



**NOVA**  
NOVA SCHOOL OF  
SCIENCE & TECHNOLOGY

DEPARTMENT OF  
ELECTRICAL AND COMPUTER  
ENGINEERING

ARTUR MIGUEL REIS GONÇALVES COLLINGRIDGE  
SEABRA

Undergraduate Degree in Electrical and Computer  
Engineering

# IMMERSIVE TRAINING THROUGH VR AND AI

A PROOF OF CONCEPT FOR ROBOT CONTROL  
TRAINING

MASTER IN ELECTRICAL AND COMPUTERS ENGINEERING

NOVA University Lisbon  
September, 2024





# IMMERSIVE TRAINING THROUGH VR AND AI

## A PROOF OF CONCEPT FOR ROBOT CONTROL TRAINING

**ARTUR MIGUEL REIS GONÇALVES COLLINGRIDGE SEABRA**

Undergraduate Degree in Electrical and Computer Engineering

**Adviser:** João Rosas,  
Assistant Professor, NOVA University Lisbon

**Examination Committee:**

**Chair:** André Teixeira Bento Damas Mora,  
Assistant Professor, FCT-NOVA

**Rapporteurs:** Patrícia Alexandra Pires Macedo,  
Assistant Professor, EST/IPS

**Adviser:** João Rosas,  
Assistant Professor, FCT-NOVA



## **Immersive Training through VR and AI**

Copyright © Artur Miguel Reis Gonçalves Collingride Seabra NOVA School of Science and Technology, NOVA University Lisbon.

The NOVA School of Science and Technology and the NOVA University Lisbon have the right, perpetual and without geographical boundaries, to file and publish this dissertation through printed copies reproduced on paper or on digital form, or by any other means known or that may be invented, and to disseminate through scientific repositories and admit its copying and distribution for non-commercial, educational or research purposes, as long as credit is given to the author and editor.

This document was created with Microsoft Word text processor and the NOVAtesis Word template [1].







## ACKNOWLEDGMENTS

I would like to start by expressing my gratitude to my professor, João Rosas, for his guidance, support, and consistent feedback throughout the process of writing this dissertation. Despite his often busy schedule, he always made time to provide me with his feedback and support whenever I needed it.

I also want to extend my appreciation for Arsenal do Alfeite S.A.'s support. Without the resources and tools they provided, the completion of this dissertation would have never been possible.

My heartfelt thanks go to my family, whose constant encouragement, love, and understanding have been the foundation of my perseverance. Their support has made it possible for me to reach this point in the first place, and for that, I am truly thankful.

I would also like to acknowledge my friends, who have been a source of motivation and positivity. Their companionship has helped me maintain balance and perspective throughout this demanding process.

Finally, I am thankful to everyone else who, in one way or another, has supported and contributed to this dissertation.



“It’s going to be interesting to see how society deals with artificial intelligence,  
but it will definitely be cool.” (Colin Angle).



## ABSTRACT

We live in a time in which the importance of efficient teaching methods in the professional world is immense and continuously growing. In the face of this, numerous approaches and ideas on this subject have emerged.

However, with today's technology, it is now possible to integrate generative artificial intelligence with virtual reality and digital twins to create efficient and autonomous learning environments. This dissertation demonstrates this concept, within the context of teaching the user how to control a manufacturing robot.

While there are existing ideas on using these tools for educational purposes, the concept of combining them to create a fully immersive learning environment has not been sufficiently explored. One reason for this is that most of these tools, particularly generative AI, are relatively new, and some have been expensive or difficult to gain access to in recent years. However, this is expected to change over time as these technologies improve and become more accessible.

The focus of this project was then on developing a proof-of-concept system that demonstrates the feasibility of integrating generative AI, Digital Twins (or some other type of virtual replica) and VR technologies, all within a single system, to provide the user with an immersive learning experience, in this case, for teaching robot controls. The system immerses the user in a VR environment, allows the user to control a virtual replica of a manufacturing robot, and supports the user with AI functionalities, in the form of an AI tutor or assistant.

The development process required a thorough analysis of available tools and methods for integrating them into one application. The main challenge was to make these tools function cohesively within the context of teaching and immersing the user in the virtual world.

By the end of development, the proof-of-concept system successfully demonstrates that creating such an immersive learning environment is indeed possible and it highlights the profound potential of these new technologies to revolutionize teaching methods.

**Keywords:** Generative AI, Virtual Reality (VR), Digital Twins, Virtual Replica, Immersive Learning, Autonomous Learning Environments, Manufacturing Robot Control, Integrative Technologies

## RESUMO

Vivemos numa época em que a importância de métodos de ensino eficientes no mundo profissional é imensa e tem vindo a crescer. Perante isto, têm surgindo diversas abordagens e ideias sobre este tema.

Contudo, com a tecnologia atual, torna-se agora possível integrar inteligência artificial generativa com realidade virtual e digital twins para criar ambientes de aprendizagem eficientes e autónomos. Esta dissertação irá demonstrar este conceito, dentro do contexto de ensinar o utilizador a controlar um robô de manufatura.

Embora já existam ideias sobre o uso destas ferramentas para fins educativos, o conceito de as combinar para criar um ambiente de aprendizagem totalmente imersivo não foi suficientemente explorado. Uma das razões para isso é que muitas destas ferramentas, particularmente a IA generativa, são relativamente recentes e algumas têm sido caras ou difíceis de obter nos últimos anos. No entanto, espera-se que isso mude à medida que, com o passar do tempo, estas tecnologias melhorem e se tornem mais acessíveis.

O foco deste projeto consistiu, portanto, no desenvolvimento de um sistema de prova de conceito que demonstre a viabilidade de integrar IA generativa, Gémeos Digitais (ou outro tipo de réplica virtual) e tecnologias de Realidade Virtual, todas num único sistema, com o intuito de proporcionar ao utilizador uma experiência de aprendizagem imersiva, neste caso, para o ensino de controlo de robôs. O sistema imerge o utilizador num ambiente de realidade virtual, permite-lhe controlar a réplica digital de um robô de manufatura e apoia o utilizador com funcionalidades de IA, na forma de um tutor ou assistente artificial.

O processo de desenvolvimento exigiu uma análise detalhada das ferramentas disponíveis e dos métodos para as integrar numa única aplicação. O principal desafio foi fazer com que estas ferramentas funcionassem de forma coesa no contexto de ensino e imersão do utilizador no mundo virtual.

Ao fim do desenvolvimento, o sistema de prova de conceito demonstra com sucesso que é de facto possível criar um ambiente de aprendizagem imersivo deste tipo, e destaca o potencial profundo que estas novas tecnologias têm para revolucionar os métodos de ensino.

**Palavras-chave:** IA generativa, Realidade Virtual, Digital Twins (Gêmeos Digitais), Replica Virtual, Aprendizagem Imersiva, Ambientes de Ensino Autónomos, Controlo de Robôs de Manufatura, Tecnologias de Ensino, Integração de Tecnologias

# CONTENTS

<b>1</b>	<b>INTRODUCTION.....</b>	<b>1</b>
1.1	Motivation.....	1
1.2	Main Objectives.....	2
1.3	Document Structure.....	2
<b>2</b>	<b>STATE OF THE ART.....</b>	<b>5</b>
2.1	Innovations in the Growing L&D Industry.....	5
2.1.1	Contextualization of the Current Needs in the L&D Industry.....	6
2.1.2	Applications in Lean Learning.....	8
2.2	Contextualization with Industries 4.0 and 5.0.....	9
2.3	Technologies Supporting Learning Paradigms.....	11
2.3.1	Digital Twins.....	12
2.3.2	Virtual Reality.....	14
2.3.3	Chat Bots.....	19
2.3.4	Game Engines.....	27
2.4	Related Works.....	32
2.4.1	Industrial Trends.....	32
2.4.2	Academic Research.....	35
2.4.3	Key Takeaways.....	37
<b>3</b>	<b>DEVELOPMENT.....</b>	<b>39</b>
3.1	System Architecture.....	39

3.1.1	Main Requirements.....	40
3.1.2	General Simulation Architecture.....	42
3.2	Realvirtual.io Framework .....	63
3.3	VR Integration.....	69
3.4	AI Integration .....	74
3.4.1	Chat Bot Integration.....	74
3.4.2	TTS and STT Integration.....	78
3.4.3	Chat Bot Preparation.....	78
<b>4</b>	<b>TESTS AND RESULTS .....</b>	<b>83</b>
4.1	Presentation of the Developed System.....	83
4.2	Testing Phase .....	89
4.2.1	General Tests.....	89
4.2.2	Tutorial Sequence Test.....	100
4.3	Results Phase.....	103
4.3.1	General Results.....	104
4.3.2	Possible Improvements .....	106
<b>5</b>	<b>CONCLUSION .....</b>	<b>109</b>
5.1	General Overview of the Development Process .....	109
5.2	Main Challenges.....	110
5.3	Most Relevant Results.....	111
5.4	Future Work.....	111

## LIST OF FIGURES

Figure 2.1 - Moodle is an example of an e-learning tool, in this case, a LMS [4] .....	6
Figure 2.2 - Evolution of the average cost associated with professional training between 2008 and 2021 [5].....	7
Figure 2.3 - Learning management system (LMS) [9].....	7
Figure 2.4 - Industry 4.0 [11] .....	10
Figure 2.5 - Industry 5.0 [14] .....	11
Figure 2.6 - Digital twin of a robotic arm [15] .....	12
Figure 2.7 - Digital twins in VR [18] .....	13
Figure 2.8 - Virtual Reality [19].....	14
Figure 2.9 - The Sword of Damocles, the first virtual reality helmet [20].....	15
Figure 2.10 - Augmented Reality [22].....	16
Figure 2.11 - Fully immersive VR [24] .....	17
Figure 2.12 - Graphical representation of the point at which American companies are, regarding the adoption of virtual reality technologies, according to a PwC study [26].....	18
Figure 2.13 - Graphical representation of the time required to complete training according to a PwC study [26] .....	18
Figure 2.14 - Google Bard (Gemini's ancestor) [27] .....	19
Figure 2.15 - The first Chat Bot, Eliza [28] .....	20
Figure 2.16 - Articles and studies associated with the terms chatbot, conversation agent, and conversation interface according to Scopus between 2000 and 2019 [29] .....	21
Figure 2.17 - Conversation with an airport's Task-Based Chat Bot [30] .....	22
Figure 2.18 - AI generated advertisement images [31].....	23
Figure 2.19 - Artistic drawing generated by AI [32] .....	24
Figure 2.20 - Chat Bot being used as a learning tool [34] .....	25

Figure 2.21 - Codewhisperer [36] .....	26
Figure 2.22 - Robotstudio [40] .....	27
Figure 2.23 - Virtual Factory in Omniverse [42] .....	28
Figure 2.24 - Robotic models in an Unreal Engine environment [47] .....	29
Figure 2.25 - Most used game engines in the United Kingdom by game developers in 2019 [49] .....	30
Figure 2.26 - Robotic arm in a Unity environment [51] .....	31
Figure 2.27 - Example of a Realvirtual.io robot visualized in a Unity 3D scene [53].....	32
Figure 2.28 - Volkswagen's Digital Reality Hub [56] .....	33
Figure 2.29 - Predicted increase in VR's market size [60].....	34
Figure 2.30 - Digital twin of a real laboratory [63].....	35
Figure 2.31 - The growth in publications about virtual reality in education from 1995–2016..	36
Figure 3.1 - General System Architecture.....	43
Figure 3.2 - Main Manager Block .....	45
Figure 3.3 - Snippet of code from the TutorialSequence coroutine function .....	47
Figure 3.4 - Pause screen during instruction delivery in guided mode .....	48
Figure 3.5 - Disabled buttons on the handheld controller in guided mode.....	48
Figure 3.6 - Sequence diagram representing part of the guided tutorial's progression .....	50
Figure 3.7 - AI Block.....	52
Figure 3.8 - Microphone selection .....	53
Figure 3.9 - Scene Block.....	53
Figure 3.10 - Recording time selection .....	54
Figure 3.11 - Cube repositioning screen .....	54
Figure 3.12 - XR Block.....	55
Figure 3.13 - User's hands inside the virtual environment .....	56
Figure 3.14 - FlexPendant [68].....	57
Figure 3.15 - Coding in RAPID within Robotstudio [69] .....	57
Figure 3.16 - Yaskawa controller DX200 [70] .....	58
Figure 3.17 - Virtual handheld robot controller (editor view) .....	59
Figure 3.18 - SLURBT [71].....	60
Figure 3.19 - Hand Controller Block.....	61
Figure 3.20 - Realvirtual.io Block .....	62
Figure 3.21 - Realvirtual.io packages available on the Unity asset store [72].....	64
Figure 3.22 - Realvirtual.io's Unity scene editing features (in editor view).....	65

Figure 3.23 - The robot and its components/scripts as seen in the Unity hierarchy (in editor view).....	66
Figure 3.24 - One of the robot's joints, and its corresponding drive components (in editor view) .....	67
Figure 3.25 - The robot's gripper and its sensor component (in editor view) .....	67
Figure 3.26 - List of Realvirtual.io signals as seen in Unity's hierarchy (in editor view) .....	68
Figure 3.27 - Meta Quest 3 equipment that was used for development and testing.....	69
Figure 3.28 - Guided tutorial mode's controller configuration.....	70
Figure 3.29 - Meta Quest 3 connected through a USB cable to the computer.....	71
Figure 3.30 - Unity test scene provided by the tutorial .....	72
Figure 3.31 - XR Rig Simulator Keyboard Shortcuts [74].....	73
Figure 3.32 - Hand animation.....	74
Figure 3.33 - Early communication test with RASA .....	75
Figure 3.34 - Tokenizer [77].....	76
Figure 3.35 - Example project from GitHub package .....	77
Figure 3.36 - Example of a dataset for fine-tuning [79].....	80
Figure 3.37 - First prompt defined for the system.....	81
Figure 3.38 - Auxiliar functions designed to update the AI bot on what's happening in scene .....	82
Figure 4.1 - First screen of the main menu.....	84
Figure 4.2 - Description of the guided tutorial (in editor view) .....	84
Figure 4.3 - Description of the free tutorial (in editor view) .....	85
Figure 4.4 - Tutorial room (in editor view).....	86
Figure 4.5 - Wall with tips and AI related functionalities (in editor view) .....	86
Figure 4.6 - Auxiliar text boxes (in editor view).....	87
Figure 4.7 - Cube bases and the cube repositioning screen (in editor view) .....	87
Figure 4.8 - Robot Controls (in editor view).....	88
Figure 4.9 - Instructions pause screen, in guided tutorial mode.....	89
Figure 4.10 - VR hands interacting with controller.....	91
Figure 4.11 - Example of a user interacting with the UI from far away.....	91
Figure 4.12 - Example of an inaccurate detection of the pointer by the cube-moving screen	92
Figure 4.13 - Gripper going through the cube.....	93
Figure 4.14 - Representation of a successful robot movement sequence .....	94
Figure 4.15 - Example of a bullet point list response by the AI assistant .....	95

Figure 4.16 - Example of a very long response by the AI assistant.....96  
Figure 4.17 - Final improved prompt.....98





## GLOSSARY

<b>Artificial Intelligence</b>	Computer system that can perform tasks that normally require human intelligence, such as understanding language and recognizing patterns.
<b>Augmented Reality</b>	Technology that overlays digital information (like images or sounds) onto the real world, often through a smartphone or special glasses.
<b>Chat Bot</b>	A system designed to simulate conversations with humans, typically through text or voice interactions.
<b>ChatGPT</b>	An AI language model (Chat Bot) developed by OpenAI that can generate human-like text based on the input it receives.
<b>Digital Twin</b>	A virtual model that mirrors a physical object or system, allowing for analysis and simulations.
<b>E-Learning</b>	Learning conducted via electronic means, typically through online courses and digital resources.
<b>Game Engine</b>	Software system used for creating and developing video games.
<b>Industry 4.0</b>	The current trend in manufacturing that uses smart technology to create more automated and efficient production processes.
<b>Industry 5.0</b>	The next phase in industry, focusing on the collaboration between humans and machines, emphasizing sustainability and personalization.
<b>Lean Learning</b>	An approach to learning that minimizes waste and maximizes efficiency, often through streamlined processes and focused training.

<b>Learning and Development</b>	Organizational activities aimed at improving the skills, knowledge, and competencies of employees.
<b>Learning Management Systems</b>	Software applications that help organizations manage, deliver, and track training and educational courses.
<b>Natural Language Processing</b>	A branch of AI that enables computers to understand, interpret, and respond to human language.
<b>OpenAI</b>	One of the major AI research companies at the moment, that focuses on developing and promoting AI.
<b>Proof of Concept</b>	A demonstration to prove that an idea or theory is feasible and can be developed into a real application or product.
<b>Robotstudio</b>	ABB's software that allows users to simulate, program, and optimize robot operations without needing physical robots.
<b>Society 5.0</b>	A conceptual paradigm for a smart society that integrates advanced technologies to improve quality of life and solve societal issues.
<b>Speech-to-Text</b>	Technology that converts spoken language into written text.
<b>Text-to-Speech</b>	Technology that converts written text into spoken voice output.
<b>Unity</b>	One of the most popular game engines, used for creating 2D and 3D video games and simulations across various platforms.
<b>V-Learning</b>	A form of learning that uses virtual environments or platforms to enhance educational experiences.
<b>Virtual Environment</b>	A computer-generated space where users can interact with a simulated world, often through VR or AR technologies.
<b>Virtual Reality</b>	An immersive technology that creates a simulated environment, allowing users to experience and interact with a 3D world using VR headsets.





## ACRONYMS

AI	Artificial Intelligence
AR	Augmented Reality
L&D	Learning and Development
LMS	Learning Management Systems
NLP	Natural Language Processing
TTS	Text-to-Speech
STT	Speech-to-Text





# INTRODUCTION

## 1.1 Motivation

With the emergence of Industry 4.0, a technological revolution in manufacturing processes began, marked by the extensive integration of digital technologies to develop more efficient intelligent systems. This new era of industry led to the popularization of the use of new crucial tools in manufacturing, based on, for example, AI, virtual reality and digital twins [1].

Following Industry 4.0, Industry 5.0 now emerges, with its focus on human-machine integration, representing an expansion in which collaboration between humans and machines is prioritized, integrating human innovation and flexibility in the production process.

Given this context, the industrial world finds itself in a phase where human-machine interaction technologies have great relevance and immense potential. One technology that surprised the world in 2022 was the Chat Bot developed by OpenAI, ChatGPT.

ChatGPT is an AI tool, capable of interacting with a human almost as equals, with access to an enormous amount of information. If an AI tool like this were to be used together with some of the main technological tools and innovations that have defined Industries 5.0 and 4.0, the potential applications and solutions to explore would be vast.

The combined use of tools like digital twins, virtual reality and AI could help solve various challenges in a more advantageous way. For example, the application of these tools in industry or academic training could allow for efficient learning without the need to reallocate resources typically allocated for production. Humans can thus interact virtually with these resources in an immersive learning environment that is also safer and more comprehensive, thanks to the combination of the mentioned technologies. However, such applications are still in an embryonic phase, not yet having reached a maturity that allows for widespread use. This exploratory work

aims to contribute to progress in this direction, refining the combination of these technologies, driving the development of solutions, and their eventual incorporation into industrial environments. To this end, a virtual training environment will be developed in which the concept will be illustrated, combining AI, virtual replicas, and virtual reality technologies.

## 1.2 Main Objectives

The objective of this dissertation is, therefore, to demonstrate the potential that Chat Bot tools and immersive technologies bring when combined and applied in the context of learning and professional training, as well as in the context of industrial manufacturing.

Establishing the concept's credibility and utility is important, as is studying the different technologies involved and elaborating on how they relate to and interact with each other.

More concretely, what will be developed in practical terms is a virtual environment that can be experienced using virtual reality equipment. In this environment, the user can learn the controls of an industrial robot with the support of an artificial assistant, a Chat Bot.

This project thus serves as a prototype of a virtual environment dedicated to professional training. In this environment, the controls of a real robot are simulated, with the support of a Chat Bot that autonomously assists the user in a precise and dynamic manner.

One of the main advantages that a system of this type can bring is allowing users to learn with virtual equipment instead of real ones. This approach avoids the risk of damaging the equipment. It also reduces the number of resources necessary for training new employees, as there is no need to use real equipment for training, and fewer human resources are required for these activities. The potential of using this popular new technology, in combination with other tools, to make professional training more efficient and practical will be demonstrated.

Ideally, this dissertation could serve as proof of concept for other similar systems, where Chat Bots are also used to assist and communicate with workers, whether in a training context or otherwise.

## 1.3 Document Structure

This dissertation is structured into five chapters. The first chapter outlines the motivation and primary objectives of the study.

The second chapter reviews the state of the art related to the project, examining its connection to the current industrial landscape, relevant technologies, and similar systems that either exist or have been explored.

In the third chapter, an in-depth analysis of the development process is presented. This chapter covers how various technologies were integrated into a single system, the system's structure, and how its components work together.

The fourth chapter provides an overview of the testing process and the conclusions reached, assessing whether the system meets the desired requirements. It also includes an analysis of potential improvements and explores how systems similar to this one could be developed further.

Finally, the fifth chapter offers a concise conclusion. It reviews the key challenges encountered during development and reflects on the project's future potential, wrapping up the dissertation with insights on the concept's outlook.



## STATE OF THE ART

### 2.1 Innovations in the Growing L&D Industry

Learning and Development (L&D) refers to the department within an organization dedicated to preparing and training employees, by investing in their skills to improve the organization's performance. It's an area of growing relevance today, which has led to the emergence of new solutions and technologies over the years, aimed at enhancing training processes.

Microlearning is an example of a training method that has gained popularity in recent years. In this method, training is delivered through short, focused lessons, provided gradually and divided into modules to make them more accessible, thus facilitating knowledge retention.

Other terms have also gained prominence, such as Social Learning, Remote Learning, Personalized Learning, Gamification, and Continuous Learning Culture. Social learning refers to the importance of collaborative learning, valuing the exchange of knowledge and teamwork among colleagues. Remote learning became especially relevant during the COVID-19 pandemic, referring to the concept of learning via the internet. Personalized learning recognizes the individual needs and characteristics of each learner, emphasizing the creation of personalized learning paths tailored to their specific needs. The term Gamification describes the implementation of game logic in the learning process, for example, promoting competition and implementing reward systems for learning. Continuous Learning Culture refers to the value placed on the ability to learn and relearn throughout a career in today's work cultures [2].

These terms represent some of the most relevant topics in L&D sectors today, and to realize them, new technological tools are starting to gain popularity. Currently, there are various tools of this type capable of supporting professional training. E-learning, for example, is a term that emerged to describe the new electronic teaching platforms that have appeared.

Learning Management Systems, or LMS, are software systems dedicated to managing, planning, and assessing students' preparation and performance, and they are also an example of a type of e-learning technological tool that is already being used in the context of L&D [3]. Moodle is one popular example of a LMS system, as is seen in Figure 2.1.

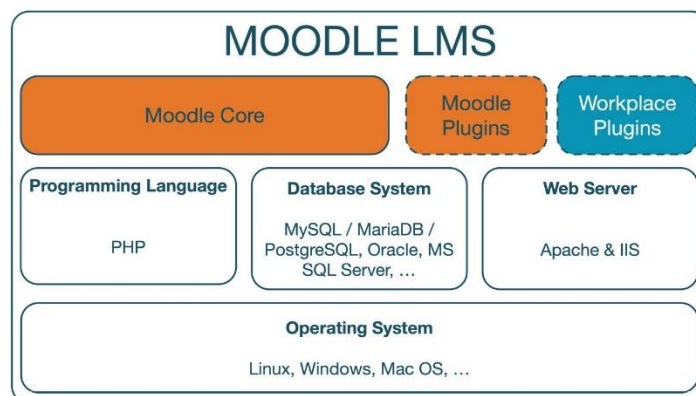


Figure 2.1 - Moodle is an example of an e-learning tool, in this case, a LMS [4]

AI-based tools and virtual reality are increasingly used in L&D contexts, as they are capable of offering functionalities that can address many of the needs mentioned earlier.

In further chapters we will explore in greater detail how AI tools and virtual environment creation can benefit the L&D industry, and this topic will be a major focus throughout this dissertation.

In the following section, section 2.1.1, we will examine how the L&D industry's current needs are leading to the emergence of these new solutions.

### 2.1.1 Contextualization of the Current Needs in the L&D Industry

In a dynamic and constantly evolving world, the relevance of L&D sectors has increased considerably in recent years, becoming one of the fundamental pillars for a company's success. The costs of professional training, excluding a decline during the COVID-19 pandemic, have been consistently rising since 2008, as indicated by a study published by Statista [5] and shown in Figure 2.2. This increase could be inferred from the growth in the number of training hours in the workplace worldwide, with the professional training industry being one-third larger in 2021 than it was in 2009 [6].

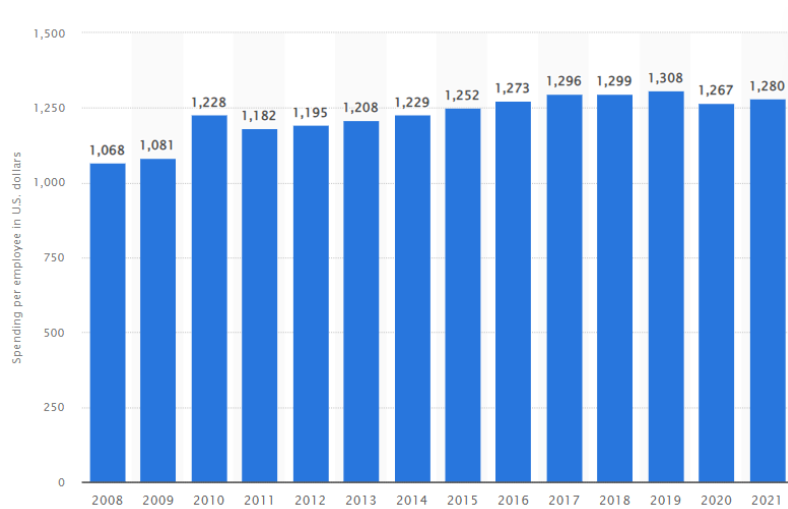


Figure 2.2 - Evolution of the average cost associated with professional training between 2008 and 2021 [5]

In 2022, training costs in the U.S. exceeded 100 billion dollars for the first time, and according to more recent studies, they rose to 101.8 billion dollars in 2023. This reflects the growing importance of the L&D industry today. However, it is also important to mention the exceptions, cases where L&D costs decreased. Often, this occurred due to economic uncertainties, reductions in the scope of training, or decreased investments. However, a common reason was also the adoption of autonomous processes that reduce costs compared to other training programs, such as the use of e-learning tools [7].

LMS (represented in Figure 2.3) are especially popular among companies experiencing rapid growth, with exponential increases in workers and partners. In such cases, individual training or human instruction for small groups may no longer be viable [8].



Figure 2.3 - Learning management system (LMS) [9]

The growth of L&D industries in recent decades demonstrates the increasing importance of the sector. This fact has led to the emergence of new solutions and the reinforcement of certain paradigms mentioned earlier. The rise of autonomous, accessible, and efficient solutions in the provision of training services shows a growing interest in optimizing these

processes, and technological tools, such as the use of artificial intelligence (e.g., Chat Bots), demonstrate significant potential in this area.

In work areas that require training in the use of complicated, often scarce and expensive equipment, virtual training options through realistic simulations would be useful. New professionals could learn how to use the equipment safely without the risk of accidents. If possible, automating the process by using artificial intelligence could potentially diminish the human and financial costs required for training.

All of this indicates a growing interest in cost-effective and efficient solutions for professional training and shows that there is a space in today's markets where a concept like the one promoted in this dissertation has potential.

In section 2.1.2, we will look at an example of a training methodology that would greatly benefit from the use of these new tools.

## **2.1.2 Applications in Lean Learning**

Lean learning is a learning and training methodology initially popularized by Toyota, primarily focused on maximizing results and reducing costs. When applied in companies, the concept aims to increase the adaptability and flexibility of its employees.

The main steps for applying lean learning are as follows:

1. Learn the basics of the subject of study.
2. Immediately apply what was learned in real situations.
3. Receive immediate support and suggestions to refine what has been learned.
4. Repeat the cycle.

This learning approach requires constant contact with the practical aspects of what is being learned and demands a high level of customization in training, so that it adapts to the specific needs of each learner. In this methodology, it is crucial for the learner to receive constant support from an instructor or, at the very least, from something capable of providing constructive feedback whenever necessary [10].

These characteristics make it a methodology frequently recommended by critics of traditional training approaches in L&D sectors. It is becoming increasingly common in various fields and serves as a clear example of the potential benefits that simulation technologies and learning support can provide.

VR Simulation systems applied to professional training in robot control, capable of providing autonomous support through an intelligent Chat Bot, could be highly useful in the context of lean learning.

The interactive virtual environment, together with the Chat Bot, would be able to offer instant support to the user in an efficient and almost human-like manner (given the capabilities of modern Chat Bots). Such a virtual environment could also offer various training options and scenarios, meeting the adaptability needs mentioned earlier. This way, the user could learn and test their skills in a realistic, safe, and controlled environment, fulfilling the necessary requirements for the application of lean learning.

This training methodology thus serves as a good example of a scenario in which training systems like the one proposed in this dissertation could be useful.

## **2.2 Contextualization with Industries 4.0 and 5.0**

As mentioned in chapter 1.1, we are immersed in the digital era of Industry 4.0, also known as the Fourth Industrial Revolution.

Industry 4.0 (illustrated in Figure 2.4) marks the era of digitization in the manufacturing sectors, driven by an increase in data exchanges, connectivity, advancements in human-machine interaction, and progress in robotics. During this transition, tools such as machine learning, artificial intelligence, augmented reality, and virtual reality have demonstrated their utility in various ways within manufacturing industries [1]. This new paradigm in manufacturing processes emerged in the 2010s and continues to be extremely relevant today.



Figure 2.4 - Industry 4.0 [11]

However, there are already talks of 5.0 paradigms, from which Society 5.0 emerges, leading also to the emergence of Industry 5.0. The 5.0 social paradigm proposes a human-centered society, promoting a balance between economic advancement and the pursuit of solutions to social problems through a system that integrates the cyberspace and the physical world [12]. Industry 5.0, then, arises as a direct consequence of Society 5.0. In Industry 5.0, the reintroduction of humans in the production process is emphasized, not to diminish the importance of the technological advancements of Industry 4.0, but rather to apply them in a way that supports humans. The focus shifts from surpassing and replacing humans through extensive automation to achieving full cooperation between humans and machines to maximize human adaptability, creativity, and intelligence [13]. In Figure 2.5 we can see a representation of the main focus points of Industry 5.0, sustainability, human centricism and resilience.



Figure 2.5 - Industry 5.0 [14]

Thus, it becomes evident that the development of new forms of interaction and collaboration between humans and machines is one of the most crucial topics today and for the future of the industrial landscape.

In this context, the proposed concept fits perfectly, as the development of a virtual environment where humans can learn and train with the support of an intelligent, artificial assistant, represents a form of human-machine collaboration taken to a more advanced level.

## 2.3 Technologies Supporting Learning Paradigms

Previously, we have seen how, in the current context of the private and industrial world, L&D sectors are increasingly valued. The growing demand for accessible solutions for training new workers was highlighted, and it was discussed how, in the context of lean learning, a training methodology that is increasingly adopted, technological tools focused in providing immersive and interactive learning experiences show great potential.

After a general contextualization of the current industrial landscape, it was concluded that there is a clear trend towards increasing collaboration and interaction between humans and machines, something that aligns with the value of the concept promoted in this dissertation.

The next step will be to explore the technological tools necessary for the implementation of the proposed project.

To recap, the project in question is a simulation system designed for teaching the controls of a robot, which can be experienced in virtual reality. This virtual environment will feature a digital replica of a robot, controllable through a control system that is inspired by real robot control systems. Additionally, the user will have the support of an artificial "instructor" in the form of a Chat Bot.

