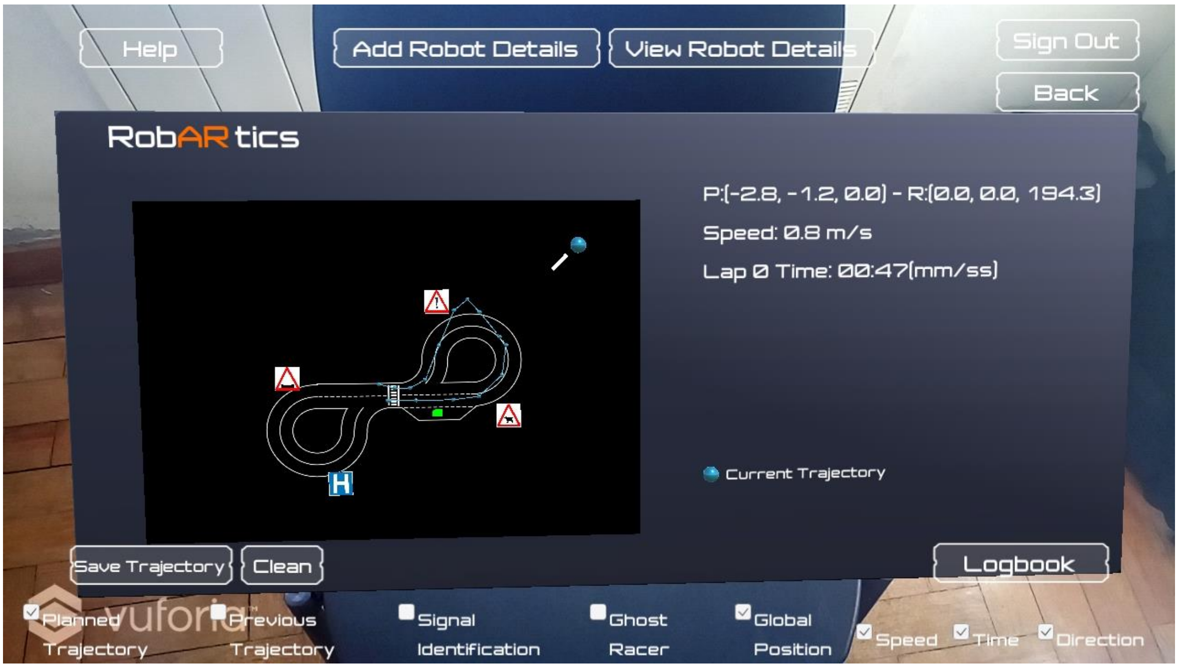


Figure 4. 13: Planned Trajectory view

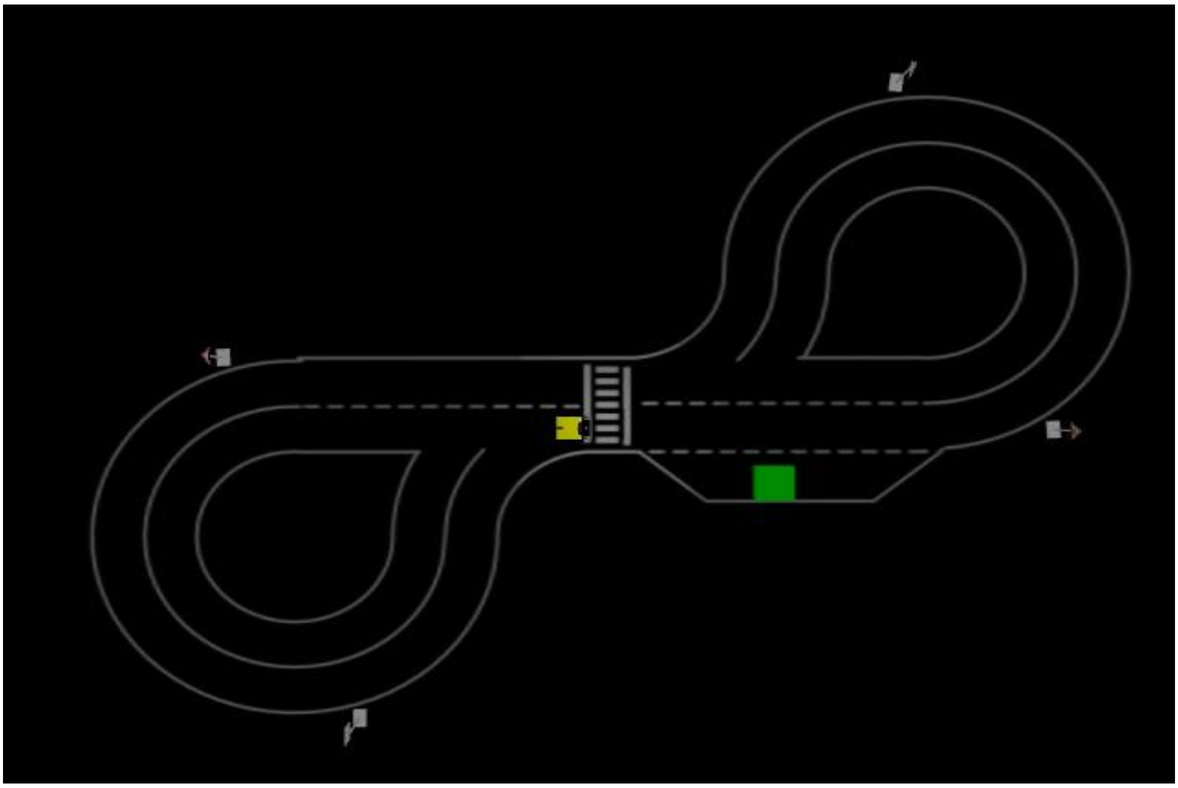


Figure 4. 14: Overview of the track in the ROTA simulator

The result of this feature is the simulation of the robot's global position, as he traverses the track. The global position, retrieved from the odometry topic, starts to be drawn on top of the track, represented by several small, connected spheres.

While the trajectory drawn on the track allows the public to get an idea of the robot's position while it traverses it, the right top of the screen provides several useful information that can aid both the referees and researchers. The first line gives the position and the rotation of the robot, received from ROS. The rotation was converted from the original quaternion format to Euler Angles, to be more accessible to the public and referees. This new format is measured using Roll (X axis rotation), Pitch (Y axis rotation) and Yaw (Z axis rotation), commonly used in airplane aircraft, but also applied to robotics (Figure 4.15). This feature was designed to be used by all the personas.

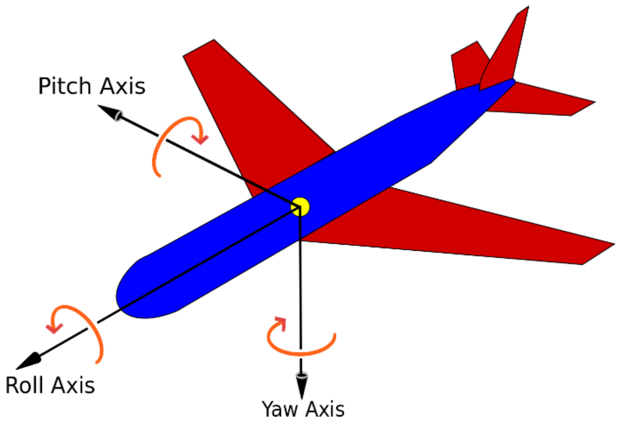


Figure 4. 15: Demonstration of the Roll, Pitch, Yaw angles[16]

The position is represented by the traditional cartesian coordinate system (X,Y,Z). The second line shows at which speed is the robot currently going, read from the base velocity topic. The following line fetches again details received from the odometry messages, prominently the timestamps and indicates the users in which lap is the robot currently in and how much time is it taking.

While the coordinates are being generated as spheres, they are also being stored in PlayerPrefs, alongside the respective rotation. To store them as such, the content of the arrays where they were being stored was converted to the *string* format. With the values

[16] https://en.wikipedia.org/wiki/Aircraft_principal_axes Visited in 27/10/19

stored as *string,* it was possible to replay this info, incorporated in *Ghost Racer* and *Previous Trajectory,* using a reverse operation to convert them back to their original formats.

4.3.5. Previous Trajectory

Previous Trajectory shows the trajectory of the robot from the previous lap, in case the user decides to check the Logbook or perform other actions (Figure 4.16). The information displayed from the *Planned Trajectory* stays on screen and this function quickly rebuilds the trajectory with a new color, displayed on the right bottom.

Figure 4. 16: Previous Trajectory page

The *Previous Trajectory view* will always keep the coordinates, allowing users to enable other features and check the trajectory again in case they missed something from the previous run.

4.3.6. Signal Identification

Signal Identification is a feature useful for every user that simulates the competition's challenge of signal identification through the sensors of a robot, for road signs like *STOP* and *Tunnel* warnings. This option may be selected in conjunction with *Planned View* and when it's active, a message displaying *Displaying signals…* will appear. The signals are represented on the application's track by blue spheres for simplifying reasons. These signs can be viewed in the video recorded from the ROTA simulator. As the trajectory is being drawn, each time the robot detects a signal, an image of the signal detected is displayed on the right and next to it, the signal's name. The signal images were taken directly from the simulator's files (Figure 4.17). Currently, the way the signs are detected is through the robot's position and how close it's to the signs.

