

Politecnico di Torino

Master of Science in Mechatronic Engineering



Master of Science Thesis

Design and Implementation of a Mixed-Reality Robotic Training System to Support Multiple Learning Models

Supervisor

Prof. Fabrizio Lamberti

Candidate

Francisco Navarro Merino

July 2020

To my parents

Acknowledgments

I would like to express my gratitude to Prof. Fabrizio Lamberti for the opportunity provided to develop this project within the Department. Special mention to F. Gabriele Praticò for all the help throughout the year and particularly during the last few weeks to finish the project on time.

Abstract

This thesis aimed to develop a Mixed Reality-based robotic training platform able to support multiple learning modalities. A use case was performed to the system on two paradigms, the traditional method and learning by teaching, where the learner reinforces his or her own knowledge by giving lessons to a teachable agent, role in this case played by the Cozmo robot.

First, careful research was conducted to understand the current state of this field of study and the related technology, searching for possible lacks with the aim to take a step forward in the tackled domain. Then, an application was designed and implemented in Unity including a specific mode for each learning paradigm, whose topic was based on the Thèvenin Theorem from the electrical domain, aside from a tutorial to master the user interface.

The robot acted autonomously during the traditional mode, whereas in the second mode it was manually controlled following the Wizard of Oz approach. For fairness purposes, in both modes, the robot was endowed with identical movement and action capabilities. The Mixed Reality environment was implemented using a tabletop projection and enabling touch interaction applying a background subtraction algorithm to images captured from a depth camera sensor. Both the robot and the sensor were controlled using the Python language. The devised system was evaluated by several participants, whose experience was collected via a standard questionnaire with the objective to possibly validate the advantage of the studied methods under certain conditions.

In conclusion, the present thesis managed to design and implement a Mixed Reality robotic training system able to support two different learning models. The main concern during the development of the system was its usability, as the goal was to implement an intuitive interface that, after a short learning period to master it, could be capable to let the user focus on the learning objectives rather than on technology, a goal which was achieved to a certain extent.

Keywords: Mixed Reality – Cozmo Robot – Touch Screen – Unity – Wizard of Oz – Kinect – Human Robot Interaction – Learning by Teaching

Contents

Acknowledgements	V
Abstract	VI
Contents	VII
List of Figures	IX
List of Tables	XI
Introduction	1
1.1. Serious Games	1
1.2. Motivation.....	2
1.3. Scope of the Project	2
1.4. Organization of the Document	3
State of the Art	4
2.1. Mixed Reality	4
2.2. Learning.....	5
2.2.1. Robotic Teachable Agent	12
2.3. Wizard of Oz	16
2.4. Touch Surface	18
2.5. Robots in the Market	20
2.5.1. Cozmo	20
2.5.2. Sphero	20
2.5.3. Ozobot's Evo	21
2.5.4. Jumping Sumo	21
2.5.5. Vortex.....	22
2.5.6. Drones	22
2.6. Cameras in the Market.....	22
2.6.1. Kinect and Kinect v2.....	22
2.6.2. Intel RealSense Depth Cameras.....	23
2.6.3. Orbbec Astra Series	23
2.7. Previous Works	24
2.7.1. Station Master.....	24
2.7.2. Colour Pong.....	24
2.7.3. Protoman Revenge	25
2.7.4. Protect Treasures.....	26
Technology	27
3.1. Software	27
3.1.1. Unity	27
3.1.2. C# and Visual Studio	27
3.1.3. Python	27
3.1.4. HTML and Java	28
3.1.5. OpenCV.....	28
3.1.6. ZeroMQ.....	28
3.2. Hardware.....	28
3.2.1. Cozmo	29
3.2.2. Kinect v2	29
3.2.3. Projector	29
System Design	30
4.1. Brainstorming.....	30
4.2. System Overview.....	34
4.2.1. System concept	34
4.2.2. Training Modes.....	35

4.3.	Elements.....	36
System Implementation		38
5.1.	Overview.....	38
5.2.	Unity.....	39
5.2.1.	Interface.....	39
5.2.2.	Learning by Teaching (LxT)	40
5.2.3.	Scripts.....	40
5.2.4.	Methods.....	41
5.2.5.	Canvas.....	43
5.3.	Python.....	44
5.3.1.	Threads and Methods	44
5.3.2.	Wizard of Oz	45
5.4.	Touch Surface	47
5.5.	Robot	54
5.6.	IPC communication	55
5.6.1.	Functions	55
5.6.2.	Camera – Unity.....	56
5.6.3.	Unity – Robot	57
5.6.4.	Robot – Camera	59
5.6.5.	Camera – Robot – Unity.....	60
5.7.	System Deployment.....	62
5.7.1.	Manual	63
Experimental Evaluation		65
6.1.	Methodology	65
6.1.1.	Research Design and Sample	65
6.1.2.	Instruments	67
6.2.	Analysis of Results	72
Conclusions		75
7.1.	Final Considerations	75
7.2.	Lesson Learned.....	76
7.2.1.	System Concept	76
7.2.2.	Redesign.....	76
7.2.3.	Hand Recognition	77
7.3.	Future Work	77
7.3.1.	Hand Gestures.....	77
7.3.2.	Comparisons	78
Bibliography.....		79

List of Figures

Figure 1. Group Learning with Autonomous Robot ^[2]	6
Figure 2. Experiment with robot Keepon solving a nonogram ^[3]	6
Figure 3. Bloom’s Taxonomy ^[6]	7
Figure 4. Procedure FC comparison with BL, EL and TL ^[8]	7
Figure 5. Independent sample t-test for pre-test and post-test between the groups based on the achievement level ^[8]	8
Figure 6. Students’ mean grades in organic chemistry in control and experimental groups ^[9]	8
Figure 7. Comparison of quiz and exam questions grouped by topic ^[11]	9
Figure 8. Outline of the differences between both methodologies ^[12]	10
Figure 9. Results of the engineering laboratories and metrics ^[12]	10
Figure 10. Exam performance as a function of condition (control vs SERT) for high and low knowledge students ^[13]	11
Figure 11. Affective Learning Technology Research: State-of-art and future directions ^[17]	12
Figure 12. TAG system and Quinn’s facial expressions ^[19]	13
Figure 13 Quinn attribution messages and dimensions ^[19]	13
Figure 14. Three TAG condition: rTAG, eTAG and vTAG [1].....	14
Figure 15. Means and standard deviations for social perception and learning variables ^[1]	14
Figure 16. Categorized Behaviours ^[20]	15
Figure 17. Experimental flow chart and Post-test results ^[20]	15
Figure 18. Demonstration of the robot Nao simulating the writing of a word with its finger ^[21]	16
Figure 19. WoZ technique	16
Figure 20. Hands-Free Virtual Reality WoZ Test ^[23]	17
Figure 21. Hands-Free Virtual Reality WoZ Architecture ^[23]	18
Figure 22. Infrared grid system	18
Figure 23. Capacitive and resistive systems	19
Figure 24. Xperia Touch application	19
Figure 25. Cozmo	20
Figure 26. Sphero.....	21
Figure 27. Ozobot’s Evo	21
Figure 28. Jumping Sumo	21
Figure 29. Vortex.....	22
Figure 30. Kinect and Kinect v2.....	23
Figure 31. Intel RealSense D435	23
Figure 32. Orbbec Astra	23
Figure 33. Station Master	24
Figure 34. Colour Pong.....	25
Figure 35. Protoman Revenge	25
Figure 36. Protect Treasures.....	26
Figure 37. Cozmo	29
Figure 38. Kinect v2	29
Figure 39. Projector Acer	29
Figure 40. Brainstorming	32
Figure 41. Initial Interface Draft.....	34
Figure 42. Design Panel	36
Figure 43. Selecting Area.....	36
Figure 44. Customization Area.....	36
Figure 45. Interface Buttons	37
Figure 46. Menu Screen	37
Figure 47. Robot starting position	37
Figure 48. Restart Screen	37
Figure 49. System Architecture	38

Figure 50. Interface.....	39
Figure 51. Input Tool for LxT	40
Figure 52. Additional Buttons for LxT mode.....	40
Figure 53. Wizard of Oz interface	45
Figure 54. Wizard of Oz’s Flowchart	47
Figure 55. Touch Surface Architecture.....	48
Figure 56. Hand Detection Flowchart.....	48
Figure 57. RGB image example	49
Figure 58. Depth image example.....	49
Figure 59. Perspective Transformation	50
Figure 60. Example of ROI	51
Figure 61. Background Subtractor Procedure	51
Figure 62. Detection of the fingertip.....	52
Figure 63. Fingertip’s Threshold	52
Figure 64. Algorithm to discern click or slide.....	53
Figure 65. Cozmo configuration.....	54
Figure 66. Socket pattern	55
Figure 67. Depth Camera and Unity reference’s systems	56
Figure 68. Unity and robot reference systems	58
Figure 69. Depth camera and robot reference systems	59
Figure 70. Unity’s Socket Flowchart	61
Figure 71. Python’s Socket Flowchart.....	62
Figure 72. Deployment example.....	63
Figure 73. Domain Knowledge Cheatsheet.....	64
Figure 74. Experimental Procedure	66

List of Tables

Table 1. Techonologies related.....	67
Table 2. Learning preferences I.....	67
Table 3. Learning preferences II.....	67
Table 4. Evaluacion preference	67
Table 5. Learning expentancy	68
Table 7. Pre-Test I, general and laws concept	68
Table 8. Pre-Test II, Thèvenin concept	69
Table 9. Pre-Test Circuits.....	69
Table 10. User’s perception of the robot I.....	69
Table 11. User’s perception of the robot II	69
Table 12. User’s perception of the robot’s performance and inclusion level	70
Table 13. System Usability Scale (SUS).....	70
Table 14. Learning expectancy.....	70
Table 15. Post-Test	71
Table 16. Post-Test Circuits	71
Table 17. Post-Experience.....	71
Table 18. Open Questions	71
Table 19. Pre-experience significant differences	72
Table 20. Post-training significant differences	73
Table 21. Pre-Test results	73
Table 22. Post-Test results	73
Table 23. Observed learning gains.....	74
Table 24. Time distribution	74
Table 25. Open Questions Feedback.....	74

Chapter 1

Introduction

In this chapter, it is provided a brief preface covering the basic aspects regarding the present thesis such as the gaming, the motivation, and the scope of the project. Finally, a short organization of the document is shown going through all the chapters mentioning the content of them, which assists in the comprehension of the scope of the thesis.

1.1. Serious Games

The games are considered the principal leisure activity of kids and have a main role in their growth because through it you can stimulate and cultivate excellent progress in essential areas such as cognitive, psychomotor, and social-emotional skills, among other things. It usually has educational purposes which maximizes creativity, memory, attention, and focus capabilities, alongside with decision making in adverse situations. Therefore, the games assist in actively experimenting, transforming, and learning the reality and, so that, is treated as an effective method for understanding it.

Along with the preceding benefits, there are several interesting advantages worth mentioning that depending on the age and, as a result, on the game, since when we grow we seek different stimuli compared to our childhood. These skills and development can be divided into the following categories.

- **Social:** an old perspective of the games perceives them as an individual and meaningful pastime which promotes loneliness. However, nothing could be further from the truth in the current state of games, especially in the videogames. These types of activities are usually played with friends enhancing conversation and leadership skills so the team can work together, indirectly improving their real-life social groups when practicing pain, loss, joy, perseverance, and overcoming adversity inside the game. A clear example of this would be the eSports, where professional players are completely dedicated to compete cooperatively in tournaments. For that reason, social gaming creates an exceptional stimulating environment.
- **Cognitive:** playing games can bring enduring beneficial outcomes in such a large number of areas. A study has proved that gaming can improve the recognition ability and spatial memory to a certain point, as well as the reaction time and mental flexibility and processing, which could positively affect the player capabilities when it comes to working under pressure, adapting to changes and discerning among important and irrelevant information.
- **Emotional:** the entertainment activities such as the games have been linked to effective manners of generating positive feelings and abating detrimental emotions like frustration, anxiety, and grief, increasing relaxation depending on the content. This empowerment can influence the player's real-life, feeling more confident with an overall more desirable mood and attitude improving the job or school behaviour. So, games could take advantage of failure as a motivation tool to face adversities with reinforced approaches, increasing the critical reasoning.
- **Physical:** aside from the mental benefits of gaming, it also contributes physically to the player in many aspects as some studies have proved. The games present therapeutic help

for certain illnesses along with pain relief, as well as improving the motor skills if it has interactive characteristics. A study has shown that playing first-person shooter games could improve the vision of people with cataracts.

1.2. Motivation

Given the wide variety of learning methods, going from habituation and imitation to active approaches, the need to develop and deepen new forms of acquiring knowledge more efficiently emerges in order to stimulate the users in a non-traditional way to avoid losing their interest.

Also, *“It is often said that the best way to learn something new is to teach it to someone else”*. From the 50’s theory of Dale’s “Cone of Experience”, lots of experiments were performed to test this commonplace assumption up to the point where an entire body of research nice is nowadays available. Thus, it is not surprising that *learning by teaching* (LxT) is prevalent among contemporary educational practices, and a relatively well-studied topic [4].

On the other hand, another emerging trend that has raised its popularity recently is the Mixed Reality (MR) experience, which consists in the development of applications in a mixed environment, combining digital content on the real world either superimposing the objects or interacting with the physical ones. Within this environment, physical elements can be included such as robots, either cooperatively or separately, using them as tools or as any type of complement. Despite of whatever the robot’s role is, the system mechanics not only have to prevent the robot from being unnoticed and unnecessary within the ground but also have to provide the robot an essential task. However, this does not mean that the robot is the centerpiece of the system, which can be, but needs to be far from being dispensable.

Consequently, the present project intends to take a step forward and combine these trends in an application that integrates the mechanisms and movements of robots in a MR environment which could let the user deepen his knowledge in a concrete topic under the LxT model, procuring an adequate and balanced form by using the required technology.

1.3. Scope of the Project

The scope of the present thesis is to study the design, dynamics, and interactions of a system that exploits the fact that one or more robots are present within the MR environment. It should also stimulate and engage the users and encourage physical movement. The present thesis explores what are the possibilities of learning, and how robots can improve the experience w.r.t. classic methods. It also explores what kinds of roles the robot can take in the system e.g. opponent, companion, tool, etc.

A MR robotic gaming platform is currently being developed on the Department of Control and Computer Engineering, at the Politecnico di Torino, Italy. It features a virtual playground that is projected on the floor, on which both human and robot players can move around and interact. Robot players are controlled by the game’s AI itself, which is implemented in Unity. Therefore, this project adds a new tool to the platform, which could be used for students in the next projects that require similar technology.

From the review of the state of the art, two observations can be made. First, although the LxT model has been extensively analyzed in different contexts and disciplines (even including robots in the experience) there is still a lack of studies conducted for university subjects such as Science, Technology, Engineering, and Mathematics (STEM) high-level education. Second, it has not been found any study that investigated this topic on a full MR environment, since the most similar project [1] did not create a real interaction with the projection. Therefore, this

project decided to take a step forward in this field of study and examine the impact of the LxT approach by combining the MR technology with a robot as the teachable or teacher agent respectively, developing the required tools to carry out the experiment. Then, the research hypotheses that guided this work are:

- Is there a significant difference in the user's learning between the TL and LxT models in a MR robotic environment?
- Is the developed system able to effectively support the user's learning in both modes?