### 2.3.1 Digital Twins

For the development of this idea, it is crucial to study what a digital twin is, in what contexts they are used, and what role this technology plays in the context of professional training.

A digital twin is a detailed virtual replica of an object or system, capable of simulating its real behaviour. They are typically created based on data collected from various sensors over time, associated with the real system that is being replicated. The use of this data allows for the creation of a virtual copy that can closely mimic the system with high precision. An example of what a virtual model of a real object may look like is seen in Figure 2.6.

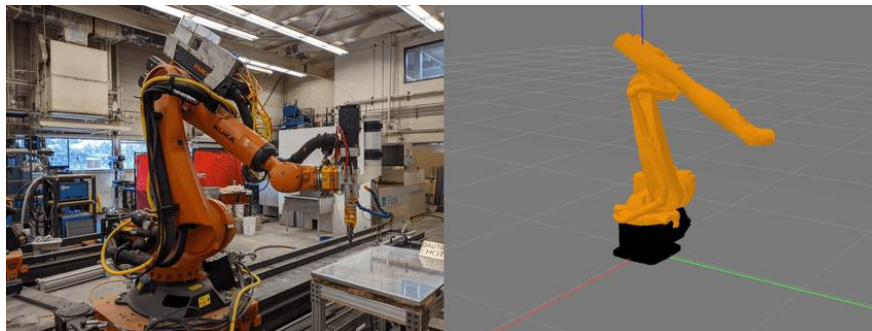


Figure 2.6 - Digital twin of a robotic arm [15]

The distinction of the digital twin concept from more traditional simulations, comes mainly from its considerably larger scale. While conventional virtual simulations typically focus on a single process, the term digital twin refers to simulations generated from a massive amount of data, enabling the simulation of a vast number of scenarios more realistically.

Another distinguishing aspect of digital twins compared to more traditional forms of virtual simulation is the access to real-time data, allowing for the updating of the virtual model and the exchange of information between the real and virtual systems. This feature makes it possible to explore a wider variety of phenomena with greater realism and adaptability [16].

Digital twins have a wide range of applications across various industries, being useful in sectors such as healthcare, agriculture, construction, manufacturing, and more. They are

widely used to optimize systems and to make predictions through simulations. For example, suppose a digital twin of a factory is created and used to simulate its operations. In that case, it becomes possible to detect risks and problems that would be difficult to identify by only analysing physical processes. Additionally, finding ways to optimize the factory's processes becomes feasible, something that would be challenging to visualize without the use of a digital twin.

Moreover, the potential of digital twins in training areas is also notable. Given the importance of training competent professionals, as mentioned earlier, due to the increasing competition in today's markets, more and more digital tools are being integrated in the context of L&D, with digital twins being one of these tools.

The use of digital twins, especially in conjunction with virtual reality (illustrated in Figure 2.7), can allow professionals to acquire valuable skills by virtually experiencing the scenarios for which they are being trained. This solution saves resources and, in many cases, protects professionals from potential risks that could arise if they were learning in real environments.

Practical examples include Volkswagen's Digital Reality Hub platform, made to provide a collaborative training environment in virtual and augmented reality for workers, with various training scenarios. Companies in automation, like Volkswagen, illustrate how immersive tools with digital twins can be beneficial in the context of worker training.

Walmart and INVISTA are two other companies that have adopted technologies based on digital twins and VR for training and preparing their employees [17].



Figure 2.7 - Digital twins in VR [18]

Thus, it becomes evident that this technology is already valued across multiple industries, and its potential in the context of professional training is taken seriously by various entities. It

is a credible tool in teaching and training fields and it's particularly relevant when combined with VR or AR tools.

### 2.3.2 Virtual Reality

After addressing digital twin technologies, it is now essential to provide context on what virtual reality is, the types of experiences available, and the applications of this technology.

Virtual reality, sometimes also referred to as immersive computing technology, offers a unique way to interact with the virtual world. It is a technology that allows for an immersive experience in a world beyond our everyday reality. A representation of this concept can be seen in Figure 2.8.

Over the years, we have witnessed the continuous progression of these technologies, which have gained increasing prominence, currently being predominantly used to convey sensations and experiences to our senses, particularly to our vision, touch, and hearing.



Figure 2.8 - Virtual Reality [19]

Typically, these immersion systems include at least one of the following technologies, a single large projection screen, multiple connected projection screens, stereo-capable monitors with desktop tracking or head-mounted displays (HMDs). It's also common to use headphones or a surround sound system to ensure auditory immersion. Various monitoring systems allow for the real-time calculation of the position and orientation of objects in relation to the user's perspective. The user's own movements can be tracked and transformed into interactions with the virtual world, often including handheld controllers that allow for the navigation and manipulation of virtual objects. It is also possible to incorporate sensations of vibration, wind, temperature, and pressure to make the experience even more immersive.

In essence, virtual reality provides a human experience, where the technologies involved take advantage of our senses to mimic how we interpret and experience the world around us. When well executed, virtual reality experiences can persuade users that they are physically in a virtual world.

The term "Virtual Reality" first emerged in the 1960s with the publication of Ivan Sutherland's "The Ultimate Display". In it, the author described a monitor capable of transmitting information not only visually but also through sound, smell, taste, and touch. Sutherland proposed many technologies that did not yet exist in his time, including eye-tracking systems and dynamic 3D rendering. His work laid the foundation for subsequent developments over the decades. In Figure 2.9 we have a photo taken of the first ever virtual reality helmet, developed by Sutherland, "The Sword of Damocles".



Figure 2.9 - The Sword of Damocles, the first virtual reality helmet [20]

However, it was only in the 1990s that the technical capabilities to realize Sutherland's proposals began to emerge. New studies started to be published, supporting the viability of a future where virtual reality technologies would become a reality, this led to an increase in interest in the use of virtual reality in industry.

In 1999, Fred Brooks, a professor of computer science at the University of North Carolina, published the study "What's Real About Virtual Reality?" in which he described the key challenges for the development of virtual reality, concluding that virtual reality had finally arrived and that it "barely works".

Since then, this field of study has expanded considerably, with significant contributions from both the academic and industrial communities, bringing this technology to a point where it can be truly useful in various industries. Although technical challenges still exist, many of the

problems identified by Brooks have already been overcome, and today we have the computational resources necessary to truly benefit from these technologies [21]. With this context in mind, let us now explore the differences between the different types of virtual reality that exist.

It is important to begin by distinguishing augmented reality from virtual reality. Both concepts are related, and both technologies aim to enrich or replace the physical environments around us. However, augmented reality focuses on adding virtual elements to the physical reality we experience, by combining the virtual world with the real world (a representation of this concept can be seen in Figure 2.10). Virtual reality, on the other hand, refers to the complete replacement of the real world with the virtual world, creating a fully virtual and immersive experience.

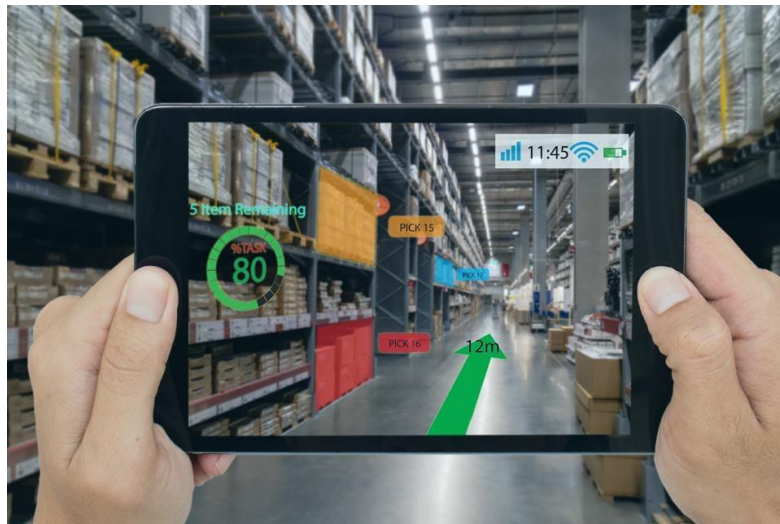


Figure 2.10 - Augmented Reality [22]

Despite the conceptual difference between the two technologies, there is often some overlap in terms of functionalities and utilities. Much of what is discussed regarding virtual reality applications can be similarly applied to augmented reality.

That said, there are three main types of virtual reality, non-immersive, semi-immersive, and fully immersive virtual reality.

Non-immersive virtual reality refers to a computer-generated environment where the user can control activities without direct interaction with the environment. This type of virtual reality is commonly used in the daily life of an ordinary person, as it relies only on a computer or video game console to create the environment. Most of the video games that exist today are good examples of non-immersive virtual reality.

Semi-immersive virtual reality lies somewhere between non-immersive and fully immersive virtual reality. It allows the user to experience the virtual environment directly, but without the ability to experience physical sensations. This type of virtual reality is common in learning and training contexts.

Finally, fully immersive virtual reality corresponds to the most realistic experience possible, incorporating various forms of immersion. This includes, for example, special gloves for interacting with the environment, body detectors, and other special sensors beyond the more commonly used virtual reality HMDs (head mounted displays). This type of virtual reality is used for environments that require deeper immersion [23]. In the following figure, Figure 2.11, we see an example of this, with a person experiencing fully immersive VR, with various tools beyond just the VR headset.

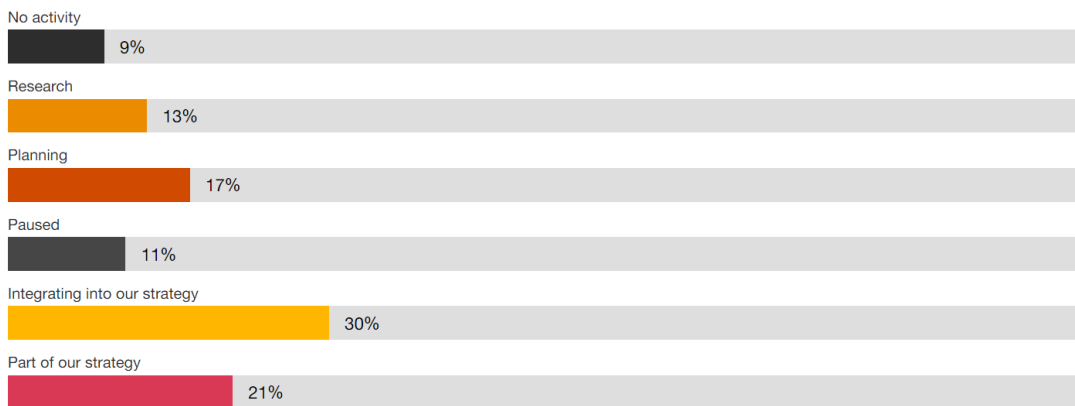


Figure 2.11 - Fully immersive VR [24]

Virtual reality tools are currently used in various fields, such as engineering, sports, and healthcare. It is estimated that the market value for augmented reality and virtual reality will increase from \$8.94 billion in 2020 to \$80.80 billion by 2028 [25].

This technology is growing substantially today in various areas, but it stands out particularly in the field of education and training. According to a PwC survey in 2022, 51% of American companies are currently integrating or have already integrated virtual reality into at least one of their business lines. This data can be seen in Figure 2.12. Additionally, 34% of companies consider that one of the greatest advantages of integrating virtual reality technologies is the development of more efficient training methods.

### Where companies are at with VR adoption



Source: PwC 2022 Metaverse Survey, July 2022. Q: To what extent are the following technologies embedded in your company's strategy? virtual reality. Base: 1,004

Figure 2.12 - Graphical representation of the point at which American companies are, regarding the adoption of virtual reality technologies, according to a PwC study [26]

The study also compared the performance of students who participated in a soft skills course in a classroom setting, with students who did the same but through e-learning and with v-learning (using virtual reality). Overall, students who learned using VR demonstrated better performance. Those trained with virtual reality learned up to four times faster (as seen in Figure 2.13), showed greater confidence in applying their new knowledge, and demonstrated greater focus during the course [26].

### Time to complete training



Source: PwC VR Soft Skills Training Efficacy Study, 2020

Figure 2.13 - Graphical representation of the time required to complete training according to a PwC study [26]

Thus, it becomes evident that the topic of virtual reality will likely be a recurring one in the coming years, with the L&D industry being one of the areas where this will be the case,

whether in the context of simulations with digital twins, as discussed earlier, or even in the context of soft skills courses.

### 2.3.3 Chat Bots

In addition to digital twins and virtual reality, it is equally essential to explore the concept of Chat Bots, understand the different types of Chat Bots that exist, analyse the potential applications of this technology, and reflect on how we have reached the current point, with tools like ChatGPT and Gemini (Figure 2.14) revolutionizing various aspects of our lives.

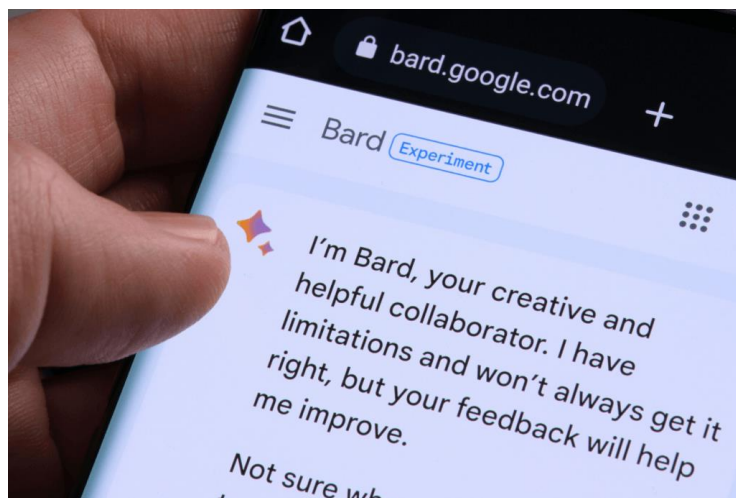


Figure 2.14 - Google Bard (Gemini's ancestor) [27]

A Chat Bot is a common artificial intelligence system. It is a computer program capable of interacting intelligently with humans, using voice or text, and it is often capable of understanding one or more human languages through natural language processing (NLP). It is a program designed to simulate conversations with human users.

This concept began to gain popularity after Alan Turing proposed the Turing Test in 1950, which is a simple test to demonstrate whether a machine has human-like intelligence.

The first Chat Bot, known as Eliza, was developed in 1966 using simple pattern recognition algorithms with responses based on predefined models (as can be seen in Figure 2.15). Over the years, more Chat Bots emerged, such as PARRY in 1972 and ALICE in 1995, the latter winning several awards. ALICE was the first Chat Bot to earn the title of "most human computer", still functioning solely with simple pattern detection algorithms.

As we entered the 21st century, the evolution of this technology accelerated even further. Chat Bots like SmarterChild, in 2001, began to be incorporated into messaging applications,

and later, virtual assistants such as Apple's SIRI, Microsoft's Cortana, Amazon's Alexa, and others became common presences.



Figure 2.15 - The first Chat Bot, Eliza [28]

This constant growth, both in technological terms and in interest, can be visualized in Figure 2.16, which represents the increase in the number of articles and studies associated with the terms "chatbot", "conversation agent", and "conversation interface" according to Scopus, between 2000 and 2019.

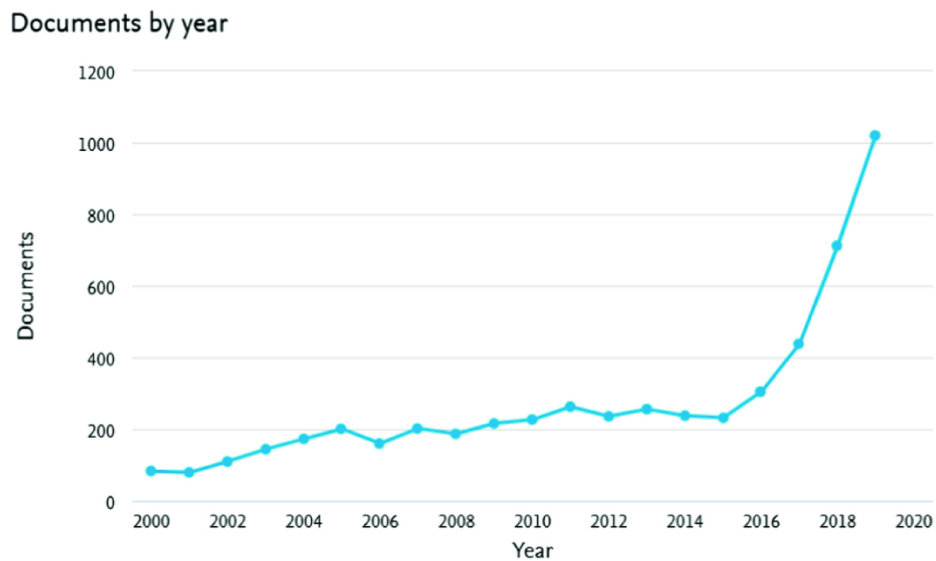


Figure 2.16 - Articles and studies associated with the terms chatbot, conversation agent, and conversation interface according to Scopus between 2000 and 2019 [29]

However, we are currently experiencing a revolution in the field of Chat Bots, primarily marked by the emergence of tools like ChatGPT. These Chat Bots are based on generative artificial intelligence, a type of AI capable of generating text, images, videos, and other content in response to user requests, using a vast database.

Chat Bots can be classified based on various parameters, such as the domain of knowledge, the types of services provided, the purpose, the method of generating responses, the level of human assistance involved, and the type of permissions offered by their development platform.

The classification by domain of knowledge refers to the types of data the Chat Bot can access and be trained on. An open-domain Chat Bot has access to a wide range of data and can address multiple topics, while a closed-domain Chat Bot is restricted to data that is related to one specific topic.

The classification in terms of the services offered evaluates the emotional proximity that the Chat Bot maintains with the user. Interpersonal Chat Bots, although capable of having a personality and offering pleasant interactions, do not function as consistent companions. On the other hand, intrapersonal Chat Bots operate within the user's personal domain, understanding them similarly to a human.

The purpose of a Chat Bot can also be classified. Informational Chat Bots are used to provide information stored in a fixed source. Conversational Chat Bots are designed to interact with the user, responding to questions in a human-like and understandable manner. Finally,

task-based Chat Bots have the specific function of performing a particular task, with one example of this type of Chat Bot being depicted in Figure 2.17.

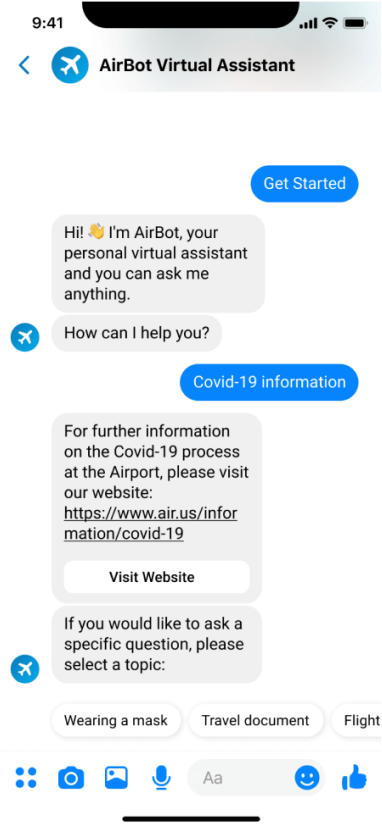


Figure 2.17 - Conversation with an airport's Task-Based Chat Bot [30]

Regarding the methods involved in processing inputs and generating responses, there are three main categories in which a Chat Bot can be classified.

Rule-based Chat Bots follow the architecture on which the vast majority of early Chat Bots were based. The responses given by the system are grounded in predefined rules. Lexical recognition methods are applied to the messages sent by the user, and based on this reading, a response is generated according to the established rules. The quality of this type of Chat Bot heavily depends on the robustness of its rule database, a solid rules database allows the Chat Bot to handle a wider variety of inputs. However, this type of Chat Bot tends to struggle with grammatical errors in user messages.

Retrieval-based models are like rule-based models but offer greater flexibility as they can search and analyse external resources. Like rule-based models, they rely on a database to support their responses, but instead of being restricted to its rules, they also have access to external databases with multiple potential responses of various types.

Finally, generative models are based on machine learning and deep learning algorithms, offering significantly superior quality, especially in terms of language and adaptability. They can respond based on previous inputs more easily than other models.

Chat Bots can also be classified as human-aided if they involve at least one element of human intervention. The platforms on which they are developed can be classified as open-source or closed-source, with open-source platforms allowing the Chat Bot designer to intervene in most aspects of the implementation, unlike closed-source platforms where this is not possible.

Naturally, it is often difficult to classify Chat Bots directly. However, these categories are generally relevant to any Chat Bot to some extent [29].

Chat Bot technologies have had various applications over time, some of which have already been mentioned. However, with the emergence of ChatGPT and other similar generative AI tools, the potential of these technologies has increased significantly.

One area where generative artificial intelligence can be useful is in business, covering sectors such as marketing, sales, engineering, finance, human resources, among others. For example, a Chat Bot can provide customer service, serve as a virtual assistant to help customers resolve issues, manage the allocation of human resources within an organization, and generate ideas and advertisements for marketing (as seen in Figure 2.18). The applications of generative Chat Bots in business areas are vast.



Figure 2.18 - AI generated advertisement images [31]

Another area where generative Chat Bots have potential is in healthcare. Systems like ChatGPT can transform the healthcare industry in aspects related to patient interaction, clinical diagnostic support, telemedicine services, health education, and overall health promotion.

In content creation, this technology also has various uses. We have seen that in the marketing industry, generative artificial intelligence can create appealing and personalized advertisements for consumers. Additionally, we have observed significant changes, for example, in journalism, caused by the introduction of these technologies in news production, with robots like Quill and Xiaomingbot being examples of this.

In artistic fields, there is also a noticeable impact, with the emergence of tools like ChatGPT, Stable Diffusion, and Midjourney, capable of generating art and assisting artists with ease (an example of this can be seen in Figure 2.19).



Figure 2.19 - Artistic drawing generated by AI [32]

In the video game industry, AI is already used to optimize game development and diversify creation possibilities.

However, an area where such technologies also have significant potential, more related to the topic of this dissertation, is in education. Intelligent Chat Bots can be used as assistants in learning and teaching activities. They can help students with research, answer questions related to specific topics, assist with writing in different languages, and facilitate the learning process (Figure 2.20). For teachers, Chat Bots can aid in planning and creating teaching materials, such as quizzes, slides, and more. Additionally, they can be useful in reviewing and evaluating students' work, providing constructive feedback.

Chat Bots of this type are valuable in creating educational content and can provide personalized learning experiences, enhancing student engagement with the material, which can

improve the efficiency and effectiveness of education. By offering support, guidance, and feedback, intelligent Chat Bots can also be very helpful for self-directed learners in open education [33].

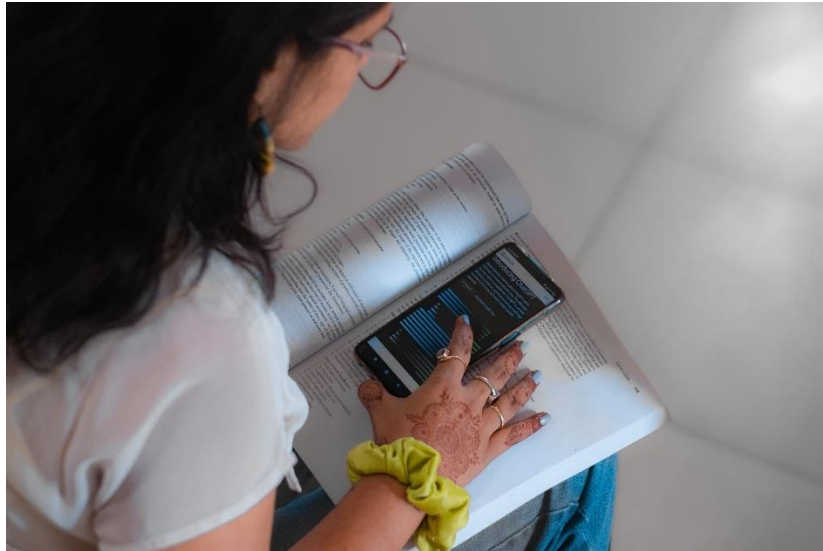


Figure 2.20 - Chat Bot being used as a learning tool [34]

This wide range of possible uses has led many stakeholders to enter the market for these technologies. OpenAI, with its ChatGPT, has truly started this technological revolution and has since remained the most popular provider of generative artificial intelligence. However, competitors such as Google, Amazon, and DeepMind have been trying to gain relevance by enhancing their own Chat Bots, by making them more appealing to users, efficient, and secure.

Currently, OpenAI offers various applications beyond the well-known Chat Bots, GPT-3.5 and GPT-4.o. For example, there is Whisper, a voice interpreter capable of listening and transcribing what the user says (Speech-to-Text). Additionally, there is DALL-E 2, which can generate images on request.

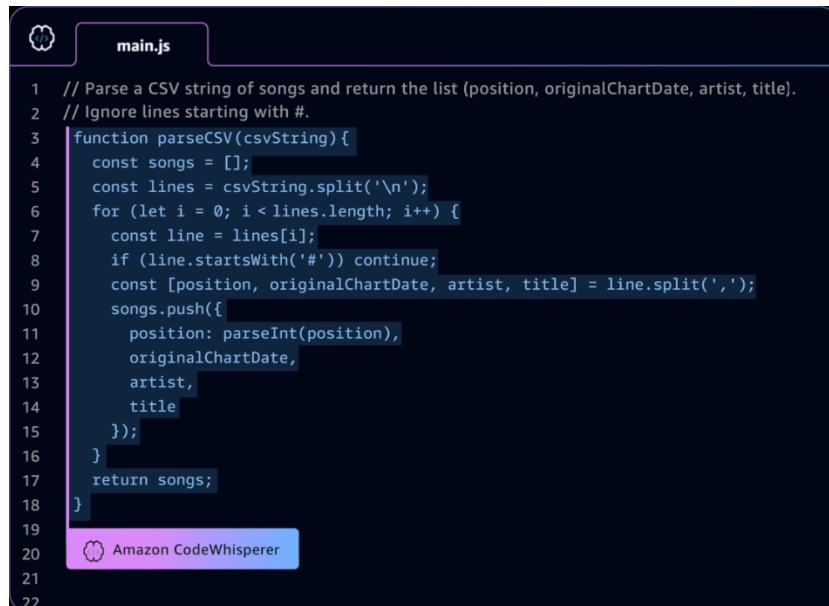
Any of these services can be used at the cost "tokens", and they can be integrated into various systems through specific APIs.

However, there are also competitors like Google's Gemini, which operates similarly to ChatGPT but processes data from public web sources and other Google applications, while ChatGPT processes data from a mix of its own training data and Bing search results.

Google AI is one of the biggest competitors that OpenAI currently faces, also being a viable provider of AI APIs that can be used in various systems.

There are also Task-Based generative AI tools, such as Microsoft's BingChat and Amazon's Codewhisperer. BingChat, although versatile and capable of interacting directly with

users, mainly focuses on assisting with searches, providing links, and collaborating with the Bing search engine. Codewhisperer, similarly, primarily concentrates on its role as an assistant for programmers [35], as is seen in Figure 2.21.



```
main.js
1 // Parse a CSV string of songs and return the list (position, originalChartDate, artist, title).
2 // Ignore lines starting with #.
3 function parseCSV(csvString) {
4   const songs = [];
5   const lines = csvString.split('\n');
6   for (let i = 0; i < lines.length; i++) {
7     const line = lines[i];
8     if (line.startsWith('#')) continue;
9     const [position, originalChartDate, artist, title] = line.split(',');
10    songs.push({
11      position: parseInt(position),
12      originalChartDate,
13      artist,
14      title
15    });
16  }
17  return songs;
18 }
19
20 Amazon CodeWhisperer
21
22
```

Figure 2.21 - Codewhisperer [36]

RASA is another example of a platform that offers Chat Bot and virtual assistant services, being an open-source option that allows for the development and training of personalized intelligent agents. It uses a combination of generative machine learning algorithms and rule-based algorithms for its operation [37].

The emergence of these generative AI technologies represents one of the greatest technological disruptions in recent years. With the ability to communicate in a human-like, direct manner and with potential access to vast databases, generative Chat Bots have potential across various industries. However, in this dissertation, the area in which this technology will be explored is in education, more specifically, in the context of professional training.

We have seen how digital twins and virtual reality technologies are used together to create training simulations, where students can learn effectively and efficiently while also avoiding various risks. But with the advent of this technological revolution in artificial intelligence, it becomes possible to expand this concept to a more advanced level, allowing for the introduction of a virtual and intelligent assistant into the system.

## 2.3.4 Game Engines

There are several platforms that can be used to create virtual reality environments, and in this project, it is necessary to choose a platform that allows for the integration of the three main technological tools and concepts involved in this project, a digital replica, virtual reality, and the virtual assistant.

One option is A-Frame, an open-source web program that facilitates the development of virtual reality experiences based on Three.js. This platform allows for the creation of 3D scenes and WebXR using HTML, making it a convenient tool for developing virtual reality environments on the web [38].

Another alternative is RobotStudio, developed by ABB, which is used for the simulation and programming of robots. Although it's mainly known for its offline programming and simulation functions for robots, it also has relevant virtual reality functionalities, providing a more immersive analysis of the 3D environment where robots are visualized. It also allows, for example, for the use of hand tracking in virtual reality to define the movement of robotic arms [39]. In Figure 2.22 we can see what the Robotstudio environment looks like.

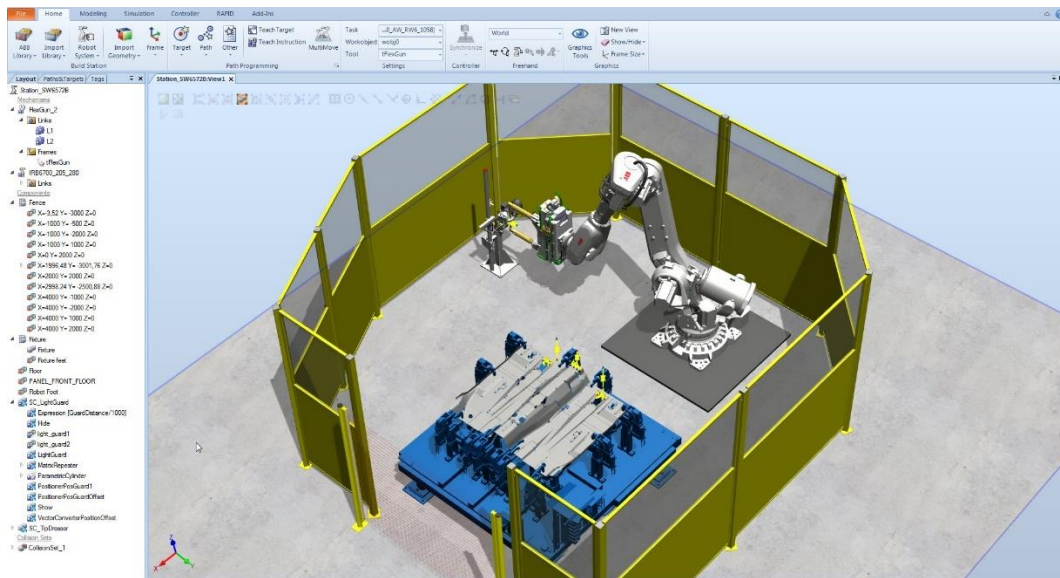


Figure 2.22 - Robotstudio [40]

NVIDIA's Omniverse is another tool used for creating virtual environments. This platform is well-suited for the collaborative development of 3D workflows and applications, excelling at simulating complex digital twins and at creating VR environments. For example, it was the tool chosen by BMW for creating the "world's first virtual factory", by utilizing digital twins and virtual reality technologies [41] (Figure 2.23).