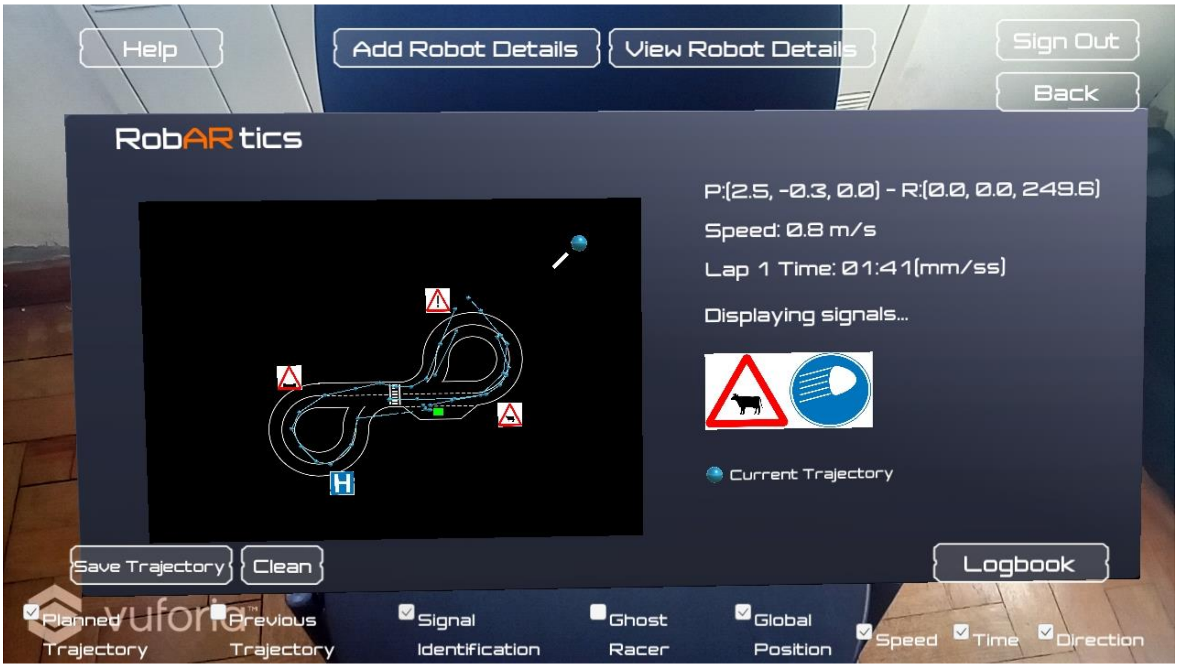


Figure 4. 17: Signal Identification activated. Detecting "Danger" signal

Additionally, while this option is active, users can press the signal images on the track to amplify them and view the name of the signal pressed and its coordinates. Tapping them again deactivates this functionality.

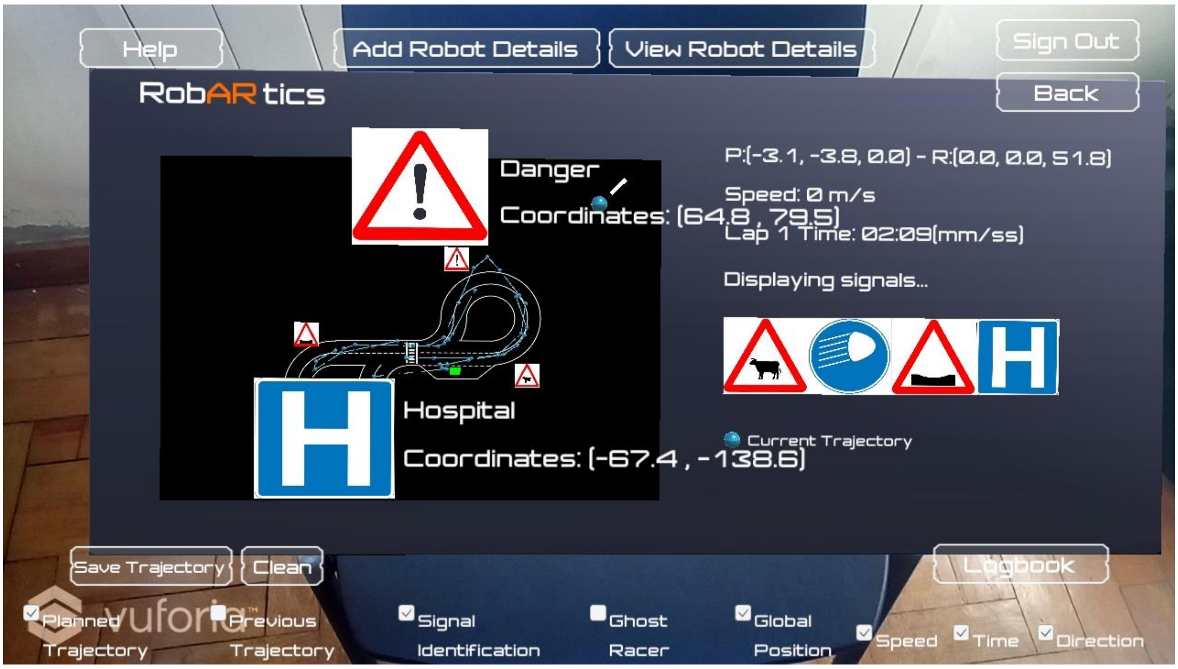


Figure 4. 18: Signal augmentation, displaying the name and coordinates of two signs

4.3.7. Ghost Racer

Enable Ghost Racer is a feature most commonly found in racing video games. Similar to the implementation in such games, it allows the user to view and compare various previously saved trajectories. To save a trajectory, there's a *Save Trajectory* button at the right bottom of the track in the *Planned View* feature, which after being pressed will bring the user to the *Save Slot* screen (Figure 4.19).

The user is then allowed to save a trajectory into one of two slots. Once again, the *PlayerPrefs* are called each time the user presses the slot button. If there are no previously saved trajectories in any of the slots, the trajectory's coordinates will be saved normally, while the following message is displayed. (Figure 4.20).

Otherwise, a message asking the user if they want to overwrite the previous save is displayed (Figure 4.21). They may choose yes if they want to overwrite the coordinates or no to choose other save slot. The *Delete* button empties the *PlayerPrefs*, effectively erasing any saves in every slot.