To conclude, a critical analysis will be carried out on the work as a whole and the conclusions drawn, as any other investigation project. This final assessment should be as objective as possible and will collect all the impressions that have arisen during the work, highlighting both the problems and the achievements of this proposal.

1.4. Organization of the Document

In this section, the organization of the document is described with the objective of exposing a general notion about the content of each chapter. The structure is divided into seven chapters and it is detailed as follows:

- 1) Introduction: it is shown a brief preamble of the project, where it is discussed its motivation, its context, and the final target scope.
- 2) State of the Art: it has been done detailed research about the previous works of the same field of study in order to understand the current state of the MR and to advance on a concrete direction based on it.
- 3) Technology: it is shown all the tools involved in the project, briefly describing the corresponding software and hardware capabilities.
- 4) System Design: it is illustrated all the phases of the proposed system design process from the brainstorming taking into account the system requisites and recommendations to the final conception and rules, describing all the required elements that are involved in the system mechanics and the way they interact with each other.
- 5) System Implementation: it is shown all the phases of the implementation of the present project covering from the system mechanics' modeling and the logic developed in Unity to the deployment process on the target device, including how the integration of each part has been done.
- 6) Experimental Evaluation: it has been done an experimental evaluation in order to assess the value and effectiveness of the project. In this case, a preliminary user study was carried on to collect feedback and analyze their impressions about the implemented training system by completing a posterior standard questionnaire.
- 7) Conclusions: it has been done a global critical assessment about the project from different perspectives highlighting the objectives met and the main results obtained, as well as explaining the major problems encountered and the possible future lines.

Chapter 2

State of the Art

In this chapter, previous works of other authors are approached with the objective of understanding the current state of the present technology. This research is considered essential to properly comprehend the existing limitations and, consequently, appropriately orientate the scope of the project in a specific direction with the aim of advancing in the field of automated MR robotic learning systems.

2.1. Mixed Reality

Within the real environment, the following blending levels of digital content can be distinguished.

- Virtual Reality (VR): it is a fully immersive technology that deceives the user's senses transporting it to a different environment aside from the real world. With the use of a headset, the user can experience imaginary conditions generated by a computer, where he can interact with the objects and complete distinct
- Augmented Reality (AR): superimposes digital information on the real world, amplifying the user's vision of his environment with new layers of digital objects. However, digital and real world cannot interact with each other.
- Mixed Reality (MR): brings together digital content and the real environment similar to the AR but, in this case, the user can interact with both digital and physical objects and surroundings.

The hardware requirements increase with the level of immersion and, therefore, the virtual reality applications need specialized VR headsets and technology to track spaces, boundaries and motion, while the AR can be run on less demanding devices such as smartphones.

Moreover, the MR has the ability to interact with the digital images and data, being able to move and adjust them using different command prompts such as hand gestures, while the AR just superimposes the digital content over the real world. However, in VR there is almost no overlap with reality since the user is transported to a computer-generated world.

On the other hand, there is an existing trend to exploit the full potential of the robots redesigning their initial purposes for completely different applications, growing the number of robotic systems that include Artificial Intelligence (AI), resulting in a more flexible and autonomous system. One of these new applications lies in introducing robots into the gaming scenario bringing a new characteristic to them, the Human-Robot Interaction (HRI). However, this paradigm usually creates poor experiences due to actually designing minimal interaction between both sides.

Then, from this needs arises the so-called Physically Interactive RoboGames (PIRGs) which is defined by the authors of [2] as *“a number of autonomous agents (including software, hardware, and physical agents) of which at least one is an autonomous robot, and one is a person. These agents interact with each other in a possibly variable and unknown environment, by following some game rules, so that the human players can have fun”*. Furthermore, another

branch regarding the games focuses on introducing digital content into the gaming environment, a paradigm called phygital play.

For that reason, the natural step forward consists of combining both procedures resulting in incorporating robots into a mixed environment where each agent, the robot, the user, and the digital and physical content, can interact with each other creating a more engaging experience. This trend is called Mixed Reality Robotic Gaming (MRRG) [3].

2.2. Learning

Learning is defined as “*the process of acquiring new, or modifying existing knowledge, behaviours, skills, values or preferences*”. The ability to learn is inherent to humans, animals, certain machines, and some studies shown evidence about plants. Moreover, the learning may occur as a result of classical or operant conditioning, habituation or other activities such as play. In fact, certain theorists have approached the play as the first form of learning, since the children learn to interact with the environment by playing educational games, which is considered pivotal in their development. There are numerous models to approach the learning of new knowledge, of which it is proceeded to describe the main and most expanded styles.

The first model lies in the habituation to a concrete experience, which occurs when the agent is continuously exposed to a stimulus and, consequently, the innate response to it diminishes, resulting in non-associative learning.

It is also quite spread the associative learning, where the subject learns through an association between two events or stimuli. In this case, the learning lies in that a behaviour that is punished or reinforced in the presence of an event is more or less likely to occur again when that event is repeated.

There is also active learning, when the agent takes control of his own learning experience, monitoring their own paths and, therefore having control not only about how they learn but also about what. This is characteristic to student-centered learning, conversely to what occurs in traditional education, where the teacher takes the main role while the students are the passive agents, having no control over what they learn.

Regarding the robotic field, several studies have studied the possible influence the robots could have during the learning process of children, is essential that the subject of the lesson is adequate for the capabilities of the robot.

Authors of [4] explored a group learning scenario using an autonomous robot, which is illustrated in Figure 1, researching the capabilities of an emphatic robot to foster collaborative learning and to sustain positive educational outcomes with groups of students. The comparison carried out, in this case, contrasted the learning conditions in children: with an empathic robot, with a robot and without it, resulting in the confirmation of the hypothesis that a robot with emphatic behaviors fosters meaningful discussions about sustainability, which is a learning outcome in sustainability education. It also studied the introduction of this robot into a classroom for two months, not providing any significant evidence or correlation regarding the learning gains.



Figure 1. Group Learning with Autonomous Robot^[4]

Within this trend of human-robot interaction, the project developed in [5] researched the potential benefits of personalizing a robot's actions to an individual's strengths and weaknesses. In this study, 80 participants, aged between 18 and 40, were asked to complete nonogram logic puzzles with the help of a personalized or non-personalized robot tutor comparing the solving times between four one-on-one robot tutoring conditions. An experiment is shown in Figure 2 where a participant is solving a puzzle in the computer while the robot Keepon analyses the moves and delivers brief gameplay strategies and advice. The user and tutor interface during the gameplay is also used.

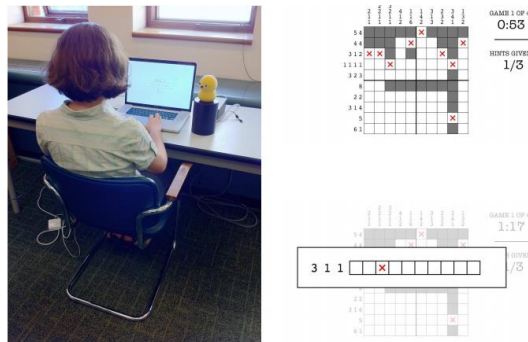


Figure 2. Experiment with robot Keepon solving a nonogram [5]

Their results showed a “one-sigma” improvement (68th percentile) in post-tests for participants who received personalized lessons, evidencing that relatively simple personalization can yield huge benefits in assistive-robot interactions or educational topics.

Moreover, the developed thesis in [6] explored the hypothesis if the augmented reality could improve communication of a social robot's intentions when interacting with children. It concluded the potential of AR to teach different behaviours, improving the intended communication of a robot, but also the robot movements were enough clear in some cases.

On the other hand, the flipped classroom reverses the traditional lecture-based classes supplying educational content outside the classroom. In this case, the student takes an active role while the teacher becomes a facilitator that supports him with feedback in a less didactic way but also more personalized. Therefore, the main part of the learning process happens during the professor's absence, which usually results in the exploration of topics in greater depths since the lecture information is not disseminated by him. From this methodology arises the blended learning[7], another option within the spectrum of learning models, an educational approach based on the combination of online interactions and materials with traditional methods placed in classroom lectures by the professor.

Motivated by the rapid advancement in technology, researches have been exploring numerous strategies and methods to engage and motivate the students during the learning process taking

into account the standard creative process and Bloom's Taxonomy. The creative process consists of a four-step procedure which includes the phases of preparation, incubation, illumination, and verification. It covers the process from gathering information needed to overcome a task through research to refining the proposed idea of studying its viability.

On the other hand, the Bloom's hierarchical classification, illustrated in Figure 3, was proposed in 1956 by Benjamin Bloom and sets the desired objectives and skills for students learning a new topic, explaining its process[8]. It is represented with a pyramid structure to reinforce that each category is built on a foundation of the previous levels. However, not every new concept requires going through the entire taxonomy or starting from the base level and, therefore, each task should be identified and aligned with a concrete level and continue escalating.

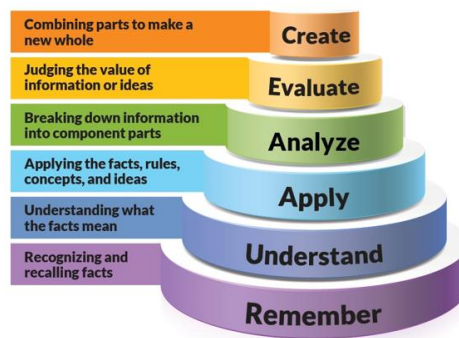


Figure 3. Bloom's Taxonomy [8]

The authors of the research in [9] from the Can Tho University (Vietnam) examined the motivation, flexibility, performance and self-efficacy beliefs perceived on four different settings: Flipped Classroom (FC), Blended Learning (BL), E-Learning(EL) and Traditional Learning(TL). The sample was formed by 90 undergraduate students enrolled in an Invertebrates Course and was randomly spread on the target settings. The methodology followed is illustrated in Figure 4.

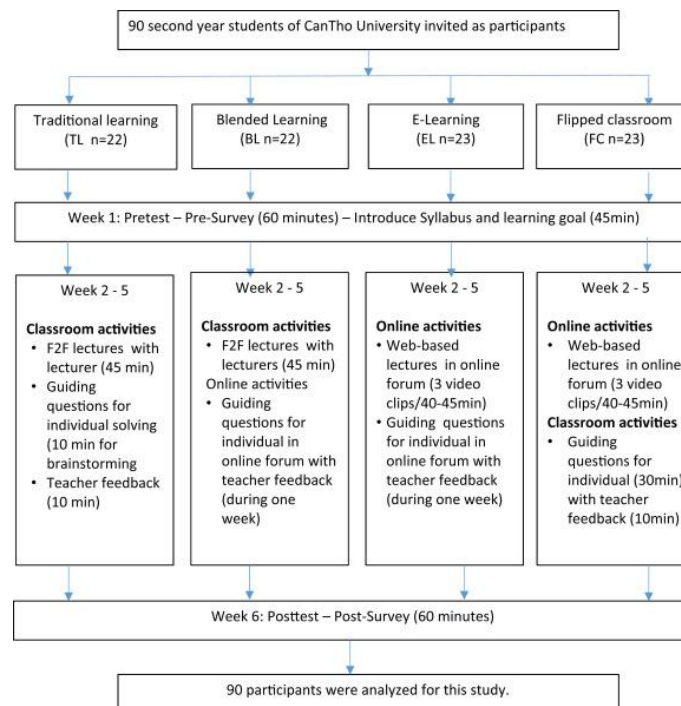


Figure 4. Procedure FC comparison with BL, EL and TL [10]

The analyses performed showed superior learning in the FC compared to the rest, Cohens' d equal to 1.58 (TL), 1.01 (EL), and 0.71 (BL), being the BL also superior to EL. Regarding the qualitative results, it was observed higher levels of motivation and self-efficacy beliefs, but no relation with the parameter of perceived flexibility. As a result, these findings were interpreted as the FC environment could be a promising context of enriching students' learning performance.

Moreover, the authors of this study[10] aimed to assess the student's learning achievement levels and motivation, creating a flipped classroom environment with the objective of teaching mathematics concepts. These parameters were measured by Mathematics Achievement Test (MAT) and Course Interest Survey (CIS), employing a pre-test post-test quasi-experimental design. The 82 high-school participants were divided into two groups, which were taught trigonometry using the flipped classroom (experimental) and traditional (control) teaching models, respectively. The results for the achievement level parameter are shown in Figure 5.

Achievement level	Group	Group	Mean	SD	MD	t-value	p
Pre-test	High	Experimental	13.11	1.45	0.54	1.62	0.10
		Control	12.57	1.55			
	Average	Experimental	10.12	2.15	1.07	2.96*	
		Control	9.05	0.84			
	Low	Experimental	5.25	2.12	-0.50	1.26	
		Control	5.75	1.38			
Posttest	High	Experimental	10	2.54	0.13	0.25	0.80
		Control	9.87	2.03			
	Average	Experimental	10.75	2.40	1.81	3.93*	
		Control	8.94	1.71			
	Low	Experimental	9.18	2.71	1.56	2.83*	
		Control	7.62	2.26			

Note. * $p < .05$. SD = Standard deviation, MD = mean difference.

Figure 5. Independent sample t-test for pre-test and post-test between the groups based on the achievement level [10]

The results demonstrated their initial hypothesis since a significant difference in motivation and learning achievement was observed in the group taught by the flipped classroom model, which is especially noticeable in the average achievers during the pre-test and the average and low achievers during the post-test.

In [11] it was assessed the improvements in final grade scores in the subject organic chemistry at a postsecondary level, motivated by the lack of involvement created by the traditional way of teaching in such a difficult course for students and the lack of research on this topic. Then, the premise of the work relied on the possible performance improvement induced by active pedagogy, in this case, flipped classroom. The sample was composed of two groups, the control and experimental with 66 and 151 people, respectively, and the participants were stratified depending on their ability in college, from low to high achieving. The quantitative results are shown in Figure 6.

	Entire sample		Low achievers		Average students		High achievers	
	N	Mean grade (SD)	N	Mean grade (SD)	N	Mean grade (SD)	N	Mean grade (SD)
Traditional classroom	66	72.9 (14.7)	21	59.8 (11.0)	18	68.5 (10.6)	27	86.0 (6.6)
Flipped classroom	151	77.1 (11.2)	51	69.8 (10.4)	55	75.6 (8.6)	45	87.1 (7.1)
t Score	2.053*		3.663**		2.868*		0.651	
Effect size	Small effect (Cohen's $d = 0.32$)		Large effect (Cohen's $d = 0.94$)		Moderate effect (Cohen's $d = 0.73$)		No significant difference	

*Statistically significant ($p < 0.05$).