Figure 2.23 - Virtual Factory in Omniverse [42]

But one type of system that has been gaining relevance in the world of simulated virtual environments is definitely the game engine. Game engines are software development environments optimized for creating video games, that usually come with 3D and 2D graphics rendering engines, physical simulation tools, artificial intelligence tools, and more [43]. There are several options, with Unity, Unreal Engine, and Godot being some of the most popular.

Godot is a relatively recent game engine known for being easy to use, versatile, and open source. These characteristics have led to an increase in its popularity in recent years, but compared to the other options, it has a smaller community and fewer resources for creating complex 3D virtual environments [44].

Nonetheless, it is a game engine with growing influence not only in the video game development market but also in developing programs for professional contexts, gaining popularity, for example, in AEC (Architecture, Engineering, and Construction) industries [45].

Unreal Engine is also an open-source game engine, but unlike Godot, it has a stronger established reputation. Founded in 1995, it stands out for its robustness in all aspects, featuring the most realistic and detailed 3D rendering system of the three, making it especially popular in high-budget game development [46]. Additionally, it is frequently used for developing professional applications beyond video games. We can see an example of an Unreal Engine 3D environment in Figure 2.24.

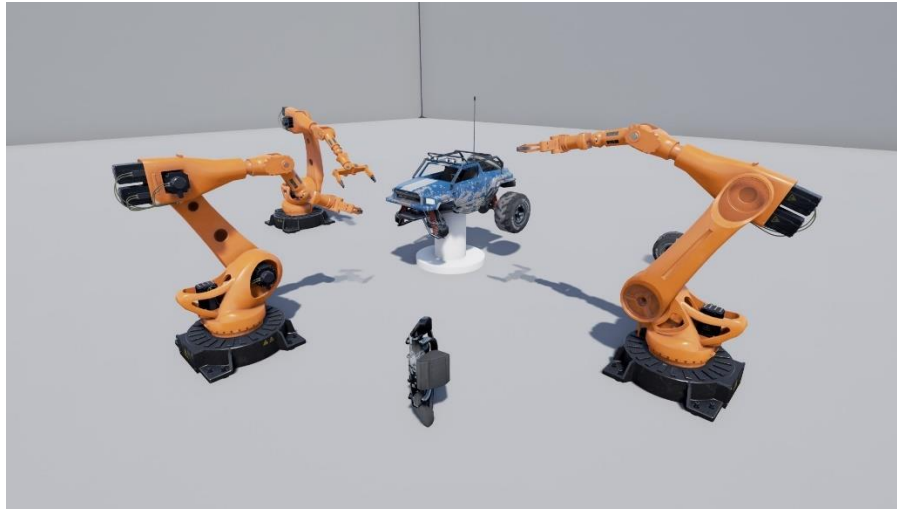


Figure 2.24 - Robotic models in an Unreal Engine environment [47]

Unreal Engine, for example, is used for the development of flight simulation systems for pilot training by Lockheed Martin [48]. It's a widely used game engine for creating virtual environments across multiple contexts, including digital twin simulations, offering various tools for that purpose.

Unity is also recognized as a game engine that provides a wide range of tools. Unlike Unreal Engine, it is more commonly used in the production of indie video games due to its accessibility, ease of use, and graphical capabilities, which, while not as advanced as those of Unreal Engine, meet the needs of this category of games. However, among all game engines, Unity stands out particularly due to its abundance of online resources available, many of which are free, and exist because of Unity's extensive community. This prominence is evident in Figure 2.25, which shows the popularity of Unity among British video game developers in 2019.

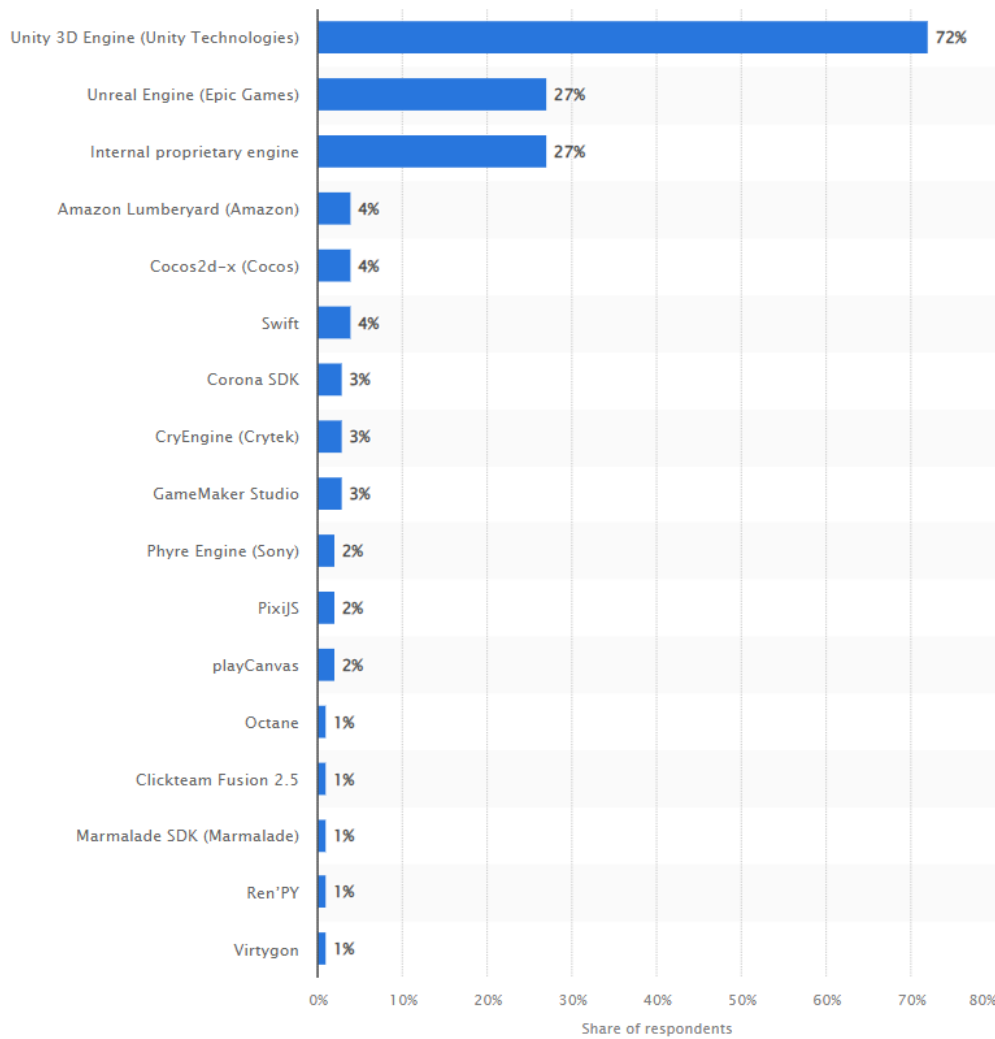


Figure 2.25 - Most used game engines in the United Kingdom by game developers in 2019 [49]

Unity, like Unreal Engine, is widely used in professional environments, including in the field of simulation systems, digital twins, and virtual reality [46]. In 2021, Unity was proposed as a platform for developing a digital twin simulation system for automated vehicles during an international conference on digital twins and parallel intelligence [50]. Additionally, there is the case of the industrial simulation company, Realvirtual.io, which recently chose Unity as a development platform, recognizing its potential to challenge traditional industrial simulation tools. An example of a robot being represented inside a Unity virtual environment can be seen in the following, Figure 2.26.

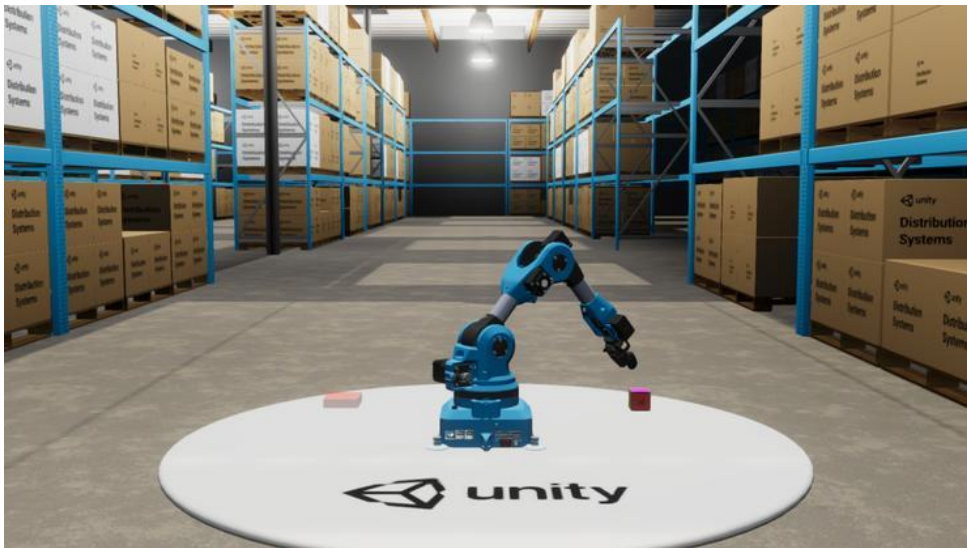


Figure 2.26 - Robotic arm in a Unity environment [51]

According to Thomas Strigl from Realvirtual.io, significant changes are being observed in the field of industrial simulation. Platforms such as Unity, Unreal Engine, and Omniverse are gaining increasing relevance in this sector, standing out for their potential to challenge conventional simulation and industrial engineering solutions from companies like Siemens and Dassault. All three systems have the capability to easily import CAD data and can be integrated with a wide variety of systems. Realvirtual.io's investment in Unity services responds to an emerging trend of adopting Unity that is seen in various industries, such as engineering and architecture. This preference is primarily due to Unity's aforementioned features, such as the extensive range of available resources, ease of use, and compatibility with multiple platforms [52]. In Figure 2.27 we see an example of a robot visualised within Realvirtual.io's Unity framework.

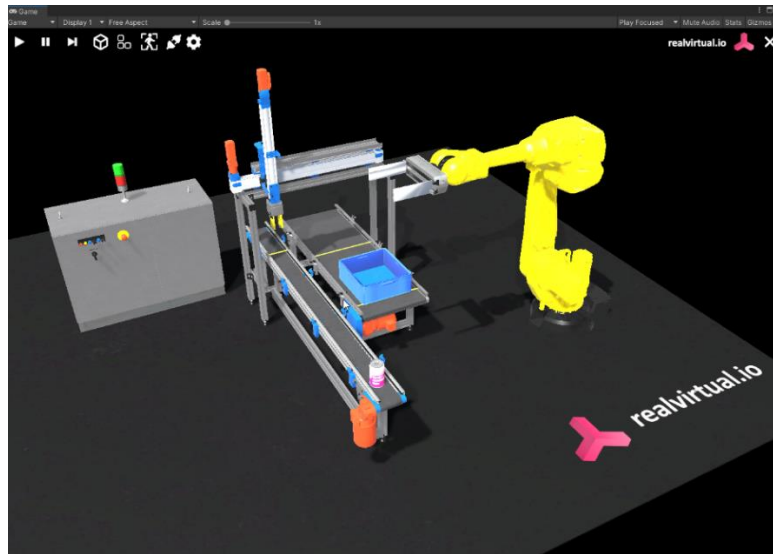


Figure 2.27 - Example of a Realvirtual.io robot visualized in a Unity 3D scene [53]

## 2.4 Related Works

Having analysed the main technologies necessary for this project and studied their value in a general and modern context, especially regarding their applications in teaching and training areas, this chapter will conclude with a broader overview of a few works related to this subject that already exist. This overview will analyse how these works highlight trends in the industry and academy, and expose a shared knowledge gap that this dissertation aims to address.

### 2.4.1 Industrial Trends

In section 2.1, we examined the current state of the Learning and Development (L&D) industry, emphasizing the importance of discovering new methods, techniques, and paradigms for professional education. We saw how the rising costs of training, alongside new technological learning tools and the need for teaching methods centered around efficiency and results, have underscored this necessity.

We also analysed the broader context of the incoming transition from Industry 4.0 to the emerging Industry 5.0. A shift that emphasizes human-machine collaboration, focusing on augmenting rather than replacing human capabilities. A transition that gives credence not only to the concept proposed in this document but also to related systems and ideas.

Having this context in mind, the actions from companies like Volkswagen become unsurprising.

Volkswagen's Digital Reality Hub serves as an excellent example of how major corporations are now utilizing these technologies for learning and collaboration. This platform, introduced at the Digility conference in Cologne, in 2017 [54], is a cloud-based environment that facilitates global collaboration in training, meetings, and prototyping without requiring physical materials or travel [55] (Figure 2.28).

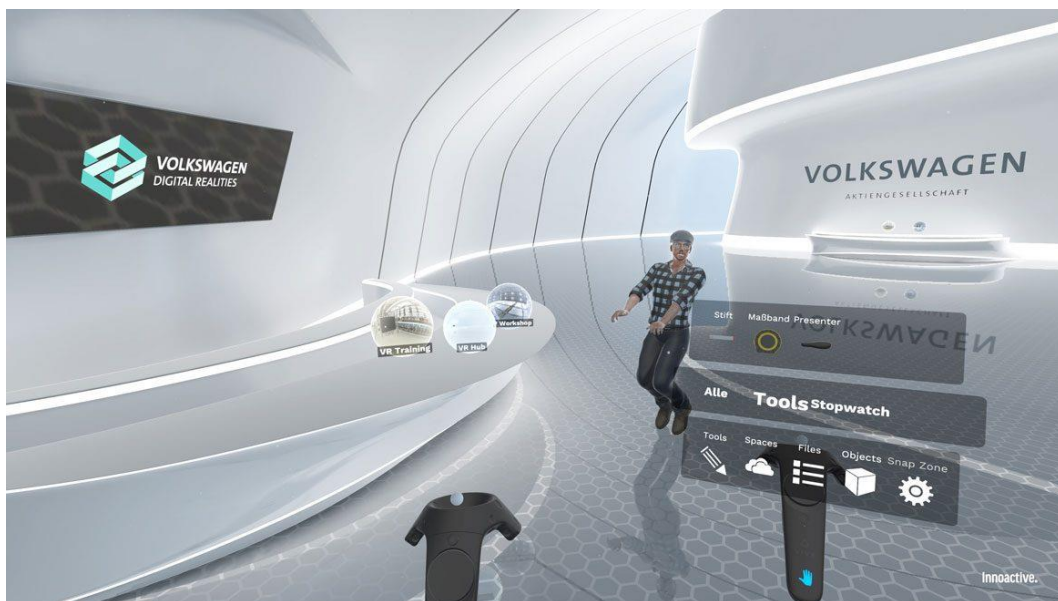


Figure 2.28 - Volkswagen's Digital Reality Hub [56]

Walmart has also embraced VR-based solutions for its L&D sector. Faced with the challenge of training its 1.5 million workers, the company transitioned from traditional methods, such as lectures and quizzes, to a VR-based pilot program. This program has led to a 10-15% improvement in test scores and to a general increase in worker confidence, showcasing the effectiveness of VR for large-scale training initiatives [57].

Many companies are following similar paths. For example, ArborXR is a company that focuses on providing XR (VR and AR) services to other companies. It provides its services to UPS, a logistics company, through a VR based training system for drivers, which has led to a training time reduction of 75%. All of this without even reducing the effectiveness of drivers, who demonstrate equal, if not better competency than those who have underwent traditional training methods.

Delta Airlines is another company that has incorporated a VR based simulation system into its L&D program. The system was developed by a partnership between ArborXR and

Futurus, and ever since the implementation of this system, Delta Airlines has saved millions of dollars annually from minimizing the use of aircraft de-icing fluid during training, from eliminating travel expenses for trainers and technicians and from a diminished amount of fuel consumed by training aircraft. It's a system that puts technicians in a realistic virtual environment to practice de-icing procedures on virtual aircraft, allowing for the training of technicians year-round, more frequently and with less costs.

Companies like Pfizer, Bank of America, and MDA Space have also integrated ArborXR's VR systems into their L&D programs with great success [58].

These examples align with the trends we've discussed earlier in section 2.3.2. They align with the fact that VR has emerged as a key tool for corporate learning. This trend, which began gaining momentum in the 2010s, was further accelerated by the COVID-19 pandemic, which demonstrated the value of remote learning and working environments [59]. In Figure 2.29 we can see how much VR's market size is predicted to increase.

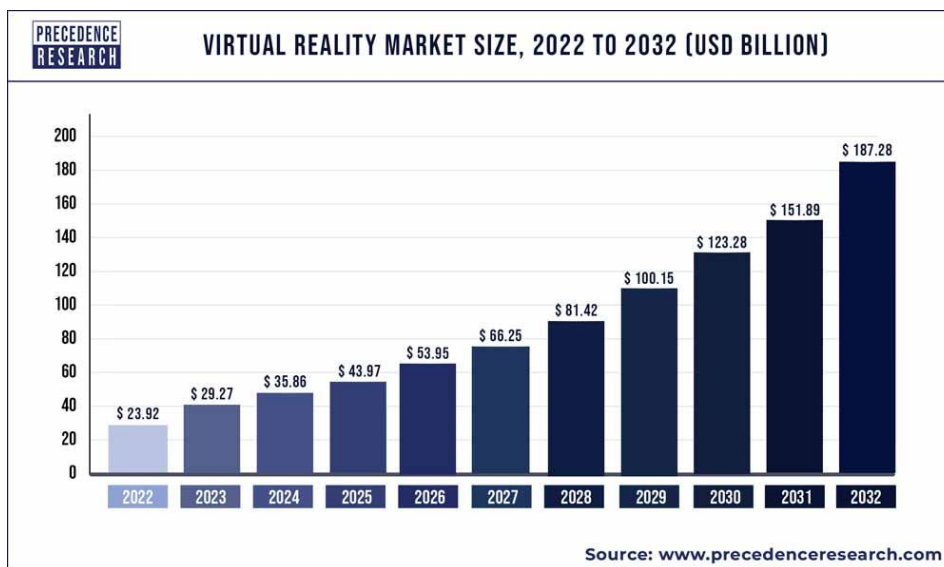


Figure 2.29 - Predicted increase in VR's market size [60]

Digital Twin technology naturally complements these VR trends, as one of the key characteristics of VR learning environments is the simulation of real-world systems or tools. Digital Twins, as the most accurate virtual replicas of physical systems, are particularly useful when high levels of realism are required.

However, one technology not yet mentioned in these examples is generative AI. While VR and Digital Twin tools have already been incorporated into various L&D programs in the past decade, generative AI, especially in the form of advanced NLP models, is a more recent innovation, only gaining widespread popularity in the last few years.

That said, generative NLP models are beginning to make their way into L&D solutions. For instance, Starbucks has introduced a Chat Bot to engage staff in quick training sessions via text, and Volvo uses a Chat Bot to help technicians gain hands-on repair knowledge through conversational simulations. Accenture and Unilever have also employed Chat Bots to address employee questions. However, these examples are relatively simple and do not yet integrate VR or Digital Twin technology [61].

## 2.4.2 Academic Research

At the academic and scientific levels, though, there are a few research papers that describe the combination of VR with AI-powered Chat Bot assistants in a learning context.

One notable example is the system described in a study called "VR Job Interview Simulator: Where Virtual Reality Meets Artificial Intelligence for Education", which provides users with a VR environment where they can simulate job interviews with an AI bot acting as the interviewer. The system even analyses stress levels based on the user's heart rate and skin conductance [62].

Another example is the system explained in the study "The Rise of the AI Scientist: Unleashing the Potential of Chat-GPT Powered Avatars in Virtual Reality Digital-twin Laboratories". This study stands out particularly, because it showcases the use of VR, Digital Twin and Chat Bot technologies all within the same system. This system allows users to learn and collaborate inside the digital twin of a real laboratory (Figure 2.30), where AI-powered assistants help with various tasks[63].



Figure 2.30 - Digital twin of a real laboratory [63]

The study "Exploring the Role of Artificial Intelligence for Augmentation of Adaptable Sustainable Education" [64] gives a general overview of the role that generative AI might play in the world of Education, emphasizing its potential as a tool for supporting personalized learning, adaptive learning methods, and its ability to provide the learner with real-time

feedback, being able to point out mistakes, suggesting alternative approaches and offering particular explanations when a learner struggles with a particular concept.

This potential is also reinforced by the study "Next-Generation Education: The Impact of Generative AI on Learning" [65], which not only, once again, reinforces the role of generative AI in the context of personalized learning, but it also specifically covers the potential advantages of autonomous AI assistants for teaching, and within virtual environments. According to this study, "Generative AI will play a crucial role in creating immersive virtual learning environments that enable students to interact with educational content, simulations, and virtual tutors in a more engaging and interactive manner".

In comparison, the idea of utilizing VR systems as educative tools, is much more well established, as can be asserted from the analysis made in chapter 7 of the book "The Potentials and Trends of Virtual Reality in Education" [66], which reinforces the trends described in chapter 2.3.2 of this document with the graph shown in Figure 2.31. The graph depicts an exponential increase in publications related to virtual reality in education, between 1995 and 2016, showing how there were already hundreds of papers on this subject almost 10 years before the writing of this dissertation.

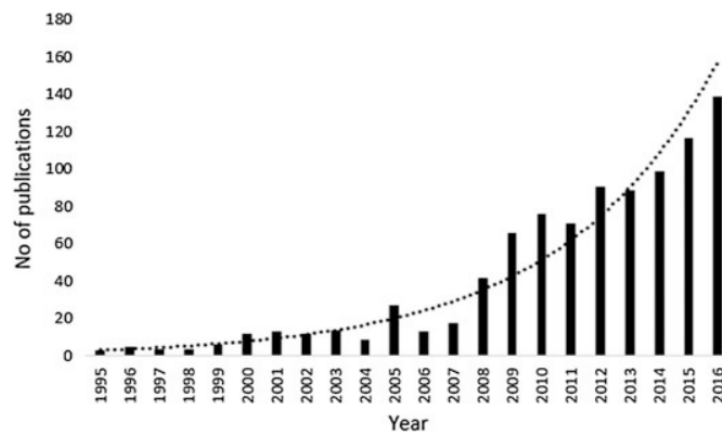


Figure 2.31 - The growth in publications about virtual reality in education from 1995–2016

The potential applications of Digital Twins in education are also not a new concept, with studies like "Automation in Education with Digital Twins: Trends and Issues" stating that it's a technology that is already applied in E-Learning and distance education contexts. The value that Digital Twin technologies bring to students, especially in technical courses where the systems are complex and hands-on training on these systems is important, is also reinforced in the paper "Applications of Digital Twin across Industries: A Review" [67].

### 2.4.3 Key Takeaways

From these examples, several conclusions can be drawn. VR and Digital Twin technologies have already secured their place as valuable tools for L&D. This is clear from the numerous already existing examples proving their effectiveness and utility in various educational contexts, as well as the large number of already existing academic papers on this subject.

However, the introduction of AI Chat Bots into this space, particularly in combination with VR and Digital Twins, is still in its early stages. The concept isn't new, but there aren't as many examples of its efficacy.

This lack of case studies on the integration of AI Chat Bots in VR and Digital Twin based educational systems is what represents the main knowledge gap that this dissertation aims to address. The goal of this project is to explore the concept of creating learning tools that not only leverage immersive VR technologies and realistic virtual replicas, but also incorporate the latest generative AI technologies, particularly in the form of Chat Bots acting as instructors or assistants in training.



## DEVELOPMENT

Now that we have reviewed the state of the art for the technologies and concepts involved, analysing each tool, its general purpose, and its potential applications within the context of this dissertation project, we can proceed to examine the development process of the system.

In this chapter, we will explore how the three main technological tools of the system were integrated into Unity. We will also explore the system's architecture, its various components, how they interact, and the rationale behind their development choices.

### 3.1 System Architecture

As mentioned earlier, this project serves as a proof of concept for a VR-based teaching tool, with its standout features being the integration of a digital replica and, most notably, an AI Chat Bot assistant. With the digital replica emulating the role of a digital twin, of providing a precise, virtual representation of an industrial robot, while the AI Chat Bot acts as a teaching assistant, enhancing the learning experience.

The system was developed as a Unity application, requiring a computer and VR tools, which in this case, for the development and testing process, were Meta Quest 3's VR tools.

Unity was selected as the primary development platform for this system primarily due to its free access and ease of use. Unity is also a widely adopted tool with an extensive, supportive community and numerous external resources, which can be easily integrated.

Unity applications are built within one or more scenes, which serve as the primary virtual assets where systems are developed. Typically, various scenes are created, each containing a distinct and significant part of the application's architecture.

For this system, three scenes were developed, the Main Menu scene, the Free Tutorial scene, and the Controlled Tutorial scene. The Main Menu scene functions as the starting point of the application and as the central hub from where users can select their preferred training mode. The two tutorial scenes offer different simulation environments, one provides users with complete freedom to train as they wish, while the other guides users through a sequence of pre-designed tutorial steps.

### 3.1.1 Main Requirements

Before delving deeper into the development process of this project, it's important to re-emphasize its main requirements and goals.

The primary objective is to create a proof of concept for a VR learning application, utilizing, as mentioned earlier, a digital replica of a robot (emulating a digital twin), and AI in the form of an AI assistant. Our aim is to demonstrate the feasibility of using these technologies together to enhance the learning process. The key requirements are:

- The system must feature some sort of digital replica of a manufacturing robot, controllable by the user within the virtual environment using realistic controls.
- It must provide an immersive environment that the user can experience in VR.
- The AI assistant must be accessible throughout the learning experience and capable of providing meaningful feedback.
- The application must be capable of teaching the user.

More specifically, the system should also have an intuitive starting menu serving as a user-friendly interface from where the user can decide what to do inside the application. It should be the initial environment the user enters upon launching the application. From there, the user gets presented with two options, Guided Robot Tutorial and Free Robot Tutorial. This first interface should then serve as a central hub for navigating the application's features, that the user can experience in VR.

The main menu should also provide the user with enough context and information for them to be able to understand the basic simulation controls (that the user will need to be aware of before they enter the actual simulation). The main simulated environments, Guided Robot Tutorial and Free Robot Tutorial, should then be accessible from the initial main menu.

Both simulated modes can share the same key features, both could share the same 3D environment, the same robot and the same user control setup. They don't need to be significantly different. The only real difference between these modes must simply be the type of learning experience that they provide to the user.

The Guided Robot Tutorial will provide the user with a more restricted and controlled experience. The user will be guided by the system through sequence of steps they must complete. Ideally the user receives frequent feedback through the system's text-to-speech (TTS) functionalities, on what task they must complete and on whether they have completed said task. This means the system in this simulated mode will have to analyse closely the user's actions, so that it can determine when the user has completed the required task, and when the user requires assistance,

The Free Robot Tutorial on the other hand won't offer the user any guidance, the user will be free to train however they please, they won't be under any kind of restriction, and they won't have to complete any specific task.

Both modes must allow the user to communicate with the AI assistant with ease, with the AI assistant being able to give useful tips to the user when asked. In the case of the Guided Robot Tutorial, the AI assistant should also be aware of the user's progression within its sequence of tasks (or steps).

The AI assistant will must be able to communicate with the user through voice, meaning that speech-to-text (STT) methods will be used to interpret what the user says, and then TTS tools will be used to provide the user with a verbal response from the Assistant. Due to the characteristics of the STT system used, it's expected that the user will need the means to choose which microphone to use, and the time window size during which the STT system will be able to record the user's inputs.

The robot control system must be realistic, it must have functionalities similar to real robot control systems, based on real controller frameworks. It should also, if possible, emulate the feeling of holding a real handheld controller.

In the following subchapters, the focus will be on explaining the general organization and architecture of the application, as well as how the integration of the mentioned technologies was accomplished.

### 3.1.2 General Simulation Architecture

Now that a broad overview of the system's requirements has been provided, the next step is to analyse in detail how the system functions, and how its components interact with each other. The focus will be on the Unity scenes dedicated to the simulated training environments, as they are the most critical for demonstrating the proof of concept. It is within these simulation scenes that both the AI assistant and the digital replica of the robot are present.

To effectively study the architecture of the system while maintaining a reasonable level of detail, a UML-based diagram was created. This diagram represents the key blocks of the system, their components, and how they interact with one another. The complete UML diagram is shown in Figure 3.1 below.

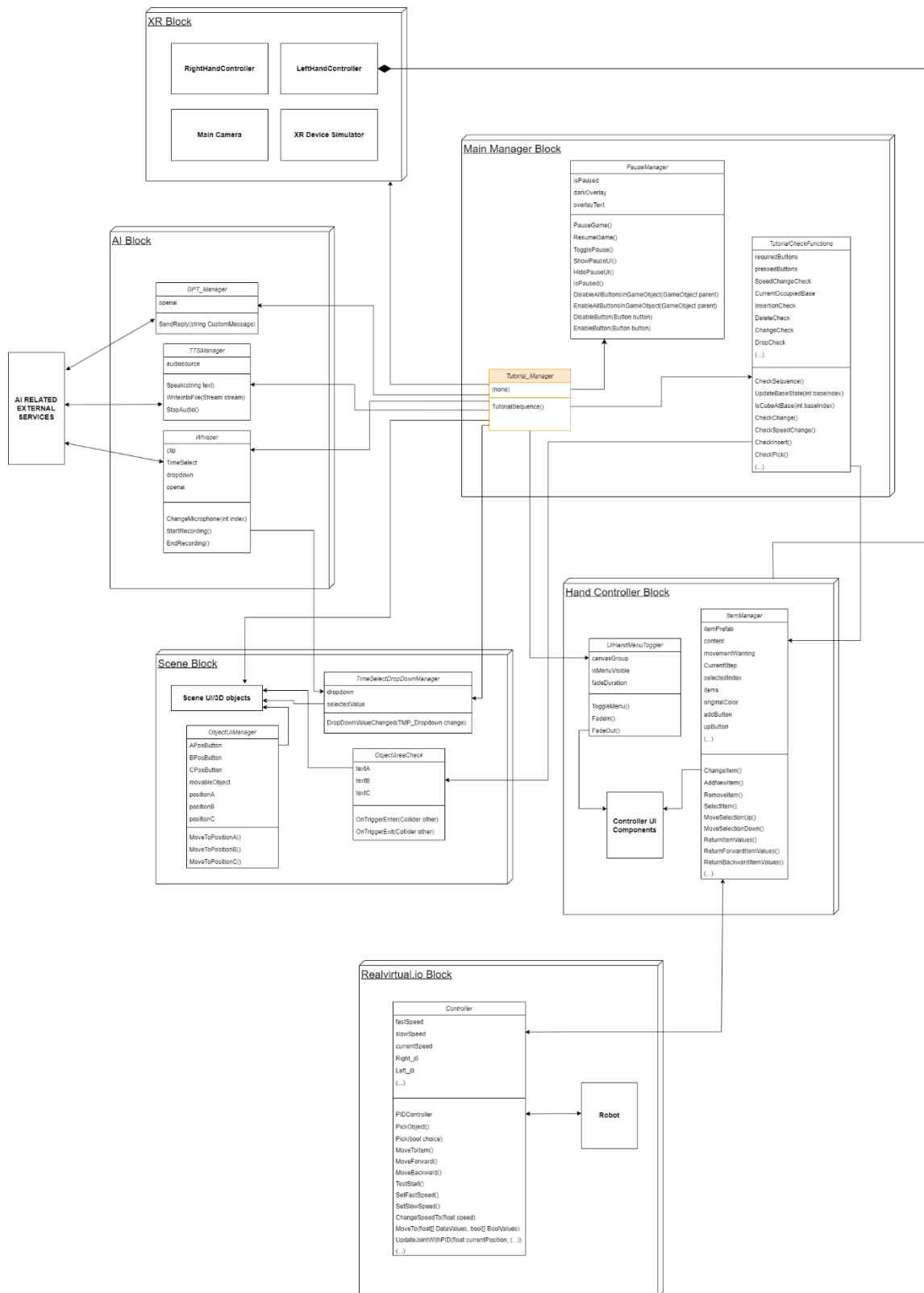


Figure 3.1 - General System Architecture

This system is divided into the following six main blocks, each containing various scripts and components:

- **Main Manager Block:** This block contains the main manager script and two other scripts that work closely with it. The main manager script of this system is the Tutorial\_Manager (highlighted in yellow in the diagram). As the name suggests, its purpose is to serve as a central manager that helps all the blocks and components of the system work together.
- **AI Block:** This refers to the API scripts responsible for AI services in the system. It contains the APIs required for the integration of ChatGPT, Whisper and AWS's TTS (which will be analysed in more detail in section 3.4).
- **Scene Block:** This block includes the scripts and game objects that define and control the 3D room in which the simulation takes place.
- **XR Block:** This contains components related to the integrated VR tools of the system, including the XR device simulator, each hand controller, and the main camera component.
- **Hand Controller Block:** This block includes the scripts and corresponding UI components for managing the user's robot control system, and it maintains a composition relationship with the LeftHandController since it's integrated into it.
- **Realvirtual.io Block:** This block refers to the digital robot and its main controller script. It contains all the components related to Realvirtual.io's framework, which is responsible for handling the digital replica and defining its functions (further analysis on it will be conducted in section 3.2).