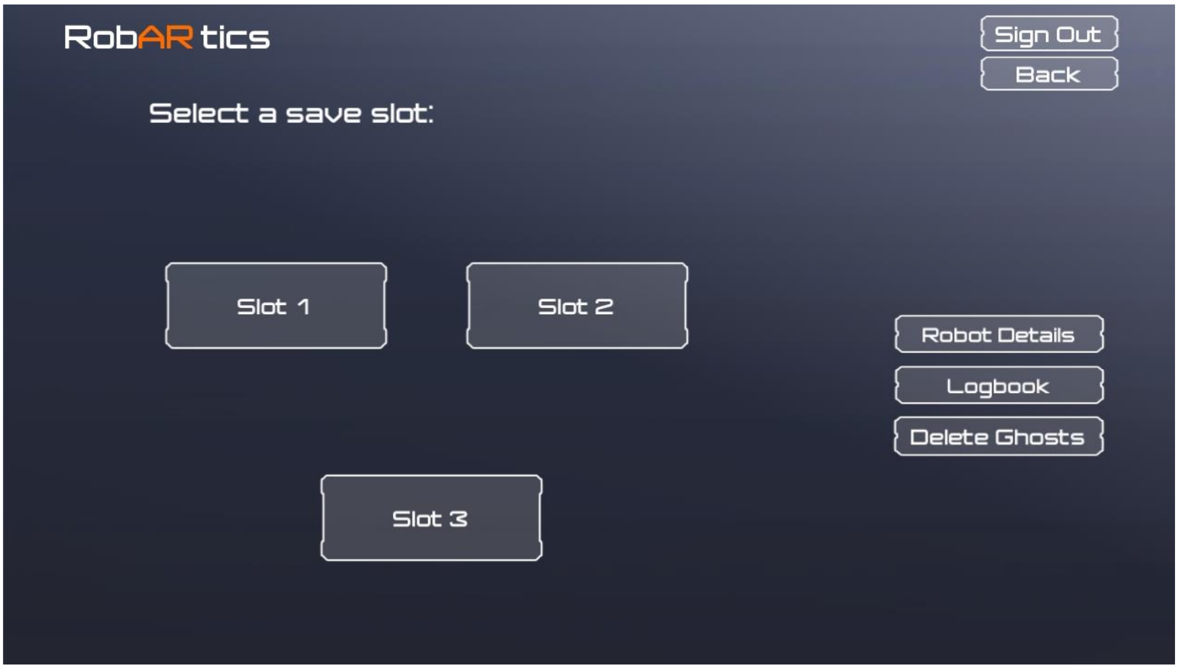


Figure 4. 19: Save Slot screen

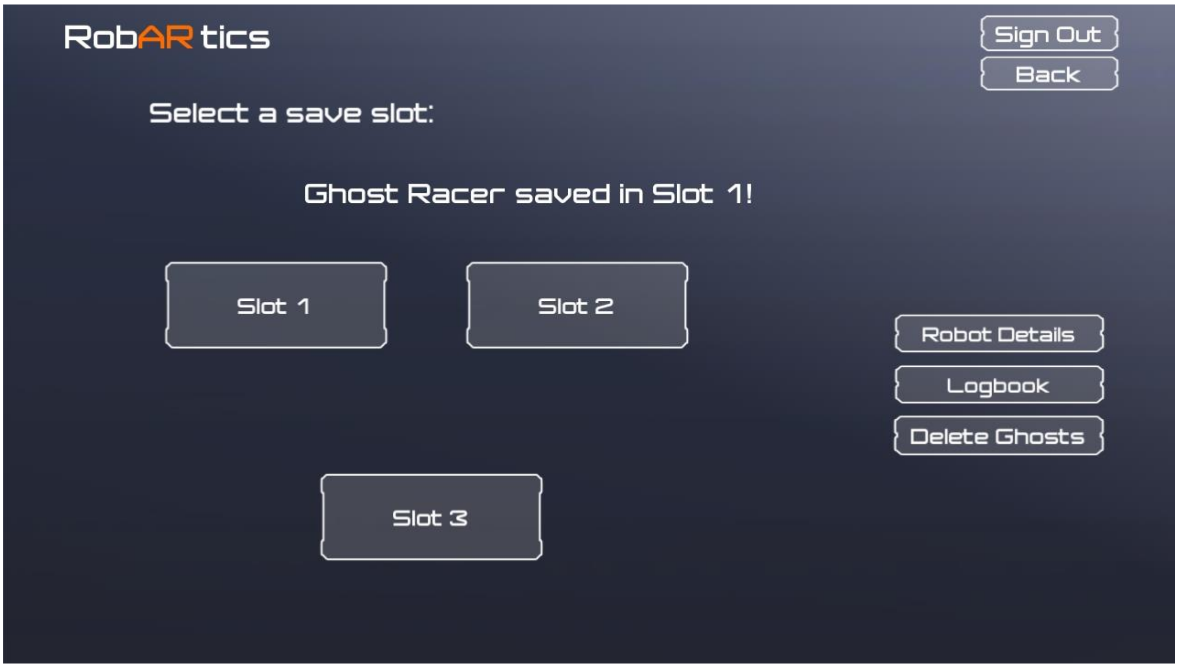


Figure 4. 20: Ghost Racer successfully saved on slot 1

After returning to the main screen, by checking the *Ghost Racer* option, three buttons appear, giving the user the option to play previously recorded trajectories. Both saved trajectories can be played at the same time, with the following results (Figure 4.22).

The ghosts are identified by different colors, corresponding to the symbols on the bottom right (Figure 4.23).

The ghost racer is a feature with appeal to all users, since it allows referees and researchers to play other laps from different robots or compare a current lap with a previous one. This also suits the public for entertainment purposes.

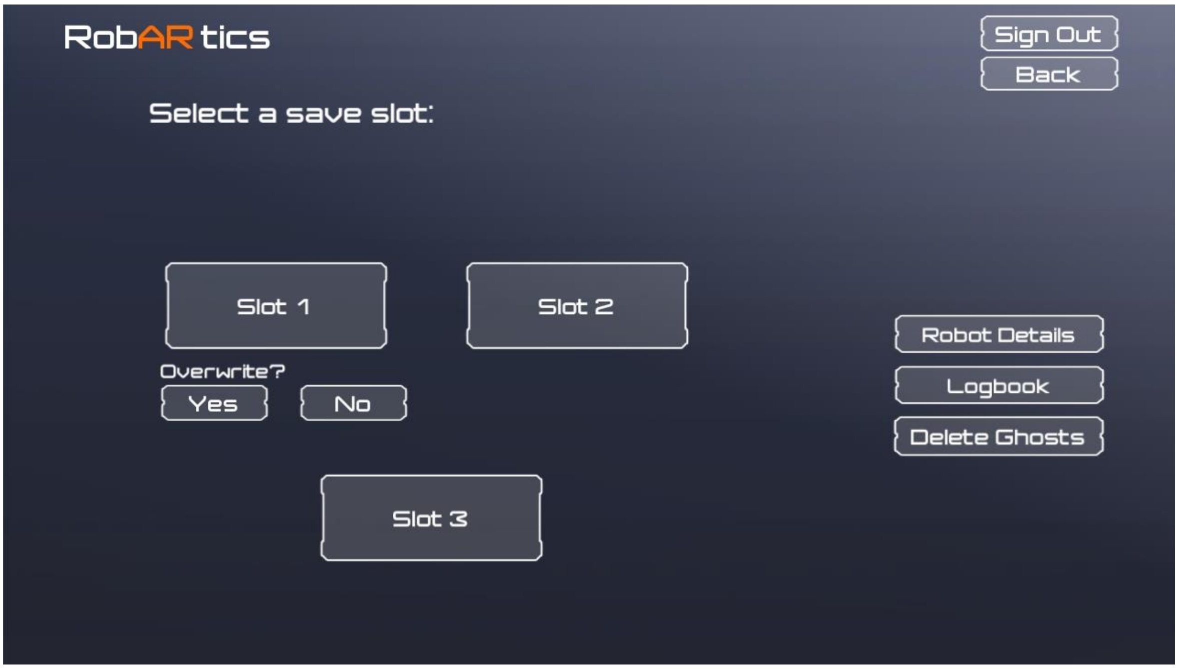


Figure 4. 21: Overwrite dialog

4.3.8. Logbook

Located above the *Speed, Time* and *Direction* checkboxes, the *Logbook* feature allows researchers and referees to view the specific coordinates of a robot by displaying a list with robot's coordinates collected from previous runs (Figure 4.24). Additionally, it also displays the rotation, given again in Euler Angles, the signs detected by the robot, how many laps it completed around the track and the corresponding times.