**Statistically significant ($p < 0.005$).

Figure 6. Students' mean grades in organic chemistry in control and experimental groups [11]

These results showed that the experimental students, those attending the flipped group, obtained significantly higher final grades in organic chemistry, especially for the low and average students, while the high achievers presented similar grades. Therefore, this study illustrated the higher impact of the flipped model for students with learning difficulties

A similar project [12] regarding chemistry topics examined the impact of minimizing the seat time of traditional lectures reducing its length by two-thirds and employing that time in an active learning environment. Therefore, related content was available online. Moreover, BL and FC settings were introduced as a replica to measure standard parameters such as subject-matter learning, using multiple-choice test, and students' perceptions, using a certified survey instrument. The sample utilized was formed by 1100 chemical students across three semesters.

As a result, their findings demonstrated that the students could achieve similar or significantly better learning outcomes using an active environment and, at the same time, improve their learning perceptions. Therefore, that study suggested a more efficient use of physical space even though fewer students are seat per square foot.

The study [13] took place during a Control System course in the Department of Mechanical Engineering at Seattle University. This study compared the effectiveness of an inverted classroom to a TL model in three areas: content coverage, student performance and student's perception of the flipped format. The methodology followed consisted of evaluating the course in two successive years, one for each learning model, using the same conditions such as duration, time-slot, professor, assignments and topics. Both years, the 20 senior mechanical students attended the course and performed weekly 15-min quizzes, 50-min midterm exam, and 110-min final, having a similar number, type and difficulty of questions in both cases. This project obtained the results shown in Figure 7.

Problem	TC Mean (SD)	IC Mean (SD)	P value
<i>Ladder Logic (1 problem)</i>	8.20 (2.21)	8.15 (2.01)	0.941
<i>Open Loop (5 problems)</i>	8.34(0.70)	9.09 (0.71)	0.002*
Open Loop 1	8.25 (1.37)	8.75 (1.71)	0.315
Open Loop 2	8.90 (0.97)	9.30 (0.73)	0.149
Open Loop 3	7.13 (1.73)	8.65 (0.93)	0.001*
Open Loop 4	9.40 (1.01)	9.07 (1.07)	0.318
Open Loop 5	8.03 (2.20)	9.70 (6.30)	0.004*
<i>Root Locus (2 problems)</i>	9.04 (0.78)	9.30 (0.70)	0.277
Root Locus 1	9.38 (0.57)	9.83 (0.42)	0.007*
Root Locus 2	8.70 (1.59)	8.77 (1.36)	0.887
<i>Root Locus Design (3 problems)</i>	7.79 (0.96)	8.79 (0.97)	0.003*
Root Locus Design 1	9.00(0.9)	9.00 (1.38)	1.000
Root Locus Design 2	7.88 (1.11)	7.87 (1.26)	0.986
Root Locus Design 3	6.50 (2.01)	9.46 (1.27)	<0.001*
<i>Algebraic Design (1 problem)</i>	9.55 (1.00)	9.25 (0.85)	0.313
<i>Bode-Based Design (3 problems)</i>	6.72 (0.94)	8.25 (1.30)	<0.001*
Bode-Based Design 1	5.60 (1.85)	8.45 (1.61)	<0.001*
Bode-Based Design 2	5.57 (1.74)	7.50 (2.06)	0.006*
Bode-Based Design 3	8.80 (1.44)	8.80 (1.47)	1.000
<i>Nyquist Stability (2 problems)</i>	7.68 (1.41)	7.52 (1.94)	0.770
Nyquist Stability 1	8.00 (1.81)	7.33 (2.56)	0.347
Nyquist Stability 2	7.35 (2.21)	7.70 (2.03)	0.605
<i>Design (Root Locus, Algebraic Design, Bode Based)</i>	7.58 (0.75)	8.61 (0.98)	0.001*

* Statistically significant results, $p < 0.05$

Figure 7. Comparison of quiz and exam questions grouped by topic^[13]

This study illustrated how the inverted model allowed to cover more material that was what initially expected and slightly better performance on a quiz and open-ended design questions, as well as a quick adaptation to the model after initial minor struggles. The participants also qualitatively described the format as satisfactory and effective.

The authors of [14], from the Department of Physics at the Polytechnic University of Valencia, described the effect of the flipped methodology on the students' performance when applied to laboratory practice in two subjects, physics and electricity, of a technical degree. The experiment sample was formed by 1233 students enrolled from 2013 to 2017. In this case, the laboratory and final grades were compared during four consecutive years on these subjects. Every course operated with the same characteristics, applying the traditional way to the first two years while the last two were taught inverted.

The methodology employed is shown in Figure 8, whose boxes in the centre represented points in common for both cases. As it can be seen, the main changes rely on the addition of videos to the materials available before the lab session and that the teacher's explanation was replaced by the student's preparation at home. Consequently, the students had more time during the lab to discuss and organize the session.

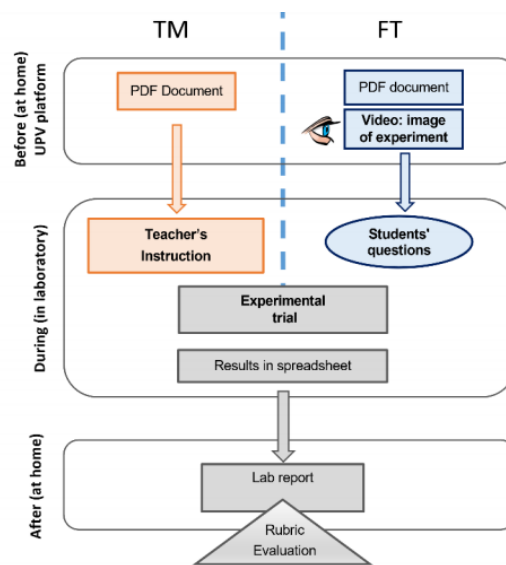


Figure 8. Outline of the differences between both methodologies [14]

Also, the sample was stratified in high and low performance levels with respect to the median of the course. The data obtained is illustrated in Figure 9.

	Average final grade	Standard deviation	Average laboratory grade	Standard deviation	t	p-value
PhysTM (2013/14)	7.17	1.31	7.06	1.31	-1.36	0.18
PhysTM (2014/15)	6.92	1.53	7.32	1.92		
PhysFT (2015/16)	7.20	1.28	7.50	1.58	0.18	0.86
PhysFT (2016/17)	7.44	1.19	7.47	1.21		
ElecTM (2013/14)	5.51	1.46	6.79	1.49	-1.28	0.20
ElecTM (2014/15)	5.69	1.74	7.02	1.67		
ElecFT (2015/16)	6.13	1.46	7.17	1.53	-1.07	0.29
ElecFT (2016/17)	6.27	1.29	7.36	1.50		

Figure 9. Results of the engineering laboratories and metrics [14]

The analysis showed higher academic results during the years operating under the flipped model, with a difference statistically noticeable, since both courses presented significant variations, as well as globally between both methodologies. Moreover, it also demonstrated how the academic improvement does not affect all participants equally since it affected to a greater extent the physics' lower level and the high level in electricity. This discrepancy was interpreted with the difficulty level of both subjects, due to the high level in electricity did not

saturate, allowing room for improvement when changing the methodology, which did not occur with the physics related topics.

In [15] it was demonstrated the benefits of the Self-Explanation Reading Training (SERT) to college performance. The sample of this experiment was composed of 265 participants enrolled in an introductory Biology course, of which 59 were set with SERT settings. The final scores and corresponding statistical analysis are shown in Figure 10, where the sample is stratified by group and science knowledge.

	Control		SERT		Statistics	
	Mean	Se	Mean	Se	ANOVA	Cohen's <i>d</i>
All participants						
Exam 1	.75	.008	.80	.016	$F(1,262) = 7.69, MSe = .014, p = .006$.47
Exam 2	.74	.009	.77	.016	$F < 2$.25
Exam 3	.71	.008	.72	.015	$F < 1$.10
High knowledge						
Exam 1	.79	.011	.82	.024	$F(1,139) = 2.28, MSe = .014, p = .142$.34
Exam 2	.77	.011	.79	.023	$F < 1$.12
Exam 3	.75	.009	.74	.020	$F < 1$	-.09
Low knowledge						
Exam 1	.71	.012	.79	.020	$F(1,119) = 12.55, MSe = .012, p < .001$.74
Exam 2	.70	.013	.76	.022	$F(1,119) = 5.77, MSe = .015, p = .018$.50
Exam 3	.67	.011	.71	.019	$F(1,119) = 4.14, MSe = .011, p = .044$.43

Figure 10. Exam performance as a function of condition (control vs SERT) for high and low knowledge students [15]

As can be seen, it was concluded that the SERT was more beneficial to students with less knowledge, while the high achievers were barely influenced. Moreover, it was also observed how low achiever students could perform very similar to high achievers with the SERT approach, which did not occur within the control group, where the difference was significant. Consequently, self-explain and comprehension strategies could be promising methods in order to overcome knowledge deficits, assisting students to effectively understand and process topics and, in this case, science.

Finally, it is also a well-known trend the learning by teaching model (*LxT*), which was designed by the German teacher Jean-Pol Martin at the beginning of the '80s. It grounds on the naïve practice of the peer-tutoring, in which students tutor other students by teaching them self-learned domain knowledge from traditional (or not) sources. From peer-tutoring, a lot of diversified pedagogical models were branched out (e.g., cross-age tutoring, reciprocal tutoring, cooperative learning environments), including the so-called teachable agents (TA), which are usually computer agents built to be taught, based on the pedagogical concept of *LxT*.

During the tutoring process, where students take on the role of the tutor to teach a computer agent PUPIL (tutee), learners have been observed to gain a deeper understanding of the subject matter. TA can be of various kinds, ranging from an interactive application (a piece of software) to, of course, robots. It is worth to mention that TA are referenced in literature with 2 main branches, the first was the final goal is to teach the agents (in the sense of programming it) using ML approaches, and a second (the one of true *LxT*) were the final aim is to let the user gain a better understating of the topic through the process of teaching to someone else. Several specializations of TA can be found, such as Affective TA and Persuasive TA.

In the *LxT* model, there are few [16] key steps that the student must undergo and were proved to be effective.

- Preparing to teach (expectation)
- Explaining to others/PUPIL (teaching)
- Interacting with others/PUPIL (Q&A, feedback to PUPIL)
- Observing the Pupil spending the acquired knowledge (Recursive Feedback)[17]

2.2.1. Robotic Teachable Agent

Regarding Robotic TA (RTA) there are a fairly number of studies on the topic, and some of them are mere “call to action”. For example, the authors of [18] from the Harvard University claimed a new direction for the intelligent autonomous systems community is education, developing theory and practice for embodied learning agents that could be taught by students to gain personal benefit, combining the established advantages of LxT and robots as education and social tools. Moreover, in [19] it was highlighted three emerging areas of research within this field of study, which are briefly described in Figure 11, as well as suggestions for future learning models and work environments. Therefore, it claimed the importance of the affect’s role within both designing and work environments, where immersive and intelligent technology understand and use the emotion in its partnership with the user. Such systems should be able to successfully communicate with nonverbal behaviours, gesture, and emotional expressions.

Research area	Current state-of-the-art	Future directions
Human-like technology (<i>Simulating</i> affect)	artificially intelligent virtual agents; social robots	Increasing realism, rapport, and social responsiveness; systems emoting to inform learner of intention and generating interest and buy-in
Affect-aware systems (<i>Sensing</i> learner affect)	affective intelligent tutoring systems; dashboards with multimodal data; MOOCs that assess learner engagement	Generating more meaningful learning-related information from multi-channel multimodal data; role of emotion AI and learning; more sensitive detection and modeling of learner affect for learning at scale
Affective learning outcomes (<i>Supporting</i> learner affect)	Support flow/engagement; cognitive disequilibrium as integral to the learning process	Discovering new strategies to elicit emotions such as empathy, delight, curiosity; modeling the complex relationship with learning and cognition; deeper investigation of the affordances VR

Figure 11. Affective Learning Technology Research: State-of-art and future directions [19]

Within the psychology domain, the authors of [20] from the Autonomous University of Barcelona reviewed a body of relevant research in order to gather evidence about the potential and limits of the LxT pedagogical mechanism. It also created an explanatory framework about this approach, promoting these kind of practices and providing opportunities for students, such as peer tutoring, peer assessment or cooperative learning. It was also concluded that the more complex the teaching activity is, the more opportunities there are to learn by teaching. In addition, it highlighted the importance of teaching skills for future professional needs due to the fact the continuous learning is highly valued in the current society.

Regarding experimental research about LxT a few projects are worth mentioning. First, in [21] it was examined the influence of causal attributions made by a teachable agent, on 19 participants from 5th and 6th grade that had little exposure and knowledge to the target topic, geometry. The system used was called *Tangible Activities for Geometry* (TAG) which aimed to help students learn about geometry providing a projected space where the participants could move in and solve problems while concurrently interacting with a robotic agent called Quinn. The main components are depicted in Figure 12, which included the robot, the problem space projected on the floor, a hanging pointer to interact with it and the corresponding mobile interface. It must be noted that Quinn consisted of a LEGO Mindstorms robot with an iPod mounted on top of it representing its face.

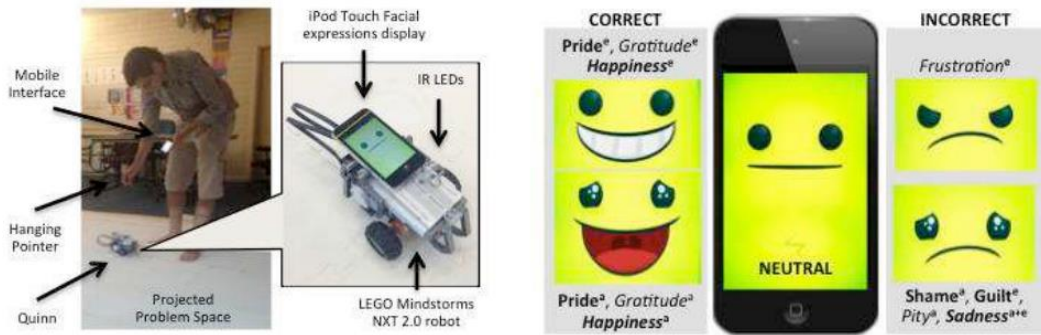


Figure 12. TAG system and Quinn's facial expressions [21]

Also, the robot's social behaviours are based on attribution theory to ability & effort, as well as to the student, itself, or both. Therefore, they are generated after in response to TAG's feedback, emulating its reaction to whether it got the correct answer, displaying emotion on the iPod and sending a short spoken message. Several samples of the investigated dimensions are illustrated in Figure 13.

<i>Outcome</i>	<i>Cause (Agent/Source)</i>	<i>Quinn's Message</i>
correct	I / ability	That was right. Oh man, I am smart at math.
correct	you / ability	Yay! I got that right because you are a good teacher.
correct	we / ability	That was correct! My gosh, We are good at this.
incorrect	I / effort	Oh boy. I got that wrong because I did not try hard to learn.
incorrect	you / effort	You did not put in much effort into teaching me that problem.
incorrect	we /effort	Dang it, that was wrong. We did not work hard to solve that problem.