Most relationships in this UML diagram are associations because scripts within the system typically utilize functions from one another, and they do so regularly. In cases where there's a nearly equal, mutual exchange of data or use of methods from both sides, the relationship is a bidirectional association. This is the case between the Robot and the Controller, between the ItemManager and the Controller, and between the AI scripts and the external AI services they communicate with.

In the following subchapters, each of these blocks will be explained in more detail, with additional context and close-up images of their respective interactions in the diagram.

### 3.1.2.1 Main Manager Block

As mentioned earlier, the Main Manager block (Figure 3.2) refers to the Tutorial\_Manager script and its key satellite scripts.

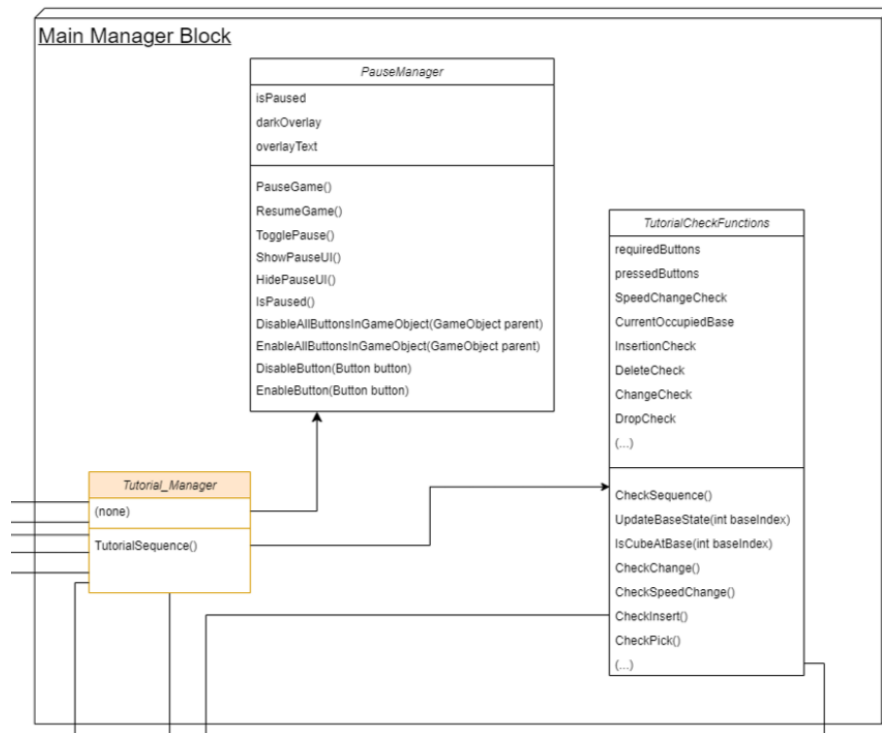


Figure 3.2 - Main Manager Block

The Tutorial\_Manager script is responsible for assigning key binds to the main VR simulation controller actions. For example, the user communicates with the AI assistant by pressing "Y," skips instructions by pressing "A," silences the AI voice with "X," returns to the starting menu with "B," and toggles the hand controller with the left grip button. These key bindings are all defined in the Tutorial\_Manager.

In addition, this script utilizes the AI APIs to provide the user with a dynamic AI Interaction experience, handling STT recording, when messages are sent to GPT-3.5 and when TTS is used.

In general, it acts as a middleman to connect the user with the various functionalities of the simulation scene.

However, the most complex role of this block is managing the guided tutorial sequence for the simulation's guided tutorial mode.

Since the primary objective of this project is to create a learning tool, it was essential to determine the type of learning experience to implement within the system. With the ability to communicate with users through an AI voice, the system offers the flexibility to provide both

a free, open learning experience and a more structured, guided learning experience. This led to the development of the two distinct learning modes, which were previously described.

The free learning mode is straightforward, giving the user complete control over the simulation. In this mode, the AI assistant only offers help when explicitly requested by the user. However, the guided mode is more complex, as it required the development of a structured learning plan, broken down into steps.

The primary goal of this learning plan was to gradually familiarize the user with the robot controls, progressing from the simplest functions to the more advanced ones. With this in mind, the following nine steps were defined for the guided tutorial:

- **STEP 1:** Learn how to move the robot with the twelve joint movement buttons.
- **STEP 2:** Learn how to change the robot's speed mode.
- **STEP 3:** Learn how to pick up and drop objects with the robot's gripper.
- **STEP 4:** Learn how to return the object to its original base.
- **STEP 5:** Learn how to save and delete robot positions.
- **STEP 6:** Learn how to change saved robot positions and how to move the robot to saved positions.
- **STEP 7:** Learn how to make a sequence of saved positions.
- **STEP 8:** Learn how to move the robot forward and backward In the sequence, step by step.
- **STEP 9:** Learn how to test a whole sequence with the Test Start button.

This guided tutorial is a coroutine function that is divided into these steps, with each step requiring certain conditions to be met before progressing. Coroutines are a special type of function that can be paused using a "yield" statement, which prevents the routine from advancing until a specific condition is fulfilled. The beginning of this function can be seen in Figure 3.3.

Understanding how coroutines work helps explain the structure of the guided tutorial and the significance of the TutorialCheckFunctions script. Each step in the tutorial is separated by yield statements, which act as checkpoints that depend on the conditions defined in the TutorialCheckFunctions script. This script is crucial for checking whether the user has completed the necessary steps to progress through the tutorial. It does so by retrieving information from both the scene block and the hand controller block.

```
Tutorial_Manager.cs [X]
Arquivos Diversos Tutorial_Manager
307
308 private IEnumerator TutorialSequence()
309 {
310     Debug.Log("Tutorial Initiated!");
311     // Wait for the specified amount of time
312     yield return new WaitForSeconds(5f);
313     ActivateAPressCheck = true;
314
315     // using the basic joint moving buttons
316     #region STEP 1
317     tts.StopAudio();
318     pauseManager.PauseGame();
319
320     // Disable all menu buttons outside of the manual joint movers
321     pauseManager.DisableAllButtonsInGameObject(SpeedButtons);
322     pauseManager.DisableAllButtonsInGameObject(ArrowButtons);
323     pauseManager.DisableAllButtonsInGameObject(FwdBwdButtons);
324     pauseManager.DisableAllButtonsInGameObject(InsDelModButtons);
325     pauseManager.DisableAllButtonsInGameObject(MoveChange);
326     pauseManager.DisableButton(PickDropButton);
327     pauseManager.DisableButton(TestStartButton);
328
329     //Debug.Log("1 - TRY THE SIMPLE JOINT MOVE BUTTONS");
330     tts.Speak("Welcome to the guided tutorial for learning the basic controls of the test robot");
331     instructions.text = "Welcome to the guided tutorial for learning the basic controls of the test robot";
332
333     yield return new WaitUntil(C => AWasPressed);
334     tts.StopAudio();
335     AWasPressed = false;
336     pauseManager.ResumeGame();
337
338 }
```

Figure 3.3 - Snippet of code from the TutorialSequence coroutine function

The TutorialSequence coroutine relies on various system components, including the constant use of text-to-speech (TTS), of scene block components, and of TutorialCheckFunctions' methods to verify when the user is ready to move to the next step. Additionally, it updates the AI assistant of what the current step is, every time the user advances in the tutorial.

Another critical script used in the tutorial sequence is the PauseManager. It manages the pausing of the simulation that happens while tutorial instructions are being delivered to the user. At the start of each new step, the simulation is paused, and a notice appears in front of the user with the message "Instructions" and "Press A to skip" (which can be seen in Figure 3.4). During this time, TTS is used to explain the objectives of the next tutorial step. The methods in the PauseManager script control these pauses and also handle the enabling and disabling of buttons on the hand controller.

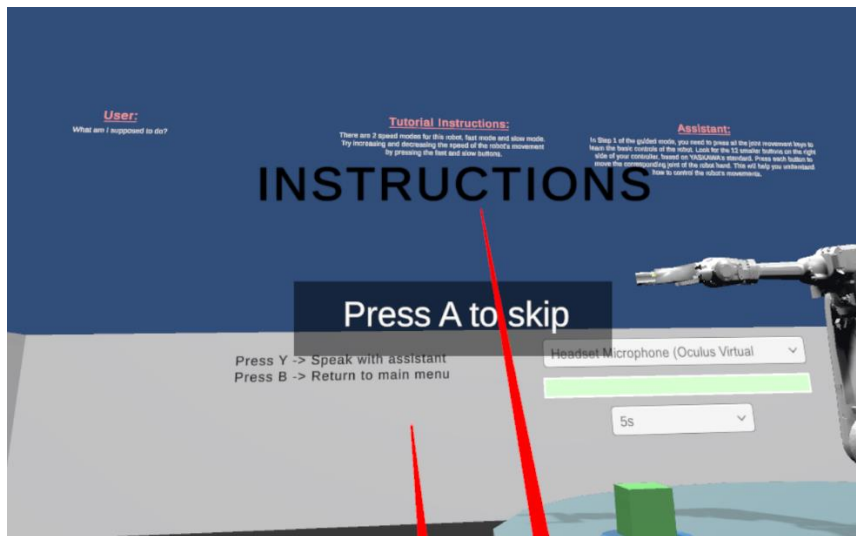


Figure 3.4 - Pause screen during instruction delivery in guided mode

The tutorial begins with simple commands, gradually introducing more complex tasks. To help users stay focused, unnecessary buttons are disabled during the early steps. For instance, at the beginning of the training sequence, all buttons except for the joint control buttons are disabled (Figure 3.5). As the tutorial progresses, more buttons are enabled, allowing the user to access all available robot controls by the end of the tutorial.



Figure 3.5 - Disabled buttons on the handheld controller in guided mode

The general interactions within the guided tutorial can be illustrated using a UML sequence diagram. Figure 3.6 provides a simplified representation of the key processes that occur during the first three steps of the guided tutorial. For the sake of clarity, not every procedure is included in this diagram. Instead, only the most relevant interactions are depicted, offering

a general overview of how the guided tutorial is structured and the types of interactions it involves.

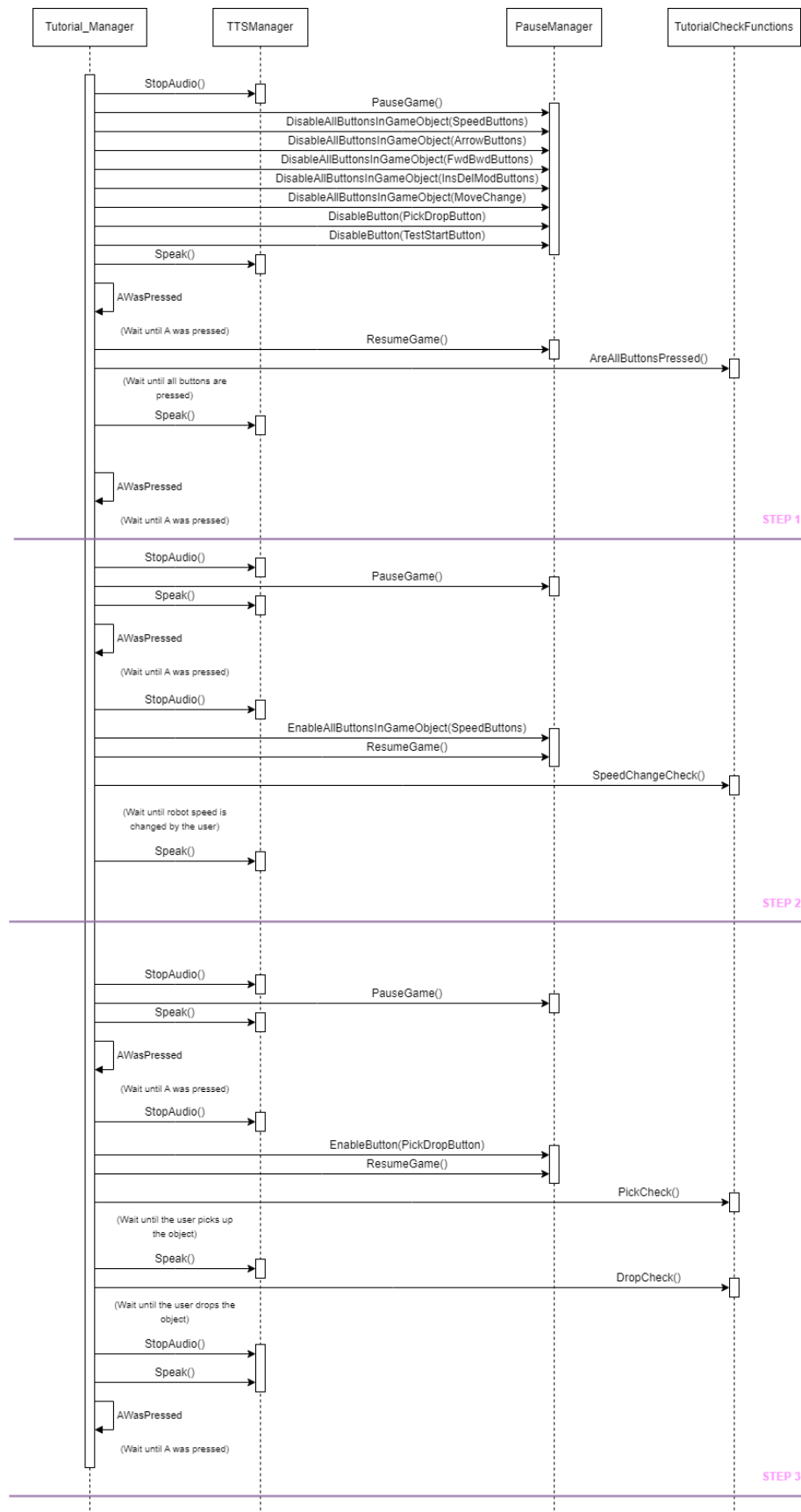


Figure 3.6 - Sequence diagram representing part of the guided tutorial's progression

From analysing the diagram, it becomes evident how crucial the PauseManager, TutorialCheckFunctions, and TTSManger scripts are to the system. The TTSManger's Speak and StopAudio methods are frequently invoked to either provide new instructions to the user or to notify them of when a task has been completed. The PauseManager's functions are Invoked, whether to pause or resume the simulation when new instructions are told, or to disable or enable buttons from the handheld robot controller. And the TutorialCheckFunctions script is called at the end of each step to verify that the user has successfully completed the required tasks before progressing to the next step.

Overall, the Main Manager block defines the user's controls within the scene, coordinates them with other scene components, and defines the tutorial sequence for the guided tutorial mode. This makes it one of the most Important blocks in the system.

### **3.1.2.2 AI Block**

The AI block (Figure 3.7) is composed of three key scripts, TTSManger, GPT\_Manager, and Whisper, each responsible for integrating external AI services into the system. TTSManger connects with AWS's text-to-speech service, with the necessary AWS credentials embedded directly in the script. On the other hand, both GPT\_Manager and Whisper rely on OpenAI's services, both requiring the same OpenAI credentials, though these are stored in a separate file on the computer rather than within the API scripts themselves.



Figure 3.7 - AI Block

These scripts' functions play a crucial role in the system, frequently being called by the Main Manager block.

GPT\_Manager's SendReply function is responsible for sending a user message to the AI model and receiving a response, TTSManger's Speak function converts written text into spoken audio and the Whisper script handles audio detection and its conversion into text through its StartRecording and EndRecording functions.

Each of these scripts maintains a bidirectional relationship with their respective external services, as they rely on continuous communication with them. Although all three scripts are extensively used by the Tutorial\_Manager script, Whisper, unlike the other two, also interacts with other system components.

For instance, Whisper needs to know for how long it's meant to record the user's audio before starting its audio to text conversion process, information that is provided through the user's interaction with the scene. The user selects the desired recording duration, ranging from 5 to 20 seconds, via a dropdown UI menu. Additionally, the system must determine which microphone to use, another choice the user makes through an interface element. As a result, the Whisper script is associated with the TimeSelectionDropDownManager script, which

handles the time selection logic, and with the drop-down selection interface that manages which microphone will be used (Figure 3.8).

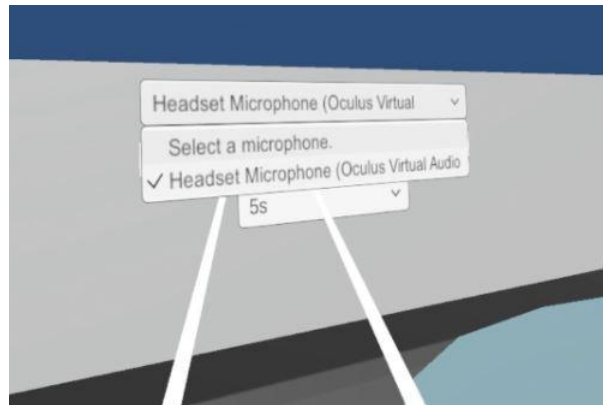


Figure 3.8 - Microphone selection

### 3.1.2.3 Scene Block

The Scene block (Figure 3.9) includes all the primary 3D and UI game objects related to the virtual environment, such as the floor, walls, and various interactive UI components from the virtual room.

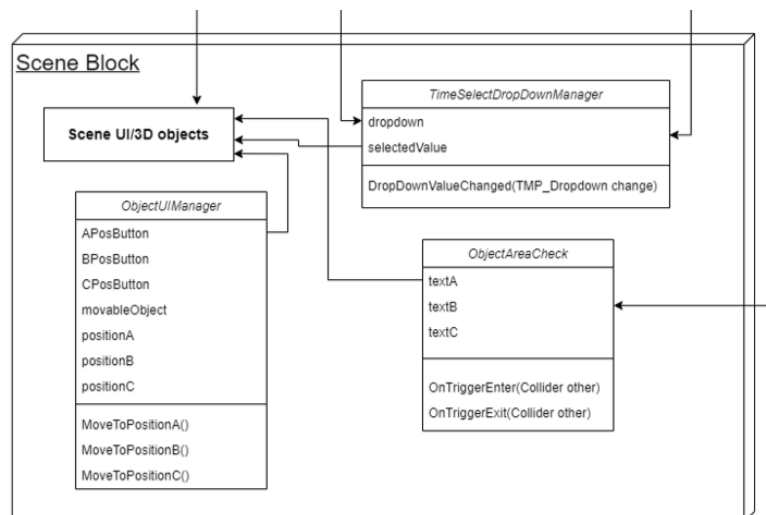


Figure 3.9 - Scene Block

One key script, TimeSelectDropDownManager, manages the dropdown selector that determines the recording duration for the STT (Speech-to-Text) functionality (as seen in Figure

3.10). This script is used by both the Tutorial\_Manager and Whisper scripts, which work in tandem to control the recording process.



Figure 3.10 - Recording time selection

Another important script, ObjectUIManager, oversees the interaction between the user and the UI screen that allows them to move a cube between different bases in the virtual environment. It manages the relationships between the cube, the three bases, and the screen interface, ensuring the user can easily shift the cube's position through an interactable UI screen, next to the robot's platform, as seen in Figure 3.11.

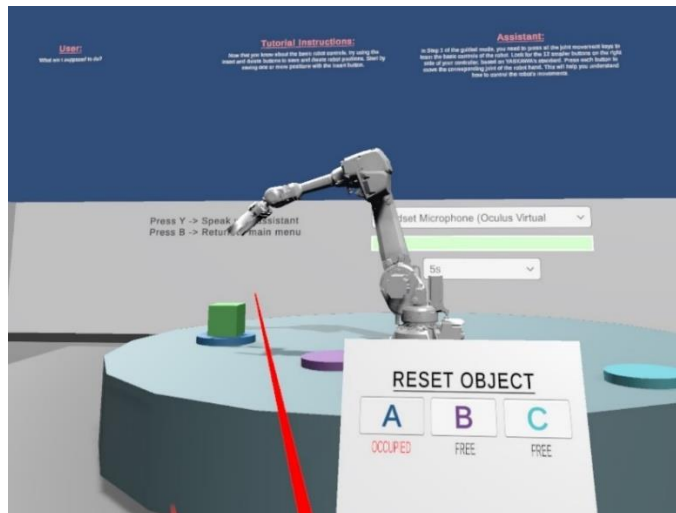


Figure 3.11 - Cube repositioning screen

ObjectAreaCheck contains the logic responsible for detecting when the pickable cube is placed on any of the bases. It signals whether or not the cube is present at a particular base, and this information is frequently used by the TutorialCheckFunctions script, as many of the tutorial steps depend on the cube being positioned at or moving between specific bases.

Together, these scripts interact with the 3D and UI components of the virtual scene, providing information to the Main Manager bloc and allowing the user to engage with the system through the scene's interface.

### 3.1.2.4 XR Block

The XR bloc (Figure 3.12) refers to the system components responsible for the VR functionality. These components were imported from an official Unity source and have remained mostly unmodified since their integration.

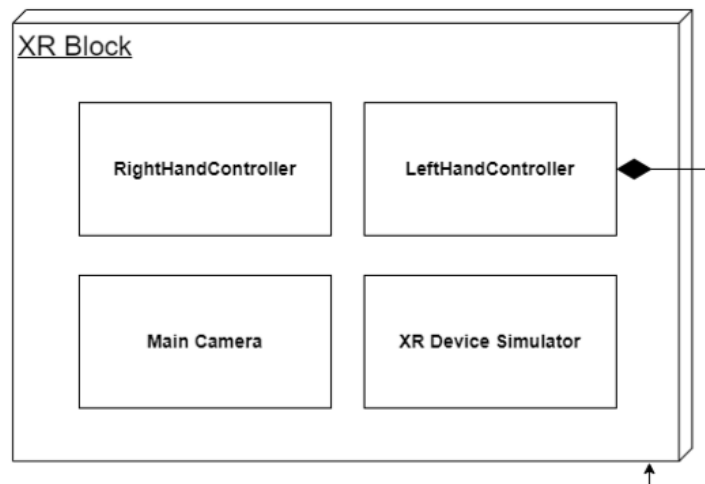


Figure 3.12 - XR Block

One key component of this bloc is the XR Device Simulator, which provides a framework for testing VR controls without the need for actual VR equipment. This feature was especially useful during the early stages of development, although it was disabled for the final product.

The RightHandController and LeftHandController are also part of this block, as they represent the VR hands that the user sees and interacts with inside the system (visible in Figure 3.13). These components define the VR functionalities for the hand controllers. The LeftHandController is particularly notable because the entire Hand Controller bloc was assigned to it, as reflected by its composition relationship. This setup ensures that the hand controller feels like a natural extension of the user's hand within the virtual environment.

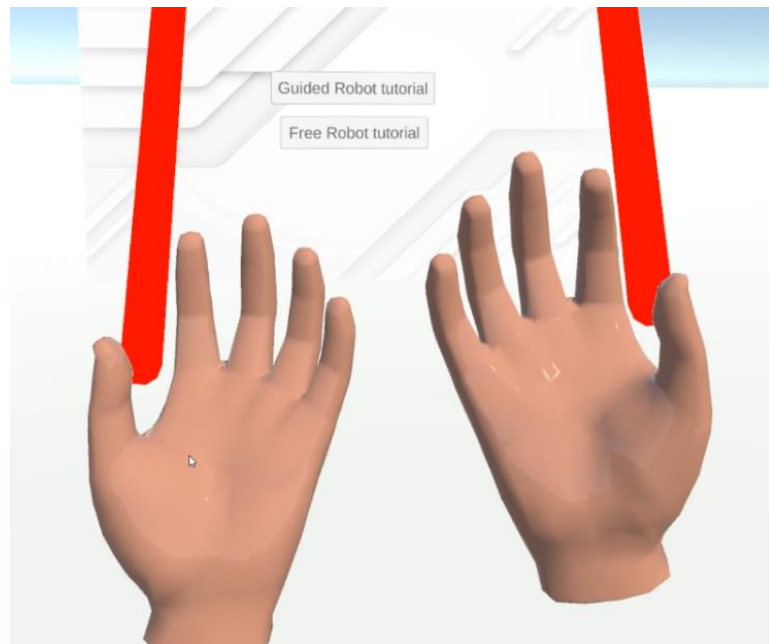


Figure 3.13 - User's hands inside the virtual environment

Finally, the XR block includes the main camera, which, as the name suggests, represents the player's field of view.

The Main Manager block interacts with the XR block by utilizing its features to map VR controls to relevant system functions.

### 3.1.2.5 Hand Controller Block

The hand controller block refers to the scripts and system components that are closely connected to the robot's virtual controller, allowing the user to control the robot.

Deciding on which robot control types to train users on was a key part of this project. Most robot control systems include methods for moving individual joints, saving robot positions, and moving the robot along those positions, among other features. These controls are closely linked to how the robot is programmed, and the methods can vary depending on the controller's framework. This means that it was important to study about some of the robot control frameworks that exist, in order to have some idea about which framework(s) the simulated control system would be based around.

One of the frameworks studied was ABB's, which involves the FlexPendant controller (Figure 3.14) and Robotstudio software. The FlexPendant is a controller with a screen on its left side and several buttons, along with an analog stick, on the right. With the FlexPendant, users can move the robot in three modes:

- **Linear mode:** Controls the robot's end-effector position along X, Y, and Z coordinates, with the robot's base as the center.
- **Axis mode:** Controls the individual joints of the robot.
- **Reorient mode:** Adjusts the robot's orientation without changing the end-effector's position.



Figure 3.14 - FlexPendant [68]

The FlexPendant also allows users to save robot positions, adjust movement speeds, and more. These saved positions can be transferred to a computer, where they can be used to program the robot in Robotstudio using RAPID, ABB's programming language, as illustrated in Figure 3.15.

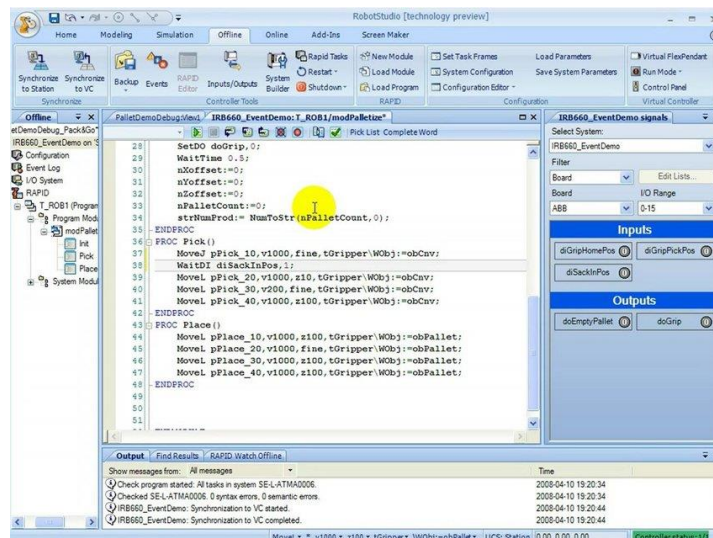


Figure 3.15 - Coding in RAPID within Robotstudio [69]

However, this framework presented challenges for the project. The FlexPendant relies heavily on its analog stick for moving the robot in different modes. During the early stages of



As a result, the final controller design was based mostly on Yaskawa's framework, specifically controllers like the DX200 and YRC1000. The physical shape of the controller, though, is more similar to the FlexPendant, with a screen on the left and buttons on the right.

Since there was no real robot controller available as a digital twin, the controller ended up being represented as a virtual projection (Figure 3.17), which can be toggled on and off on the user's left hand to simulate the holding of a real handheld controller.

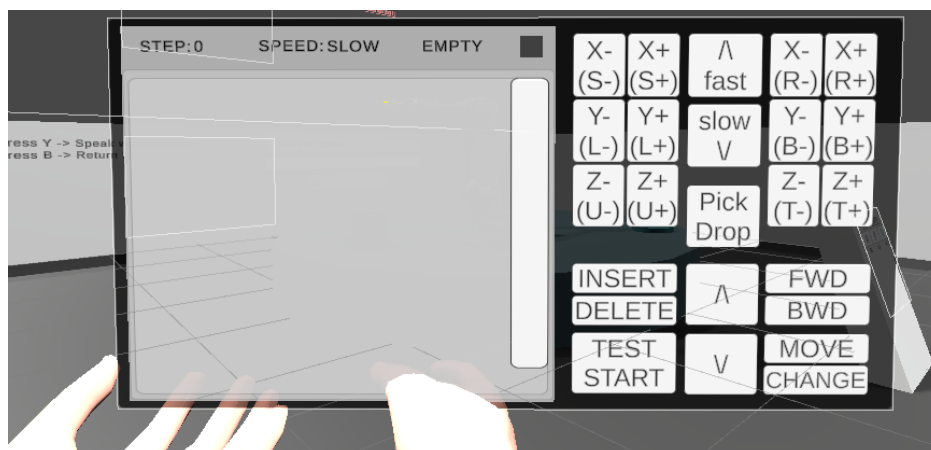


Figure 3.17 - Virtual handheld robot controller (editor view)

The controller in the training simulation ended up being a mix between Yaskawa's framework and the FlexPendant's design.

The simulated controller allows for individual joint control with twelve buttons, based on Yaskawa's SLURBT framework (Figure 3.18). This nomenclature refers to the six typical axes of joint movement in industrial robots [71]:

- **S:** Swivel base
- **L:** Lower arm
- **U:** Upper arm
- **R:** Arm roll
- **B:** Wrist bend
- **T:** Tool flange

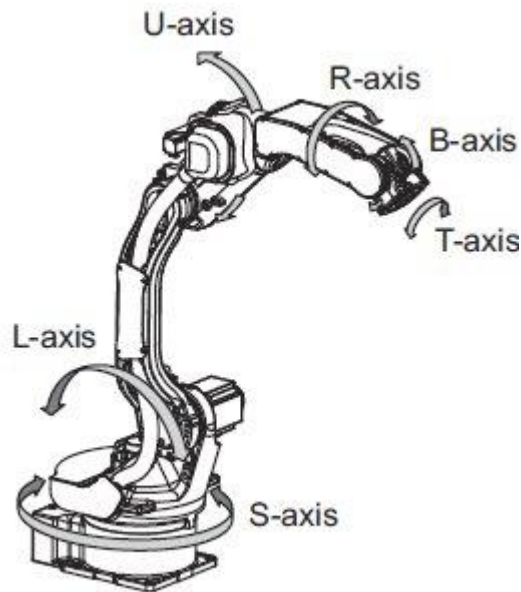


Figure 3.18 - SLURBT [71]

In addition to these 12 buttons, there are also buttons for changing the robot's speed (FAST and SLOW), and another button for picking up or dropping objects.

The position-saving system works similarly to a Yaskawa controller. Users can save the robot's current position with a button press, and saved positions appear as a list on the left screen of the controller. Each entry shows the joint values, the gripper state (whether it is holding an object or not), and the speed mode at the time of saving. Users can navigate through the list with two arrow buttons, and there are buttons assigned for the insertion, deletion, or changing of positions.