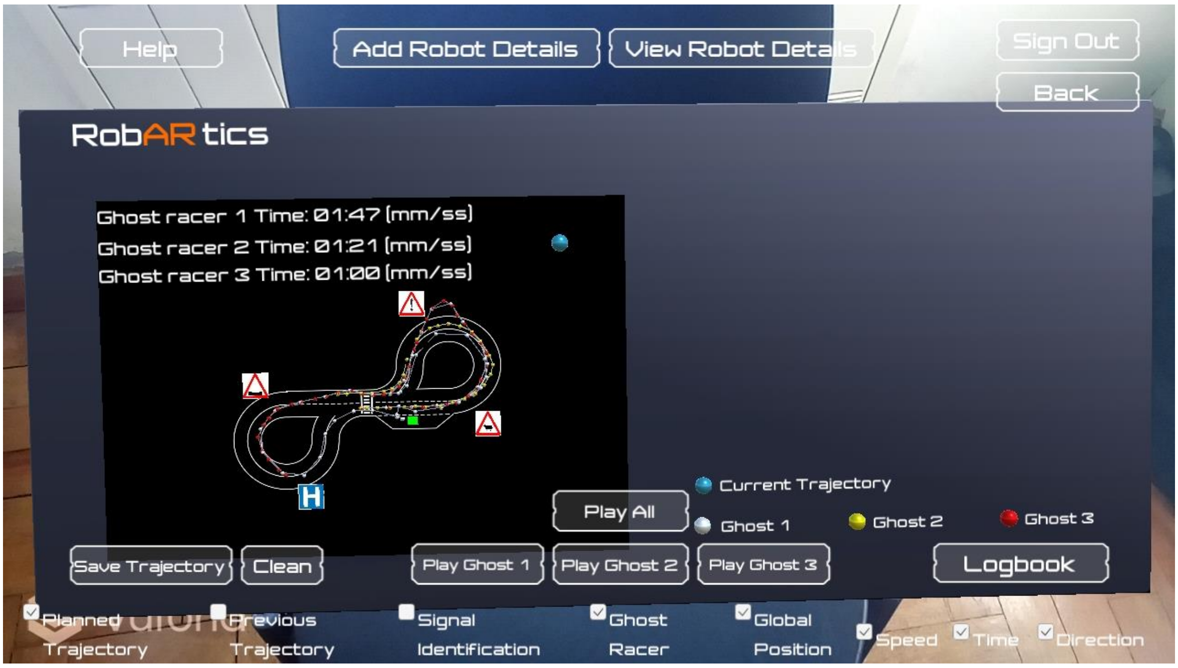


Figure 4. 22: Ghost Racers along with a planned trajectory being displayed

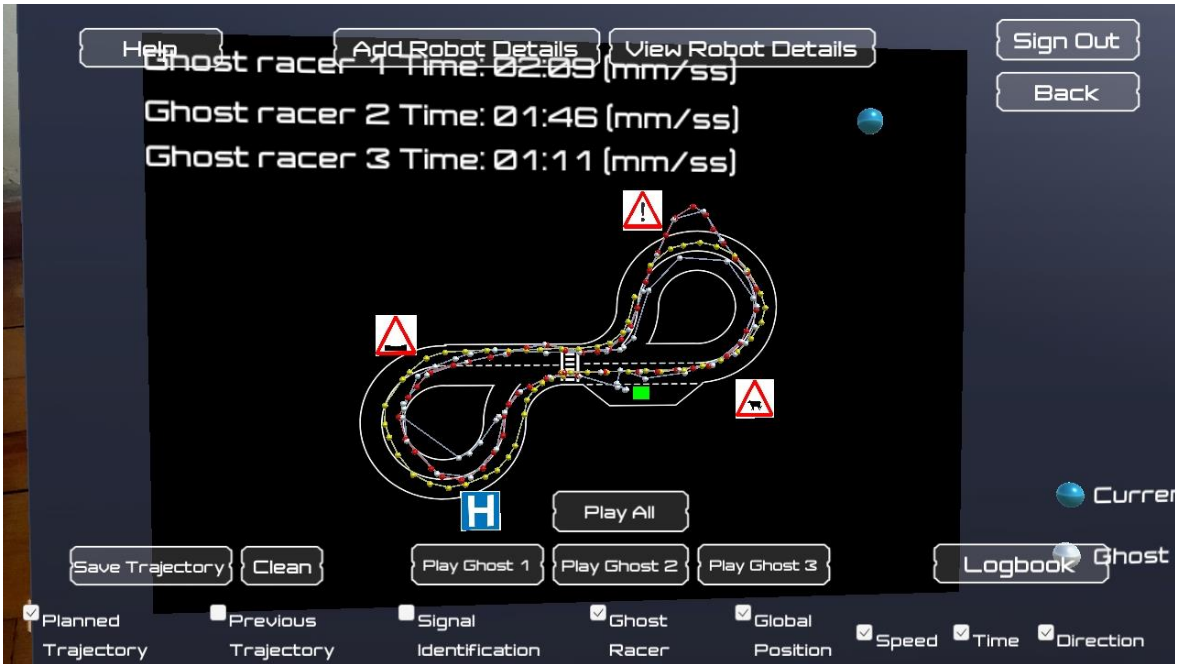


Figure 4. 23: Closer look at 3 ghost racers played simultaneously

Figure 4. 24: Logbook page

Users can choose between the different options by using the buttons above (Figures 4.25 & 4.26). The availability of each option is related to its occupied space. For example, since the rotation coordinates are rather extensive, that feature is disabled to give room for the other information. This feature is more geared towards the researchers and referees, given it provides more technical information like coordinates of the robot's position and rotation values, although the signs detected and the lap times may be of the public's interest as well.

Figure 4. 25: Logbook showing Position and Rotation

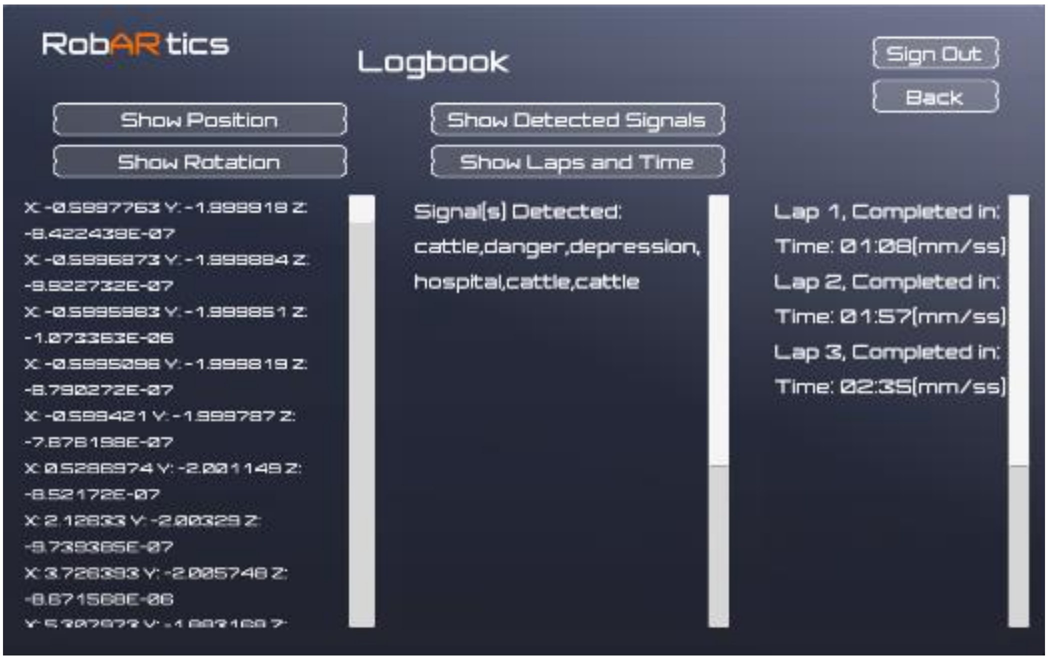


Figure 4. 26: Logbook showing Position, Signs Detected, Laps and its respective times

5. User Tests

To test the usefulness and usability of RobARtics, a usability test was conducted with several participants to check how well it helped users without any background in robotics understanding what is happening during competitions by giving them additional information, how well they reacted when interacting within AR environments, especially with mobile devices and what benefits could it provide to the more experienced users when assisting a competition.

5.1. Protocol

The test was presented to 5 different users, with varied knowledge about robotics and AR, aiming to represent the personas described in the third chapter. One user had no background knowledge in neither robotics nor AR, two users had previously interacted in AR environments and remaining two users were both familiar with robotics and AR. They were all male students, aged from 23 to 24 years old.

Before starting the questionnaire, users were presented with a privacy declaration, where it was established a set of statements, such as data gathering, that they had to agree, in order to proceed. This declaration is written in Portuguese and can be viewed in Appendix 1.

The test was composed by 3 parts. The first part was a questionnaire to get some demographic information about the users, such as how familiar they were with robotics or if they had interacted with AR before, and acquaint users with robotic competitions and what they were going to see while using the application. Initially, they watched a video of ROTA performing some laps around a track, which included an obstacle and road signs located around the track, similarly to the competitions this robot is used in. This video was representative of a simulation done in the Gazebo simulator and it intent was to represent a possible point of view from someone attending a robotic competition. While watching it, users were given an overview of what they were seeing on the video, the drawbacks of not having any information related to the competition or the robot besides what they were able to see right away and the benefits that the application would provide during competitions.

In the second part, users had to perform several tasks which tested all the fundamental application features. As participants were performing the tasks, much information, like the time it took to complete them, was recorded, alongside the accomplishments during each task, if there were any errors, whether they were caused by the users or issues during the application's usage, difficulties each of the participants had completing the tasks and if they required help understanding or performing what was asked.

At the end of each task, there were several questions, which could be answered with information obtained as users were performing the task. This meant to assess the user's perception of each task and if they understood what they were asked to do. The third part of the test was composed by 6 statements, based on the SUS[17], which are the following:

1. "I think that I would like to use this system frequently."[17]
2. "I found the system unnecessarily complex."[17]
3. "I thought the system was easy to use."[17]
4. "I think that I would need the support of a technical person to be able to use this system."[17]
5. "I found the various functions in this system were well integrated."[17]
6. "I thought there was too much inconsistency in this system."[17]
7. "I would imagine that most people would learn to use this system very quickly."[17]
8. "I found the system very cumbersome to use."[17]
9. "I felt very confident using the system."[17]
10. "I needed to learn a lot of things before I could get going with this system."[17]

In the context of this application, it didn't make sense using these exact metrics for evaluation, with statements like number 9 being somewhat redundant to ask, given the peculiar nature of AR systems. The 6 chosen statements were found to be the most relevant metrics to take notes from and analyse how usable was the application.

[17] https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html Visited in 23/10/19

Users had to rate them from 1 to 5, according to their perception of the system, with opportunities for them to leave some feedback on what it could be improved, along with what they found most difficult to follow or complete. The full questionnaire can be view in Appendix 2. Answers could be given in Portuguese or English.

After the users were done, their answers were all collected and are further analysed in the following section.

5.2. Results

This section presents the main results obtained from the observation of the users while they performed the tasks as well as from their answers to the questionnaires.

5.2.1. Task 1 – View Robot Details

The first task consisted on users logging in the application using their participant number. Then, the app main screen would appear and they were given the opportunity to explore the interface before starting. Once they felt ready, the objective was to indicate what were the robot's sensors, which involved their ability to navigate through menus.

All participants were able to find the correct answer, which was "Laser", without much trouble. They found the button, which took them to the screen where it was the information they needed to answer the question to be clearly visible. None of the participants asked for clarifications or help to perform the task.

5.2.2. Task 2 – Planned Trajectory, Signal Identification, Speed and Direction

In the second task, users had to observe a simulation of the robot, similar to the video presented before starting the test, being drawn on top of the application's display track, and before the robot completed the first lap, they were asked to turn on *Signal Identification* and observe the results.

The results for this task were varied. For the first question, which asked participants what the robot's speed was before stopping, some of them had trouble giving an answer, since they didn't pay attention the speed value before the robot had stopped. As such, they were given a replay of the simulation (Figure 5.1). Overall, all the given answers were correct, although not every user identified the units the speed was measured.

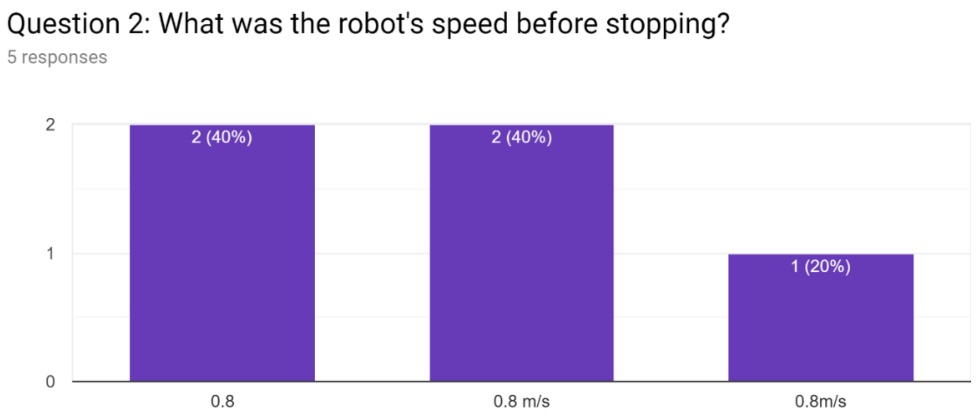


Figure 5. 1: Answers for the first question of Task 2

The second question of this task asked participants to indicate the robot's direction before stopping, by giving the corresponding cardinal direction. The responses were mostly correct. Users had no issue locating where the robot's direction was being displayed, although some users didn't remember the intercardinal directions. As such, they answered with "East", the closest direction to "North East", which was the correct answer. The others answered with the initials NE.