Figure 13 Quinn attribution messages and dimensions [21]

The procedure carried out by participants consisted in reading a domain cheat sheet (~5 min), filling questionnaires (~15 min), being trained on how to use TAG (~20 min), using TAG to teach the robot (~45 min) and finally participating in a semi-structured interview (~20 min). As conclusion, for incorrect outcomes the analysis suggested that attributions to effort are safe for a social robot to express, maintaining positive student affect and motivation to teach, since the we dimension got the most enthusiastic responses, associating failure as a collective sign taking part of the responsibility. However, students showed frustration and irritation when the robot attributed its failure to them with the messages. In contrast, regarding the correct outcomes, all dimensions were positively received (with different degrees), increasing believability and students' bonds with the agent.

Following the previous study, the same authors carried out a subsequent project [1] studied the impact of the feedback mood and tangible/embodiment on the learning performance and experience pleasure. The investigated hypotheses relied on, firstly, if robots may better engage students in the learning activity and, secondly, if students that take embodied actions to teach the robot may develop deeper knowledge. The sample utilized was formed by 37 4th-6th grade, and the rTAG system was compared to the other two conditions, where the students used embodied action to teach a virtual agent (eTAG) or teach a virtual agent on a computer (vTAG). These configurations are illustrated in Figure 14.

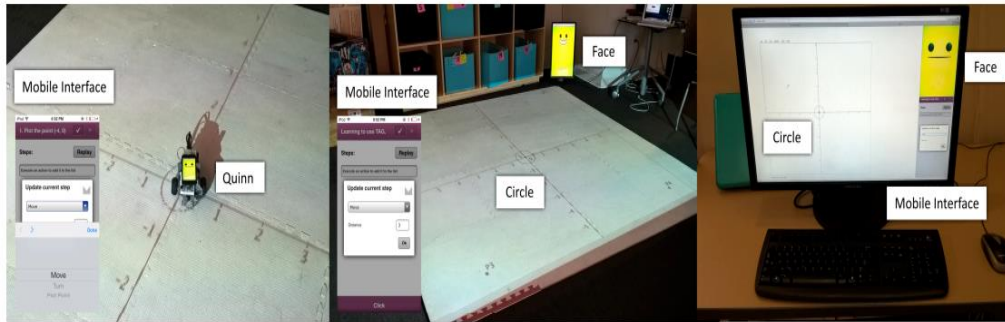


Figure 14. Three TAG condition: rTAG, eTAG and vTAG ^[1]

The followed procedure was similar to their prior research, consisting of a pre-test, study of the cheat sheet, a questionnaire about attributional style, system training, experiment to solve exercises, a post-test and self-report tests and experience perception, lasting approximately 2 hours per individual. The corresponding results analysed are shown in Figure 15, which shows means and standard deviations (in parenthesis) for social perception and learning variables. Pre-test and post-test scores are reported as percentages, while the social scores are out of 5.

	Animacy	Likability	Intelligence	Trustworthiness	Pretest	Posttest
vTAG	3.92 (0.94)	4.46 (0.96)	4.19 (0.73)	4.33 (0.73)	38.8 (24.1)	57.0 (25.6)
eTAG	3.87 (0.76)	4.62 (0.36)	4.40 (0.55)	4.64 (0.39)	45.8 (32.0)	58.7 (30.1)
rTAG	4.53 (0.67)	4.66 (0.49)	4.56 (0.46)	4.53 (0.67)	40.3 (23.0)	55.9 (17.2)

Figure 15. Means and standard deviations for social perception and learning variables ^[1]

This study did not observe significant differences between conditions on learning, while there was evidence that perceptions impacted learning. They also admitted that this project was limited by the small sample size, preventing the generalizability of the results. Moreover, the short duration of the experiment (2 hours) could difficulty the interpretation of them, since it may not be enough time to let the players learn how to use the system and focus on the priority of it, the content and the problem-solving, losing the initial engagement after the first interactions.

Another experiment worth mentioning would be the project presented in [22] carried out in the University of Tsukuba (Japan), which researched the scenario where robots receive instructions and/or care from children. The main hypothesis examined was if a care-receiving robot could promote the learning of children using English verbs, sustained by the idea of *LxT* in contrast with the conventional teaching models where the teaching agents (including the robots) are designed to play the active role of teachers. The sample studied was formed by 18 Japanese children (3-6 years) from an English language school, and two main experiments were carried out.

The first research evaluated if a care-receiving robot could induce children's care-taking behaviours during a 6 days' trial. For that reason, the robot NAO was introduced to several games with the children in a classroom performing two different roles by the condition of error rate, giving always incorrect or correct answers, and the participants were equally split for each condition. The analysis was centred in behavioural actions of the children, whose categories and results are shown in Figure 16. As it can be seen, significantly more caretaking actions were observed with the robot error rate 100% configuration, which reinforced the feasible hypothesis of inducing caregiving behaviours.

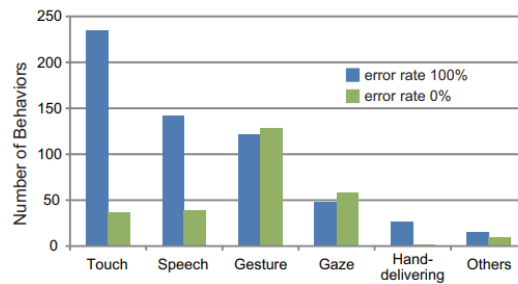


Figure 16. Categorized Behaviours [22]

The second one assessed if a care-receiving robot could promote children’s LxT. In this case, the participants performed a pre-test to verify their current knowledge, followed by a lesson period where a teacher is involved with the robot and the child, who will later abandon the classroom to let the child some free time with the robot. Finally, a post-test is performed to evaluate the children learning with the experience, whose results are illustrated in Figure 17.

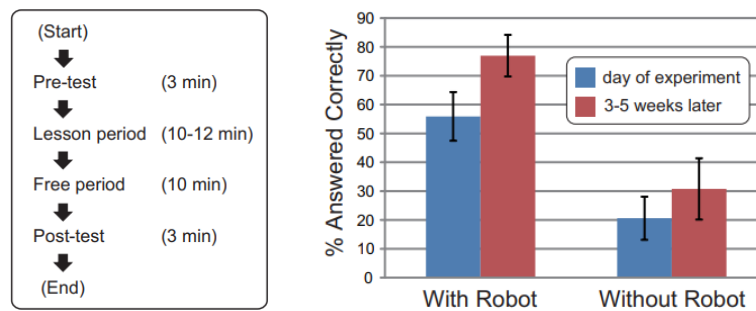


Figure 17. Experimental flow chart and Post-test results [22]

The analysis suggested an effective trend to increase the children's knowledge of English verbs when the robot was involved, since the participants picked up more correct graphic cards for verbs learned with it. In addition, it was also monitored their actions 3-5 weeks later conducting another set of post-tests following the same procedure, obtaining a higher percentage of questions answered correctly. This result was associated with the fact that after the successful experiment, the children continued playing the game at home.

In a similar trend, the authors of [23] from the École Polytechnique Fédérale de Lausanne also utilized the robot NAO to examine the physically-induced benefits encountered during human-led handwriting interventions, under the *LxT* model. This humanoid robot was configured as a partner, which purposefully drew deformed letters analysing a dataset of adult-written letter trajectories. The experiment carried out involved children aged 6-8, who successfully incorporated positive feedback with user demonstrations, allowing the robot to learn the optimal shape trajectories for each letter. An example of a robot writing task is shown in Figure 18. As a result, the nature of this project validated the observed engagement of the participants with the robot, improving its writing to a successful level.

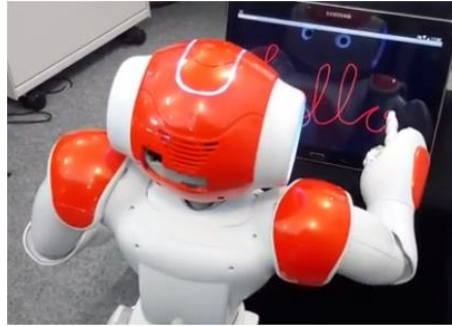


Figure 18. Demonstration of the robot Nao simulating the writing of a word with its finger^[23]

Then, it has been proven that the students are more predisposed to this type of learning one they have already tried it, sharing the same concerns about it: the lack of guidance and the need to rigorously teach yourself the contents, that could result in a larger amount of work and/or confusion on related topics. Consequently, there are several advantages attributed to the LxT approach since it has been proven that the experimental models are more efficient than the passive ones when it comes to learning new knowledge, since the experience is practised instead of being read or listened. These reviews of studies provided evidence that the LxT model has a great potential to enhance learning performance and academic gains among learners.

In conclusion, research evidence, on the use of the LxT approach in the teaching and learning of various disciplines, including statistics, chemistry, English, nursing, general engineering and pharmacy is available, and the usage of robots to assist the learning, but said research using MR, in university, appears to be limited. To fill this gap, the present study examined the effectiveness of the LxT model compared to the TL on electrical concepts in university students, using a robot and assisted by MR technology.

2.3. Wizard of Oz

Within the field of human-robot interaction, Wizard of Oz (WoZ) is a research exploration in which the subjects interact with a robotic system believing to be autonomous, but it is certainly being remotely conducted by an unseen person, also known as “wizard”. Then, with this configuration, the operator simulates the behaviour of a theoretical autonomous robot such as the movements, the gestures, the speech, etc. typically from a different room tracking all the communications and actions involved in the experiment in order to act accordingly in real-time. Consequently, it encourages natural interactions from the participant even though sometimes the experiments may require its prior knowledge about the wizard’s existence. Nevertheless, this technique includes distinct varieties within the autonomy spectrum, either from fully autonomous to fully tele-operated, just like a combined interaction. This technique is illustrated in Figure 19.

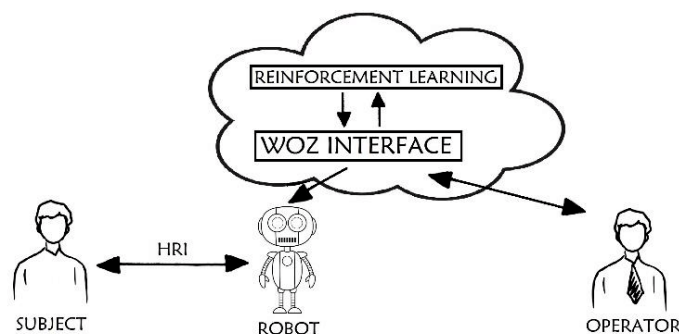


Figure 19. WoZ technique

The motivation that stimulates the use and advancement of this approach lies in the inability of current robots to fully autonomously interact with people in a socially and/or physically safe manner, despite a few exceptions. Therefore, it allows the testers to visualize and experiment how the future interaction could be like, or simply as a part of an iterative design process to improve specific aspects. For that reason, the experiments involved may employ WoZ in a repeatable manner in order to ease the transition to a more capable and autonomous future system.

On the one hand, the authors of the study in [24] presented a review of the HRI literature between 2001 – 2011 that involved the WoZ method, analysing a total of 54 papers. Regarding the control type, the majority of experiments were focused on natural language (72.2%) and non-verbal behavioural processing (48.1%), followed by navigation and mobility tasks (29.6%), with a low use for manipulation or mapping. It also contrasted the differences constraining what the wizards could recognize (11%) or produce on the robot (90.7%), being most of them (76%) humanoid and/or mechanical type, as well as measuring the wizard error (3.7%) and training (5.4%). Finally, it illustrated how the vast majority of experiments specified a priori the capabilities of the system to the testers providing them a clear scenario (94.4%), exposing the behavioural hypothesis for the user (79.6%) and for the robot (98.1%).

On the other hand, in 2018 the project presented in [25] from the Colorado School of Mines investigated the use of Virtual Reality technologies to provide the perspective of the robot to a WoZ tele-operator, motivated by the lack of experimental effectiveness and validity of current teleoperation methods such as joysticks, GUIs and sensors, regarding the naturalistic human-robot interaction. They proposed a WoZ teleoperation interface (Figure 20) combining technologies for hands-free robot control along with a VR display with the aim of providing the tele-operators an immersive VR experience with enhanced visual feedback.

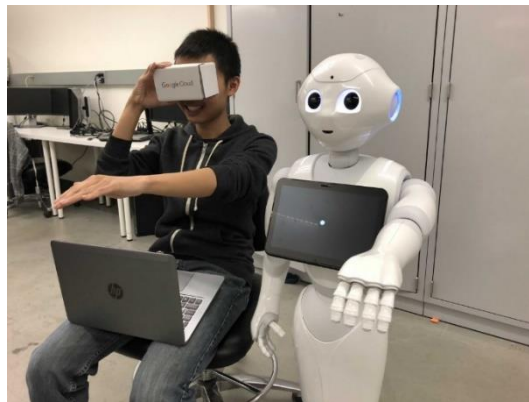


Figure 20. Hands-Free Virtual Reality WoZ Test [25]

The architecture of this project (Figure 21) consisted of the user directing interacting with a VR headset and a Leap Motion gesture sensor, which sent and received data from a humanoid robot running ROS on a standard Linux laptop.

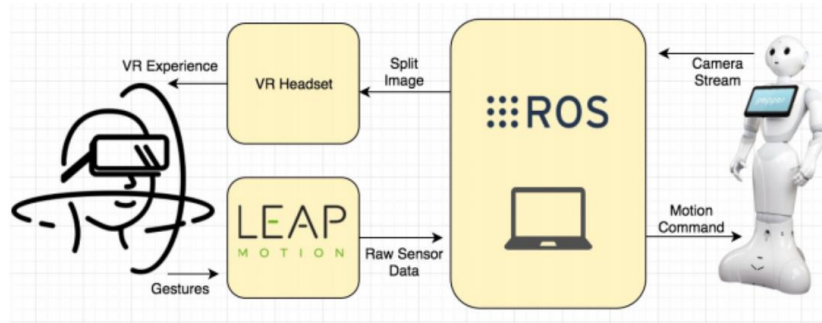


Figure 21. Hands-Free Virtual Reality WoZ Architecture [25]

2.4. Touch Surface

Nowadays, a touch surface is considered both input and output technology that lets the user control the processing of the information usually on an electronic device by using several touch gestures, such as clicks or slides, and by using one or more fingers or a special stylus or pen. Therefore, the touch screen allows the user to directly interact with the device.

Its versatility has increased the popularity of the touch user interfaces, integrating them into numerous product designs. It is quite common to see them incorporated into computers, game consoles, smartphones and point-of-sale systems.

Consequently, their accuracy plays a very important role in its functioning, since the users must be able to perform the desired actions precisely such as selecting a specific target, execute a smooth slide or simply press a button and, therefore, avoid selecting nearby elements or creating unstable situations. This also means that the interface design is really important because it must fairly indicate the system capabilities.

During the beginning of this technology, the first devices did make use of guidelines for the design, when the vast majority were inclined to incorporate infrared grids (Figure 22) to detect where the touch was made. This system is developed with Light-Emitting Diodes (LEDs) and several sensors, so when the user touches the surface the light is disrupted in this area, being detected by the sensors and indicating a touch command.

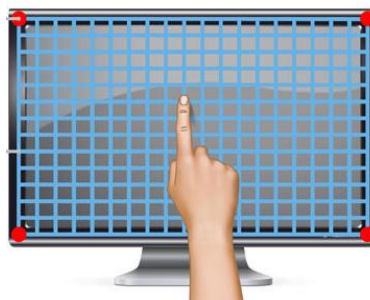


Figure 22. Infrared grid system

However, the modern consensus focus on the utilization of capacitive or resistive methods for sensing touch, which are illustrated in Figure 23. The capacitive systems apply an electrostatic charge to the top layer and when the user touches the area he absorbs part of this current, detecting the touch command. From tablets to smartphones, along with most smart devices and applications are powered by this technology.