Inspired by Yaskawa's controllers, the simulation also includes forward (FWD) and backward (BWD) buttons, which allow the user to move the robot step by step along saved positions, a useful feature for reviewing a movement sequence. The Move button moves the robot to a selected position, and the Test Start button allows users to test an entire sequence of positions.

This controller lets users move the robot, pick up objects, and program basic movement sequences. However, compared to real controllers like the FlexPendant or Yaskawa's controllers, this virtual version is very simplified. Features such as manually editing saved positions, saving/uploading movement sequences, or copying/pasting positions are not available. Additionally, coordinate-based movement is not included, mainly because this would require inverse kinematics calculations, which are complex and would require more advanced tools,

which exist in more expensive Realvirtual.io Unity packages, but not in the starter kit used for this project.

Despite these limitations, the simulated control system still provides enough functionality and realism to demonstrate the project's key concepts.

Now that we've reviewed the reasoning behind the hand controller's design and framework, it becomes easier to understand how it integrates into the system's architecture as the Hand Controller block (Figure 3.19). This block includes the controller's UI components and two scripts.

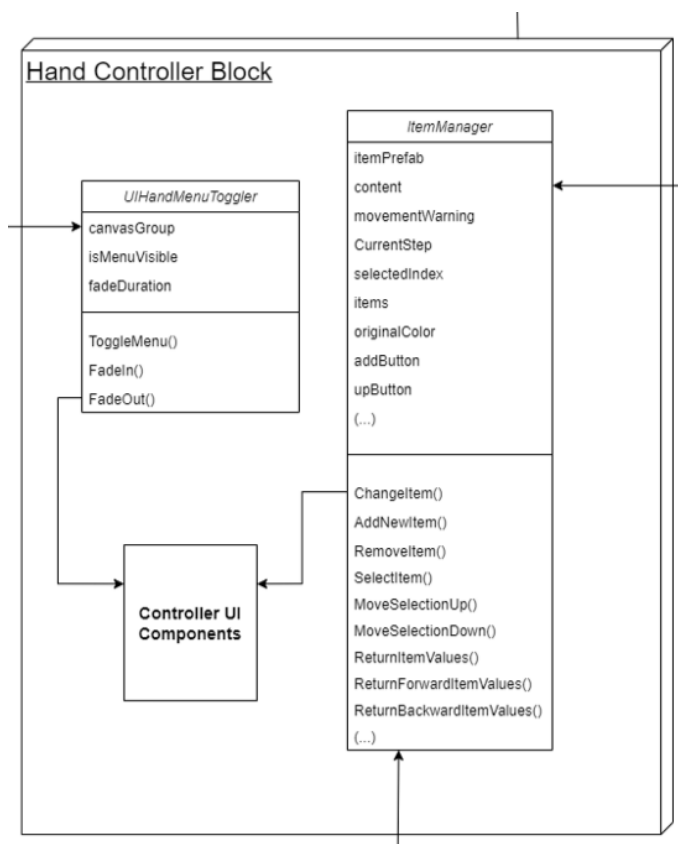


Figure 3.19 - Hand Controller Block

The ItemManager is the main manager for the entire handheld robot controller system. It contains a dynamic list of items for storing robot positions, and its functions are focused on managing the controller's UI and facilitating communication between the controller and the main Realvirtual.io robot control script. The ItemManager maintains a bidirectional relationship with the robot's main controller (the Controller script), representing the constant trade in data and instructions that is established between the two. The ItemManager is also connected to the TutorialCheckFunctions script, which relies on its data for some of its conditional methods.

The second script, UIHandMenuToggler, manages the toggling of the Hand Controller’s UI menu. It is linked to the Tutorial\_Manager, which assigns this script's methods to be triggered when the user presses the left VR controller's grip button.

**3.1.2.6 Realvirtual.io Block**

The Realvirtual.io block (Figure 3.20) refers to the components that directly relate to the system’s digital replica and its functions.

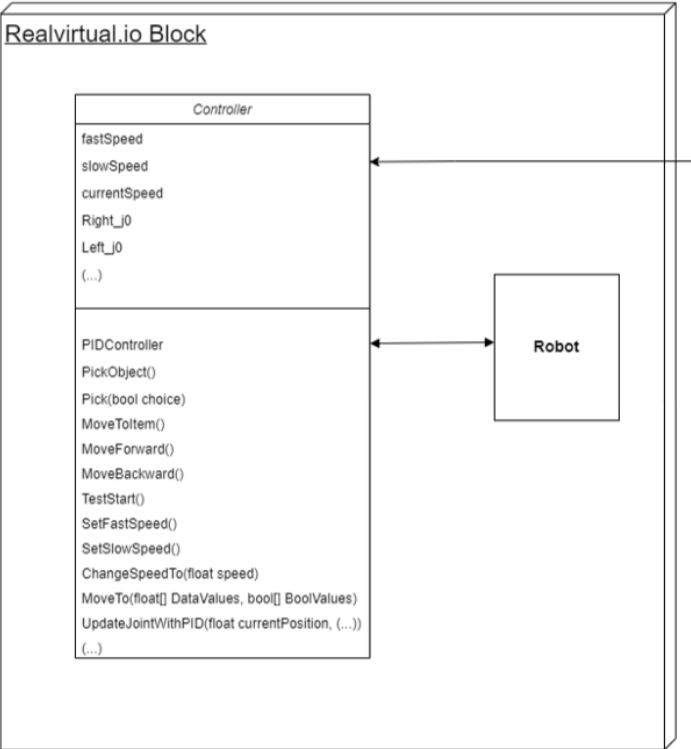


Figure 3.20 - Realvirtual.io Block

Realvirtual.io is the company behind the robot control framework used in this project. It was already previously mentioned in the state-of-the-art chapters and will be discussed in further detail in a dedicated section.

As noted earlier, during the development process, there was no actual digital twin available to work with. Instead, a digital replica of a manufacturing robot from Realvirtual.io’s package was used to emulate the role that would be fulfilled by a digital twin in a more complete system.

This block includes the Controller script and the digital robot. The Controller script acts as a PLC (Programmable Logic Controller) for the robot's logic. This script defines the various

adjustable settings and possible movements of the robot, and its functions are primarily called by the ItemManager script, as explained earlier.

Because the Controller script frequently sends information to and receives instructions from the ItemManager, it maintains a bidirectional association with it. In addition to this, the Controller script also has a bidirectional relationship with the digital robot itself, as it controls its movements and receives valuable data from it via Realvirtual.io's methods and signals.

The specific methods and signals used within the Realvirtual.io framework will be discussed in greater detail in chapter 3.2.

## 3.2 Realvirtual.io Framework

To demonstrate the concept of an immersive VR learning experience dedicated to teaching professionals how to control a robot, it would be ideal to incorporate a digital twin, a realistic representation of the robot that behaves similarly to its real-world counterpart.

As discussed in the state-of-the-art chapter of this document, a digital twin is a detailed virtual replica of a physical object or system, often created using extensive data collected from sensors connected to the real object over time. However, given the relatively small scale of this project, specifically in terms of resources availability, it wasn't possible to work with a fully featured digital twin paradigm, so the decision primarily involved either creating a custom virtual robot using 3D modelling tools like Blender or downloading pre-made virtual models from the internet. Ultimately, the final decision was to use the framework, tools and a digital replica from a company that was previously mentioned in this document, Realvirtual.io.

Realvirtual.io is an industrial simulation company that recently adopted Unity as a development platform, recognizing its potential to rival traditional industrial simulation tools [52]. The tools and framework offered by Realvirtual.io are available on the Unity Asset Store, with various packages providing different capabilities (Figure 3.21). For this project, the Realvirtual.io Digital Twin Starter package was selected due to its affordability and its ability to meet the primary objectives of the project.

Results 1-6 of 6 for realvirtual.io







 <p>realvirtual.io Simulation 2022 REALVIRTUAL €552.02</p>	 <p>realvirtual.io Digital Twin Starter 2022 ★★★★★ (25) REALVIRTUAL €9.20</p>	 <p>realvirtual.io CADLink REALVIRTUAL €414.02</p>	 <p>realvirtual.io Digital Twin Professional 2022 REALVIRTUAL €837.24</p>
 <p>realvirtual.io Digital Twin Professional 2021 REALVIRTUAL €763.63</p>	 <p>realvirtual.io Industrial Metaverse 2022 REALVIRTUAL €174.80</p>		

Figure 3.21 - Realvirtual.io packages available on the Unity asset store [72]

The Realvirtual.io starter package includes several example scenes that demonstrate how the framework operates. It provides realistic robot models and significantly modifies the Unity development environment, adding features that enhance how scenes are viewed in development mode, along with shortcuts for easily adding Realvirtual.io specific components, as seen in Figure 3.22.

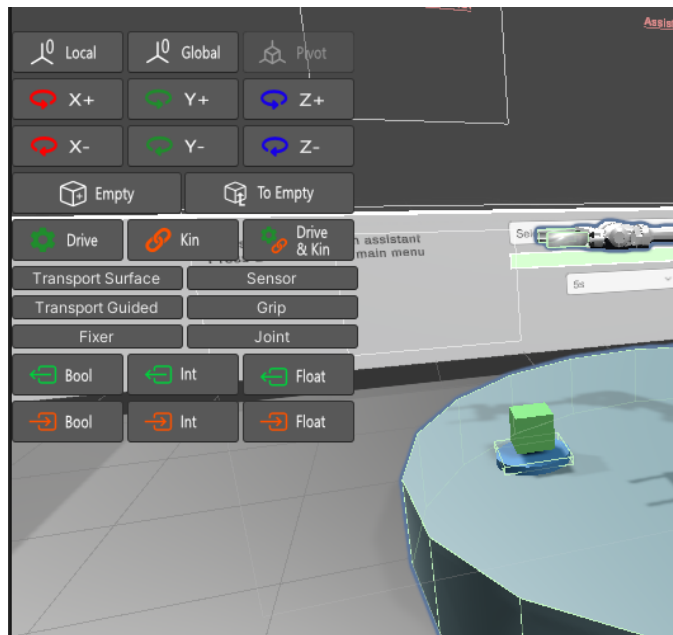


Figure 3.22 - Realvirtual.io's Unity scene editing features (in editor view)

The Realvirtual.io framework offers numerous functionalities, many of which were not utilized in this project. The primary features that were used include drive scripts for controlling the robot's joint movements, a main control logic script, and input/output signals managed by the control logic to operate the drives, and thus, the robot's movements. A sensor script was also applied to the robot's end-effector.

This framework provides a wide range of script types, each identifiable by specific symbols visible in Unity's inspector on the game objects that contain them (Figure 3.23). These scripts serve as the foundation for developing the robot's movements and functionalities within the Unity environment.



Figure 3.23 - The robot and its components/scripts as seen in the Unity hierarchy (in editor view)

The control logic script, referred to as the Controller script in this project, functions similarly to a PLC (Programmable Logic Controller). It defines the robot's logic and plays a central role in controlling the robot's movements and functionalities.

A drive script designates a robot's joint as a movable part, and while the framework offers various types of drive scripts, only simple drive scripts were used in this project, given the basic nature of the joint movements required. In Figure 3.24 we can see one of the joints of the robot, as well its associated drive scripts.

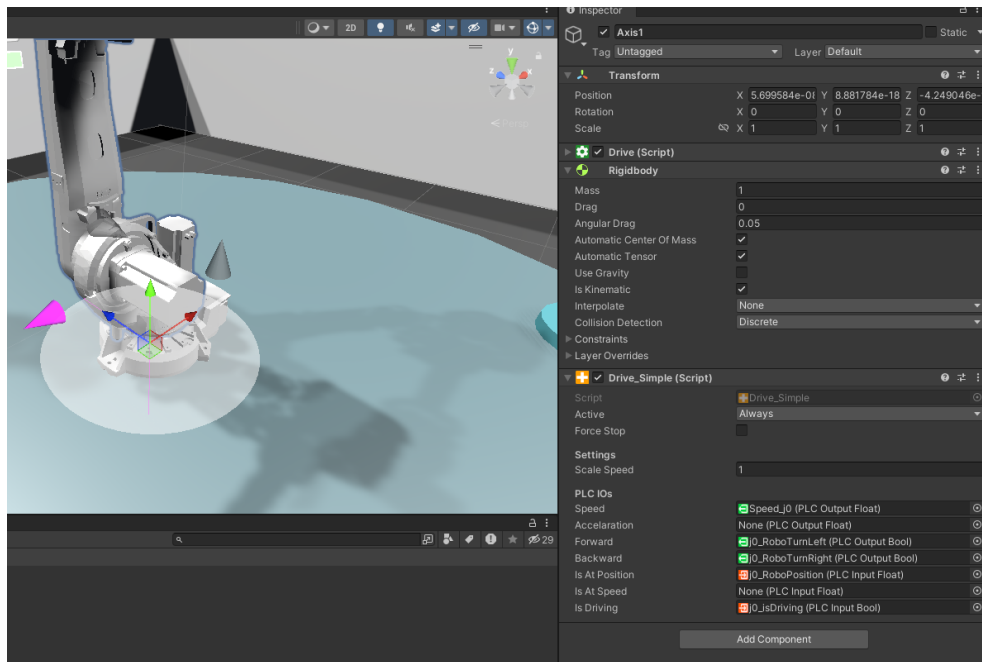


Figure 3.24 - One of the robot's joints, and its corresponding drive components (in editor view)

The sensor script enables the implementation of sensors within the system, allowing other scripts to access the sensor's data (visible in Figure 3.25). In this case, the sensor informs the Controller script on whether there is an object within the gripper's range for picking up or releasing.

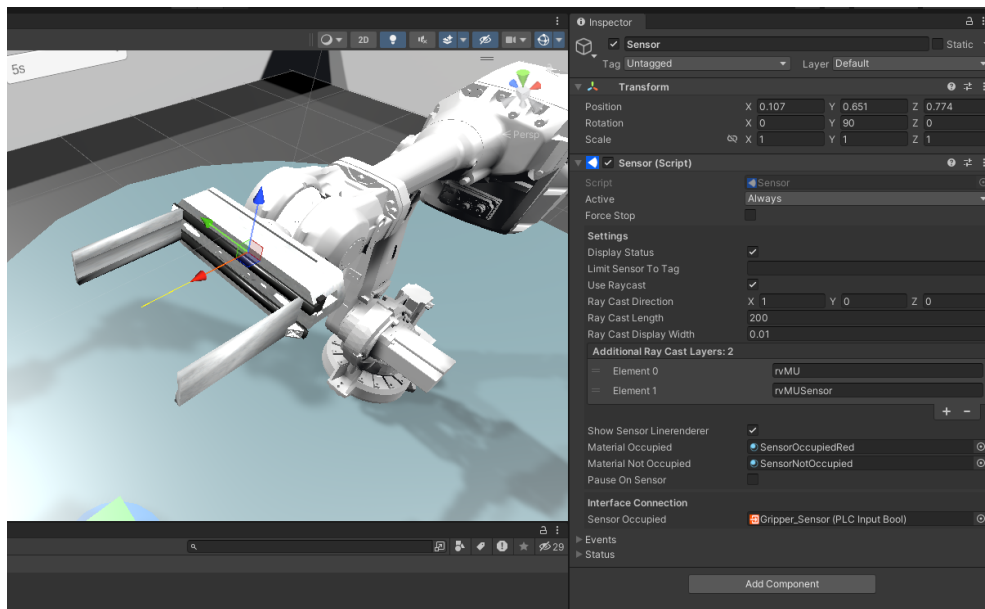


Figure 3.25 - The robot's gripper and its sensor component (in editor view)

Signals are specific to the Realvirtual.io framework and are crucial for establishing connections between drive and sensor components and other scripts. For example, you can assign

an input signal to a joint's drive to obtain its current position or speed, or you could also use it to obtain data from a sensor script. Output signals, on the other hand, are typically used to command actions. In the Controller script, for instance, joint movements such as rotating left or right are controlled by defining values for output signals assigned to specific drives.

Input signals are represented in orange, while output signals are represented in green. As was illustrated in Figure 3.24 and Figure 3.25, the drive script utilizes output signals for controlling movement and speed, and it uses input signals to provide information about its current position and state. Additionally, the sensor script only requires an input signal to convey information to other scripts. A list of these signals can be seen in Figure 3.26.

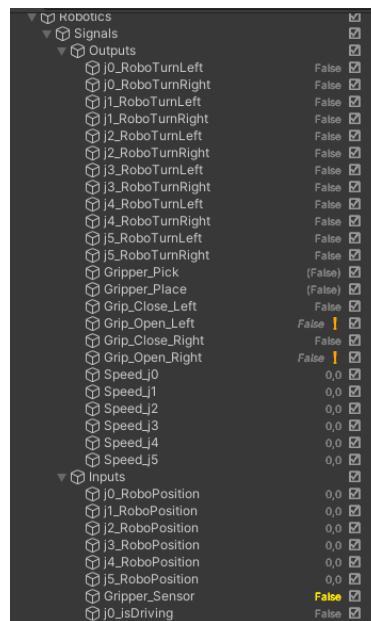


Figure 3.26 - List of Realvirtual.io signals as seen in Unity's hierarchy (in editor view)

In conclusion, by leveraging Realvirtual.io's tools and resources, a realistic and controllable robot was successfully integrated into the system. The controller script acts as the central manager for the robot's movements, containing various functions and methods that use signals to control the robot's actions. These functions are then invoked by the ItemManager script, which handles the hand controller used by the user to operate the robot. This setup effectively bridges the connection between the user's hand controller and the robot, thereby allowing the user to experience controlling the robot in a realistic way.

### 3.3 VR Integration

To understand the integration of VR tools into this project, it's important to first grasp what these tools consist of and how they are meant to be used. VR technologies aim to provide users with a realistic virtual world experience, which can range from non-immersive and semi-immersive to fully immersive.

For practicality, this project focused on delivering a semi-immersive VR experience, utilizing the most common VR tools, a VR headset (or HMD) and hand controllers. While there are many versions of these products on the market, made by various companies, Meta Quest 3 became the primary device used for the development and testing of this project.

The choice of Meta Quest 3 was primarily driven by it being one of the most modern options that were available. Beyond that, the popularity of Meta's VR systems is also advantageous, since many tutorials and resources for VR integration in Unity are based on the Meta platform, thereby easing the development process. A photo of the Meta Quest kit used can be seen in Figure 3.27.



Figure 3.27 - Meta Quest 3 equipment that was used for development and testing

The Meta Quest 3 system consists of a headset and two hand controllers. The headset immerses the user in the virtual world visually, while the hand controllers enable interaction and movement within the environment. An important aspect of the development process was configuring the buttons on the hand controllers to suit both the virtual environment and user

convenience. The controller configuration for guided mode can be seen illustrated in Figure 3.28.

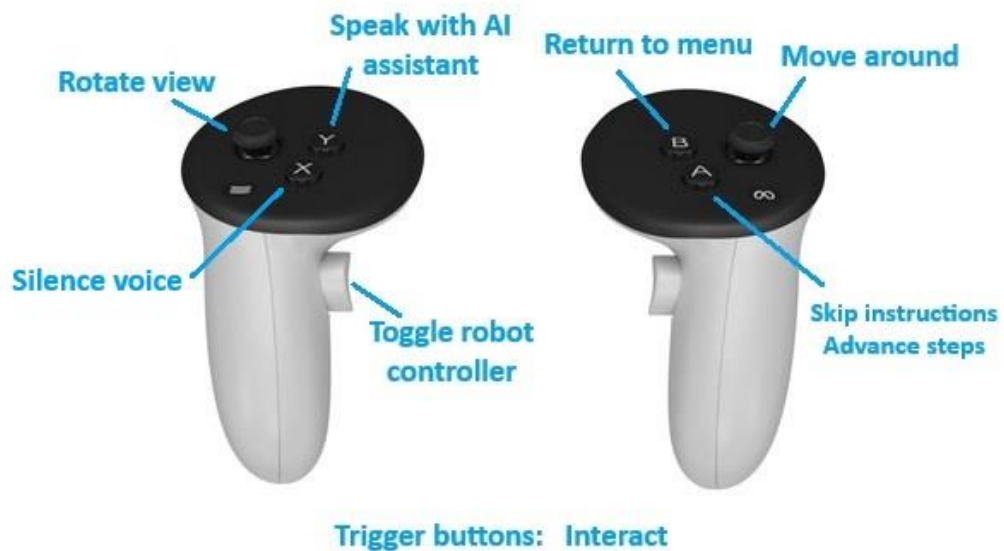


Figure 3.28 - Guided tutorial mode's controller configuration

To operate the Meta Quest 3, Meta Quest Link needed to be installed on the computer. This software allows the headset to connect to the PC, either via a USB cable (Figure 3.29) or wirelessly through Air Link, enabling users to run VR applications installed on the computer.



Figure 3.29 - Meta Quest 3 connected through a USB cable to the computer

The integration of VR tools into the project was based on the official VR Integration Unity tutorial [73], available online. This tutorial is divided into several steps, with both text and video instructions provided. It includes an example Unity project and the necessary extensions to incorporate VR functionalities into Unity. The example project served as a foundation for understanding how VR operates within Unity.

In Figure 3.30 we can see one of the Unity test scenes provided by the tutorial, where two grey spheres with red lines are visible. These spheres represent the user's hand positions, while the red lines indicate the direction they are pointing.

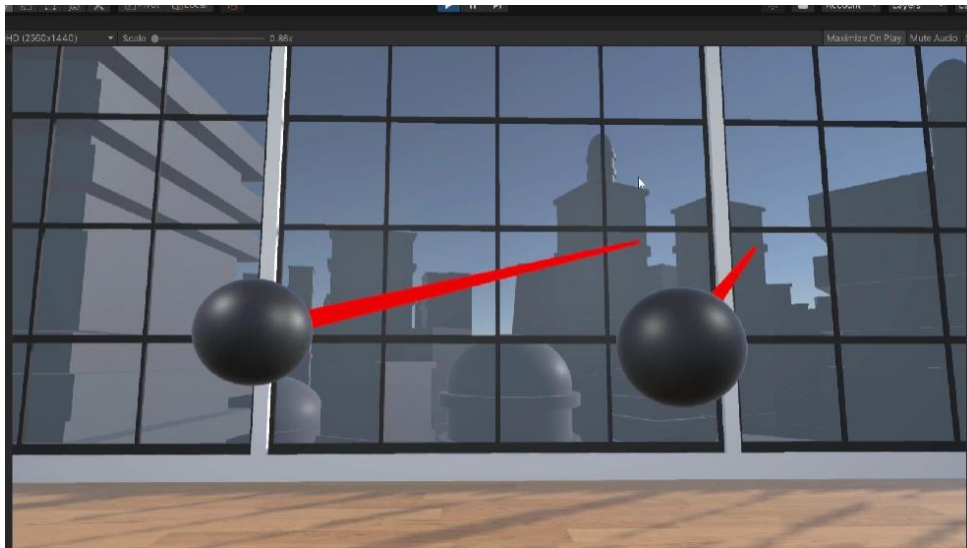


Figure 3.30 - Unity test scene provided by the tutorial

However, for most of the development process, actual VR equipment was unavailable. It wasn't until the final stages of development that real VR hardware could be used. As a result, much of the VR development and testing was conducted using a VR simulator called the XR Device Simulator.

The XR Device Simulator, obtained from the official Unity VR tutorial page, allowed for simulating the controls and systems of real VR equipment using a mouse and keyboard. The key scheme for the simulator is shown in Figure 3.31.

# XR Rig Simulator Keyboard Shortcuts

## Overview

Since there are three devices in VR, all of which can be simulated independently, the process for using the rig simulator is:

- Step 1: Activate the device you would like to control – multiple can be selected at once
- Step 2: Use the control(s) for that device

Use the shortcuts for Step 1 and Step 2 (A and B) below as reference.

## 1. Activate devices



Left controller	Headset	Right Controller
Hold Left Shift or toggle with T	Hold Right-click	Hold Spacebar or toggle with Y

## 2A. Basic controls

<b>Pan</b>	Move mouse
<b>Rotate</b>	Hold middle mouse
<b>Inverse controls</b>	R
<b>Reset transform</b>	V
<b>Snap turn</b>	A [left] or D [right] (with controller active)

## 2B. Input controls

<b>Grip</b>	G
<b>Trigger</b>	Left-click
<b>Primary Button</b>	B
<b>Secondary Button</b>	N

Figure 3.31 - XR Rig Simulator Keyboard Shortcuts [74]

However, it wasn't until access to the Meta Quest 3 was gained that testing the application as a whole became more feasible. Having direct contact with the real equipment allows the developer to have a better feel for what they are developing, for example, it made it a lot easier to assign specific interactions to the buttons on the hand controllers in a way that felt intuitive for the user.

By the end of the development process, the user could move around the virtual scene using the right-hand controller's analogic stick, while the one on the left allowed them to turn. The "Y", "A", and "B" buttons were assigned specific functions within the simulation scenes, and the trigger buttons on the front of the controllers served as the main interaction tools, enabling the user to easily interact with UI menus by pressing them with their index fingers.

Additionally, the simple grey spheres that previously represented the user's hands were replaced with more realistic hand models, complete with a few animations to reflect the user's interactions (Figure 3.32).

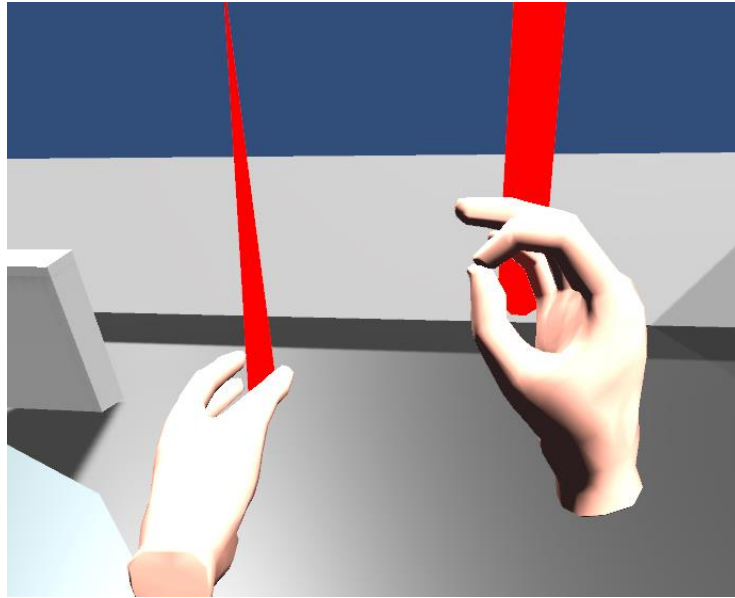


Figure 3.32 - Hand animation

## 3.4 AI Integration

### 3.4.1 Chat Bot Integration

As mentioned in the state-of-the-art chapter, generative AI is one of the most rapidly evolving technologies today. It has given rise to numerous concepts and ideas on how these technologies can optimize economic productivity and improve people's quality of life. The core idea of this dissertation is deeply rooted in this trend, with its goal being to prove the concept of an AI-assisted VR learning environment. This means that one of the key challenges in developing this project will naturally be integrating an AI chatbot into a Unity application that could effectively fulfil the project's main objectives.

Initially, the Chat Bot service that seemed most promising was RASA. RASA operates as a platform that allows the creation of custom Chat Bot models. This option was particularly attractive due to it being free and easily accessible, as well as having a wealth of available information and tutorials. The official RASA website itself offers a very detailed tutorial, outlining the steps for installation and use of the service [75].

Additionally, it was relatively easy to find information on how to integrate RASA services with Unity [76]. The fact that the service is free and that detailed tutorials were available on

official websites, as well as on platforms like YouTube, led to the first tests being carried out with the goal of integrating a RASA Chat Bot into a Unity project.

This goal was achieved after creating a test project, following the instructions from the Unity integration tutorial mentioned earlier. In this simple project, there was a button, and upon it being pressed, a message saying "hi" was sent to the Chat Bot, which responded, as shown in Figure 3.33.

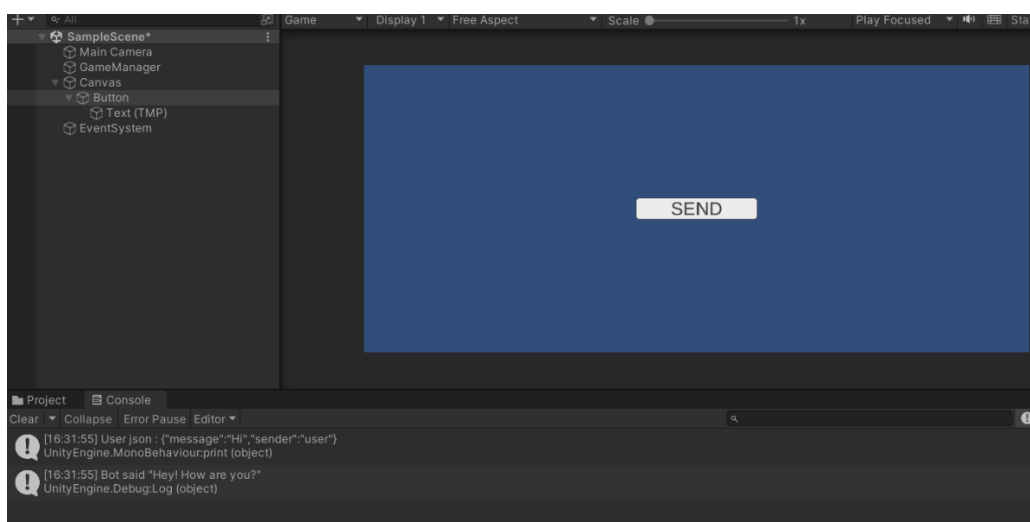


Figure 3.33 - Early communication test with RASA

However, some challenges quickly became apparent when using this service. One of them is related to the fact that RASA Chat Bots require the creation of data and "stories" in a YAML file format to communicate in a more realistic and adaptable manner. This is because RASA Chat Bots are largely built based on custom instructions provided by the user.

The process of developing these files and creating RASA models, in general, is relatively complex and time-consuming, especially if a particularly intelligent Chat Bot is needed. Although the level of customization offered by RASA services can be very useful in certain circumstances, given the time constraints for the development of this project, it was decided to explore other options.

This led to the initial tests for the integration of ChatGPT with the Unity project. Unlike RASA, the use of OpenAI services is not free, so it was first necessary to investigate the associated costs.

The costs of using OpenAI's services are measured in terms of tokens. A token is defined as a common sequence of characters found in a text. This division of text into tokens can be

easily visualized on an OpenAI website called Tokenizer, as shown in the example in Figure 3.34.

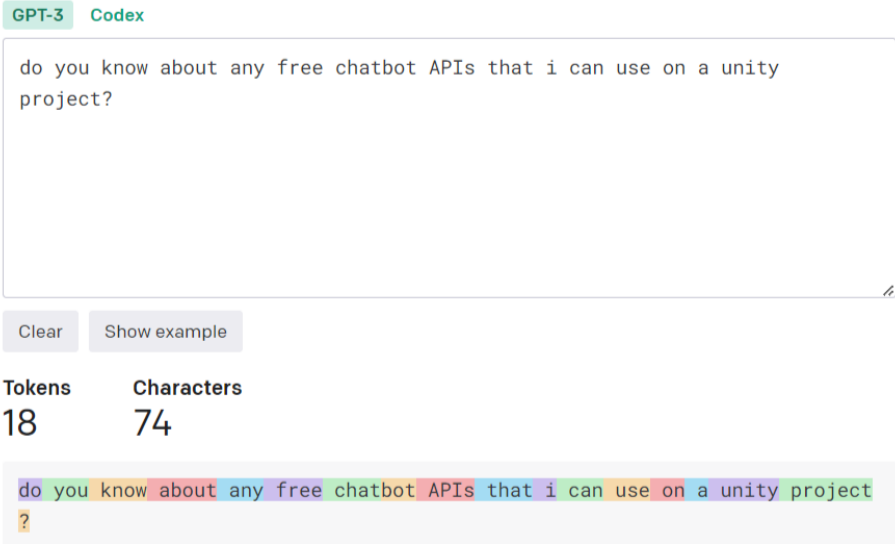


Figure 3.34 - Tokenizer [77]

With an understanding of what a token is, it is easier to comprehend the pricing of OpenAI’s services. It becomes clear that these services are relatively affordable. For example, the cost of using ChatGPT 3.5 (via API) is \$0.001 per thousand tokens.