The last question asked users how many of the signs detected by the robot were incorrect. The responses were all correct, with every user answering one signal. They were able to tell the difference between the signs placed on the track and the ones detected by the robot.

A common issue that occurred during this task was the fact that most of the participants forgot they had to turn on the *Signal Identification* option before the robot completed the first lap, which led to distinct answers in some of the latter questions. The less experienced participant only focused on the augmented track and didn't notice that information was being displayed on the right side of the track, resulting in a repetition of simulation.

5.2.3. Task 3 – Previous Trajectory and Signal Details

The third task consisted in users recreating the track they just had seen through the simulation and identify what was the nearest signal when the robot goes off track, indicating its name and coordinates, using the applications capabilities of displaying increasing information about the road signs on track. Every user was able to correctly identify the right signal. While some of them first clicked on the signs detected by the robot, they eventually realized that the signal details asked were referring to the ones located on the track.

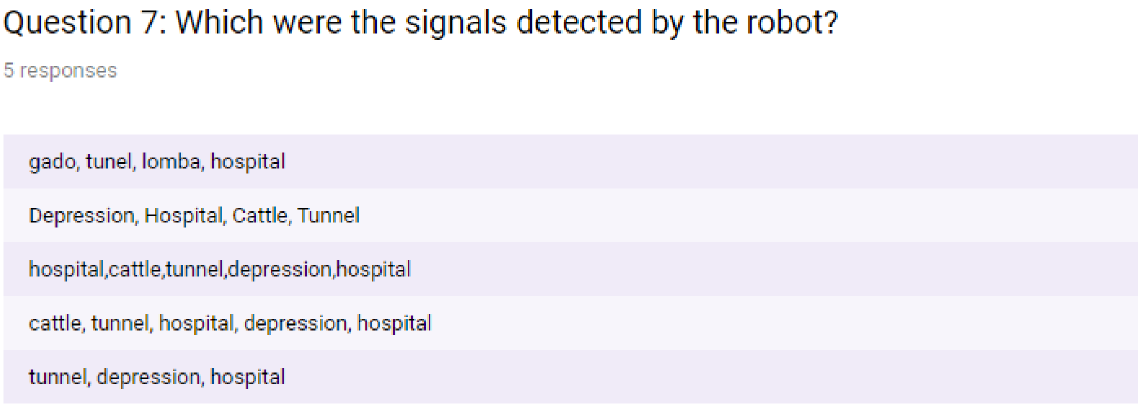
One criticism given by a user during this task was the fact that the information about the signs when they were pressed could overlap the information displayed on the application's canvas, such as the robot's speed and position coordinates. Besides this, no interventions were made during this task.

5.2.4. Task 4 – Ghost Racer and Lap Times

The fourth task involved users playing a record of 3 different "ghosts", each representing distinct laps from the robot and observing the time each "ghost" took to complete a lap. After the "ghosts" reached the finish line, users had to point out what was the color of the "ghost" representing the robot's fastest lap. Once again, every user was able to answer correctly the grey color. While the users experienced in AR and robotics found really easy to locate the colors of the "ghosts", some of the other users took a bit more time due to the fact that this information wasn't being displayed on track, where they were putting more focus on, but on the right side of it, below the other information related to the robot. All users were able to complete the task without help.

5.2.5. Task 5 – Logbook

The final task inquired users to visit the *Logbook* and answer the names of the signs that were detected by the robot and what was the time of the lap it completed during the first task simulation. The answers given in the first question were rather disparate, since users activated the *Signal Identification* option in Task 1 at different times (Figure 5.2). Indicating the lap time was correct across all users, although the repetition of the task by the same user previously mentioned caused some incongruences in the lap times


Question 7: Which were the signals detected by the robot?

5 responses

gado, tunel, lomba, hospital

Depression, Hospital, Cattle, Tunnel

hospital,cattle,tunnel,depression,hospital

cattle, tunnel, hospital, depression, hospital

tunnel, depression, hospital

Figure 5. 2: Answers for the first question of Task 5

Overall, this task was the most straight forward out of the 5. Users had no trouble finding the buttons that showed them the information they need to complete the questions and found the buttons to be an easy method of selecting which information they wanted on screen.

5.2.6. Completion times

Table 2 presents the completion times for every user in each task. The time is represented in seconds.

	Task 1	Task 2	Task 3	Task 4	Task 5	Total Duration
Participant 1	30	190	70	40	13	343
Participant 2	43	236	93	45	17	434
Participant 3	11	216	51	30	11	320
Participant 4	57	222	69	59	19	426
Participant 5	25	88	60	43	28	245
Average Time	33	172	69	43	17	335

Table 2: Time it took for each participant to complete each task, including overall time, average and standard deviation values.

Task 1 was one of the most accessible tasks to complete for all users. As such, the completion times for this assignment was relatively low, with an average time of completion of 33.34 seconds. This could be explained by the fact that nowadays, given the involvement everyone has with technology, like mobile applications and UI's in general, the navigation through menus and buttons has become rather intuitive.

Task 2 had the most diverse completion times, with an average time of conclusion of 172 seconds. Users with experience in both AR and robotics had no strife completing the assignment, getting the shortest completion times. Only one participant was able to complete this task without asking for help, as he took his time to read the instructions properly, at the expense of taking a bit more time to complete the task. This was the first task that tested some of the RobARtics core features. This suggests that the application might have a small learning curve for brand new users being introduced to AR in general, while being easy to pick up for more experienced users.

Task 3 had, once again, less scattered results, with an average time 69 seconds. This task was more straightforward than the previous one and the completion times are indicative of how much time did users spend experimenting with the track signs, as they quickly answered the question imposed during the task. The smaller conclusion times may indicate that after the first impression, user awareness to the system increased.

This point is backed by the low completion times of Task 4, being even lower than the previous tasks, where users took an average of 43 seconds to complete. The small time variations were due to the pace at which the user was able to locate where the colors of the ghosts were being shown. Nonetheless, it showed greater progress in the user's understanding of the system.

The final task was akin to Task 1 in terms of proficiency, as it was more of an interface navigation type of task. As such, it had an average completion time of 18 seconds and the lowest value of all tasks. These results showed that users had no problem interacting with information toggled by buttons. An interesting fact to point out is that the *Logbook* was one of the few components that didn't made use of AR, which suggests that users tend to feel more comfortable when they interact with a more familiar type of interface.

5.2.7. Post questionnaire Evaluation & Feedback

As it was previously mentioned, users were given at the end of the test multiple usability statements about the application they had just testing. The complete answers can be found in Appendix 3.

Most users understood well what the goals were and generally found the volume of information displayed on screen to be adequate.

The type of device which the application was built for had more varied opinions. While none of the users agreed that a smartphone was outright a poor choice to be used during robotic competitions, one of them felt that the use of mobile devices in that context was impractical, since it required users to look at the robot through another screen, which could be intrusive while trying to watch the competition normally and had inconveniences, such as having to hold them for some time and the risk of running out of battery.

Users found that navigating through the app was manageable, but one of the users felt that the checkboxes used to turn on some of the features were small and didn't contrast well enough. Another user said that he felt that the check box buttons weren't fully responsive due to their small size.

The opinion of users referring to the need of help from a more qualified person while using the application reflected the amount of help they needed while they took the test. Users whose perception of the tasks or the application itself was good gave the higher ratings, while the ones that had more difficulties gave a lower rating.

The most divisive opinion overall was related to the application of AR in the context of robotics competitions. Surprisingly, the users that didn't know anything about robotics found AR to be an effective and entertaining way of providing assistance, in this case, to the public audience of these competitions, thought that the AR components in the application were well implemented, despite some of the difficulties they found while executing the test. However, users that had grasp in both fields were more on the critical side about the usage of AR in this scenario. One user said that while he thought that mobile devices were perfectly suited for AR applications, he felt that the information presented in AR about the robot would be better suited for being displayed in a similar manner as the *Logbook*, a more traditional UI. The other highlighted the marker-based detection, saying that it was an effortless approach and that the application detected quite well the marker, but believed that AR in this context would be more useful if it provided users with an augmented version of the robot to explore, rather than information about the robot.

Almost every user gave a suggestion about what they would improve or change. They were the following:

- Add an option to see the whole canvas of the application instead of always relying on AR.
- Give users feedback by displaying a message indicative of whether an option is checked or not.

- Replace the current directional system with an arrow that rotates according to the robot's direction or add subtitles for the current method with cardinal coordinates, like North, South…
- Make the check boxes more distinguishable and have just one signal information active at a time to avoid overlapping between text.