On the other hand, resistive systems comprise several thin layers creating voltage dividers when the outer layer is pressed down, since two layers get in touch. This technology is implemented

in factories, restaurants and hospitals because of its high tolerance to liquids and tolerance, and is included, for example, by Nintendo in the DS family.

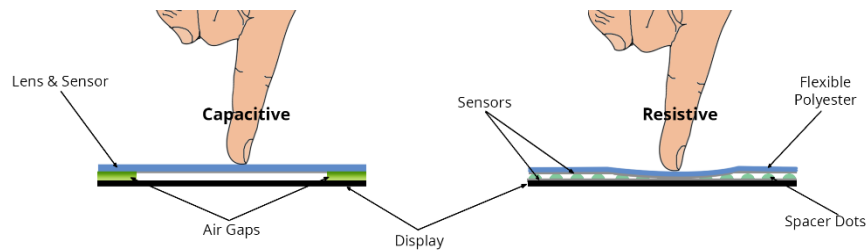


Figure 23. Capacitive and resistive systems

While the infrared and resistive systems can be used with gloves, the capacitive cannot since it must be touched with the body to conduct the electricity and be effective. However, the systems powered by capacitive technology present a superior level of accuracy and, for that, reason, they are the most widespread except for applications that need other requirements and the precision level is not the main priority such as point of sales, since it is also the most expensive among them.

Regarding the educational field, touch screens have been incorporated in classrooms and college campuses through the use of interactive projectors. This kind of devices allows the teacher to interact with the computer screen or to use it as a digital blackboard, more adequate and effective for teaching than the conventional one.

A similar example of this technology would be the Xperia Projector, also called as Xperia Touch and launched by Sony on October 2017, which is depicted in Figure 24. It is a device capable of projecting a touch screen among 23 and 80 inches on any surface by using infrared sensors. It offers an Android experience without the need for additional electronic devices.

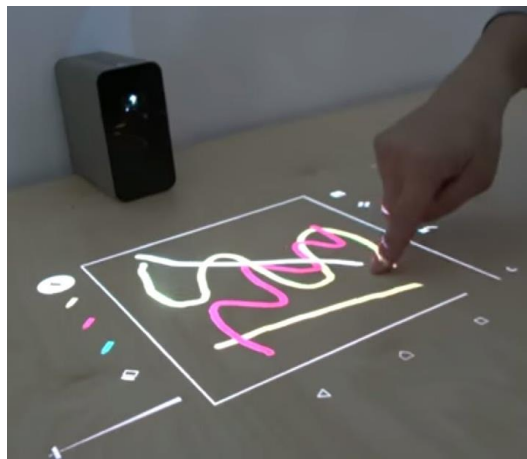


Figure 24. Xperia Touch application

Regarding its implementation, several projects have been studied. This part could be accomplished with two diverse approaches, detecting the finger's position with the RGB images to be later matched with the depth camera to obtain the distance of that pixel or using only the depth camera. Even though the most robust way to detect a hand is indeed using the first approach due to the information provided by the RGB camera, numerous studies work only with the depth sensor, due to processing a faster algorithm and the difficulties to precisely match both sensors because of the size and reference systems dissimilarities.

This project [26] demonstrates how a depth-sensing camera may be used to detect a touch on an-instrumented surface, while this one[27] proposed a method of regression-based touch detection in order to overcome the shortcomings of the Kinect and its inability to precisely detect certain coordinates or to reduce the noise. Finally, [28] sets a high-level procedure for making a touch screen out of a projector.

Even though each of them approached the fingertip's detection in a different way, all of them utilized only the depth sensor of the camera and followed the same kind of steps, consisting in a previous calibration, a hand's segmentation procedure and finally applying a threshold to the fingertip position compared to a static surface or background. Therefore, these steps will be also applied to the present project.

2.5. Robots in the Market

In this section is shown a market study in order to gather information regarding the current state of gaming robots. This research envisions the characteristics and capacities of the current robots, which could influence the design not only allowing to exploit their capabilities but also putting certain constraints in some cases. Consequently, the robot must be suitable for the system and vice versa. The most used robots are shown subsequently.

2.5.1. Cozmo

This small robot is programmed with an emotion engine, which can react to different circumstances with a wide range of facial expressions and body language imitating the human behaviour. It is empowered with a motion encoder, a face screen, a speaker, an input camera for recognition purposes and a parallel forklift that can raise small objects. The robot is illustrated in Figure 25.



Figure 25. Cozmo

Cozmo is considered a game playing machine because it counts with numerous educational applications and games for kids based on skill and speed with its power cubes. The user has also access to its core functionality and can generate his own content learning very basic coding skills.

2.5.2. Sphero

This spherical robot (Figure 26), with a 73mm diameter and 181g, is capable of rolling on the floor controlled by a smart device. It counts with a motion encoder and an IMU and, therefore, it is sometimes used as a controller for other platform games. It is also worth mentioning some of its characteristics such as 1-hour autonomy and a max speed of 2 m/s.



Figure 26. Sphero

Sphero has been completely designed for entertainment purposes, which leads to the variety of games that have been developed, even for inventing your own programs and activities since there are official and unofficial SDK (Software Development Kit). The developers have created several models such as Ollie, BB-8, R2-D2, Bolt or the version mini with different characteristics and capabilities.

The designers have also introduced the Sphero into the augmented reality world, creating some interesting content. In the game *The Rolling Dead*, using the camera of your device your area becomes a virtual zombie battlefield and the robot turns into a fire ball that can kill the infested zombies and pick up the needed ammo and power ups in order to survive the hordes. In addition, in the game *Sharky the Beaver* the robot transforms itself into a virtual pet that pursues the cupcakes that the player generates in the playing space.

2.5.3. Ozobot's Evo

This education robot (Figure 27) can move following a customized track drawn, which can be defined with two different ways, by using colour codes with markers and stickers, which teaches basic coding notions such as debugging and cause/effect without any screen or by visual programming dragging and dropping blocks of code in the tool *OzoBlockly* in order to design new games. It also counts with a speaker and RGB LEDs.



Figure 27. Ozobot's Evo

2.5.4. Jumping Sumo

This wheeled mini-drone (Figure 28) is described as a terrestrial robot that is able to perform small jumps up to 80 cm vertically and horizontally, thanks to its IMU that endowers it with flight stability analysing its current altitude and orientation. It also counts with a 640x480 pixels camera, which can record videos and images and transfer them to the controlling device connected by WiFi. Moreover, other features are a speaker, led eyes and a max speed of 2 m/s and could be dimensioned with a size of 18.5x15x11 cm and a weight of 180 g.



Figure 28. Jumping Sumo

2.5.5. Vortex

This programmable robot (Figure 29) is based on Arduino and possesses 32 eye expressions and a full range of light effects with its RGB LEDs aside from an MP3 player, which can reproduce predefined music or your own voice and sounds files by uploading them. It also counts with Bluetooth connectivity alongside a computer with proximity sensors and speed controller, which leads to intuitive maneuverability by using the control device screen and interacting with game objects.



Figure 29. Vortex

Vortex includes some pre-installed games that can be enjoyed against other players or the AI, such as Bumping Fight, which exemplifies a traditional sumo match where the other robot must be pushed away the playing field, or the Virtual Golf, where the robot turns into a ball and the player must put it in the corresponding hole by using a user interface.

2.5.6. Drones

Drones have a wide range of models varying in size, weight and autonomy, while most of them share certain characteristics such as an IMU for flight stability knowing in every moment their orientation and altitude, along with ultrasound, pressure and vertical camera sensors. They are also controlled whether manually by certain devices usually connected by Bluetooth or autonomously if programmed with the corresponding SDK.

2.6. Cameras in the Market

In this sub-chapter, the most commonly used cameras are collected in order to compare their characteristics and capabilities, whose minimum requisite lies in having a depth/infrared sensor. This technology is indispensable to execute an adequate set of algorithms that could result in the desired touch surface effect.

So that, the most commonly used 3D cameras' applications include healthcare preventing risks and supporting the rehabilitation period, the retail management monitoring inventory or people, the automation process of industrial processes, the robot autonomy and vision for real-time localization, object avoidance and interaction. It is also worth mentioning the 3D scanning used to reconstruct 3D models by depth data.

2.6.1. Kinect and Kinect v2

These movement sensors were developed by Microsoft in November 2010 and 2013 for the global launches of the video consoles Xbox 360 and Xbox One. However, with the aim of taking advantage of their full potential, they were redesigned and, therefore, improved for commercial applications, being compatible with the computers. Both sensors have colour, with 640x480px and 1920x1080px respectively, and depth cameras, with 320x240px and 512x424px respectively, along with an infrared sensor similar to the depth sensor. In both cases, the most efficient depth distance varies in the range 50 cm to 4.5 m while the field of view is a little wider on the second version. The images of both sensors are illustrated in Figure 30.



Figure 30. Kinect and Kinect v2

2.6.2. Intel RealSense Depth Cameras

This technology is actually supporting a wide range of applications, among which is worth mentioning 3D scanning, robotics, VR/AR, control of drones, facial recognition and measurement of objects and distances. This is possible due to the official open-source SDK contains a large number of wrappers for numerous platforms, that have been designed to provide the applications the ability to understand the world in 3D using vision-based approaches. A camera from the series is depicted in Figure 31.



Figure 31. Intel RealSense D435

These cameras offer the highest reliability on large scenarios since its range is theoretically stable until 10 meters, whose accuracy depends on the lighting conditions and the calibration, and therefore, are ideal for wide scenarios with low-intense light. They also provide the highest depth resolution and framerate capturing 1280x720 pixels at 90 fps.

2.6.3. Orbbec Astra Series

These cameras (Figure 32) provide computer vision algorithms easing functions such as body tracking, objects and face recognition or 2D and 3D SLAM with point clouds. Moreover, they also procure high-end responses, depth perception with the ability to discern the low-quality pixels, as well as obtaining precise and smooth contours and gradients in short and long ranges.



Figure 32. Orbbec Astra

Regarding the technical capabilities, the main difference of the PRO version with the series lies mainly in the range, which could measure stable values until 8 meters instead of 2 meters, and the RGB resolution, which captures 1280x720 pixels at 30 fps while the rest work with 640x480. So that, the principal common characteristics are their reduced size compared to the existing devices in the market, the use of its own SDK and most importantly, its charge through a USB 2.0 port which means that it does not need external battery resulting in a low power consumption for the system

2.7. Previous Works

In this subsection, similar projects from the students of Politecnico di Torino are studied in order to have a clear vision of the involved technology and the works' extension, which could serve as a starting point getting to know what has been already done. These games are also integrating a robot into a MR environment, powered by Unity.

2.7.1. Station Master

The present project[29] consisted in a MR robotic game based on floor projection, where the player took the role of the master of a train station while the robot, in this case, the Parrot's Jumping Sumo, represented the only active train, which moved autonomously over the railways split into cells. Those cells could symbolise signals, switches or stations, which were controlled by the player position standing on them. The game finished when the player managed to guide the robot through all the stations or when he lost all lives, which occurred when the robot hit the player's circle enough times. This game is illustrated in Figure 33.



Figure 33. Station Master

Regarding the involved technology and the implementation, the tracking of the robot was approached mounting a Vive Tracker on it, information that was sent to the gaming platform using the SteamVR SDK, while the player held a Vive Controller to be tracked. Lighthouse emitters were introduced to the gaming area to track both agents. Moreover, the system logic was developed in Unity synchronising the digital events with the player and robot movements.

This work was regarded as an initial attempt to relate basic fundamentals to keep in mind when developing phygital MR-based robotic games.

2.7.2. Colour Pong

In 2019, Piero Baldo[30] implemented a MR robotic game following the PIRGs philosophy, which stands for physical interactive robotic games, with the aim of analysing the influence of tangible objects on the player's perception. In this multiplayer game, the robot Cozmo arbitrated a competition between both players, who must set the right colours on Cozmo's cubes according to the specified series announced by the robot. An illustration of the game setup can be seen in Figure 34.



Figure 34. Colour Pong

Regarding the implementation, the cubes' colours and positions were tracked using a Kinect camera while the virtual area was projected on a table, where the game took place. The game was partially developed on Unity and on Python, which were connected using an open-source library.

Finally, a questionnaire was requested to several players after testing the games with the objective of collecting information about their experience, evaluating certain critical parameters such as the usability and the robot's roles. The results concluded that the usability differed greatly depending on the game mode and that different gestures by means of tangible objects do not change the perception of the robot by the players.

2.7.3. Protoman Revenge

In 2017, Thiago Resende investigated the degree of improvement possibly obtained in a game[31] involving drones by the inclusion of emotional encoding features and autonomous behaviours in quad-copters. Therefore, it is focused on emotional aspects and the player's skills to decode and attribute emotions to robots while interacting with them. The developed game, called *Protoman Revenge*, is depicted in Figure 35.



Figure 35. Protoman Revenge

The game was implemented using the A.R. Drone 2.0 from the company Parrot, while the processing of the information was made in a remote computer in a ROS environment, connected by Wi-Fi with the drone. The computational vision was executed using a colour detection mechanism using the OpenCV library to endow it the capability to distinguish between homogenous coloured objects and also, allowing it to transition between the game stages activating the motors and/or playing sounds. Therefore, the fragmentation of decisions was crucial in order to perform fast reactions, which was overcome defining different nodes,

responsible for processing the image, moving the drone processing the state of the game and playing sounds.

Regarding the experiments, the three versions of the game were tested by 16 university students, whose perceptions about it were assessed by a standard questionnaire. The results confirmed the initial hypothesis, suggesting that the autonomy altered the perception elements of the game and, as a result, a better experience was associated with the naturalness of the drone's movements.

2.7.4. Protect Treasures

In 2017, Junchao Cheng proposed a MR-based robotic game[32] with the objective of enhancing the human experience during the game progress. For that reason, multiple user interfaces were developed so as to contrast them discerning the most engaging human-robot interaction. This game recreated a turn-based adventure, where the robot or the player must protect several treasures from the other collecting them by the use of auxiliary items such as traps and missiles.

This project (Figure 36) was implemented utilizing the robot Sphero 2.0, controlled by Bluetooth on a remote computer that processed the information through the Robot Operating System (ROS), which consists of a specific framework for robotic software development. Also, the robot tracking was enhanced by two ROS tools, Rviz and rqt, as well as the implementation of a PID to smooth the robot's movements, while the player tracking was approached by two RGB-D cameras. The implementation of the game was done under the Unity environment and, therefore, the game scripts were completely coded in the C programming language, utilizing the floor-projection technique.