During the entire development process, only about \$0.24 of the \$6 that were loaded was used. These costs include not only the use of ChatGPT 3.5 services but also Whisper, which has a cost six times higher.

After confirming that OpenAI’s service pricing is affordable, the next step was to integrate them into a Unity project. To do this, a few tutorials were followed, and a pre-made package from GitHub was used as the basis for the integration [78]. This package contains the base API code, which had to be completed with an access key and OpenAI account Information, and an example project that contains a text box where it’s possible to exchange text messages with the Chat Bot. This example project can be seen working in Figure 3.35.

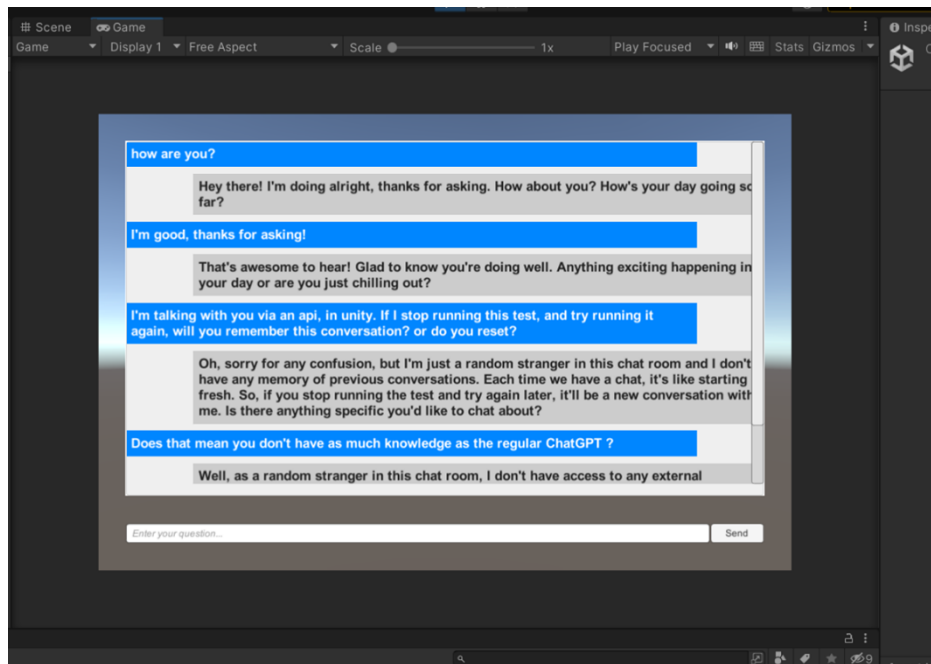


Figure 3.35 - Example project from GitHub package

Looking at the figure, it is clear that the Chat Bot is interacting as if it were a stranger, without access to any external information. This happens because, in the code that manages the Chat Bot's functionality, a prompt message is defined that instructs the Chat Bot about the role it should play. A prompt message is a message that provides instructions to an AI Bot on what to do. In this case, the prompt message was: "Act as a random stranger in a chat room and reply to the questions. Don't break character. Don't ever mention that you are an AI model."

This test immediately highlights a clear advantage that using ChatGPT has over RASA. ChatGPT can be integrated more easily, without requiring any prior setup or training to interact realistically. This test demonstrates that using a simple prompt message, like the one shown earlier, is enough to create a Chat Bot capable of dynamically communicating with the user, following instructions.

For these reasons, ChatGPT was used in subsequent tests and was the chosen Chat Bot for this project.

### 3.4.2 TTS and STT Integration

In a virtual reality environment, it is difficult to use computer keyboards or other conventional writing tools. Therefore, options for TTS (Text-to-Speech) and STT (Speech-to-Text) were explored for communication with the Chat Bot.

TTS refers to the conversion of text to speech on a computer. A TTS system can read text and convey what was written using an artificial voice through the computer's audio speakers. On the other hand, STT refers to the opposite conversion, meaning the transformation of speech into text. Using a microphone, an STT-enabled system can interpret what the user says and convert the speech into text.

Incorporating these features allows the user to interact with the Chat Bot solely through voice, providing a more immersive and practical experience, resembling human communication.

For the implementation of TTS functionalities, various options were analysed, including resources available in the Unity Asset Store and services provided by Google Cloud. After some testing however, AWS TTS services, known as AWS Polly, were chosen.

The main advantage of this service is that AWS offers free usage for one year, unlike other options that either involved immediate costs or shorter free trial periods. Additionally, it was easier to find clear tutorials for this service compared to other available alternatives.

To implement the STT functionalities, OpenAI's services were used, by integrating Whisper into the project. Whisper is a generative AI tool capable of interpreting audio and converting it into written text. Since OpenAI services were already being used through the ChatGPT 3.5 bot, it was simply a matter of reusing the available resources, adapting them to incorporate Whisper into Unity and essentially using the same API. It was also advantageous that the GitHub resources previously obtained already included the necessary tools for this integration. The cost of using Whisper is \$0.006 per minute of audio analysed, which means it's relatively cheap just like GPT 3.5.

### 3.4.3 Chat Bot Preparation

After developing functions that allow seamless integration of AI tools, such as converting detected audio into text, sending a message to GPT-3.5 to retrieve its response, and converting text into speech, one major challenge remains, teaching the AI assistant how to effectively assist the user.

By default, GPT-3.5 functions as a general-purpose assistant with broad knowledge across many domains. However, for this project, the AI must be specifically tailored to assist the user in navigating issues within the simulation. It needs to understand its purpose, the tutorial sequence, how the robot controller operates, and other crucial aspects of the system. Essentially, the Chat Bot must be context-aware and knowledgeable about the specific topics it will be questioned about by the user.

A common method used to train AI bots for specialized tasks is fine-tuning. This technique improves a pre-trained model's performance on a specific task by continuing its training with a smaller, domain-specific dataset. While models like GPT-3.5 are initially trained on vast general datasets to capture a wide array of patterns and knowledge, fine-tuning adjusts the model to excel in a particular application by exposing it to examples directly relevant to that domain.

Fine-tuning is widely used, and when executed properly, it can enable a Chat Bot to handle specific tasks and situations with high efficiency. However, this approach not only requires creating a specialized training dataset in a particular format, but it also incurs additional monetary costs. While preparing a fine-tuning dataset is significantly simpler than training an AI model from scratch, it still requires resources and effort. An example of what a fine-tuning dataset may look like is visible in Figure 3.36

```
1 {"messages": [{"role": "system", "content": "你是一个汉语语法纠错器。"}, {"role": "user", "content": "检测这个句子的语法"}]}
2 {"messages": [{"role": "system", "content": "你是一个汉语语法纠错器。"}, {"role": "user", "content": "检测这个句子的语法"}]}
3 {"messages": [{"role": "system", "content": "你是一个汉语语法纠错器。"}, {"role": "user", "content": "检测这个句子的语法"}]}
4 {"messages": [{"role": "system", "content": "你是一个汉语语法纠错器。"}, {"role": "user", "content": "检测这个句子的语法"}]}
5 {"messages": [{"role": "system", "content": "你是一个汉语语法纠错器。"}, {"role": "user", "content": "检测这个句子的语法"}]}
6 {"messages": [{"role": "system", "content": "你是一个汉语语法纠错器。"}, {"role": "user", "content": "检测这个句子的语法"}]}
7 {"messages": [{"role": "system", "content": "你是一个汉语语法纠错器。"}, {"role": "user", "content": "检测这个句子的语法"}]}
8 {"messages": [{"role": "system", "content": "你是一个汉语语法纠错器。"}, {"role": "user", "content": "检测这个句子的语法"}]}
9 {"messages": [{"role": "system", "content": "你是一个汉语语法纠错器。"}, {"role": "user", "content": "检测这个句子的语法"}]}
10 {"messages": [{"role": "system", "content": "你是一个汉语语法纠错器。"}, {"role": "user", "content": "检测这个句子的语法"}]}
11 {"messages": [{"role": "system", "content": "你是一个汉语语法纠错器。"}, {"role": "user", "content": "检测这个句子的语法"}]}
12 {"messages": [{"role": "system", "content": "你是一个汉语语法纠错器。"}, {"role": "user", "content": "检测这个句子的语法"}]}
13 {"messages": [{"role": "system", "content": "你是一个汉语语法纠错器。"}, {"role": "user", "content": "检测这个句子的语法"}]}
14 {"messages": [{"role": "system", "content": "你是一个汉语语法纠错器。"}, {"role": "user", "content": "检测这个句子的语法"}]}
15 {"messages": [{"role": "system", "content": "你是一个汉语语法纠错器。"}, {"role": "user", "content": "检测这个句子的语法"}]}
16 {"messages": [{"role": "system", "content": "你是一个汉语语法纠错器。"}, {"role": "user", "content": "检测这个句子的语法"}]}
17 {"messages": [{"role": "system", "content": "你是一个汉语语法纠错器。"}, {"role": "user", "content": "检测这个句子的语法"}]}
18 {"messages": [{"role": "system", "content": "你是一个汉语语法纠错器。"}, {"role": "user", "content": "检测这个句子的语法"}]}
19 {"messages": [{"role": "system", "content": "你是一个汉语语法纠错器。"}, {"role": "user", "content": "检测这个句子的语法"}]}
20 {"messages": [{"role": "system", "content": "你是一个汉语语法纠错器。"}, {"role": "user", "content": "检测这个句子的语法"}]}
21 {"messages": [{"role": "system", "content": "你是一个汉语语法纠错器。"}, {"role": "user", "content": "检测这个句子的语法"}]}
22 {"messages": [{"role": "system", "content": "你是一个汉语语法纠错器。"}, {"role": "user", "content": "检测这个句子的语法"}]}
23 {"messages": [{"role": "system", "content": "你是一个汉语语法纠错器。"}, {"role": "user", "content": "检测这个句子的语法"}]}
24 {"messages": [{"role": "system", "content": "你是一个汉语语法纠错器。"}, {"role": "user", "content": "检测这个句子的语法"}]}
25 {"messages": [{"role": "system", "content": "你是一个汉语语法纠错器。"}, {"role": "user", "content": "检测这个句子的语法"}]}
26 {"messages": [{"role": "system", "content": "你是一个汉语语法纠错器。"}, {"role": "user", "content": "检测这个句子的语法"}]}
```

Figure 3.36 - Example of a dataset for fine-tuning [79]

Given the relatively low complexity of the tasks expected from the AI assistant in this proof-of-concept project, a simpler approach was chosen, relying on detailed, well-structured prompting, along with supporting functions. This decision balances functionality with ease of implementation, ensuring the AI assistant is equipped to handle the required tasks without the need for extensive customization or further costs.

To make this system work, a general set of instructions and information is provided to the bot each time the simulation starts. The instructions are as follows (Figure 3.37).

```

7 public class GPT_manager : MonoBehaviour
8 {
9     private OpenAIApi openai = new OpenAIApi();
10    private List<ChatMessage> messages = new List<ChatMessage>();
11    private string prompt = @"
12    You are a virtual assistant for a VR system designed to teach users how to control a manufacturing robot.
13    If the current VR mode is Free, nothing related to tutorial steps matters, because in free mode there
14    - **General Controls:**
15      - **Right Hand VR Controller:**
16        - Walk with the analog stick.
17        - Return to the starting menu with the B button.
18        - Skip tutorial dialog and move to the next step with the A button.
19      - **Left Hand VR Controller:**
20        - Rotate view with the analog stick.
21        - Talk with the AI assistant using the Y button.
22        - Toggle the controller menu with the Grip button.
23
24    - **Guided Mode Instructions:**
25      - **Step 1:** Press all joint movement keys to learn basic robot controls (12 smaller buttons on the
26      - **Step 2:** Change robot speed mode with the fast or slow button.
27      - **Step 3:** Pick up and drop the cube using the pick/drop button. Ensure the cube is in the gripper
28      - **Step 4:** Return the cube to its original base (blue, identified as A) using joint movement and
29      - **Step 5:** Save and delete points using the insert and delete buttons. Saved points are listed on
30      - **Step 6:** Use the move button to move the robot towards a saved point and the change button to u
31      - **Step 7:** Save a sequence of points that delineate the robot moving the cube from base A (blue)
32      - **Step 8:** Move the robot along the saved sequence using the forward (FWD) and backward (BWD) but
33      - **Step 9:** Return the cube to its starting position (Base A) and press the test start button to
34
35    When instructions are shown, the user can't move or rotate their view.
36
37    - **AI Assistant Interaction:**
38      - The AI assistant can only hear the user for a specific period at a time, represented as a fill bar
39      - The user can change the microphone through the wall menu (there's a selection button).
40
41    ";

```

Figure 3.37 - First prompt defined for the system

The prompt begins with a clear description of the Chat Bot's task and is divided into sections, each covering a different topic. It explains in detail each step of the tutorial sequence, including what the user must do to proceed through each step. The prompt also includes useful information and tips on common questions users may ask, which aren't covered in detail in the pre-written tutorial instructions.

Perfecting the prompt for the bot to understand and respond effectively within its designated context requires considerable trial and error. While the prompt could likely be further improved, for the purpose of this proof of concept, it was deemed sufficient.

Beyond the initial prompt, it's essential to have a method for updating the AI assistant on the current state of the simulation. To achieve this, two supporting functions were added (Figure 3.38). These functions keep the AI informed about two particularly important details: the current step in the tutorial sequence and whether the simulation is in guided or free mode.

```

2 referências
public void SetMode(string mode)
{
    vrState.Mode = mode;
    UpdateAIWithSystemState("Mode", vrState.Mode);
}

9 referências
public void SetTutorialStep(int step)
{
    vrState.TutorialStep = step;
    UpdateAIWithSystemState("Tutorial step", vrState.TutorialStep.ToString());
}

2 referências
private void UpdateAIWithSystemState(string key, string value)
{
    var updateMessage = new ChatMessage()
    {
        Role = "system",
        Content = $"The {key} has changed to: {value}"
    };

    messages.Add(updateMessage);
}

```

Figure 3.38 - Auxiliar functions designed to update the AI bot on what's happening in scene

With the initial prompt and supporting functions in place, the AI assistant is aware of its purpose, informed enough to assist the user when needed, and updated on what is happening within the simulation.

While the AI assistant isn't perfect, due to its limited memory capacity (which can be measured in tokens), it may occasionally forget something the user has said or lose track of the simulation's state. However, it performs well enough to demonstrate the potential of systems involving close human-AI interaction, something that will be looked into in more detail in section 4.

## TESTS AND RESULTS

By the end of the development phase, the system integrated a digital replica of a robot, and generative AI and VR technologies. As we've seen in the architecture chapters, these tools work together, and the unity system can be divided into several blocks. The system proved robust enough to be compiled and run independently outside the Unity development environment, as a standalone application.

This brings us to the next major chapter of this dissertation, where detailed testing will be analysed, and conclusions will be drawn about the project's viability as a proof of concept.

Before beginning the in-depth analysis of the testing process, however, the chapter will start with a general presentation of the user's experience within the finalized system.

### 4.1 Presentation of the Developed System

When opening the application, the first interface the user encounters is the main menu (Figure 4.1), which serves as a central hub for navigating the application's features. This is the initial environment the user enters upon launching the application, where the first screen presents two options, Guided Robot Tutorial and Free Robot Tutorial.

In this scene, player movement is restricted, but the user can see their virtual hands and interact with the menu by using the VR controllers, by selecting options with the trigger buttons on the VR controllers.

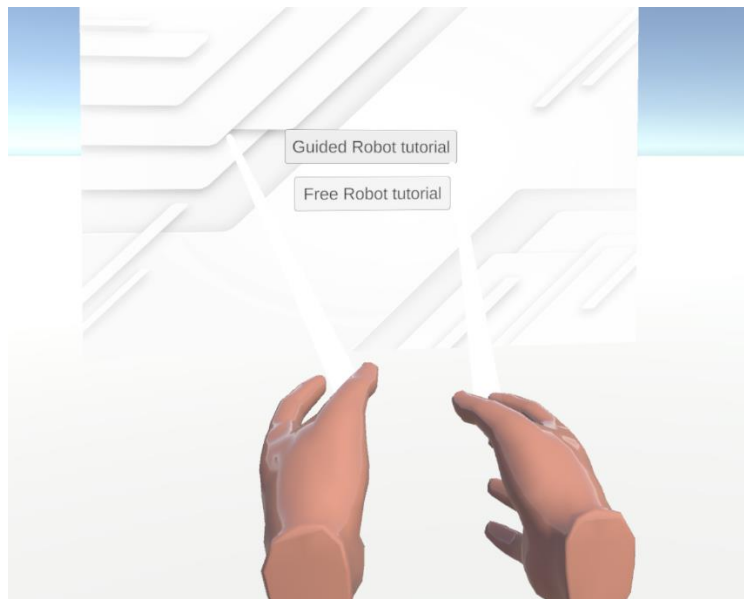


Figure 4.1 - First screen of the main menu

When the user selects one of the tutorial options, the menu updates to display a brief description of the chosen tutorial, an image showing the general layout of the VR controllers' buttons for the upcoming simulation, and two additional clickable options, one to proceed to the simulation scene, and another to return to the previous tutorial selection screen (Figure 4.2 and Figure 4.3).

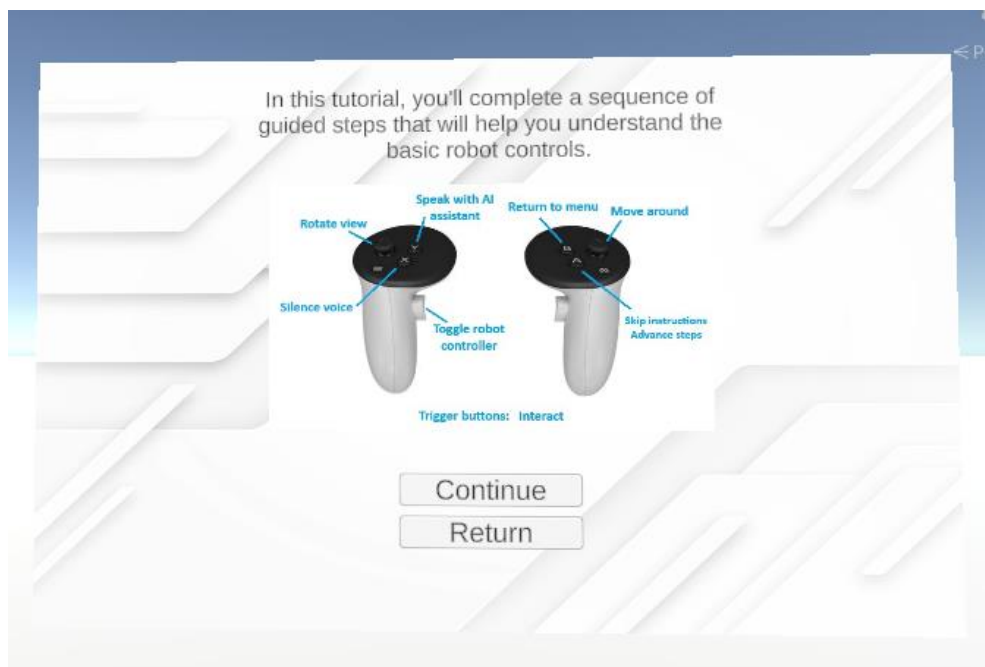


Figure 4.2 - Description of the guided tutorial (in editor view)



Figure 4.3 - Description of the free tutorial (in editor view)

This way, the Main Menu Unity scene functions as a central hub, allowing the user to access the desired simulation training scenes while also providing essential information about each option.

As the user enters either one of the simulation scenes, they notice that they share many similarities. Each features a 3D environment that the user can interact with, and this environment is identical across both scenes.

The 3D setting consists of a room with a central platform where the robot is located, as it's seen in Figure 4.4.

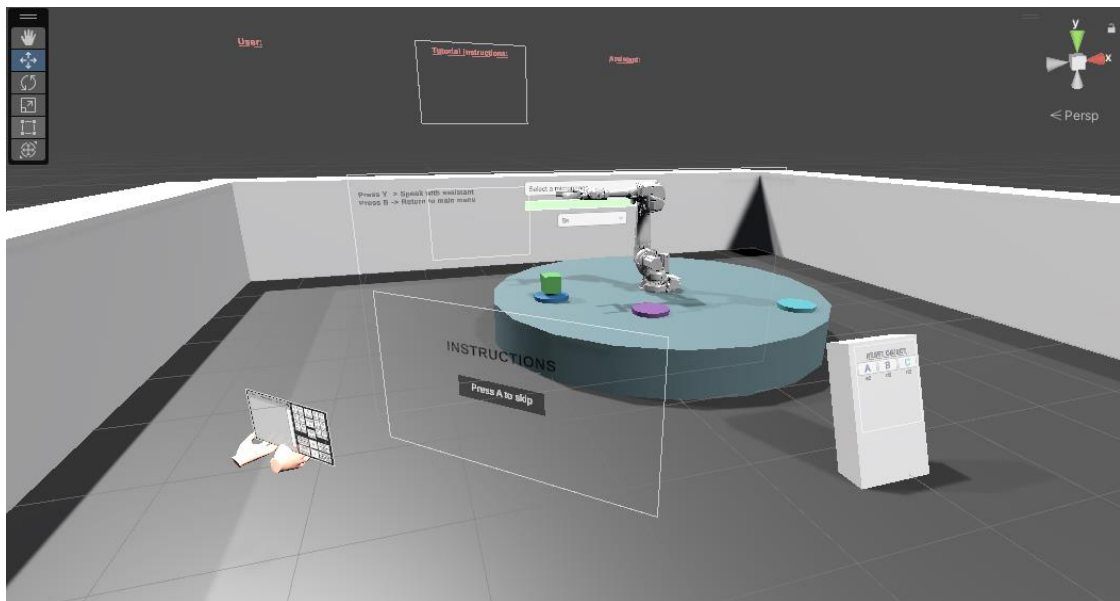


Figure 4.4 - Tutorial room (in editor view)

In one of the walls of this room, useful tips are displayed for the user (Figure 4.5), including instructions indicating that by pressing the "Y" button on their controller, they can initiate communication with the virtual assistant, and by pressing the "B" button, they can return to the main menu scene.

To communicate with the virtual assistant, the user presses the "Y" button, which starts a timed countdown during which the user's verbal input is recorded and processed by the system's AI tools. The wall also features a microphone selector, a time bar that represents the duration of the recording window, and a time window selector. Useful tools to help the user decide how they wish to interact with the AI assistant.

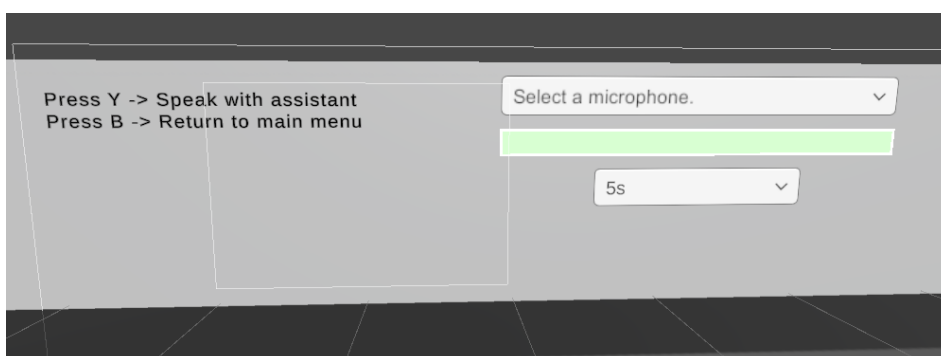


Figure 4.5 - Wall with tips and AI related functionalities (in editor view)

Above the wall, there are text boxes where the user can read written transcripts of their conversations with the Chat Bot (Figure 4.6). There's a text box for the user's input, for the guided sequence's instructions and for the Assistant's responses.

This feature is particularly helpful if the user forgets what the AI assistant has said or needs a reminder of the current task in the guided training simulation. It's also useful in situations in which the user misspeaks or has a bad microphone, as it allows the user to be aware of when the system misunderstands their inputs.

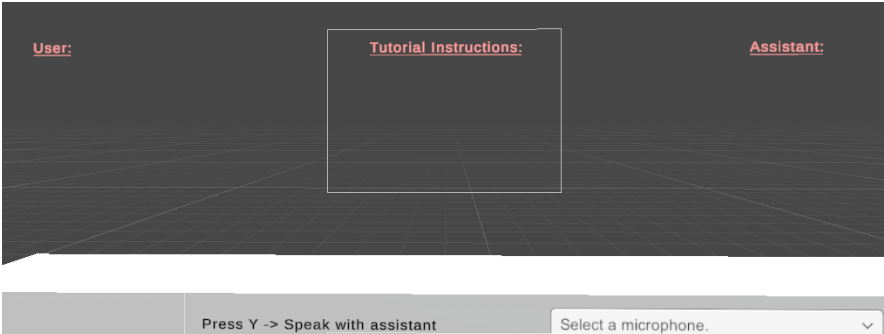


Figure 4.6 - Auxiliar text boxes (in editor view)

Additionally, the user will notice a small screen next to the robot's platform that allows them to adjust the position of a cube on the platform (Figure 4.7). This cube is designed to be picked up by the robot during training and moved between small bases on the platform, so the ability to reposition the cube using this screen is a practical feature.

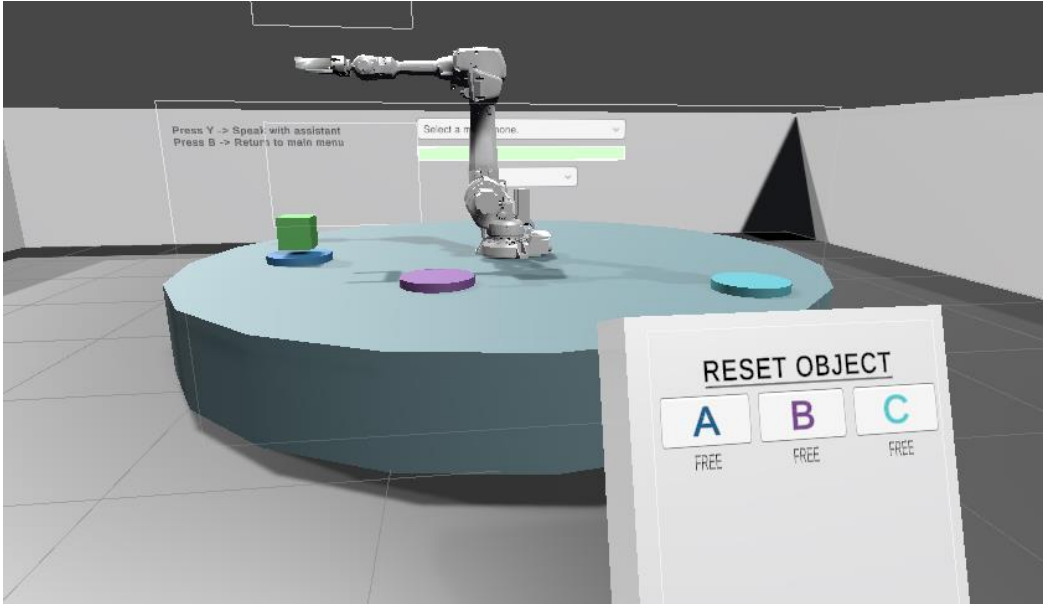


Figure 4.7 - Cube bases and the cube repositioning screen (in editor view)

Both tutorial modes share the same movement and interaction user controls, with only minor differences between them. The primary distinction between the two modes lies in their

approach to guidance. With one mode automatically dictating a sequence of guided tasks that must be completed in order, and the other allowing users to freely train without any specific guidance.

In both tutorial modes the user can control the robot's digital replica through a set of realistic controls assigned to their left hand (seen in Figure 4.8), with the only difference between the modes being that in guided mode, some of the robot's controls are locked until the user has advanced to a certain level in the guided tutorial sequence.

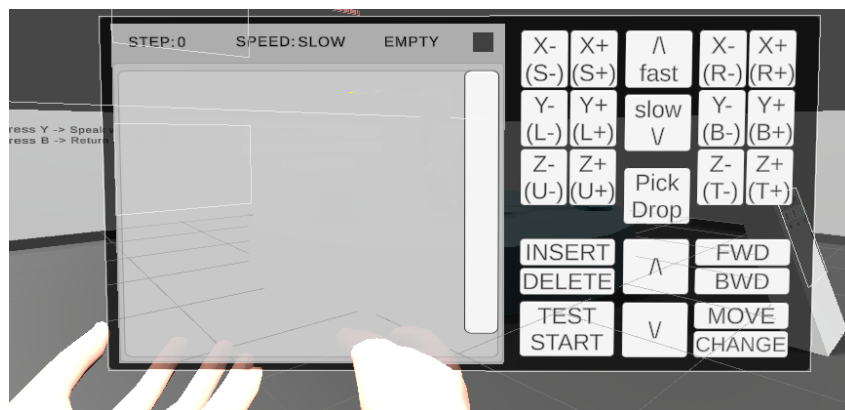


Figure 4.8 - Robot Controls (in editor view)

In guided mode, the user must complete a sequence of several steps (as was explained previously in section 3.1.2.1), with each beginning with a pre-designed instruction delivered to the user through both voice and the instructional text box mentioned earlier. During this explanation, the scene is paused (Figure 4.9), preventing the user from moving. The user is allowed to skip these instructions by pressing the "A" button on the right VR controller. The user can also silence the voice whenever they feel like by pressing "X" on the left VR controller.

In both modes, the user can communicate with the AI assistant by pressing the "Y" button. In the guided tutorial, communication with the AI assistant is disabled while an instruction is being delivered, but outside of that, the user can interact with the assistant at any time, as the assistant is even aware of what the current task is that the user is attempting to complete.

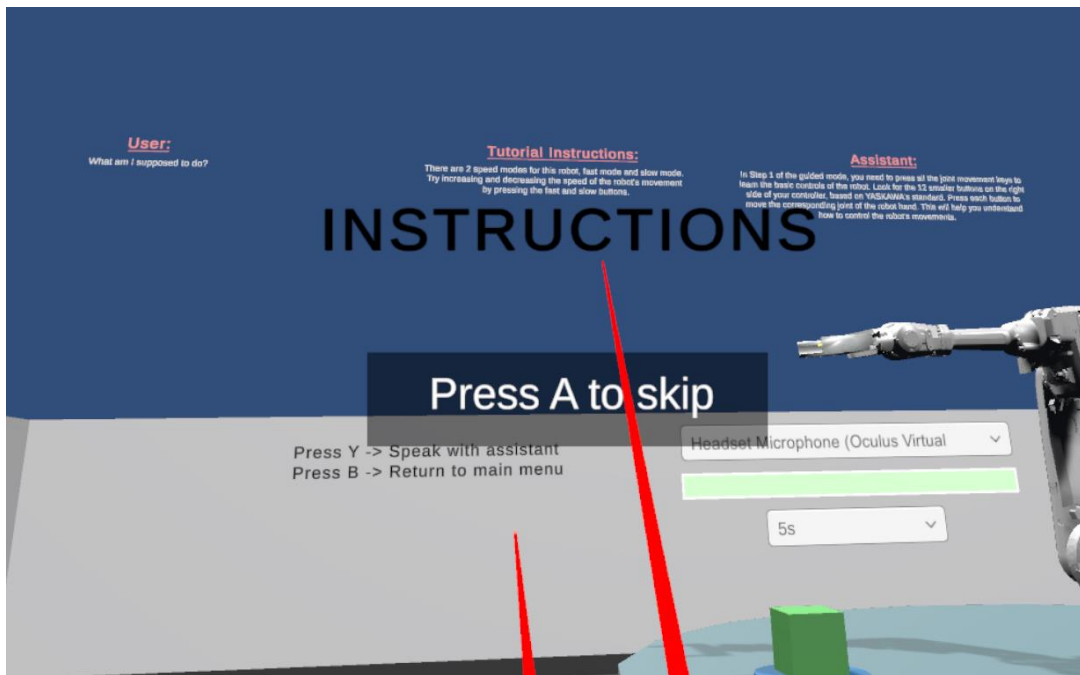


Figure 4.9 - Instructions pause screen, in guided tutorial mode

## 4.2 Testing Phase

Having analysed the general user experience inside the system, the next step is to study the different aspects of the system that were tested and how they were tested, covering aspects like the feel of the VR controls, the UI, and the robot controls. The performance of the AI will also be examined through various tests, focusing on how different prompts affect its behaviour. By the end of this subchapter, we will review the testing done on the guided tutorial sequence as a whole.