5.2.1. Experts' feedback

After the testing with users was done, the application was shown to four researchers within the field of robotics that had actually taken part in robotic competitions, particularly in autonomous driving challenges, so they could provide more insight about what may be currently wrong with the application, improvements that can be made and their overall opinions on the usage of AR within the context of robotic competitions, since they had participated in this kind of events.

Overall, every user understood the application's purpose and what contributions it could give in these environments. Some researchers commended the fact that the marker-based approach worked perfectly using a smartphone. Additionally, they thought that every feature integrated in the application improved or increased the information that could be obtained from the competition itself. One of the researchers said the *Ghost Racer* was useful when there was the need to test different path finding algorithms for the robot, to see which one had the best results in terms of lap time and the robot's on-track accuracy. Other researchers said that the *Signal Detection* feature was advantageous, since some details might escape from the team if they are not always paying attention to their robot, thus not seeing if the robot made the correct signal identification.

Some criticism included the fact that most of the application features were not parameterizable. The researchers expressed that they would like to have control over certain aspects, such as the rate the positions of the robot are updated on track and the possibility to select which topics they wanted to subscribe. The former would allow accommodate different track sizes. For example, longer tracks required the position to be updated at a higher rate, so the robot's position is displayed more accurately, while the latter would add a new layer of usability, since researchers could choose to see the position of other robots besides ROTA, which is currently hard coded.

When asked about their thoughts on the AR within the context of robotics how it was integrated with the application, the answers were varied. Two researchers thought that AR was a helpful tool, due to the fact that the tracks which the robots traverse in competitions are of large proportions and they would need to set up one or more cameras to cover the entire track. The application provided a solution to this, since they could see the track and the trajectory of the robot by simply pointing a device towards the robot's respective marker. Still, they expressed that they would rather have a desktop version of the application, usable through a webcam, since they believed that a mobile device would be better suited for the public watching the competition. The other two researchers shared the same point mentioned previously, as they said that for software development of the robot, a mobile device wasn't as practical as a computer, since they had to pick it up and point the device towards the robot each time they wanted to see the information.

Conversely, they concluded that the application's concept was well thought out, especially when applied to public, given that the integration of AR provided an innovative and interactive way of clarifying what happens during the robot trials and gave the spectators a means to get more involved in the event. They also found the application to be easily understandable with a bit of practice.

As for feedback and future work improvements, the researchers gave several suggestions:

- Improve the *Signal Identification* feature by displaying the real signal and the one detected by the robot side by side, to make the comparison more visible and if the robot hadn't detect a signal in a long time, the icons of the signs it detected previously would be cleaned, to make the interface less cluttered.
- Add the option to select between an assortment of tracks, representative of other challenges or tracks from previous editions.
- Add the possibility to track multiple markers, each representative of a different robot, to give users the ability to check information from more robots than ROTA.
- Expand the AR functionalities by adding the opportunity of inserting virtual obstacles on track for the robot to get around or making the information available proportional to the current zoom (i.e. the displayed information varies the closer or the farther the user is pointing at the marker).
- Show an augmented representation of the robot's laser.
- Develop an API to make the application usable by robots programmed with other frameworks besides ROS.

6. Conclusions and Future Work

The work on this dissertation presented a possible alternative to help users understand what happened during a robotic competition using AR, which was found to be a medium uncommonly used in this area. Most of the research done showed many of the possible AR applications, and more akin to the theme of the dissertation, AR was found to be more prominent in human robot interaction.

The first and second chapters gave a general overview of the objectives tackled throughout the dissertation, its structure and gave more insight about how robotic competitions work, the rules, challenges and how they are organized in Portugal, as well as providing a state-of-the-art review for AR, presenting how it has been applied in the field of robotics, including some of the evaluation methods utilized.

The third, fourth and fifth chapters helped to further contextualize the usage of RobARtics, through personas and scenarios, while presenting the main technologies used for its development, detailed the development of the application, the first draft and user feedback and illustrated every feature integrated, discussed the evaluation method used and discussed the results obtained by the participants drawing conclusions based on their performance and opinions, respectively.

The development of the application was relatively smooth. Unity has a large and friendly community, with much documentation concerning its scripting API, while the robotic components, such as the simulator, were provided by a third party. The most difficult aspects that delayed its development were making sure that the application was appropriate to every persona underlined in chapter 3, which caused some redesigns to its UI multiple times.

As it was observed during testing, AR is still a relatively new concept for most people and has much to be improved before reaching a certain level of general acceptance. With the developed application it was possible to see how users would react to the application of AR, not only in a robotic competition scenario, but also to using it through mobile devices.

The results obtained were surprisingly mixed. Many of the users that performed the test without any skills in robotic had some trouble orienting themselves while performing some of the tasks and often asked for help. In contrast, the users that previously had knowledge in robotics had also experienced AR in any other form. As such, they found the application interface easy to navigate and had little to no trouble completing the tasks. Among participants of such competitions, there were also several opinions that showed and discussed how AR should be applied in this kind of scenario, with them overall agreeing that AR was indeed a good choice for new method of assisting robotic competitions, although they would rather have it applied to desktop over mobile devices.

While this investigation sparked interesting results, there is still much room for improvement. The first and foremost important aspect that should be addressed is to develop a version of this application applicable to desktop devices. This way, users that preferred the info displayed on a monitor could have an alternative if they didn't feel like using a mobile device. Next, one of the application's weaknesses was the fact that it was confined to a single marker at a time. Further development should include the possibility of multiple tracking, allowing the visualization of information relative to various robots at the same time. The heatmap feature, which was originally scrapped, could be implemented on top of the track while a robot is still running or after it finished the competition. This would make for a more visually interesting enhancement of the track and could provide researchers and referees attending competitions with another mean to study the robot's trajectory.

The idea of using AR during robotics competitions still has many uncharted territories to be explored. The general acceptance for AR over the years has generally increased, but there are still multiple barriers that need to be overcome in order to take its full potential. In contrast, robotic competitions still tend to be more of a niche event, where many of its attendants tend to be people involved in robotics. The work done by this dissertation is hoped to provide some basis on future attempts at developing a system with analogous intentions.

References

Alkhamisi, A. O., & Monowar, M. M. (2013). Rise of Augmented Reality: Current and Future Application Areas. *International Journal of Internet and Distributed Systems*, *01*(04), 25–34. https://doi.org/10.4236/ijids.2013.14005

Azuma, R. T. (1997). A Survey of Augmented Reality. *Presence: Teleoperators and Virtual Environments*, *6*(4), 355–385. https://doi.org/https://doi.org/10.1162/pres.1997.6.4.355

Azuma, R. T. (2017). Making Augmented Reality a Reality. *Applied Industrial Optics: Spectroscopy, Imaging and Metrology 2017*. https://doi.org/https://doi.org/10.1364/3D.2017.JTu1F.1

Bartneck, C., Soucy, M., Fleuret, K., & Sandoval, E. B. (2015). The robot engine - Making the unity 3D game engine work for HRI. *Proceedings - IEEE International Workshop on Robot and Human Interactive Communication*, *2015-Novem*, 431–437. https://doi.org/10.1109/ROMAN.2015.7333561

Brizzi, F., Peppoloni, L., Graziano, A., Di Stefano, E., Avizzano, C. A., & Ruffaldi, E. (2018). Effects of Augmented Reality on the Performance of Teleoperated Industrial Assembly Tasks in a Robotic Embodiment. *IEEE Transactions on Human-Machine Systems*, *48*(2), 197–206. https://doi.org/10.1109/THMS.2017.2782490

Chiu, H. P., Murali, V., Villamil, R., Kessler, G. D., Samarasekera, S., & Kumar, R. (2018). Augmented Reality Driving Using Semantic Geo-Registration. *25th IEEE Conference on Virtual Reality and 3D User Interfaces, VR 2018 - Proceedings*, (March), 423–430. https://doi.org/10.1109/VR.2018.8447560

Cooper, A., Reimann, R., & Cronin, D. (2007). *About Face 3.0: The Essentials of Interaction Design*. *Information Visualization* (Vol. 3). Indianapolis: Wiley Publishing, Inc. https://doi.org/10.1057/palgrave.ivs.9500066

Craig, A. B. (2013). *Understanding Augmented Reality*. *Understanding Augmented Reality*. 225 Wyman Street, Waltham, MA 02451, USA: Morgan Kaufmann. https://doi.org/10.1016/c2011-0-07249-6

De Pace, F., Manuri, F., Sanna, A., & Zappia, D. (2018). An Augmented Interface to Display Industrial Robot Faults. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *10851 LNCS*, 403–421. https://doi.org/10.1007/978-3-319-95282-6_30

Dünser, A., & Billinghurst, M. (2011). Evaluating Augmented Reality Systems. *Handbook of Augmented Reality*, 289–307. https://doi.org/https://doi.org/10.1007/978-1-4614-0064-6_13

Elsdon, J., & Demiris, Y. (2018). Augmented Reality for Feedback in a Shared Control Spraying

Task. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 1939–1946. https://doi.org/10.1109/ICRA.2018.8461179

Furht, B., Ivkovic, M., Anisetti, M., Damiani, E., Ceravolo, P., & Carmigniani, J. (2010). Augmented reality technologies, systems and applications. *Multimedia Tools and Applications*, *51*(1), 341–377. https://doi.org/10.1007/s11042-010-0660-6

Grandi, J. G., Debarba, H. G., Nedel, L., & Maciel, A. (2017). Design and Evaluation of a Handheld-based 3D User Interface for Collaborative Object Manipulation. *CHI '17 Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, 5881–5891. https://doi.org/10.1145/3025453.3025935