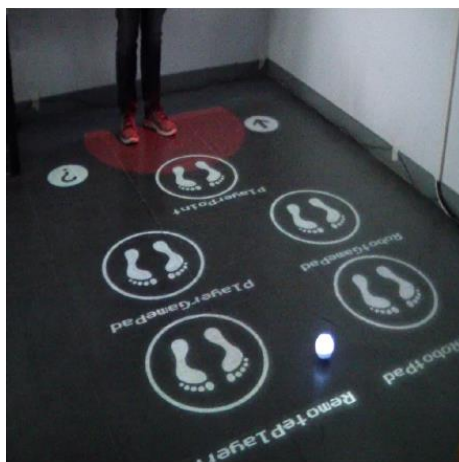


Figure 36. Protect Treasures

These game modes were tested by 15 players, whose experience and perception were analysed by a questionnaire, whose objective was to substantiate that the presence of a robot in a game was pivotal and enhanced the player experience. It quantitatively evaluated, by rating several statements from 0 to 4 (strongly disagree or agree, respectively), essential aspects such as the preferred mode, the game quality and experience as well as the system and player behaviours.

The analysis of the questionnaire showed that the inclusion of the robot brought noticeable benefits since fourteen out of fifteen people lean towards the game mode that included it, bringing a richer experience characterized by a higher intimacy and, consequently, confirming the initial project hypothesis, even though its presence created several error circumstances and certain instabilities when it executed high-speed motions.

Chapter 3

Technology

In this chapter, all the technologies involved in the project will be detailed, highlighting the corresponding main software and hardware capabilities, for a richer understanding of the design and implementation.

3.1. Software

For the realization of the proposed project the following software elements have been used: Unity game engine, Visual studio, OpenCV, the bridge ZeroMQ and the programming languages C#, Python, HTML and Java.

3.1.1. Unity

Unity[33] is a cross-platform game engine with a built-in IDE (Integrated Development Environment), developed by Unity Technologies, which is used to develop video games for consoles, web plugins, mobile devices and desktop platforms. Its workflow consists in putting together the sounds, graphics and animations in the IDE, write the required code for the logic and the assets to create a playable application that can be run on different platforms.

This game engine is endowed with numerous features that give it a clear advantage when it comes to game design and implementation compared to the rest of its competitors. Unity is a very effective rendering 2D and 3D scenes, which results in obtaining a high quality; it is also designed for cross-platform development, which allows the application to be compiled and run in several platforms efficiently. It also counts with a skilled community and tech support, who offer extensive high-quality documentation and official tutorials along with the assets store, which facilitates the learning and development process.

In this project, its main core is built in Unity integrating the graphic interface and the scripts developed in C# to control the events with internal and external signals.

3.1.2. C# and Visual Studio

Unity supports three programming languages for scripting: C# and JavaScript, being both object-oriented. It also uses MonoDevelop, which is a cross-platform implementation of Microsoft's .NET framework and, therefore, C# is considered the most powerful and suitable language since it is the primary language of .NET along with the fact that all the Unity's libraries are written in C#.

In this project, the Unity's code has been developed in visual studio using the programming language C#. These scripts transform the information received from the camera and the robot, which are both written in Python, into events, aside from managing the system's progress, which will be detailed in section 5.6.5.

3.1.3. Python

It is considered a high-level general-purpose programming language, which stresses the significance of the whitespace and the code readability, supporting several paradigms such as functional, procedural and object-oriented programming. Moreover, it is open-source, being

free for any type of application, resulting in a large community and a significant number of accessible projects and software.

In this project, it has been built in Python the needed script to control the robot and the camera, along with the language bridge with Unity.

3.1.4. HTML and Java

HTML is considered the standard language to design documents and applications displayed in a web browser, whose behaviour and content are controlled by technologies such as CSS or Java/PHP triggering the corresponding events.

Then, the browsers receive the HTML information from certain servers (local or web) rendering the documents into multimedia web pages. Also, the elements are divided in blocks, which are delineated by tags aiming a clear structure denoting the semantics for components such as paragraphs, links, buttons or images. Therefore, HTML is in charge of describing semantically the appearance of the web

In this project, the wizards' web browser interface is written in HTML while the associated functions are assisted by the programming language JavaScript. This tool is encapsulated within the Python script in order to develop a single file containing all the external information from Unity.

3.1.5. OpenCV

It is a software library[34] that provides a platform for computer vision algorithms with machine learning integration. Due to being open source along with a large community ease its code utilization and modification for companies' commercial applications, governmental bodies and research groups, most of them with real-time purposes.

This library counts with numerous algorithms specific for a wide range of applications such as face recognition, red eyes removal, identification of objects and humans, movements tracking or creation of 3D models. Moreover, it supports Windows, Linux, Android and Mac OS and even though It is natively written in C++, it also offers Python, Java and MATLAB platforms.

In this project, the software is utilized to detect and process the hand movements of the user, resulting in simulating a touch screen on a surface, in this case a table. The required functions will be detailed in section 5.4.

3.1.6. ZeroMQ

The ZeroMQ[35] software is a networking library consisting of several sockets that carry messages asynchronously across various applications, allowing the communication between both sides. It also counts with four different coding patterns such as task distribution, request-reply, pub-sub and fan-out.

In this project, the pattern request-reply is executed to transmit information between Unity and the python script. This kind of arrangement allows Unity to trigger events with signals obtained from the camera and to control Cozmo adequately transforming C# scripted messages into Python commands intelligible by the robot.

3.2. Hardware

For the realization of the proposed project the following hardware elements have been used: the robot Cozmo, the camera Kinect 2 and a projector, alongside a computer to process and integrate every part.

3.2.1. Cozmo

Among all the accessible robots, Cozmo (Figure 37) has been chosen as the main actor due to its qualities are the most appropriated to the system design and specifications. Its great maneuverability allows concise and precise movements, which is appropriate since the robot's position is considered essential for the system's purpose. In addition, its speaker and facial expressions increase the interaction's possibilities with the player, causing a positive impact on the training experience.



Figure 37. Cozmo

The commands that control its movements and sounds have been built based on the robot's Application Programming Interface (API), which is developed in the programming language Python.

3.2.2. Kinect v2

This camera is considered a motion input sensor device produced by Microsoft, which initially was developed as an accessory for the Xbox video consoles. However, and due to its characteristics, it has been included in numerous commercial projects away from the gaming scene performing as a low-cost sensor, able to interact with the environment through gestures and spoken commands.



Figure 38. Kinect v2

In this project, the camera is responsible for recognising the player gestures, which is scripted in Python and sent to Unity through the ZeroMQ library. Its essential role is to simulate that the projection is touch.

3.2.3. Projector

The displayer involved in this thesis is the projector Acer X1161P DLP, which offers a maximum resolution of HD 720 (1280x720) and a native aspect ratio of 16:10. In this case, the projector is in charge of displaying the gaming scene on a table so the player is able to see the training interface and interact with them using his hand, triggering events in Unity. Moreover, it must be located above the table like the camera, so the image is adequately shown.



Figure 39. Projector Acer

Chapter 4

System Design

In this chapter, it is proceeded to design the basics of the training system and, therefore, all the elements involved in it and the way they interact with each other. Moreover, the lesson's rules must be completely defined taking into account the possibilities offered by the current technologies of the market, which could constrain the scope of the system. Its design procedure will consist of a brainstorming followed by the definition of the concept and the required elements.

4.1. Brainstorming

The first step of the system design process lies in a brainstorming. This powerful iterative technique lets the user to generate new ideas around a specific domain of interest by eliminating all types of constraints, keeping in mind the requirements the system must fulfil.

Then, some minimal specifications about the system have been defined with the aim that the majority of them are integrated in the final version of the proposed system. These requirements have been defined based on the current state of the art of robotic gaming, taking into account which approaches and features have obtained better results and what have not been investigated yet in this field of study. The desired requisites are the following:

- Educational topic: it is considered interesting that the purpose/content of the system is educational, aside from offering only entertainment, letting the user expand concepts, learn about a concrete topic, reinforce an already assimilated development or simply assist them in learning a new skill. However, it also must satisfy the need for enjoyment providing enough motivation to continue playing and, therefore, finding an adequate balance between learning and engagement.
- Several modes: it would be beneficial that the system admits more than one mode in order to execute different comparisons such as changing the robot's role to evaluate its performance variation, modifying different rules to assess which offers more engagement or comparing diverse MR technologies such as projection against holographic content. These comparisons provide consistency to the project and reinforce the system design quality.
- Target audience over 18: it is recommended that the designed concept is not considered easy because of the fact that the final experimental evaluation will be tested with unbiased university students and, therefore, its content must be enough challenging for the target audience.
- Main robot's role: it is a must to reinforce the role of the robot inside the system and, therefore, it cannot be unnoticed by the user resulting in being dispensable within the system mechanics. Moreover, the robot's role could vary through different modes or as the training progress.

The system must be also comfortable for the user, since they are the final target who use the application, so the designer must perform a safe and rapid exploration taking advantage of the physical space and developing a deep understanding of the MR.

Then, keeping in mind the aforementioned considerations and accessible technologies the following ideas came up with the first brainstorming, which were analysed and compared.

- 1) Follow the sequence: the robot creates a random sequence going through different points and the player has to recreate it walking through the same spots/using an AR panel.
- 2) Memory Cards: a player must make pairs touching cards face down and then, an interesting fact about its content like animals, professions, materials, etc. is shown with voice/AR letters. The robot can be either another player or part of your team.
- 3) Tele-operated Firefighting: different fires appear randomly on a city and the player must take actions in an AR panel to control the situation. With this panel, the player controls the drone to rescue people and put out the fires with water, set fire alarms, evacuate people, send fire trucks, etc.
- 4) Protect the city: a city is attacked by a robot launching missiles or direct hits from different angles and the player must raise shields in certain zones to protect the citizens and buildings. The player and robot roles could be swapped.
- 5) Climb the mountain: the robot and the player work together to climb a mountain following different paths. In each turn both must decide between 3 random boxes that may contain lives, shields, etc. At certain heights bears appear and must be defeated to continue taking into account that both members must reach the top to win the game.
- 6) Guess the country: country characteristics like flags, monuments, typical food, etc. are displayed with AR and the player must place the robot on a projected map on the floor within a certain range of kilometres of the corresponding country. If it is right, the country and an interesting fact about that characteristic are shown with voice/AR letters while if wrong the robot gives you a hint and therefore the player loses a life.
- 7) PAC-MAN: the robot must collect all the coins inside a labyrinth and then go to the exit avoiding the enemies that move randomly. There are some special elements like lives, temporary invincibility and the ability to eat the enemies within a certain time.
- 8) Football battle disc: 2 robots are faced in a football match projected on the floor. Each turn a player can move his robot to shoot the ball trying to score a goal and the robot must remain where it is after hitting the ball. One team can be totally projected if there is only one player.
- 9) Cruel race: the robot must move through narrow paths avoiding pendulums, spikes/traps on the floor, fireballs, etc. in order to reach the finish. If the robot is hit by any of those, it falls to the cliff and, therefore, the game is over and must start again.
- 10) Human body: an empty AR body must be filled with different organs, muscles and bones. The player must collocate them using an AR panel and when it fails the robot gives a hint to help placing the corresponding part. After each correct collocation, an interesting fact is shown about that body part.

In this first iteration, most of the proposed ideas' purposes focused entirely on entertainment while others brought more educational or learning topics such as developing the memory. Moreover, all of them were centred in executing a comparison with different game modes stressing the differences between projected and holographic MR, and the roles of the robot.

However, none of them was interesting enough to be worth developing due to many factors such as their simplicity, their topic or simply for being already well explored areas in this field

of study. For these reasons, a second brainstorming was performed mixing some of the previous ideas and including new ones.

- 11) Triviasphero: 2+ players must answer correctly different trivial questions that appear in an AR panel. The options appear with AR letters above different areas and the player must place the robot in the corresponding area. When a player fails, the turn moves to the next one and the game finishes when a player achieves the needed points. The players can also ask for hints, and then the robot will position itself between the correct answer and another discarding the rest.
- 12) Conquer the top: The scene is divided in two, half for the player and half for the robot. In the path of the robot appear different doors which hide different task that the player must complete. Answer trivial questions, avoid fireballs, score a goal, guess a country by one characteristic, repeat an established sequence with the feet etc. If the player success the robot continues to the next door, if he fails he loses a life.
- 13) Cozmoboard: A board with 6x6 spaces is projected on the floor and 2 teams are confronted confronting the player and the robot. Each turn one of each team can move one position in any direction starting in the corners and the goal is to circle the enemy's robot/player with at least 2 characters, one in front and the other behind. Also every time a character reaches the other side of the board a new character joins their team starting on their side.

Once again, it was not possible to create a game concept that meets the desired requirements that could bring something new to this work environment. Consequently, a final brainstorming is performed, which is illustrated in Figure 40, obtaining the following ideas. As can be seen, most of the games proposed to combine at least three of these ideas.



Figure 40. Brainstorming

14) Manager: 2+ players are the managers of different airline companies and must compete with pricing their seats having both the same number of seats. The objective is not to sell the highest number of seats but to maximize the profit of the business. The interface would be projected on a table and both players would have access to the same information. The ticket sales last 5 days and the players must price their tickets at the start of each day with the info given when the day finishes.

15) Rhythm: A rectangular playing field is projected on the floor and the player must step with the feet on musical notes, which go from one side to the other of the field through 4 different lanes, when they reach a certain position. The player loses if the notes are not touched within a certain time or touches several wrong notes during the same song.

The musical notes, which are part of well-known songs, are played by the robot when the player step on them. The robot position itself on the other side of the field where the notes appear and moves at the rhythm of the music with each correct note.

16) Mirelab: A test bench is projected on a table simulating a practice lab from the university such as electrical or pneumatic circuits. The projection would be divided into 3 areas, the bigger one corresponding to the circuit, 2 columns on the right side with all the electrical components such as resistors, sources or ammeters, and a row below with the simulation buttons like start, stop or show the values.

The robot acts as the lab teacher. It moves along the table giving instructions to the player to finish the lab through voice, sounds, and greetings.

17) Escape room: The player and the robot are locked in a room and must inspect it looking for objects to escape from it. These objects are half physical and digital such as keys, papers, boxes, balls or padlocks and the player must interact with them in the correct way to complete the game within a certain time.

Therefore, it could give you hints if the player is stucked or if it has been too long without advancing, positioning itself close to the next needed object or through voice, and it could also show emotions when the objects are found and when they are used adequately/wrong.

18) Compound: This player vs player game consists in developing chemical compounds, whose difficulty could vary from water to TNT or nitro-glycerine with the corresponding value points, collecting the needed atoms. In the first phase the board game is projected on a table and the players have 3-5 cards per round of compounds that must be discovered selecting and dragging on the cards the C, O, N, H, S, P atoms that appear randomly before the time runs out.

The robot is the game master. The players are sitting on opposite sides of the tables and the robot starts the game moving from the centre to one of the rest sides.

This time, two ideas were interesting enough to be developed, the Escape Room and the Mirelab, since both ideas have the level of complexity required for such a project and include the vast majority of the specifications predefined. Moreover, the one and the other take a step forward in this field of study since neither of them are areas well explored in the gaming and MR environment. Therefore, both proposals were properly concretized in greater depth with the aim of selecting the best possible idea to be later implemented.

While the Mirelab option could stress the differences between two different learning models such as the LxT model against the traditional lecture-based classes, the Escape Room would assess the digital content impact on the system modifying the amount of non-physical objects.

The progression could cover from being all the objects physical, manually entered into the room, to all digital content created by the hololens.