### 4.2.1 General Tests

#### 4.2.1.1 Initial UI

Regarding the UI aspect of the system, particularly the main menu scene that users first encounter upon entering the application, feedback after multiple uses and from other users suggests that it is indeed simple and easy to navigate. Since the main menu offers only two options, it is not surprising that most users find it straightforward to understand.

However, after allowing individuals who were not directly involved in the project's development to try it out, certain issues with this simplicity became apparent. The primary issue identified was that the screen presented after the user selects a simulation mode does not adequately prepare them for the upcoming simulation.

As mentioned earlier, the main menu scene displays a screen with a general description of the selected simulation mode, along with an image illustrating the VR controls assigned to the user. The problem observed was that users often skimmed over this control image and quickly proceeded to the simulation. As a result, many users, especially those unfamiliar with VR, tended to forget some of the control instructions during the simulation.

In such cases, the only way for users to view the complete control scheme again was to interrupt the simulation and return to the main menu scene, which proved inconvenient.

This issue, however, is easily fixable and does not pose a critical problem for the overall project. Additionally, since the controls for interacting with the AI assistant are displayed within the simulation scene, the user can easily ask the assistant any questions related to the VR controls during the simulation. The AI assistant is generally well-informed about these controls, making it a useful resource for users in these situations.

#### **4.2.1.2 VR Controls**

After several sessions in the VR environment, it became possible to confirm the success of certain design choices, while also identifying areas that could be improved.

Replacing the original dark spheres with actual hand models significantly enhanced the system's level of immersion. Additionally, adding hand animations for certain interactions further contributed to this immersive experience, even if these animations were only implemented for a few specific actions and do not serve a functional purpose beyond enhancing immersion. An example of a user interaction, triggering an animation can be seen in Figure 4.10.

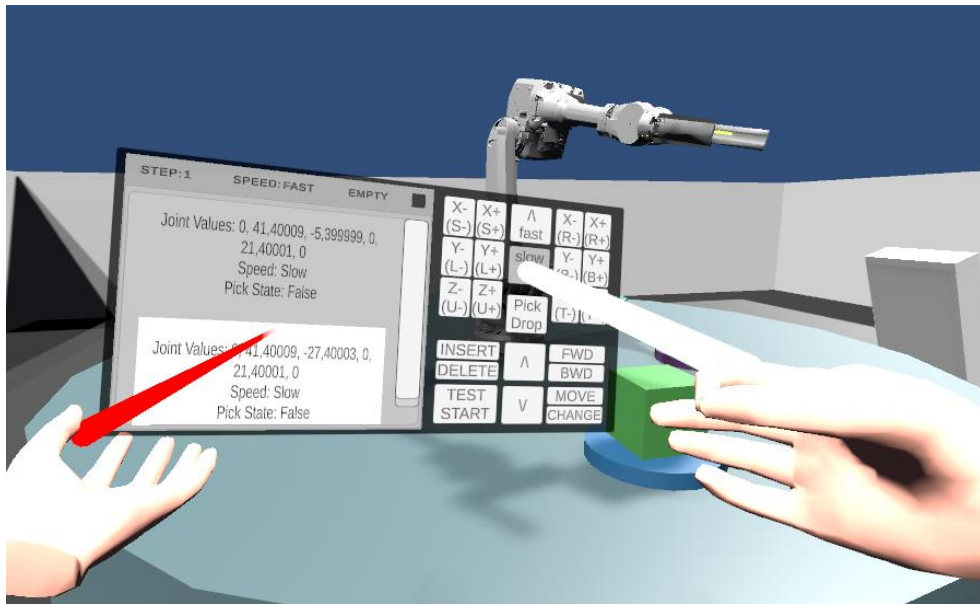


Figure 4.10 - VR hands interacting with controller

Another successful decision was to keep using the original red lines as pointers for interacting with the UI systems in the scene. This approach proved effective because it allows users to interact with UI elements that are out of reach, enhancing the practicality of the experience.

For instance, if the user is controlling the robot and wants to stay near it to observe it up close, they don't need to walk over to the wall to adjust the AI assistant's UI recording time duration settings. Instead, they can simply point at the UI with their hand from far away, making the process more convenient, as is exemplified in Figure 4.11.



Figure 4.11 - Example of a user interacting with the UI from far away

However, relying solely on the VR hand interaction lines also had a few downsides. One issue is that it becomes harder to interact with UI screens if the VR hand is too close to them. Some testers initially made the mistake of trying to interact with the handheld controller while

their VR hands were too near the screen. Fortunately, however, after some trial and error, it becomes easy to understand the optimal distance for interacting with these UI elements.

Another downside is the inconsistency of the interaction lines, particularly when using the cube-moving screen. When the user points its interaction line to one of the cube base options, the selected cube often isn't the one the user is directly pointing at, it feels as though the system detects the interaction slightly to the right of where the pointer is actually aiming (as is noticeable from Figure 4.12). While it's not difficult to adapt to this after a few attempts, it's definitely an area that could benefit from improvement.

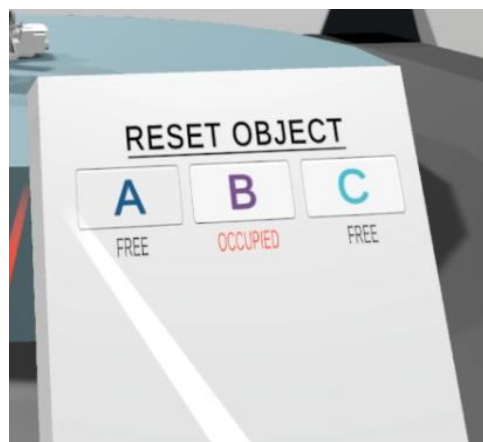


Figure 4.12 - Example of an inaccurate detection of the pointer by the cube-moving screen

Aside from the few issues mentioned, the overall VR experience feels robust and meets the key expectations of the project, and the controls assigned to the VR controllers are practical and intuitive, making them easy to use.

#### 4.2.1.3 Robot Controls

Regarding the robot control experience, after testing the system multiple times with both, users familiarized and unfamiliarized with the system, the overall feedback is positive.

The handheld controller, positioned over the left VR hand, accurately simulates the feeling of holding a real handheld controller. The buttons and functionalities of the controller work accurately and with satisfying precision, achieving the goal of imitating a simplified robot controller. The robot model also appears realistic and responds accurately to the user's commands.

Several modifications were made following initial testing. For instance, the handheld controller initially appeared automatically when the user rotated their wrist. This feature was intended to make the experience smoother, removing the need for button presses. However, it

was eventually removed as it proved more practical to toggle the handheld controller using a button (the left grip button). Otherwise, users had to maintain a specific hand position for the controller to remain visible, which gets tiring over time.

However, the robot control simulation is not without its flaws, primarily due to the lack of realism in how the digital replica interacts with the virtual environment. For example, the robot lacks movement limits, allowing users to rotate its joints in ways that would be impossible in reality. The digital replica also lacks collision detection, meaning the robot can pass through physical obstacles (Figure 4.13). Additionally, the robot's gripper doesn't realistically close when grabbing objects and lacks collision physics. The opening and closing actions of the gripper aren't animated.

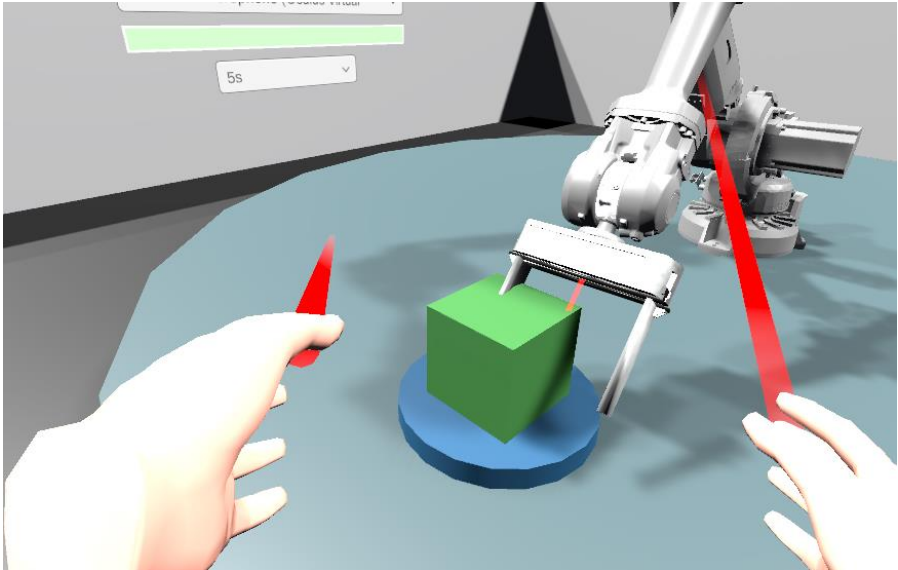


Figure 4.13 - Gripper going through the cube

While these issues could be addressed, it was decided not to invest significant time in fixing them, given that they are not crucial to the proof of concept and due to time constraints.

Despite these limitations, the robot model fulfils the core requirements of the tutorial. It can still be used to effectively teach and train users in essential robot control skills. The sequence shown in Figure 4.14 illustrates how despite these flaws, the system still allows the user to learn and perform tasks like, for example, moving the cube from base to base.

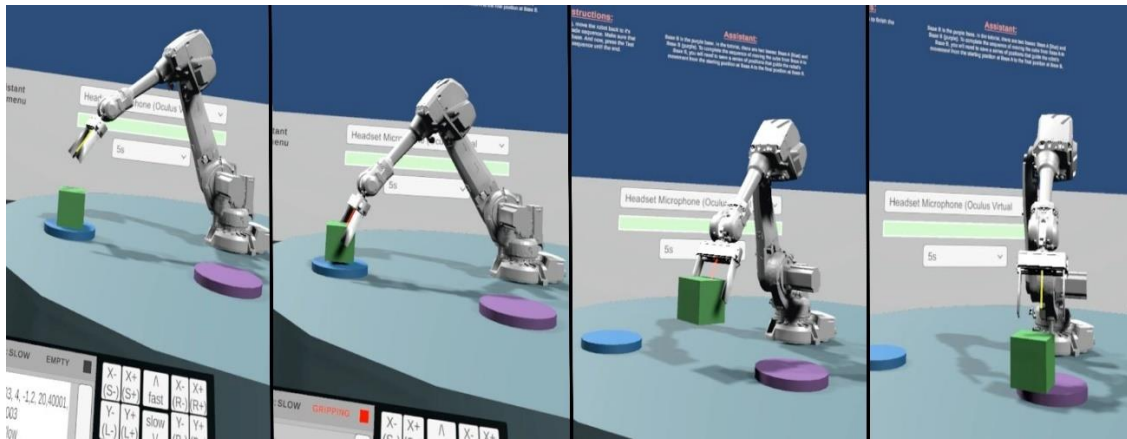


Figure 4.14 - Representation of a successful robot movement sequence

#### 4.2.1.4 AI communication

Up until now, the aspects of the system reviewed in this chapter have been evaluated solely based on general user experience. However, when it comes to the AI communication aspect, a significant amount of trial and error was required to optimize the AI assistant's behaviour to meet the project's requirements.

As mentioned earlier, preparing the AI assistant for its role in the simulation involves providing it with a detailed prompt outlining its tasks, as well as specified functions used to alert it of events occurring during the simulation. While this process is straightforward, verifying the effectiveness of the prompt and functions required extensive testing, involving multiple conversations with the AI assistant.

The initial prompt used during the first tests can be seen in Figure 3.37, in chapter 3.4.3. This prompt contains essential information about the simulation modes, each guided tutorial step, certain VR controls, and general details that could be helpful to the user.

This prompt provided the AI assistant with enough context to understand the differences between the two simulation modes and to recognize each step and its progression.

Consequently, the two auxiliary functions, `SetMode` and `UpdateAIWithSystemState` (also discussed in chapter 3.4.3), successfully updated the AI assistant on the current simulation mode and the progress within the guided simulation. However, despite these early successes, further testing revealed some issues.

One recurring problem was the AI assistant's tendency to respond with bullet point lists (Figure 4.15). Since ChatGPT was primarily designed for text-based communication, during testing, the AI assistant frequently used bullet point lists in its responses. While these lists are

practical for reading, they are less suited for verbal communication, which is the primary mode of interaction in this system.

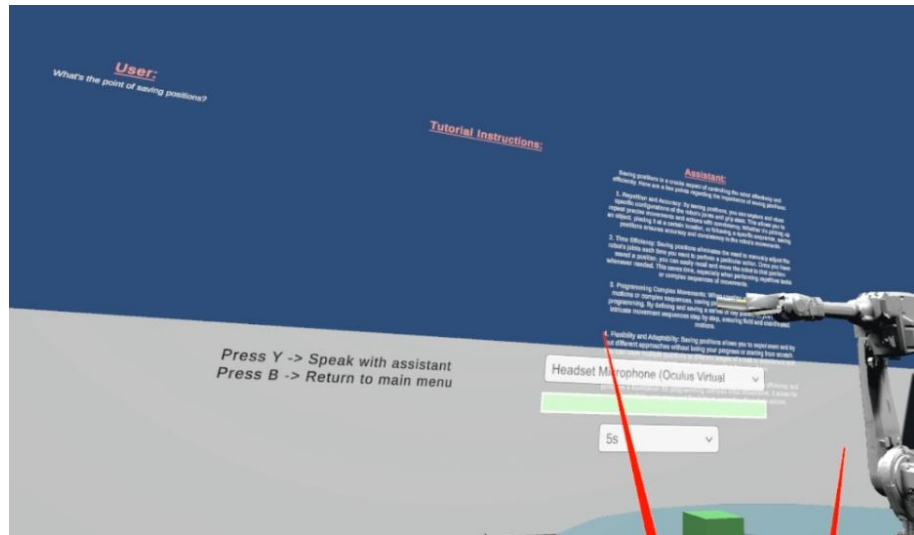


Figure 4.15 - Example of a bullet point list response by the AI assistant

The AI assistant also had a tendency to give overly long responses (Figure 4.16). ChatGPT is inherently designed to provide detailed and intricate answers, often addressing multiple points and offering additional context by bringing in external information. However, in verbal communication, this tendency can become problematic when the user is simply looking for a quick, straightforward answer. Additionally, when the responses are too lengthy, it's not uncommon for them to exceed the space limit of the auxiliary text boxes in the virtual environment.

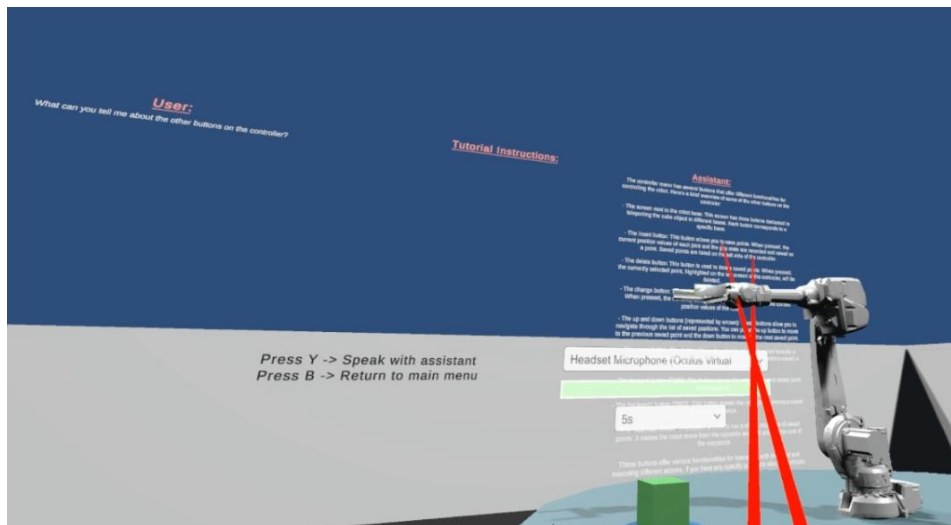


Figure 4.16 - Example of a very long response by the AI assistant

Another issue was that whenever the user asked the AI assistant technical questions, such as how to perform a task with the robot or something related to the VR controls, the AI would often respond with statements like, "In free mode, (...)" even when the question and answer applied equally to both simulation modes. Similarly, it would begin with, "In step (...)," even if the user was in free mode, where there are no steps involved.

Here is an example of one of the initial conversations with the AI assistant during testing in free mode:

- **User:** "How do I move the robot around?"
- **Assistant:** "In Free mode, you can move the robot around using the right VR controller's analog stick. Press the stick in the directions you want to move, and the robot will move accordingly."
- **User:** "Isn't the right stick for moving me around and not robot?"
- **Assistant:** "Apologies for the confusion. You are correct. In Free mode, the right-hand VR controller's analog stick is used to move your character/avatar within the virtual environment, not the robot. The robot's movement is controlled through the tutorial steps in Guided mode."
- **User:** "How do I control the robot's joints?"

- **Assistant:** "In Guided mode, to control the robot's joints, you will need to use the joint movement keys. On the right side of the controller, there are 12 smaller buttons that correspond to the joint movement controls based on Yaskawa's standard. Pressing these buttons will move the robot's joints accordingly. Experiment with pressing different combinations of these buttons to control the robot's movements."

From this example, it's clear that the AI assistant recognizes it's in Free mode, as it mentions this in its first response. We can see that it realizes its mistake in the initial response, correctly identifying that the right analog stick is used to move the player character, and it demonstrates awareness of the twelve joint movement keys designed to mimic the Yaskawa controllers.

However, the AI makes several errors. It responds incorrectly to the first question, possibly due to unclear prompt instructions or its own limitations. It repeatedly refers to the simulation mode as if the functionalities it discusses were specific to that mode. This shows that the AI interpreted the prompt as meaning that the robot controls and features outlined in the guided tutorial steps, apply only to the guided tutorial, which is not the case. These functionalities are available in Free mode as well.

Another minor mistake was assuming that the 12 movement keys are used in combinations, whereas they are intended to be used one at a time. This likely happened because the prompt did not clearly specify how these keys are used.

Considering these issues, the prompt was revised and tested several more times. By the end of this process, the final version of the prompt ended up looking like what's seen in Figure 4.17.

```

private List<Chatmessage> messages = new List<Chatmessage>();
private string prompt = @"
You are a virtual assistant for a VR system designed to teach users how to control a manufacturing robot hand. The system has two modes: Free and Guided.
If the current VR mode is Free, nothing related to tutorial steps matters, because in free mode there are no tutorial steps. Only reply to tutorial steps when in free mode, you may answer questions based on information from the guided tutorial's steps, but never mentioned from which tutorial step.
In guided mode, it's common for some buttons to be disabled on certain steps, this means those buttons aren't useful for completing the task.
Don't mention tutorial steps when you are in free mode.
Don't mention guided when you are in free mode, only if the user asks if there is a way for them to go through a sequence of instructions or steps.
In free mode, the user can do everything they can do in guided mode, except receive detailed instructions.
Try to answer questions in a simple manner, and human like way, don't give very long answers, avoid explaining things point by point.

- **General Controls:**
- **Right Hand VR Controller:**
  - Walk with the analog stick.
  - Return to the starting menu with the B button.
  - Skip tutorial dialog and move to the next step with the A button.
- **Left Hand VR Controller:**
  - Rotate view with the analog stick.
  - Talk with the AI assistant using the Y button.
  - Toggle the controller menu with the Grip button.
  - Silence the AI voice by pressing X.

- **Robot Controls Information:**
- The controller menu is from where the user can control the robot, the controller menu has a lot of buttons that the user can use.
- The user can move the robot and its joints by pressing the twelve movement keys.
- The user can change the speed of the robot's movements by pressing the fast button or the slow button, there are only these 2 speed modes.
- The user can pick up objects with the pick/drop button, The object must be in the gripper sensor's detection range (yellow line turns red when the object is picked up).
- The user can move the object from base to base by using the screen next to the robot base, which has 3 buttons dedicated to teleporting the object.
- The user can save points by pressing the insert button, the saved buttons appear on the left screen of the controller.
- The user can select saved points by pressing the Up and Down buttons on the controller, represented by arrows.
- The up and down buttons are used to navigate along the list of saved positions.
- The user can delete saved points by pressing the delete button.
- The user can change the values of a saved point by pressing the change button.
- The user can make the robot move towards a saved position by pressing the move button.
- The user can make the robot move to the next saved position by pressing the forward button (FWD button).
- The user can make the robot move to the previous saved position by pressing the backward button (BMD button).
- Actions like position deletion, change and move, are made for the position that is currently selected, which is highlighted in the left screen.
- The user can make the robot go through a whole sequence of positions, from the one selected to the final one, by pressing the Test/Start button.

- **Guided Mode Steps:**
- **Step 1:** Press all joint movement keys to learn basic robot controls (12 smaller buttons on the right side, based on YASKAWA's standard).
- **Step 2:** Change robot speed mode with the fast or slow button.
- **Step 3:** Pick up and drop the cube using the pick/drop button. Ensure the cube is in the gripper sensor's detection range (yellow line turns red when the object is picked up).
- **Step 4:** Return the cube to its original base (blue, identified as A) using joint movement and pick/drop buttons, or the screen next to the robot base.
- **Step 5:** Save and delete points using the insert and delete buttons. Saved points are listed on the left side of the controller and can be navigated using the up and down buttons.
- **Step 6:** Use the move button to move the robot towards a saved point and the change button to update the values of a saved point. The move button is used to move the robot towards a saved point.
- **Step 7:** Save a sequence of points that delineate the robot moving the cube from base A (blue) to base B (purple). At least three saved points are required.
- **Step 8:** Move the robot along the saved sequence using the forward (FWD) and backward (BMD) buttons. These buttons help visualize the sequence.
- **Step 9:** Return the cube to its starting position (Base A) and press the test start button to run the whole sequence. The test start button is used to run the whole sequence.

When instructions are shown, the user can't move or rotate their view.

- **In Real Life information**
- In real life, saving robot positions is very common, and it's crucial for programming its movements.
- It's common practice to always have saved a position that is central and safe for the robot, that works as the base point for the robot.

```

Figure 4.17 - Final improved prompt

In this new prompt, significant changes were made to address the previous issues. For example, to prevent the AI assistant from giving overly long, bullet-pointed responses, the line "Try to answer questions in a simple manner, and human like way, don't give very long answers, avoid explaining things point by point" was added. Other instructions like "Don't mention tutorial steps when you are in free mode." were included to stop the AI from referencing steps and tutorial modes whenever it answers a technical question.

Additionally, a new section called "Robot Controls Information" was added, separate from the tutorial steps list. This was necessary because the AI assistant had previously been confused about when robot controls should be applied. Since the robot control information was originally stored in the guided mode steps section, the AI mistakenly assumed these controls were specific to guided mode. With the new separate section containing all the robot controls, unrelated to any specific mode, it's now much less likely that the AI will associate them with a particular simulation mode.

Here, we see an example of a similar conversation to the previous one, but this time using the improved prompt.

- **User:** "How do I move the robot around?"
  
- **Assistant:** "You can move the robot and its joints by pressing the movement keys on the controller. They allow you to control the robot's position and orientation."
  
- **User:** "What is the current tutorial mode?"
  
- **Assistant:** "The current tutorial mode is Free."
  
- **User:** "What can I do in free tutorial mode?"
  
- **Assistant:** "In free tutorial mode, you have the freedom to explore and interact with the VR system without following specific tutorial steps. You can still use the robot controls, move around, pick up objects, and experiment with different features of the system."

From these responses, significant improvements are immediately noticeable. The conversation starts with the same question as before, but this time the AI assistance answers it correctly without mentioning any simulation mode. The AI proves that it knows in which tutorial mode the user is, and when asked to describe what the user can do in it, it answers the question correctly, in a simple and general manner.

By the end of these tests, the AI assistant's performance was deemed sufficient for the proof of concept. The AI stopped associating the robot controls with guided mode, its answers became generally shorter and more simplified, and the use of bullet points became less frequent. Despite these improvements, the AI assistant still wasn't perfect. Occasionally, if the question asked is too general and difficult to answer briefly, the AI might still produce a long response that extends beyond the bounds of the scene's text box, it might also still use bullet-points if it's asked to describe a long process. Additionally, the AI may occasionally forget certain details. This could be due to the prompt not being clear enough or due to the limitations of the AI itself.

AI models like ChatGPT have limited memory. While they are designed to retain the context of conversations, when the conversation becomes too lengthy and the memory token limit is approached, the model may start forgetting certain aspects of the conversation.

But despite these challenges, the bot successfully answered user questions, retained important contextual information about the simulation, and kept pace with the steps in the guided tutorial. Furthermore, the TTS (Text-to-Speech) and STT (Speech-to-Text) features functioned properly, with only a minor delay between the user's question, the STT interpretation of it, and the AI's spoken response through TTS.

## 4.2.2 Tutorial Sequence Test

Now that an overview of the testing processes for various individual aspects of the system has been provided, it's important to evaluate the performance of the guided tutorial sequence as a whole. Particularly because this sequence effectively showcases all the previously discussed components working together in unison to deliver an interactive learning experience to the user.

Overall, the tutorial sequence functions well. Throughout, the system maintains frequent communication with the user. TTS (Text-to-Speech) is utilized not only for the AI assistant but also for delivering any instructions to the user during the tutorial. The auxiliary text boxes on the virtual wall are easily visible and serve their purpose effectively, reminding users of the current step's instructions and summarizing previous AI interactions.

### 4.2.2.1 AI performance in Guided Mode

The quality of feedback from the AI assistant was strong. For instance, during an interaction from step 1, the user receives a clear response from the AI assistant explaining why certain buttons are disabled:

- **User:** "Why are there buttons that I can't press?"
- **Assistant:** "In guided mode, some buttons may be disabled in certain steps because they are not needed for the current task or instruction. These disabled buttons will become active in later steps as we progress through the tutorial. It helps to ensure that you focus on the relevant buttons and actions for each step, making the learning process more structured and manageable."

In another example, we can see an interaction made in step 3, where the user asks the assistant what the yellow line on the gripper signifies:

- **User:** "What is that yellow line on top of the gripper?"
  
- **Assistant:** "The yellow line on top of the gripper is the gripper's detection range indicator. When the gripper is within range of an object, the yellow line will turn red, indicating that the gripper has detected the presence of an object. This helps you determine whether the gripper is close enough to pick up the object or not. Make sure the yellow line turns red before attempting to pick up an object to ensure successful gripping."

In another instance, we see an interaction where the user asks the AI assistant how to complete the step they are on (step 4), demonstrating that the bot is aware of the current step without requiring user interference:

- **User:** "What am I supposed to do in this step?"
  
- **Assistant:** "In this step, you need to save and delete points the Insert and Delete buttons. The saved points will be listed on the left side of the controller, and you can navigate them using the arrow buttons. Saving a point records the current position values of each joint and grip state."

These are just a few examples from tests conducted throughout the tutorial sequence. Despite the simplicity of the AI's setup, which consists of a prompt and two functions, the assistant still demonstrated reasonable capabilities in assisting the user. The AI assistant is not fully aware of the situation within the VR environment, it doesn't track the robot's position, movements, or how many control buttons are enabled at any given time, but it has enough understanding of the goals and conditions for progressing through each step to offer useful feedback on how to complete the tutorial.

#### 4.2.2.2 TTS performance in Guided Mode

Another aspect of the tutorial sequence that proved valuable during testing was the consistent, pre-defined, TTS-based feedback given to the user as they progressed. Throughout the tutorial, the same voice used by the assistant is employed not only to provide instructions but also to guide the user in various ways.

A good example of these pre-defined interactions occurs during step 3. When the user first picks up the object, the voice says, "You've picked up the cube. Notice how the label on the left screen is now red, indicating a gripping state. Try moving it around and then click the button again to drop it". When the user drops the cube, the voice responds, "You've dropped the cube, good job. If you want to move on to the next phase of this tutorial, press A". Throughout various steps, the AI voice offers tips, congratulates the user on their progress, and informs them when they are ready to move on to the next tutorial step.

In some steps, if the user presses the A button without completing the necessary task to advance, the tutorial voice informs them of what needs to be done. For instance, in step 7, if the user presses A without saving at least 3 robot positions for the sequence, the voice says, "You should have at least 3 saved positions for this sequence". If the user presses A after saving 3 positions but without the cube having passed through the required bases, the voice says, "It doesn't seem like the cube has passed through both bases".

Overall, these pre-defined interactions by the AI voice enhanced the immersive experience and played a crucial role in keeping the user informed of what needed to be done and when they could advance in the tutorial.

#### 4.2.2.3 Guided Mode progression quality

In this tutorial sequence, one of the most crucial aspects to test was the effectiveness of the tutorial progression conditions. As mentioned previously, the guided tutorial consists of a series of steps that the user must complete. To ensure that the system knows when the user has completed these tasks, certain functions are used to verify the completion conditions.

With this in mind, testing focused heavily on evaluating the reliability of these functions. In most cases, particularly in steps that require simple tasks, the conditions worked well. For example, in step 2, where the goal is for the user to change the robot's speed, the system simply verifies if the robot's speed was changed to "FAST" (since it always starts in "SLOW" by default). Simple tasks like this are easy for the system to track, so there were no major issues. The more challenging steps to verify were step 1 and step 7.

Step 1 was particularly difficult because the condition for progression requires the user to click each of the twelve joint movement buttons at least once. To track which buttons the user has pressed, Unity methods and ButtonTracker logic were used. Unfortunately, these methods were not always accurate, and sometimes the system failed to register when a button was pressed. This led to confusion for users, especially those unaware of the issue. In some cases, the user had to press buttons multiple times before they were properly registered. This problem arose from a design flaw that could likely be fixed with more time.

Step 7's conditions were difficult not because they failed to verify what they were designed to verify, but because the verifications made don't fully ensure the user completed the task correctly. In this step, the system requires the user to save more than three robot positions and requires that the object has been in contact with both base A and base B. While a good sequence should indeed have at least three saved positions and the object should pass from base A to base B, the verification criteria are too simplistic. The user could, for instance, save three unrelated positions and manually move the cube to base B or use the cube movement screen to place it there, bypassing the real goal of the step and still progressing.

The conditions for these two steps lack robustness and could be improved, but due to time constraints and different development priorities, they were left as they were.

Despite these issues, the guided tutorial experience successfully achieves its primary goal of providing an interactive learning experience with close guidance and support from the system. In most cases, the AI assistant effectively assists the user and keeps pace with the tutorial sequence, helping to manage the user's learning process.

## 4.3 Results Phase

Now that an overview of the system's testing process has been provided, the next subchapter will summarize the system's overall performance during testing and assess how well it aligns with the project's initial goals. Additionally, potential improvements will be explored, analysing how the system could be further developed and scaled with additional time and resources, along with the potential benefits such advancements could bring.

### 4.3.1 General Results

The original goal of this project was to develop a VR-based learning system dedicated to teaching users how to control a manufacturing robot. The system would feature a virtual replica of a real robot, controllable by the user, and supported by an AI-based assistant. Ideally, this system would provide an immersive learning experience, allowing users to feel what it's like to control a robot through the realism of the digital robot and its controls. The AI assistant, along with other system tools, would offer effective, concise support to the user's learning process, drawing inspiration from the learning paradigms discussed in the state-of-the-art section of this dissertation.