Green, S. A., Chase, J. G., Chen, X. Q., & Billinghurst, M. (2008). Evaluating the augmented reality human-robot collaboration system. *International Journal of Intelligent Systems Technologies and Applications*, *8*(1–4), 130–143. https://doi.org/10.1504/IJISTA.2010.030195

Hartmann, M., Viehweger, M., Spitzer, M., Desmet, W., Watzenig, D., & Stolz, M. (2018). "Pedestrian in the Loop": An Approach Using Augmented Reality. *SAE Technical Paper Series*, *1*(February), 10–15. https://doi.org/10.4271/2018-01-1053

Irawati, S., Green, S., Billinghurst, M., Duenser, A., & Ko, H. (2007). Move the couch where? : Developing an augmented reality multimodal interface. *Proceedings - ISMAR 2006: Fifth IEEE and ACM International Symposium on Mixed and Augmented Reality*, 183–186. https://doi.org/10.1109/ISMAR.2006.297812

Kim, K., Member, S., Billinghurst, M., Member, S., Bruder, G., Been-Lirn Duh, H., & Welch, G. F. (2018). Revisiting Trends in Augmented Reality Research: A Review of the 2nd Decade of ISMAR (2008-2017). *Transactions on Visualization and Computer Graphics*, *24*(11), 2947–2962. https://doi.org/10.1109/TVCG.2018.2868591

Ko, S. M., Chang, W. S., & Ji, Y. G. (2013). Usability Principles for Augmented Reality Applications in a Smartphone Environment. *International Journal of Human-Computer Interaction*, *29*(8), 501–515. https://doi.org/10.1080/10447318.2012.722466

Krevelen, D. W. F. van, & Poelman, R. (2010). A Survey of Augmented Reality Technologies, Applications and Limittions. *The International Journal of Virtual Reality*, *9*(2), 1–20. https://doi.org/10.1155/2011/721827

Lebeck, K., Kohno, T., & Roesner, F. (2016). How to Safely Augment Reality : Challenges and Directions. *Proceedings of the 17th International Workshop on Mobile Computing Systems and Applications*, 45–50. https://doi.org/10.1145/2873587.2873595

Marner, M. R., Smith, R. T., & Thomas, B. H. (2015). Mapping 2D input to 3D immersive spatial augmented reality. *2015 IEEE Symposium on 3D User Interfaces, 3DUI 2015 - Proceedings*, (June), 171–172. https://doi.org/10.1109/3DUI.2015.7131755

Mekni, M., & Lemieux, A. (2014). Augmented Reality: Applications, Challenges and Future Trends. *Applied Computational Science*.

Michalos, G., Karagiannis, P., Makris, S., Tokçalar, Ö., & Chryssolouris, G. (2016). Augmented Reality (AR) Applications for Supporting Human-robot Interactive Cooperation. *Procedia CIRP*, *41*, 370–375. https://doi.org/10.1016/j.procir.2015.12.005

Milgram, P., & Kishimo, F. (1994). A taxonomy of mixed reality. *IEICE Transactions on Information and Systems*, *77*(12), 1321–1329.

Nikolaidis, E., Cope, A. J., Reina, A., Sabo, C., & Marshall, J. A. R. (2017). ARK: Augmented Reality for Kilobots. *IEEE Robotics and Automation Letters*, *2*(3), 1755–1761. https://doi.org/10.1109/lra.2017.2700059

Peng, H., Briggs, J., Wang, C.-Y., Guo, K., Kider, J., Mueller, S., … Guimbretière, F. (2018). Roma. *2018 Association for Computing Machinery*, 258–264. https://doi.org/10.4324/9781315856506

Rodrigues, J. G. das N. (2018). *Deep Learning Applied to Ackermann Steering Vehicle Driving*. MIECT, Universidade de Aveiro.

Ruffaldi, E., & Brizzi, F. (2016). CoCo - A Framework for Multicore Visuo-Haptics in Mixed Reality. *Springer International Publishing Switzerland 2016*, *1*, 267–280. https://doi.org/10.1007/978-3-319-40621-3

Ryffel, M., Zünd, F., Aksoy, Y., Marra, A., Nitti, M., Aydin, T. O., & Sumner, B. (2017). AR museum: A mobile augmented reality application for interactive painting recoloring. *Proc. GET*, (c).

Santos, M. E. C., Polvi, J., Taketomi, T., Yamamoto, G., Sandor, C., & Kato, H. (2015). Toward Standard Usability Questionnaires for Handheld Augmented Reality. *IEEE Computer Graphics and Applications*, *35*(5), 66–75. https://doi.org/10.1109/MCG.2015.94

Sprute, D., Tönnies, K., & König, M. (2017). Virtual Borders: Accurate Definition of a Mobile Robot's Workspace Using a RGB-D Google Tango Tablet. Retrieved from http://arxiv.org/abs/1709.00954

Stadler, S., Kain, K., Giuliani, M., Mirnig, N., Stollnberger, G., & Tscheligi, M. (2016). Augmented reality for industrial robot programmers: Workload analysis for task-based, augmented reality-supported robot control. *25th IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN 2016*, 179–184. https://doi.org/10.1109/ROMAN.2016.7745108

tom Dieck, M. C., & Jung, T. (2015). A theoretical model of mobile augmented reality acceptance in urban heritage tourism. *Current Issues in Tourism*, *21*(2), 154–174. https://doi.org/10.1080/13683500.2015.1070801

Ullmer, B., & Ishii, H. (2001). Human-Computer Interaction in the New Millienium, *39*(3), 579–601. https://doi.org/10.1109/TAP.2010.2055810

Valentini, P. P. (2018). Natural interface for interactive virtual assembly in augmented reality using Leap Motion Controller. *International Journal on Interactive Design and Manufacturing*, *12*(4), 1157–1165. https://doi.org/10.1007/s12008-018-0461-0

Walker, M., Hedayati, H., Lee, J., & Szafir, D. (2018). Communicating Robot Motion Intent with Augmented Reality. *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, 316–324. https://doi.org/10.1145/3171221.3171253

Weinstock, M. (2015). Checkmate. *Hospitals and Health Networks*, *89*(7), 10.

Zhou, F., Been-Lirn Duh, H., & Billinghurst, M. (2008). Trends in Augmented Reality Tracking, Interaction and Display: A Review of Ten Years of ISMAR. *ISMAR '08 Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, 193–202. https://doi.org/10.1109/ISMAR.2008.4637362

Zolotas, M., Elsdon, J., & Demiris, Y. (2018). Head-Mounted Augmented Reality for Explainable Robotic Wheelchair Assistance. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (September), 1823–1829. https://doi.org/10.1109/iros.2018.8594002

Appendix

Appendix 1 – Consentiment Declaration

Declaração de Consentimento Informado

O estudo consiste na utilização de uma aplicação com capacidades de Realidade Aumentada, para a realização de um conjunto de tarefas. Os dados obtidos vão contribuir para uma avaliação pós-experiência da solução desenvolvida, de modo a compreender as suas limitações e quais os cenários em que se enquadra.

Como participante desta investigação, declaro estar consciente que terei de participar numa experiência inserida num ambiente que irá conter conteúdo virtual aumentado, onde a minha experiência poderá ser filmada e/ou fotografada. Declaro igualmente ter conhecimento que todos os dados recolhidos neste estudo serão usados apenas para fins científicos, garantindo o anonimato dos participantes.

Compreendo que, em qualquer altura, tenho a liberdade de retirar a minha autorização ou recusar participar no estudo.

Caso tenha quaisquer questões ou problemas relativamente à minha participação, contactarei o investigador João Alves (jbga@ua.pt), o investigador Bernardo Marques (bernardo.marques@ua.pt), o Prof. Paulo Dias (paulo.dias@ua.pt), ou a Prof. Beatriz Sousa Santos (bss@ua.pt).

______________________________ ______________________________

(Data) (Assinatura do Participante)

Appendix 2 – Questionnaire: RobARtics

All information provided is entirely confidential and will not be distributed or used for any purpose other than this test.

Thank you.

Participant:

Age:

Gender:

☐ Male

☐ Female

Occupation:

	Not acquainted	***Little acquainted***	***Somewhat experienced***	***Acquainted***	***Very acquainted***
How acquainted are you with robotics?	1	2	3	4	5

Have you ever used Virtual Reality?

☐ Yes

☐ No

Have you ever used Augmented Reality?

☐ Yes

☐ No

Part 1

Now you'll be asked to perform a series of tasks, as well as answer some questions based on the respective tasks.

Pre-Task

- Open intro.mp4.
- Listen to the instructions.
- Enter your name, participant number and press “Start RobARtics”.

Task 1

- Point the mobile device towards the marker and check all the available options.
- Click on "View Robot Details" and view the information on display.

Question 1: What were the robot sensors?

Task 2

- Point the mobile device towards the marker. You'll see that the "Planned Trajectory" option is on.
- Observe the track and let the simulation run. When the speed value changes to 0 m/s, it means it has finished.
- When the robot is close to complete 1 lap, check the "Signal Identification" option and watch the results.

Question 2: What was the robot's speed?

Question 3: Question 3: What was the robot's direction in its last position? Answer by giving its cardinal direction (North, South...)

Question 4: How many signs did the robot detect incorrectly?

Task 3

- Tap the signal icons to view more information about them.
- Press "Clean" to clear the track. Then check the "Previous Trajectory" option.