Finally, it was chosen to implement the idea called Mirelab because it is considered the most adequate option, containing the vast majority of the initial requirements, being the one that best suits the expected scope of a project of this magnitude. Furthermore, even though each part has been investigated separated, such as the touch screen, a robotic system, MR, etc. the integration of all these features in the same project is an under-developed field of study in this discipline, which results in an appropriate topic to be developed in this thesis. Therefore, this application will be able to support the mentioned leaning models, whose corresponding Unity modes will be completely represented by their main attributes.

4.2. System Overview

4.2.1. System concept

This design describes the details for a MR single player educational training system based on engineering university subject laboratories, with the novelty of supporting multiple learning models, resulting in a flexible system that could better fit each student's learning style and adapt to extended blending model to foster systemic adoption. It consists of projecting a test bench on a table exemplifying a practice laboratory from the university such as electrical circuits. The user must complete the labs within a certain time following the instructions provided.

The assessment carried out will compare the TL with the LxT model, providing evidence to which method is more effective in the user's learning in the context of MR robotic teaching. Moreover, the significance and the utility of the robot within the system will be critically evaluated.

For that reason, two different modes will be developed with the aim of stressing the differences between these diverse learning models and, therefore, both the role of the robot and the user vary completely from one mode to another. These two modes will represent the main characteristics associated with the traditional and LxT models.

It is important to note that in both cases, the ultimate goal of the system is the learning of the user, one being explicitly executed taking passive lessons from the robot while the other is performed by the need of actively teaching to the robot, reinforcing the user's own concepts.

Regarding the training scene, the projection will be divided into three areas, the bigger one corresponding to the workspace, two columns on the right side with all the required components such as resistors or sources, and a row below with the supporting buttons for rotating the components or creating cables. This preliminary configuration is depicted in Figure 41.

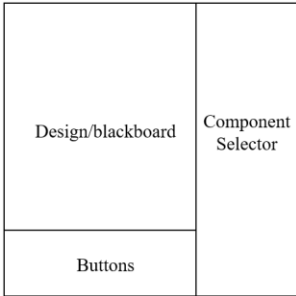


Figure 41. Initial Interface Draft

Then, and due to the training interface will be displayed on a table, the user must remain standing or sitting in front of it. Thus, the user is in a comfortable position to reach any area of the table in order to complete whatever task or action the activity requires in each moment. The robot will be also on the table and will be able to move freely, acting accordingly to the events. Moreover, the interaction with the projection will be entirely carried out with the hands of the user, either with the palms or the fingers, actions that will be recognised by the camera.

4.2.2. Training Modes

Since the ultimate goal of this project consists of providing sufficient evidence to which method is more effective when it comes to learning, designing carefully the training modes that could stress enough the advantages and drawbacks of each learning model becomes essential. Therefore, two main modes will be implemented, one representing each technique, along with a short introduction where the basics of the interface are explained to the user so he can easily interact with it during the principal modes.

- **Tutorial:** The objective of this first mode is to teach the user how to use the interface and, therefore, all the functionalities are enabled with no predefined order. In this case, the user will be taught side by side all the possible actions the system can perform and later he will have time to practice until he feels enough confident using it.
- **Traditional Learning:** In this mode, the user will learn a concept from the engineering field following the explanation received from the robot. Therefore, the user has the student's role while the robot acts as the teacher moving itself on the table giving information and feedback to the user through voice, sounds and greetings.

Therefore, the robot uses the blackboard inserting different objects such as images, electronic components and/or formulas to explain the concept and solving an example while speaking to carefully make it clear for the user. Moreover, it also asks the user for collaboration at some points like choosing the values of the components or removing certain elements suggested by the robot. These little interactions are introduced in order to keep the user's attention in the explanation resulting in a more engaging experience.

- **Learning by Teaching:** In this second mode, the roles are reversed. Consequently, the users will have to learn by themselves a concept from the engineering field, and prepare the lecture prior to teaching it to the robot. So that, the robot has the student's role while the user acts as the teacher interacting with the interface and using his voice.

As a result, being similar to the previous mode, the user uses the blackboard to clearly explain the theorem and continuously interacting with the robot through voice and the interface. On the other hand, the robot will follow the lesson asking questions and performing the tasks that the user requires in order to increase its inclusion grade on the experience and not being unnoticed.

The chosen topic for this project is the Thévenin Theorem, from the electrical engineering domain. Due to the target audience was university students from engineering, it was necessary to select a topic not too basic in order to keep the engagement of the participants and inducing a possible learning about a new subject, but also not excessively complicated so the system's capabilities do not restrict the learning's extent.

4.3. Elements

Before starting to implement the system, each needed component must be completely defined in order to ease the creation of an interface that fulfil all the training progression necessities. Once again, it must be noted that this is considered an iteration process, so even though the training core is set, there could be modifications to fit the training progression in the most adequate way, based on the user testing impressions.

- A place where the circuits could be created (Figure 42) is required and, therefore, the electric components could be placed there and joined using the cables. This must be the bigger area of the training screen since, even though the robot could move around all parts, the main actions and lectures take place in this panel.



Figure 42. Design Panel

- A selecting area (Figure 43) is also needed in order to be able to choose the desired electric components by either the user or the robot. This area should contain at least the most basic elements such as voltage and current sources, as well as resistors, capacitors and inductors. Other components more complex such as transformers or operational amplifiers could be added if the circuit or the lectures require them.

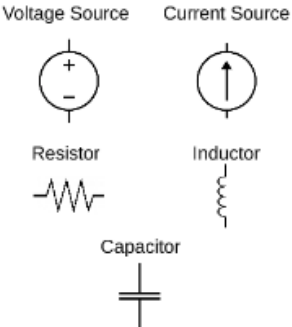


Figure 43. Selecting Area

- A customization area (Figure 44) is required where the electrical components are refined before being dragged to the panel in order to fit the circuit design. In that place, their value is assigned and it can be rotated if needed.



Figure 44. Customization Area

- Several buttons (Figure 45) are also needed to support the actions of the user during the training experience. With that said, the main actions are the rotation of the components and the creation of the cables that connect them.



Figure 45. Interface Buttons

- A menu screen (Figure 46) is also a must, shown at the beginning, that shows the platform name and the robot start position, as well as allowing the user to select the training mode.

MIRELAB



Figure 46. Menu Screen

- It must be also set the robot starting place (Figure 47). This initial position will be essential to the robot to orientate himself during the training in order to have a clear movement reference, since it does not know where is when it is turned on.



Figure 47. Robot starting position

- A restart screen (Figure 48) must also be included after each lab is completed in order to let the user go back to the title screen and select the next mode. It should also display a text to guide the user.

Lecture Finished!



Figure 48. Restart Screen

It is worth mentioning that these elements should be redesigned during the implementation due to Unity requisites or because incompatibilities of the training progression, modifying the initial art concept until designing an effective interface, since it must clearly show all the system capabilities to the user.

Chapter 5

System Implementation

In this chapter, the implementation of the designed system will be detailed. Firstly, a short introduction describes the general overview of the implementation carried out with the aim of showing how the two main parts of the projects are integrated. These parts, Python and Unity, are concisely detailed subsequently.

5.1. Overview

The process of implementing this project has been divided into four different sections: the control of the robot, the design of a touch surface using a camera, the creation of the training interface, training modes and system logic in Unity, and finally their integration. In Figure 49 is depicted the general overview of the present project, illustrating how each part is incorporated and connected with the rest.

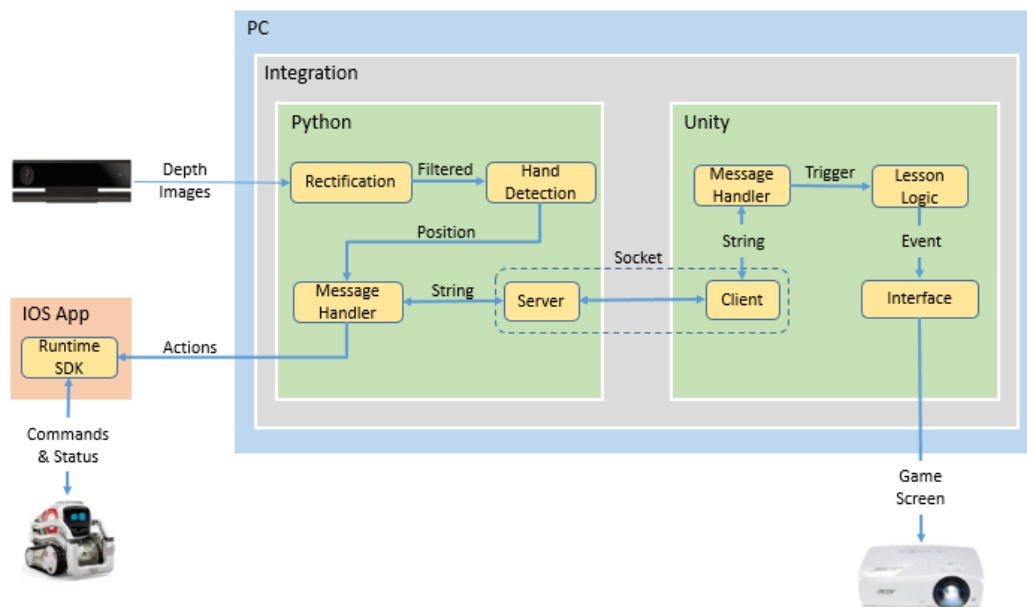


Figure 49. System Architecture

On the one hand, the touch screen has been approached using a background subtractor method, which identifies the moving objects in a static environment. First, the depth frames received from the camera are rectified to be later applied an algorithm to detect the hand of the user and calculate the fingertip's position of it, recognising events such a click or a slide, which are sent to the application through the socket by the message handler. This method is in charge of encrypting and decrypting the calculated position (both axes) together with the detected event in a string format.

Then, the application computes this information triggering the corresponding events, analysed by the logic, and displayed on the interface, which is projected to the training area, in this case a table. Moreover, the events also trigger certain movements of the robot and vice-versa, adequately matching the training progression.

Finally, the integration core is carried out by the library ZeroMQ. This library is used to send synchronously encrypted messages between python and Unity, allowing the communication between the camera, the robot and the application.

5.2. Unity

5.2.1. Interface

One of the main considerations to keep in mind when implementing a training system lies in capturing the user's attention during each phase of the training to the things that need their perception. Therefore, the design of a clear interface that properly reflects the capabilities of the system, supporting the robot actions and leading the user during the training experience is considered a critical part of the project. The main interface can be seen in Figure 50.

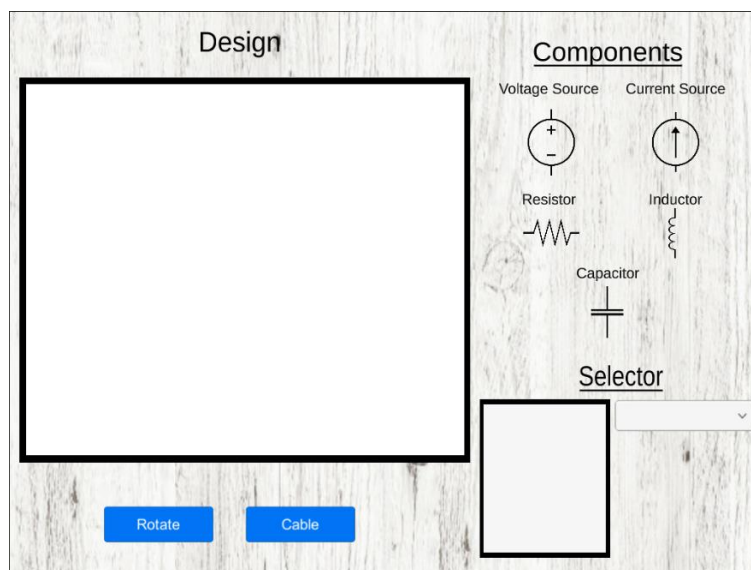


Figure 50. Interface

As can be seen, the interface has been created introducing all the needed elements conceived during the design. It has a blackboard for the designs, occupying the main space of the screen where the circuits are created and where the robot's lessons take place.

Moreover, it also has a selecting area, where the robot and the user can choose the desired components for the circuit. When the components are instantiated, they appear on the selector where a value can be assigned using the dropdown, which shows coherent values with the component selected. Then, they can be rotated if needed using the corresponding button and dragged to the panel. Finally, when all desired elements are on the panel, they can be connected using by clicking the cable button and later selecting the adequate terminals of the components.

Regarding the needed assets, which are 2D due to being a table-projected activity, they are designed carefully taking into account the requirements and art goals such as shape, colour and size. Then, these assets are imported to Unity where the application code is implemented allowing the desired interaction between the components.

From an efficiency point of view, some considerations must be taken during the application design creating the best possible experience. There are some design aspects that could affect negatively the performance of crucial parts of the application's device like the CPU, GPU and memory. For instance, the number of polygons, the textures and the shaders of the assets, this

is, its total number and their complexity, or the physics, the media and the scripts involved in the dynamics. Must be noted that the GPU is usually the most affected unit of the device.

Therefore, during the creation process for assets and designing in Unity, an efficient development should be performed minimizing the negative impact on the device.

5.2.2. Learning by Teaching (LxT)

Due to the role's change of the user during this second mode, some extra functions have been added to this part of the experience. The inclusion of these have been motivated by the necessity of endowing the user total control of the screen, in order to be able to successfully manage every aspect of the lecture and, therefore, having sufficient capabilities to effectively teach the lesson to the robot.

On the one hand, the main modification lies in an input tool whose principal functionality is to insert certain images and formulas to the screen's blackboard using an index procedure, and its numerical assignment to obtain the desired component value. As can be seen in Figure 51, its design is based on a general calculator and, therefore, contains all the digits, the operands for basic operations, a clear button and an insert button. This tool is deployed every time the user touches on the blackboard (with some exceptions) and, in the same way, it will be closed clicking outside of it.

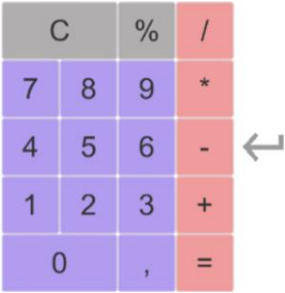


Figure 51. Input Tool for LxT

On the other hand, besides the basic functions for rotating a component and creating a cable, two additional buttons have been included (Figure 52). While the erase board method lets the user remove all the elements inserted by the input tool, the other function gives the user the possibility to restore the constructed circuit either when an error occurs during the lesson or for comparison purposes, so there is no need to build it again resulting in a smoother experience.



Figure 52. Additional Buttons for LxT mode

5.2.3. Scripts

The application code has been divided into seven different scripts in Unity due to design needs and for clarity reasons. These scripts are briefly described subsequently:

- **GameManager:** This class is the core of the program, since it defines most of the variables involved and the main methods in the system such as the creation of the static components, the UI elements and the delimitation of each area along with the progression of all training modes.