By the end of development, the system met these goals to a reasonable degree. The VR environment functions well, and the VR controls are practical and well-defined. While the VR environment may not feature the most detailed graphics, the experience is immersive and serves its purpose. Thanks to Realvirtual.io's resources, it was possible to create a realistic digital replica of a manufacturing robot in the virtual world and control it in a realistic way.

The control system assigned to the robot's movement is not as sophisticated as a real robot controller and lacks a 3D model, like the one the robot has. However, it functions effectively as a handheld controller, allowing the user to move robotic joints, pick up objects, change the robot's speed, save and manage robot positions, and even run sequences of movements. These features meet the objective of teaching users realistic robot controls, offering enough depth to provide a meaningful learning challenge.

The AI component of the system also meets its intended goal. AI tools are widely used throughout the system. AWS's TTS services facilitate communication, not just for the AI assistant, but also throughout the guided tutorial. Communication with the AI assistant was efficient and user-friendly, with the combined use of Whisper, GPT-3.5, and AWS's TTS proving successful, allowing the user to interact with the AI smoothly.

The AI assistant's performance was satisfactory. It was prepared with relevant system context and kept updated with simple functions, as demonstrated in the earlier tests. The guided tutorial sequence, with its consistent instructions and guidance, showed how the system could offer a highly interactive and immersive learning experience. Through its verification functions, TTS communication, and the availability of the AI assistant for additional support, the guided tutorial successfully demonstrated the concept of immersive AI-assisted learning using a realistic digital replica of a robot and VR, an essential part of this proof of concept.

However, despite these successes, it is also important to recognize the system's limitations.

The initial UI is straightforward but lacks complexity, offering few choices and not providing the user with sufficient exposure to the system's rules and controls beforehand.

The digital replica has a functional control scheme, useful features and realistic looks, but it lacks realism in some areas. For instance, the gripper doesn't physically close when picking up objects, and the object-picking physics isn't realistic since the gripper's fingers lack proper collision mechanics. Additionally, the robot does not have movement limits like a real robot, and it can pass through physical objects without reacting to collisions.

The robot control system could also be more realistic. It doesn't emulate most functions of a real robot controller, lacks a 3D model, doesn't simulate the process of downloading data from the controller to a computer, and isn't a replica of an actual system. Instead, it is a made-up hybrid system, inspired by Yaskawa's controllers and the FlexPendant, replicating only a few of their key functions.

Due to time and resource constraints, a true digital twin was not implemented in this project. Instead, a digital replica of a robot, without any real-time connection to a physical counterpart, was used.

The AI assistant, while functional, lacks the robustness needed to avoid occasional mistakes during interactions. Preparing the AI with a simple prompt and a few functions works well for a proof of concept, but a fully developed product would require more complex functions and much deeper preparation to ensure consistent performance.

The guided tutorial sequence demonstrates the potential of these tools in teaching users how to work with professional equipment, in this case, a manufacturing robot. However, there is room for improvement in the level and quality of assistance provided during the tutorial, especially when it comes to the tutorial progression logic.

Despite these challenges, the project successfully meets the core goals and requirements of this dissertation. The aim was to demonstrate that these tools could be integrated into a system to deliver an immersive learning experience, and the final system effectively proves that this can be achieved with these technological tools that have gained prominence in recent years.

The identified issues are not insurmountable, and most stem from the project's small scale and the limited time available for its development. As a proof of concept, the system doesn't need to be perfect in every aspect, it simply needs to demonstrate that the concept is

valuable and feasible with the current environment and available technologies, and this project is successful in those aspects.

### 4.3.2 Possible Improvements

Now that we've reflected on the system's performance, quality, and alignment with the main goals of this project, it's interesting to explore how the system could be improved. How much could a system like this proof of concept be enhanced to become genuinely useful in practice? What is the potential for systems like this in the future?

Many ways to improve this particular system have already been mentioned in general terms throughout the previous chapters.

First, the UI could be enhanced. The initial main menu could benefit from offering more options to the user. For example, a separate virtual environment could be created specifically to teach users the basic VR controls for the simulation, accessible from the main menu in a manner similar to how the Free Tutorial and Guided Tutorial scenes are accessed. This introductory VR tutorial could be recommended if it's the user's first time using the system.

The initial menu could also include an options screen where users could adjust settings such as audio levels, microphone selection, graphics quality, and other options.

Another valuable addition would be allowing for the option of experiencing the system without VR. This would require the development of a new control scheme, but it would extend the application's accessibility to users who don't own VR equipment. This could also be offered as an option from the initial main menu.

The settings screen could also be accessible from within the simulation environment, from a newly added pause option. Currently, the simulation only pauses when instructions are given in guided mode, but a separate pause system could be added, allowing the user to pause with a button and access a settings screen such as those mentioned earlier. This would, for example, eliminate the need for AI-related settings to be displayed on the walls of the virtual room.

Beyond these potential UI, practicality, and accessibility improvements, the system could be significantly enhanced in terms of the realism of the virtual environment. In the proof-of-concept system, the 3D environment was created using Unity 3D objects, but a more advanced system could include a digital twin of an entire factory. With sufficient resources, a real digital

twin not only of the robot but of an actual real place could be developed, making the simulated environment much more immersive.

Replacing the robot's digital replica with a fully realized digital twin would be ideal. Making it have realistic physics and collision simulations would be a substantial upgrade compared to the current system.

Enhancing the realism of the handheld robot controller would also represent a major improvement. The virtual robot controller could be an exact digital twin of a real-world robot controller, with all its functionalities, potentially even capable of exporting data in a manner similar to the real version.

In a scenario where a system like this is developed on a larger scale, both in depth and complexity, a more advanced AI assistant with additional interactive features would be a significant advantage. This could be achieved through fine-tuning, a method previously mentioned, where developers train the AI by providing it with detailed datasets. AI interaction with the user could also be improved by incorporating many more functions tied to events within the system.

The system could also be expanded with more training modes, environments, and simulations. There are numerous ways to build upon the concepts introduced in this project, and this concept could extend beyond robotics. Almost any discipline, especially those involving specific tools, could benefit from a system designed around the idea of a simulated VR environment with digital twins and AI assistance designed for learning.

Currently, VR equipment remains relatively expensive, and generative AI is still in its early stages. However, as time progresses, it's reasonable to expect these tools to become more commonplace and accessible, a process that is already becoming apparent, as noted in the state-of-the-art chapters of this document.



## CONCLUSION

### 5.1 General Overview of the Development Process

The topic of this dissertation emerged largely from the current landscape of industry and technology. Today, we find ourselves in a transitional phase, moving from the paradigms of the fourth industrial revolution to those of the fifth, at a time when previously niche technologies are becoming more mainstream, cheaper, and more accessible. Alongside this, we are witnessing the early stages of one of the most disruptive technologies in history, the rise and popularization of generative AI.

Additionally, the modern world of work is becoming increasingly competitive, with a growing demand for highly skilled workers. The learning and development (L&D) industry is incredibly valuable, with a constant drive for more efficient and effective teaching methods.

This context is what led to the idea of proving the concept for a system aimed at teaching professionals, built around VR, a digital replica (or ideally, a digital twin), and AI tools. The goal was to harness some of the most relevant technologies of recent years to create a system designed to educate professionals, specifically in the field of robotics.

Development of this proof-of-concept system began with a thorough investigation of available technological tools, choosing the most suitable ones given the available resources and time constraints. After this analysis, Unity was chosen as the main development framework, OpenAI's GPT-3.5 was selected as the main integrated generative AI tool, and Realvirtual.io was used for creating the robot's digital replica and programming its functionalities.

Later in development, it was decided to also integrate text-to-speech (TTS) and speech-to-text (STT) tools into the system to facilitate communication and the interactions between

the user and the system. Specifically, AWS's TTS service and OpenAI's Whisper STT service were incorporated.

The system ultimately ended up being divided into two tutorial modes: guided mode and free mode. In guided mode, the user receives consistent feedback from the simulation as they attempt to complete pre-planned tasks, which are divided into several steps. The goal of these steps is to provide a more interactive and structured learning experience. In free mode on the other hand, the user can explore the simulated environment without any restrictions.

In both modes, the user has access to a generative AI-based assistant capable of answering questions and offering assistance within the virtual environment.

## 5.2 Main Challenges

By the end of the main development phase, extensive testing and experimentation were conducted. The AI assistant was pushed to its limits to assess how well it adapted to different scenarios within the system. Issues were identified with the guided mode's progression, and through trial and error, as new problems were found during further testing, the system was systematically improved within the available timeframe.

The main challenges during the development and testing phases of this project were arguably the preparation of the AI assistant, the development of the guided tutorial sequence, and the creation of the robot control logic.

Preparing the AI assistant was challenging because it required extensive trial-and-error testing to optimize the efficiency and quality of its communication. Despite the numerous tests conducted, it's likely that further improvements could have been made with additional testing.

The guided tutorial sequence posed difficulties due to the need for careful pre-planning to design a series of tasks that would facilitate effective learning. Another challenge was developing verification functions, which had to analyse the user's actions and determine whether the task was completed to advance in the tutorial.

Developing the robot's control logic was also demanding, as it required not only an understanding of the Realvirtual.io framework but also a certain degree of knowledge about how real robot controllers operate. Building the handheld control system and linking it with the PLC script that manages the robot's movements using Realvirtual.io's tools was particularly complex.

Another relevant challenge during these phases was the limited access to VR equipment. In the early stages, testing the system relied solely on the XR Device Simulator component, as mentioned earlier.

Interestingly however, the process of integrating AI tools, Realvirtual.io, and VR capabilities into the system was relatively straightforward in comparison with these challenges.

### **5.3 Most Relevant Results**

By the end of development, while the final system was not perfect, it successfully met the primary objectives.

The three core technologies explored at the outset, VR, the digital replica, and generative AI, were successfully integrated into a unified system designed for teaching. The VR environment is immersive, the robot's digital replica is realistic and can be controlled in a manner that reflects how real robots are operated, emulating relatively well the behaviour that would be expected from a digital twin. The various AI tools work together to provide an interactive learning experience, with TTS guiding users through the tutorial, and an AI assistant well-informed about the subject matter, readily available to assist the user.

A key outcome of this proof of concept is the ease with which an AI assistant could be integrated and adapted to the system's context.

### **5.4 Future Work**

These conclusions demonstrate that systems similar to this one are likely to become increasingly relevant in the future. The fact that the AI assistant's preparation was simple and rudimentary, yet produced positive results, only highlights the immense potential of these tools moving forward.

As this dissertation concludes, it is safe to say that the project successfully demonstrates the feasibility of virtual training systems based on VR, digital twin, and generative AI technologies. It provides solid evidence that similar systems could gain traction in the future as these tools continue to improve and become more accessible. It's highly likely that future iterations of this concept, with more robust features and advanced technologies, will enhance efficiency in the L&D field.

In future work, it would be interesting to explore the simulation of real, larger-scale digital twins, with more complex systems within the virtual environment. Additionally, more advanced and robust AI systems could be developed, AI that is more knowledgeable about the topics being addressed and capable of interacting with users in a more consistent and autonomous manner.

## BIBLIOGRAPHY

- [1] 'What is industry 4.0 and the Fourth Industrial Revolution? | McKinsey'. Accessed: Jan. 01, 2024. [Online]. Available: <https://www.mckinsey.com/featured-insights/mckinsey-explainers/what-are-industry-4-0-the-fourth-industrial-revolution-and-4ir>
- [2] D. M. Theodotou, 'Optimizing Innovation In Learning And Development', eLearning Industry. Accessed: Jan. 25, 2024. [Online]. Available: <https://elearningindustry.com/optimizing-innovation-in-learning-and-development>
- [3] 'Learning management system', *Wikipedia*. Jan. 15, 2024. Accessed: Jan. 25, 2024. [Online]. Available: [https://en.wikipedia.org/w/index.php?title=Learning\\_management\\_system&oldid=1195823501](https://en.wikipedia.org/w/index.php?title=Learning_management_system&oldid=1195823501)
- [4] 'Moodle 4 Administration - Fourth Edition'. Accessed: Jan. 28, 2024. [Online]. Available: <https://subscription.packtpub.com/book/web-development/9781801816724/2/ch02lv1sec08/understanding-the-moodle-architecture>
- [5] 'Workplace training: spending per employee 2021', Statista. Accessed: Jan. 02, 2024. [Online]. Available: <https://www.statista.com/statistics/738519/workplace-training-spending-per-employee/>
- [6] 'Workplace training: training hours used per employee', Statista. Accessed: Jan. 02, 2024. [Online]. Available: <https://www.statista.com/statistics/739741/learning-hours-used-per-employee/>
- [7] S. Sedgman, 'How Much Companies Spend on Employee Training?', LearnExperts. Accessed: Jan. 02, 2024. [Online]. Available: <https://learnexperts.ai/blog/how-much-do-companies-spend-on-training-per-employee/>
- [8] T. MacDonald, 'How to choose the right LMS', LearnExperts. Accessed: Jan. 02, 2024. [Online]. Available: <https://learnexperts.ai/blog/how-to-choose-the-right-lms-selection-guide-and-tips/>
- [9] manaable, 'What is a Learning Management System?', Medium. Accessed: Jan. 25, 2024. [Online]. Available: [https://medium.com/@manaable\\_blog-en/what-is-a-learning-management-system-8fb15880f699](https://medium.com/@manaable_blog-en/what-is-a-learning-management-system-8fb15880f699)
- [10] L. M. Miller, 'How to Build Lean Systems of Organizational Learning', Management Meditations. Accessed: Jan. 17, 2024. [Online]. Available: <https://www.lmmiller.com/build-lean-system-organizational-learning/>
- [11] A. Alley, 'The Industry 4.0 Effect: Why SMEs Should Adopt Smart Technologies Today - Articles - Automation Alley'. Accessed: Jan. 25, 2024. [Online]. Available: <https://www.automationalley.com/articles/the-industry-4-0-effect-why-smes-should-adopt-smart-technologies-today>

- [12] E. Forcael, G. Garcés, and A. D. Lantada, 'Convergence of Educational Paradigms into Engineering Education 5.0', in *2023 World Engineering Education Forum - Global Engineering Deans Council (WEEF-GEDC)*, Oct. 2023, pp. 1–8. doi: 10.1109/WEEF-GEDC59520.2023.10344026.
- [13] 'What is Industry 5.0? (In-depth guide with examples)'. Accessed: Jan. 03, 2024. [Online]. Available: <https://www.clarify.io/learn/industry-5-0>
- [14] Momenta, 'Industry 5.0 | Industry 5.0 Strategy | Industry 5.0 Technologies'. Accessed: Jan. 25, 2024. [Online]. Available: <https://www.momenta.one/industry5.0>
- [15] 'Figure 1. The physical robotic arm (left) and its Digital Twin (right).', ResearchGate. Accessed: Jan. 25, 2024. [Online]. Available: [https://www.researchgate.net/figure/The-physical-robotic-arm-left-and-its-Digital-Twin-right\\_fig1\\_346222676](https://www.researchgate.net/figure/The-physical-robotic-arm-left-and-its-Digital-Twin-right_fig1_346222676)
- [16] 'What is a digital twin? | IBM'. Accessed: Jan. 07, 2024. [Online]. Available: <https://www.ibm.com/topics/what-is-a-digital-twin>
- [17] 'Digital Twin to Enhance Corporate Education - Visartech Blog', Visartech. Accessed: Jan. 07, 2024. [Online]. Available: <https://www.visartech.com/blog/how-to-create-a-digital-twin-for-education/>
- [18] 'Digital Twins and AR/VR Applications', EIT Campus. Accessed: Jan. 25, 2024. [Online]. Available: <https://eit-campus.eu/course/manufacturing/digital-twins-and-ar-vr-applications>
- [19] 'What is the Future of Virtual Reality in Education?', Computeam. Accessed: Jan. 25, 2024. [Online]. Available: <https://www.computeam.co.uk/videos-and-blog/article/what-is-the-future-of-virtual-reality-in-education>
- [20] 'History of VR – Timeline of Events and Tech Development – VirtualSpeech'. Accessed: Jan. 25, 2024. [Online]. Available: <https://virtualspeech.com/blog/history-of-vr>
- [21] L. P. Berg and J. M. Vance, 'Industry use of virtual reality in product design and manufacturing: a survey', *Virtual Reality*, vol. 21, no. 1, pp. 1–17, Mar. 2017, doi: 10.1007/s10055-016-0293-9.
- [22] Y. E. Council, 'Council Post: Augmented Reality In Business: How AR May Change The Way We Work', Forbes. Accessed: Jan. 25, 2024. [Online]. Available: <https://www.forbes.com/sites/theyec/2019/02/06/augmented-reality-in-business-how-ar-may-change-the-way-we-work/>
- [23] 'The Ultimate Guide to Types of Virtual Reality', Atria University. Accessed: Jan. 09, 2024. [Online]. Available: <https://www.atriauniversity.edu.in/types-of-virtual-reality/>
- [24] P. Miller, 'Testing the Virtuix Omni: a walk on the virtual side', The Verge. Accessed: Jan. 25, 2024. [Online]. Available: <https://www.theverge.com/2013/6/11/4419832/virtuix-omni-vr-hands-on-demo>
- [25] '(3) Global Augmented And Virtual Reality (AR VR) Market Size, Share, Trends, and Forecasted Growth for 2023-2030 | LinkedIn'. Accessed: Jan. 09, 2024. [Online]. Available: <https://www.linkedin.com/pulse/global-augmented-virtual-reality-ar-vr-market-size/>
- [26] PricewaterhouseCoopers, 'What does virtual reality and the metaverse mean for training?', PwC. Accessed: Jan. 09, 2024. [Online]. Available: <https://www.pwc.com/us/en/tech-effect/emerging-tech/virtual-reality-study.html>
- [27] B. Santos, 'What is Google Bard? An emerging chatbot | IPVanish'. Accessed: Jan. 25, 2024. [Online]. Available: <https://www.ipvanish.com/blog/what-is-google-bard/>

- [28] C. Coop, 'The first "successful" chatterbot from 1966', Caravan Coop. Accessed: Jan. 25, 2024. [Online]. Available: <https://medium.com/caravan-blog/the-first-successful-chatterbot-from-1966-17088c4c4669>
- [29] E. Adamopoulou and L. Moussiades, 'An Overview of Chatbot Technology', in *Artificial Intelligence Applications and Innovations*, I. Maglogiannis, L. Iliadis, and E. Pimenidis, Eds., in IFIP Advances in Information and Communication Technology. Cham: Springer International Publishing, 2020, pp. 373–383. doi: 10.1007/978-3-030-49186-4\_31.
- [30] 'Chatbots Powered by Conversational AI for Airport Operations', Master of Code Global. Accessed: Jan. 25, 2024. [Online]. Available: <https://masterofcode.com/conversational-ai-chatbot-for-airport-operations>
- [31] P. Kulp, 'Heinz Taps State-of-the-Art AI to Design Its Next Ad'. Accessed: Jan. 25, 2024. [Online]. Available: <https://www.adweek.com/commerce/heinz-taps-ai-design-next-ad-campaign/>
- [32] K. Knibbs, 'Why This Award-Winning Piece of AI Art Can't Be Copyrighted', *Wired*. Accessed: Jan. 25, 2024. [Online]. Available: <https://www.wired.com/story/ai-art-copyright-matthew-allen/>
- [33] F. Fui-Hoon Nah, R. Zheng, J. Cai, K. Siau, and L. Chen, 'Generative AI and ChatGPT: Applications, challenges, and AI-human collaboration', *Journal of Information Technology Case and Application Research*, vol. 25, no. 3, pp. 277–304, Jul. 2023, doi: 10.1080/15228053.2023.2233814.
- [34] M. Metropolitan, 'Chatbots in Education: Do You Need a Virtual Assistant? \* Madrid Metropolitan', Madrid Metropolitan. Accessed: Jan. 25, 2024. [Online]. Available: <https://www.madridmetropolitan.com/chatbots-in-education-do-you-need-a-virtual-assistant/>
- [35] K. Pal, 'Who are the Competitors of ChatGPT? 12 Biggest Market Players', Techopedia. Accessed: Jan. 11, 2024. [Online]. Available: <https://www.techopedia.com/who-are-the-competitors-of-chatgpt>
- [36] 'Gerador de código de IA – Amazon CodeWhisperer – AWS', Amazon Web Services, Inc. Accessed: Jan. 25, 2024. [Online]. Available: <https://aws.amazon.com/pt/codewhisperer/>
- [37] 'An Overview and Introduction to Rasa', An Overview and Introduction to Rasa. Accessed: Jan. 11, 2024. [Online]. Available: <https://www.xaqt.com/blog/an-overview-and-introduction-to-rasa/>
- [38] 'A-Frame (virtual reality framework)', *Wikipedia*. May 17, 2023. Accessed: Jan. 04, 2024. [Online]. Available: [https://en.wikipedia.org/w/index.php?title=A-Frame\\_\(virtual\\_reality\\_framework\)&oldid=1155299909](https://en.wikipedia.org/w/index.php?title=A-Frame_(virtual_reality_framework)&oldid=1155299909)
- [39] 'White Paper: Now You Can Program Robots in Virtual Reality', Engineering.com. Accessed: Jan. 04, 2024. [Online]. Available: <https://www.engineering.com/story/white-paper-now-you-can-program-robots-in-virtual-reality>
- [40] L. Eitel, 'RobotStudio — Software from ABB optimizes robot setups (with videos)', The Robot Report. Accessed: Jan. 25, 2024. [Online]. Available: <http://www.robotictips.com/robotstudio-605-industrial-software-from-abb-to-increase-robot-effectiveness/>
- [41] '(7) How BMW is Revolutionizing Industries with Digital Twin and Metaverse Technologies | LinkedIn'. Accessed: Jan. 05, 2024. [Online]. Available: <https://www.linkedin.com/pulse/how-bmw-revolutionizing-industries-digital-twin-metaverse-nh2oc/>

- [42] 'NVIDIA launches Omniverse Design Collaboration and Simulation Platform', Digitalisation World. Accessed: Jan. 25, 2024. [Online]. Available: <http://m.digitalisation-world.com/news/61213/nvidia-launches-omniverse-design-collaboration-and-simulation-platform>
- [43] A. Ltd, 'What is a Gaming Engine?', Arm | The Architecture for the Digital World. Accessed: Jan. 04, 2024. [Online]. Available: <https://www.arm.com/glossary/gaming-engines>
- [44] G. D. B. Team, 'Godot vs Unity (for making your first game)', Game Dev Beginner. Accessed: Jan. 04, 2024. [Online]. Available: <https://gamedevbeginner.com/godot-vs-unity-for-making-your-first-game/>
- [45] B. Perdigao, 'Godot Engine is growing potential for the AEC industry.', OSArch. Accessed: Jan. 04, 2024. [Online]. Available: <https://osarch.org/2021/04/27/godot-engine-is-growing-potential-for-the-aec-industry/>
- [46] 'Unity vs Unreal: Which Game Engine Should You Choose?', Hackr.io. Accessed: Jan. 04, 2024. [Online]. Available: <https://hackr.io/blog/unity-vs-unreal-engine>
- [47] 'Manipulator Robot in Blueprints - UE Marketplace', Unreal Engine. Accessed: Jan. 25, 2024. [Online]. Available: <https://www.unrealengine.com/marketplace/en-US/product/manipulator-robot>
- [48] 'Lockheed Martin engages Unreal Engine for aerospace simulation R&D', Unreal Engine. Accessed: Jan. 05, 2024. [Online]. Available: <https://www.unrealengine.com/en-US/spotlights/lockheed-martin-engages-unreal-engine-for-aerospace-simulation-r-d>
- [49] 'Game engines used by video game developers UK 2019', Statista. Accessed: Jan. 10, 2024. [Online]. Available: <https://www.statista.com/statistics/321059/game-engines-used-by-video-game-developers-uk/>
- [50] Z. Wang, K. Han, and P. Tiwari, 'Digital Twin Simulation of Connected and Automated Vehicles with the Unity Game Engine', in *2021 IEEE 1st International Conference on Digital Twins and Parallel Intelligence (DTPI)*, Jul. 2021, pp. 1–4. doi: 10.1109/DTPI52967.2021.9540074.
- [51] U. Technologies, 'Robotics Simulation | Unity'. Accessed: Jan. 25, 2024. [Online]. Available: <https://unity.com/solutions/automotive-transportation-manufacturing/robotics>
- [52] '(7) The Shifting Landscape of Industrial Simulation: Unity, Unreal Engine, and Omniverse | LinkedIn'. Accessed: Jan. 05, 2024. [Online]. Available: <https://www.linkedin.com/pulse/shifting-landscape-industrial-simulation-unity-unreal-thomas-strigl/>
- [53] 'Demo model'. Accessed: Jan. 17, 2024. [Online]. Available: <https://doc.realvirtual.io/basics/demo-model>
- [54] 'Volkswagen Group is backing virtual reality solutions for interactive collaboration in production & logistics', Volkswagen Group. Accessed: Sep. 25, 2024. [Online]. Available: <https://www.volkswagen-group.com/en/press-releases/volkswagen-group-is-backing-virtual-reality-solutions-for-interactive-collaboration-in-production-and-logistics-16361>
- [55] 'Development using VR glasses: How Volkswagen uses virtual workflows', Volkswagen Newsroom. Accessed: Sep. 25, 2024. [Online]. Available: <https://www.volkswagen-newsroom.com/en/stories/development-using-vr-glasses-how-volkswagen-uses-virtual-workflows-15786>
- [56] S. Hayden, 'Volkswagen Group is Building VR Apps to Help Employees Collaborate Across the Globe', Road to VR. Accessed: Sep. 26, 2024. [Online]. Available:

- <https://www.roadtovr.com/volkswagen-group-one-worlds-largest-companies-building-vr-apps-help-employees-collaborate-across-globe/>
- [57] 'Case study: Walmart embraces immersive learning', Chief Learning Officer. Accessed: Sep. 25, 2024. [Online]. Available: <https://www.chieflearningofficer.com/2021/03/23/case-study-walmart-embraces-immersive-learning/>
- [58] 'Here's Our List of Companies Using VR Training - ArborXR'. Accessed: Sep. 25, 2024. [Online]. Available: <https://arborxr.com/blog/companies-using-vr-training/>
- [59] J. Bailenson, 'Is VR the Future of Corporate Training?', *Harvard Business Review*, Sep. 18, 2020. Accessed: Sep. 25, 2024. [Online]. Available: <https://hbr.org/2020/09/is-vr-the-future-of-corporate-training>
- [60] 'VR Stats for the Training & Education Industry in 2024 – VirtualSpeech'. Accessed: Sep. 26, 2024. [Online]. Available: <https://virtualspeech.com/blog/vr-stats-training-education>
- [61] S. Subbarao, 'How Companies Use AI Chatbots To Support On-Demand Employee Learning', eLearning Industry. Accessed: Sep. 25, 2024. [Online]. Available: <https://elearningindustry.com/how-companies-use-ai-chatbots-to-support-on-demand-employee-learning>
- [62] I. Stanica, M.-I. Dascalu, C. N. Bodea, and A. D. Bogdan Moldoveanu, 'VR Job Interview Simulator: Where Virtual Reality Meets Artificial Intelligence for Education', in *2018 Zooming Innovation in Consumer Technologies Conference (ZINC)*, May 2018, pp. 9–12. doi: 10.1109/ZINC.2018.8448645.
- [63] M. Taylor *et al.*, 'The Rise of the AI Scientist: Unleashing the Potential of Chat-GPT Powered Avatars in Virtual Reality Digital-twin Laboratories', Chemistry, preprint, Nov. 2023. doi: 10.26434/chemrxiv-2023-t4vg7.
- [64] D. Aggarwal, D. Sharma, and A. B. Saxena, 'Exploring the Role of Artificial Intelligence for Augmentation of Adaptable Sustainable Education', *AJARR*, vol. 17, no. 11, pp. 179–184, Oct. 2023, doi: 10.9734/ajarr/2023/v17i11563.
- [65] D. Grover, 'Next-Generation Education: The Impact of Generative AI on Learning', *Journal of Informatics Education and Research*, vol. 4, no. 2, Art. no. 2, Jun. 2024, doi: 10.52783/jier.v4i2.1019.
- [66] D. Liu, K. K. Bhagat, Y. Gao, T.-W. Chang, and R. Huang, 'The Potentials and Trends of Virtual Reality in Education', in *Virtual, Augmented, and Mixed Realities in Education*, D. Liu, C. Dede, R. Huang, and J. Richards, Eds., Singapore: Springer, 2017, pp. 105–130. doi: 10.1007/978-981-10-5490-7\_7.
- [67] M. Singh *et al.*, 'Applications of Digital Twin across Industries: A Review', *Applied Sciences*, vol. 12, no. 11, Art. no. 11, Jan. 2022, doi: 10.3390/app12115727.
- [68] 'FlexPendant - Robotics - ABB | Reconditioning repairs – refurbished spare parts – new parts – service', Service und Reparatur von CNC und SPS Systemen | BVS Electronics GmbH. Accessed: Sep. 25, 2024. [Online]. Available: <https://shop.bvs-electronics.com/en/products/abb/robotics/flexpendant>
- [69] 'Figure 7.5. RobotStudio®-RAPID editor-example', ResearchGate. Accessed: Sep. 25, 2024. [Online]. Available: [https://www.researchgate.net/figure/RobotStudioR-RAPID-editor-example\\_fig60\\_318209199](https://www.researchgate.net/figure/RobotStudioR-RAPID-editor-example_fig60_318209199)
- [70] 'JZRCR-YPP21-1 | Yaskawa | Motoman DX200 Teach Pendant', NEX Instrument Inc. Accessed: Sep. 25, 2024. [Online]. Available: <https://www.nexinstrument.com/jzrcr-ypp21-1>

- [71] 'Glossary'. Accessed: Sep. 25, 2024. [Online]. Available: [https://www.yaskawa.pl/wsparcie\\_techiczne/Total%20Customer%20Support/glossary](https://www.yaskawa.pl/wsparcie_techiczne/Total%20Customer%20Support/glossary)
- [72] 'Unity Asset Store'. Accessed: Sep. 24, 2024. [Online]. Available: <https://assetstore.unity.com/>
- [73] 'Welcome to the VR Pathway', Unity Learn. Accessed: Jan. 24, 2024. [Online]. Available: <https://learn.unity.com/tutorial/welcome-to-the-pathway>
- [74] 'Rig Simulator Keyboard Shortcuts.pdf'. Accessed: Sep. 24, 2024. [Online]. Available: <https://connect-prd-cdn.unity.com/20210604/28db6ca9-aba1-4ac3-a15a-24664daff3ea/Rig%20Simulator%20Keyboard%20Shortcuts.pdf>
- [75] 'Introduction to Rasa Open Source & Rasa Pro'. Accessed: Jan. 24, 2024. [Online]. Available: <https://rasa.com/docs/rasa/>
- [76] Retro, 'Integrating Rasa Open Source Chatbot Into Unity [Part 1] : The Connection', Analytics Vidhya. Accessed: Jan. 24, 2024. [Online]. Available: <https://medium.com/analytics-vidhya/integrating-rasa-open-source-chatbot-into-unity-part-1-the-connection-9ba582c804cd>
- [77] 'OpenAI Platform'. Accessed: Jan. 24, 2024. [Online]. Available: <https://platform.openai.com>
- [78] S. (Sarge) Altundas, *srcnalt/OpenAI-Unity*. (Jan. 23, 2024). C#. Accessed: Jan. 24, 2024. [Online]. Available: <https://github.com/srcnalt/OpenAI-Unity>
- [79] *FreedomIntelligence/finetune\_chatgpt*. (Aug. 20, 2024). Python. FreedomIntelligence. Accessed: Sep. 24, 2024. [Online]. Available: [https://github.com/FreedomIntelligence/finetune\\_chatgpt](https://github.com/FreedomIntelligence/finetune_chatgpt)







2024

ARTUR SEABRA

IMMERSIVE TRAINING THROUGH VR AND AI