Question 5: Near which signal did the robot go off-track? Indicate its name and coordinates.

Task 4

- Uncheck the “Previous Trajectory” option, then press “Clean”.
- Check the “Ghost Racer” option and press the “Play All” button.

Question 6: What’s the color of the trajectory representing the robot’s fastest lap?

Task 5

- Press "Logbook", check the information available and answer the following questions:

Question 7: Which were the signs detected by the robot?

Question 8: What are the times for each of the laps completed by the robot?

Part 2 - Post-Experience Questionnaire

Please help us complete this questionnaire, based on the experience you have just completed.

All information provided is entirely confidential and will not be distributed or used for any purpose other than this test.

Rate the following statements according to your perception of the system. Thank you.

	Completely disagree	***Disagree***	***Indifferent***	***Agree***	***Completely agree***
The purpose of this system was clear	1	2	3	4	5
The type of device where the system was employed was adequate	1	2	3	4	5
The navigation interface was complex	1	2	3	4	5
The amount of information displayed was appropriate	1	2	3	4	5
The system was easily comprehensible without the help of someone more qualified within the field of robotics	1	2	3	4	5
The use of AR was useful in this context	1	2	3	4	5

What were the main difficulties?

Please add any suggestions you may consider relevant.

Please add additional comments you may consider relevant.

Monitoring Experience using Augmented Reality

Date: ______________________________

Participant: __________

Task	Accomplish goals?	Errors?	Difficulties?	Required help?
T1				
T2				
T3				

T4				
T5				

Full Duration:

____

Observations:

____

____

____

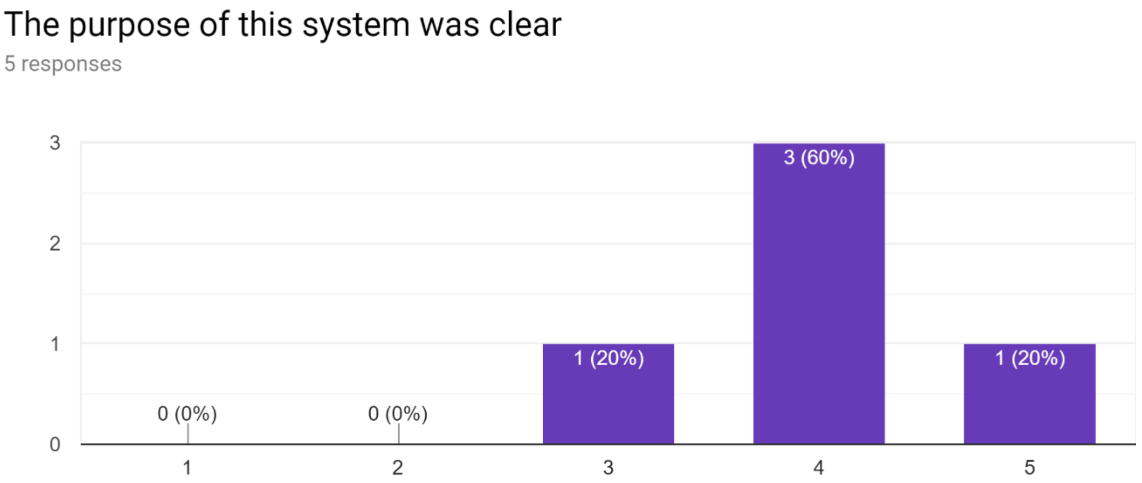
____

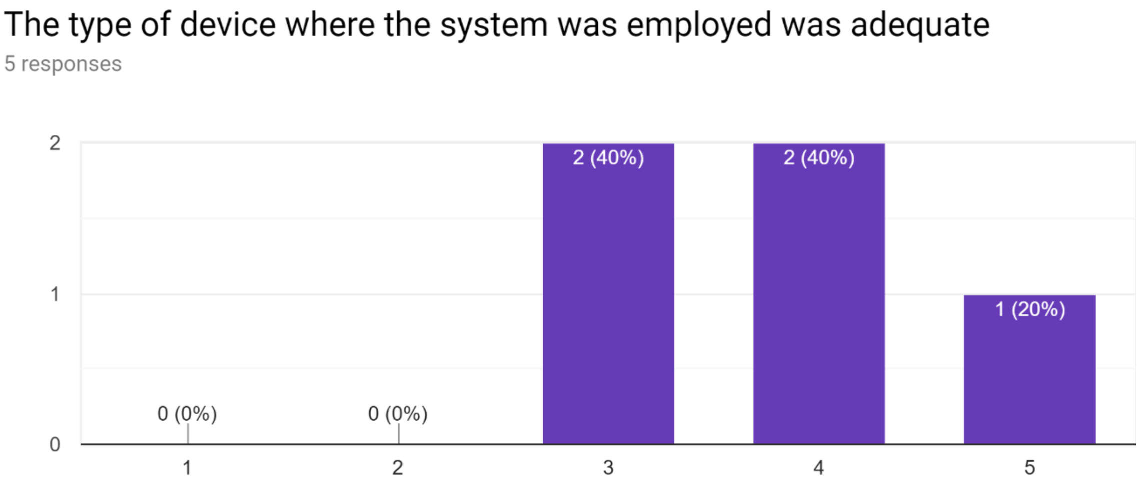
____

____

____

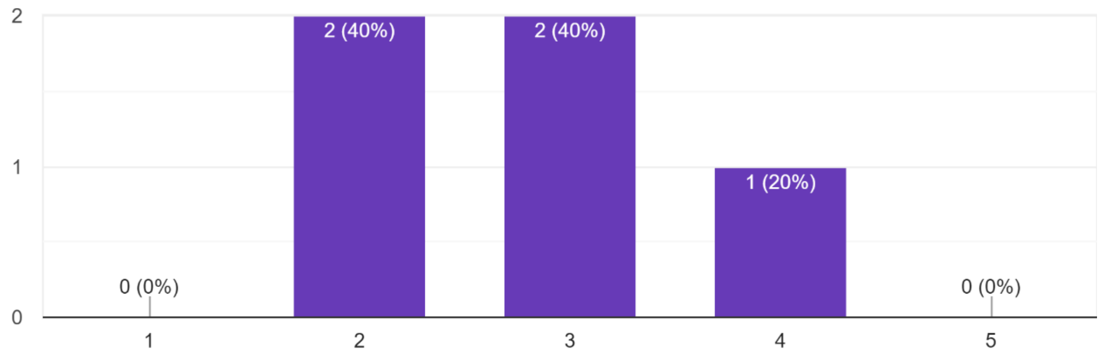
Appendix 3 – Post Questionnaire Results





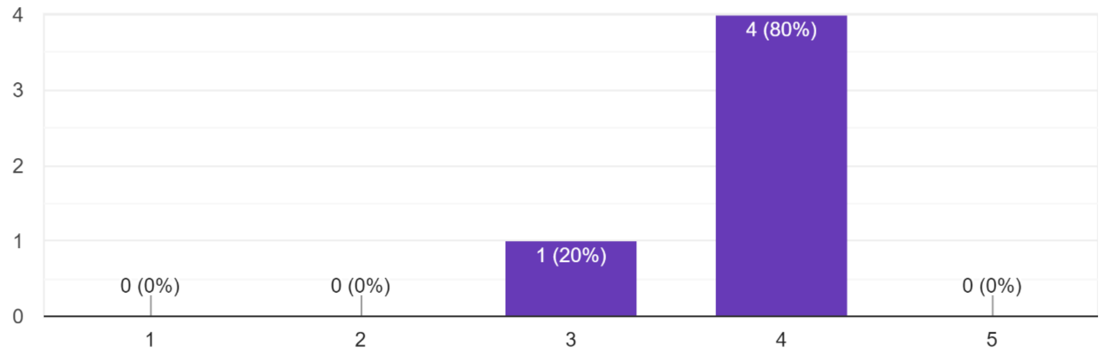
The navigation interface was complex

5 responses



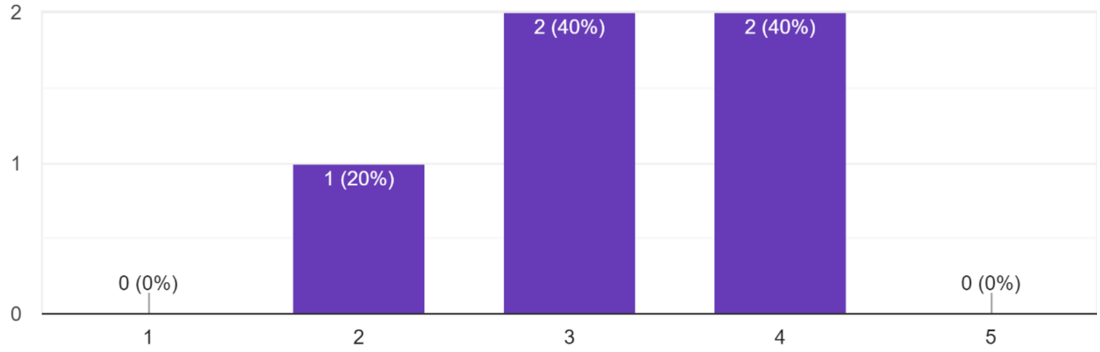
The amount of information displayed was appropriate

5 responses



The system was easily comprehensible without the help of someone more qualifed within the field of robotics

5 responses



The use of AR was useful in this context

5 responses