- Drag: It is responsible for controlling all the tasks related to the new elements created during the experience. It covers from the selection of new elements, to the dragging process along with the assignation of values.
- ScreenManager: This class defines the set of instructions that are executed when the UI elements are invoked, such as the buttons or the dropdown.
- CableManager: It is responsible for creating cables between the new components in order to connect them. First, it must find the closest terminals to the two consecutive detected clicks and then join them coherently.
- Socket: It is responsible for the communication between the application and Python, allowing the application to receive messages from the camera, handle the adequate UI elements and set the needed instructions to the robot so as to fit the training flow. This class runs in a different thread from the main process in order to avoid conflicts among both processes. This will be further detailed in section 5.6.
- Queue: The necessity of this class arises from the creation of a second thread in the socket, since numerous Unity methods can only be invoked from the main process. Therefore, it is responsible for putting these instructions in the queue of the main thread so that they can be properly executed.
- CalculatorManager: This class is in charge of managing the input tool used during the LxT mode. Therefore, it is responsible for taking actions based on the user's inputs either inserting new images or assigning values to existing formulas.

5.2.4. Methods

It is proceeded to describe the most significant functions involved in the implementation of the system in Unity, following as faithfully as possible the specifications defined during the design. Most of these functions make use of variables declared in different scripts so it is essential to establish connections between them either declaring the implicated value as public or defining getter and setter functions for particular private variables. Moreover, it has been intended to separate basic functionalities in short methods as much as possible with the aim of reusing these methods and, therefore, giving them more than a single-use. The main methods are the following ones:

- CreateInterface(): The first feature of the implementation consists of creating the interface of the training system, which has been previously described. This function sets the main panel, then selecting area for each component and the selector, along with the visible UI elements such as the two buttons, for rotation and cable, and the dropdown, as well as the corresponding texts defining each area. It must be noted that, as expected, it is called at the beginning mode after choosing it in the training system mode in the title screen.
- CreateCircuit(): This large method involves most of the functions defined in Unity to create a circuit on the panel. It can develop any electrical design predefining the components and their position and values, to be later connected by the cables. It is called with the aim of reinforcing the explanation of the robot, creating several circuits.
- GetTerminalsAndCreateCable(): This method is responsible for connecting the components of the panel. It's functioning is divided in three steps, first it recognises two clicks from the user and calculates the closest terminal of the closest component. Once it defined both possible terminals, it checks if the connection is coherent and the best way to join them to be finally connected on the third step creating a cable with the type

LineRenderer (Unity native component). It also sets an intermedium point between them if needed, with the objective of imitating the characteristic rectangular shape of the electrical circuits. This function is invoked after clicking on the cable button

- *Lesson()*: It is in charge of managing the explanation of the robot at the start of the TL, controlling the state of each component and the transitions between each part. This method has been divided into six sub-functions for clarity reasons and for a better approach during the integration of the robot and the camera, allowing a more fluid interaction between all agents.
- *LabFinished()*: This function manages the Restart Screen, which lets the user go back to the menu displaying a characteristic final text and restoring the initial scene. It is called when each training mode's requirements are met or, in other words, when each mode is finished.
- *CreateNewComponent()*: This large function covers from the instantiation of the component on the selector after choosing them in the selecting area to the assignment of coherent values. The involved sub-methods are *ModifyNewCompPosition()*, which varies the position of the element with respect to the selector centre, *PopulateList()*, responsible for filling the dropdown list with coherent values based on the chosen component, *CreateValue()*, which instantiates a text for displaying its value and *ModifyText()*, responsible for applying an offset to that text so as it matches the component orientation. Moreover, *ChangeValue()* changes its value depending on the dropdown value and *GuideManager()* is in charge of controlling an invisible guide to managing adequate positions for the circuit's components.
- *RemoveComponent()*: This method deactivates a certain component and the corresponding value's text by tapping on it twice. It is useful for opening the circuit between certain terminals removing the corresponding components which help the calculations. Moreover, there is a variation named *ShortCircuit()* that not only removes the component, but also creates a line between both terminals of the missing element re-connecting the cables and, therefore, simulating a short circuit.
- *AssignValues()*: It is responsible for specifying or changing the values of the components once they are on the panel. It is mostly used after the creation of the circuit example with *CreateCircuit()* during the robot's *Lesson()* designating each component's value with the aim of involving the user during the explanation, keeping him engaged as much as possible.
- *DestroyCircuit()*: This method destroys a complete circuit located on the panel, including the components and the associated texts that show their values. Then, this function is called to free up space in order to continue with the explanation.
- *RestoreCircuit()*: It is in charge of re-activating the components removed by the methods *RemoveComponent()* and *ShortCircuit()* so as to restore the analysed circuit. This is possible due to those methods do not destroy the components and texts, just deactivate them.
- *OnMouseUp()*: It simulates the PC version of the *OnMouseUp* method, which is invoked when the mouse stops being pressed. It is mostly used for clarity purposes, correctly reallocating the components when they are being dragged to the panel in order to avoid unaligned circuit configurations, as well as to modify certain flags.
- *ButtonTapped()*: This method is responsible for executing the adequate action when a button of the input tool is pressed. Consequently, it analyses from which one has been

pressed to the corresponding element that must be inserted to the writing of formulas and numbers, as well as the closing and clearing task associated.

- Start(): This predefined Unity's function is defined on each script and it is called one time at the beginning of launching the application. In this case, it is mostly utilized to establish connections between the scripts instantiating an object of the needed class, which allows accessing the methods and variables of that class as long as they are declared as public.
- Update(): This predefined Unity's function is called once per frame and it is defined on each script. It is only used during the dragging phase in order to modify the component's and text's position in the PC version as well as continuously calculating the distance between the invisible guide and the dragged component.

5.2.5. Canvas

Following the Unity Manual, every UI GameObject such as the buttons must be placed under the canvas in order to be intractable, as well as the text used to be read-able. It also must be noted that any element under the Canvas has a different scale than the rest of GameObjects, since Unity discerns between the screen and world scales, using dissimilar axes for UI elements. In the present project, the elements have been divided under several categories for clarity purposes, which are detailed below:

- Title Screen: the components in this category are shown when launching the application containing the text name of the system and the buttons associated with each mode. This screen is used to present the system and let the user choose the next training mode. However, it has been designed that the user must unlock the training modes by completing the precedent ones so as to follow a coherent order. Therefore, at the start, the user only can play the tutorial/basics mode in order to get to know how to use the interface and interact with the robot and unlocking the following mode, when finishing it.
- Interface: it includes every Game Object related to the static interface, even though some of them which are not required to be under the canvas. This category contains the panel where the circuit is built, the components that can be selected, the selector panel along with the dropdown and the buttons, as well as all the corresponding texts. It also includes all the new texts containing the values of the new components.
- Traditional Learning: this block contains all the GameObjects needed for this mode, including each image and text that appear during the robot explanation, as well as the intervention of the user during the exercise. Classifying all of them under the same block increases the clarity of the hierarchy and the simplicity to access to them.
- Learning by Teaching: this block includes all the required elements for this mode, which are mainly images and texts that will be invoked when the user decides to. It has been chosen this way for clarity purposes similar to the previous mode.
- Restart Screen: this screen is shown every time the user completes a training mode, letting the user go back to the menu and select the next one. Then, this category simply contains a completion text and a button which lets the user restart the menu.

It is worth mentioning that the interface is similar in both modes, except for two extra buttons that are added to the LxT mode to endow the user a higher control of it.

5.3. Python

5.3.1. Threads and Methods

Moreover, after completely defining the requirements needed by this side during the design, it has been decided to create three different threads to separate different processes. This kind of architecture was needed due to some functions were conflictive with each other, since they had to be executed in parallel, producing abnormal behaviours on the project. Then, to communicate between them some variables as well as events for synchronization purposes. These threads are detailed subsequently.

- **OpenCV:** this thread is in charge of the image processing to obtain the location of the fingertip. It covers from the receiving the RGB and depth images from the camera to delimitate the hand contour and calculate the top-forward point, alongside with several methods for a clear image calibration.
- **Socket:** this thread is in charge of receiving messages from the application in order to know what is the current state of the lesson at each moment. So, it also sends back the state of the Python processes through encrypted messages, including the finish of the robot movements or the fingertip's position. Moreover, it is also responsible for controlling the robot actions during the TL mode.
- **Robot:** this thread is responsible for following the face of the user anytime it does not interfere with other robot actions such as moving the head, moving to a certain position or executing some voice commands. Moreover, it is also in charge of controlling the robot during the LxT mode launching the Wizard of Oz's interface, which will be detailed in a subsequent sub-chapter.

It must be noted that the robot's thread is launched inside the Socket's thread to ensure that the connection with the application has been established, while the rest are executed at the very beginning of the program.

Once the structure of the Python side is completely defined using the threads, numerous methods are used to simplify the execution of repetitive actions over and over again. The main functions are briefly described afterwards.

- **FollowClick():** this method is used sometimes for setting the robot towards a specific direction when the user touches the screen. Therefore, it receives the fingertip's position and performs a transformation between the camera and robot reference systems, controlling the robot's head and angle adequately. Endowing the robot with this capability increases the human-robot interaction, which results in a richer experience for the user.
- **ShowFace():** it is responsible for showing different images on the robot's face at certain points of the lesson. The images are electrical components such as resistors that are previously saved to be later resized and converted to a specific format to be readable by the robot at execution time.
- **Lifts():** this function is in charge of controlling the robot's lift up and down to simulate its touches on the screen. It has been implemented a single, a double and a triple touch to create different actions.
- **MessageHandler():** this method is responsible for analysing the messages received from the application and acting accordingly controlling the robot actions and the touches detected from the camera. Moreover, it also sends back messages to inform about the Python side current state.

- `MouseDown()`: it is responsible for delimitating the touch area at the very beginning because of the fact that region of interest, where the screen will be projected, must be selected from the images captured with the camera, since the initial frames are larger. This action must be performed two times, one for the RGB and again for the depth images.
- `PerspectiveTrasf()`: once the region of interest is defined with `MouseDown()`, this function applies a transformation of perspective to amend the inherent deviation angle associated to the camera's position and inclination, obtaining a rectified and perpendicular vision of the screen as desired. This action is executed 2 times per frame (RGB and depth images).
- `RemoveBG()`: after applying the `PerspectiveTrasf()`, this function compares each depth frame with a static background previously captured to obtain the contour of the user's hand. Therefore, it is executed one time per frame.

5.3.2. Wizard of Oz

As has been previously explained, in the LxT mode the robot will not be any more AI controlled but manually. One of the principal premises of the project is that both versions must be as similar as possible to the user in terms of capabilities, degree of interaction, difficulty and topic. For that reason, the implementation of this tool must provide the wizard (not the user) the ability to perform exactly the same actions the robot was capable of when relying on the AI and, therefore, act in a comparable manner. Consequently, the user will not be able to differentiate if the robot is being manually controlled or not.

The developed interface was based on a similar project of the SDK Cozmo Examples[36]. This tool was developed using a webpage served by Flask, which is a framework written in Python that allows the user to create web applications with a minimum number of lines. Moreover, the web structure was written in HTML and the associated script in Java, which converts any event on the web such as mouse moving or clicking into Python messages, necessary to control the robot movements. The final interface is illustrated in Figure 53.

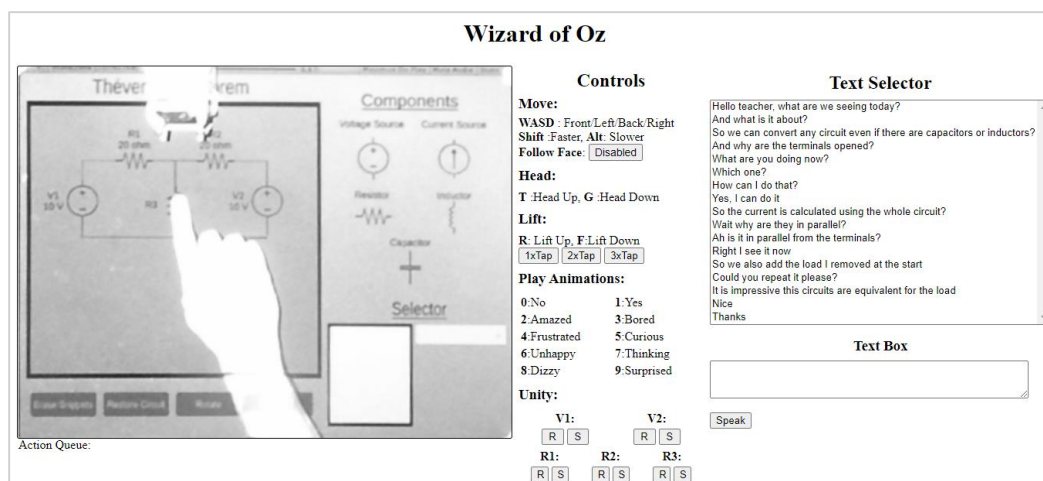


Figure 53. Wizard of Oz interface

As it can be seen, the interface was divided into three main parts adjusting to the project needs, whose purpose is described subsequently.

- Camera: a vision of the interface status is indispensable for controlling the robot. Then, the RGB image from the camera is utilized on the left part of the interface, after being calibrated, to see the region of interest where the experience takes place. It lets the wizard to see the actions of the user, how the lesson advances and to check that the robot movements are accurate enough in case a correction is needed.

It has been decided to convert the RGB camera into greyscale since the projector light reflections were severely increased by the projection's light colours such as white or light blue and, also the colour of the robot (white), which negatively affected the image shown being impossible to distinguish the majority of things. Moreover, the greyscale reduced the delay between the reality and the image seen in the wizard's tool due to less processing requirements (1x8 bits vs 3x8 bits). Therefore, in terms of functionality, it is the optimal option since the screen colour is not important for the user.

In addition, for facilities purposes, it has been added the capability of controlling the robot's position with the camera vision. Therefore, when the wizard clicks on the image, the robot will move to that location following the shortest path. Also, the final orientation can be defined clicking and dragging the mouse, which will convert the click into the final position and the created vector into orientation. If the mouse is not dragged (simple click) the robot will look forward.

- Controls: the middle section of the interface is responsible for controlling the robot. It shows all the possible actions it is capable of such as being manually controlled with the W/A/S/D keys, move faster or slower pressing certain keys or being able to control up and down either the head or the lift. It also has several buttons to execute predefined functions such as following the user's face or performing some taps on the screen.

Furthermore, several animations have been predefined assigning a number from 0-9 showing different emotions with the aim of creating a more personal and interactive experience. Finally, it has been also added some buttons to trigger events on the application. This capability is essential to simulate the feeling that the robot is actually interacting with the projection. For example, if the robot is asked to remove a resistor by the user it will have to touch it doing a double tap and then the clicking on the corresponding button to remove or short circuit such component.

- Speaking: on the right part of the interface it has been added the functionalities to control the speaking of the robot, indispensable to clearly communicate with the user. So that, it presents some predefined sentences in a text selector that are likely to be used during the lesson and, then, when they are chosen, they are copied to the text box ready to be spoken when the wizard presses the corresponding button.

Moreover, additional words can be written manually in the text box, concatenating the previously selected message or just simply writing a totally new phrase to interact accordingly to what the user is talking and/or doing.

This procedure's logic is illustrated in the flowchart shown in Figure 54, which is executed one time per frame. It must be noted that the camera vision is updated regularly in order to have a clear vision of the screen.

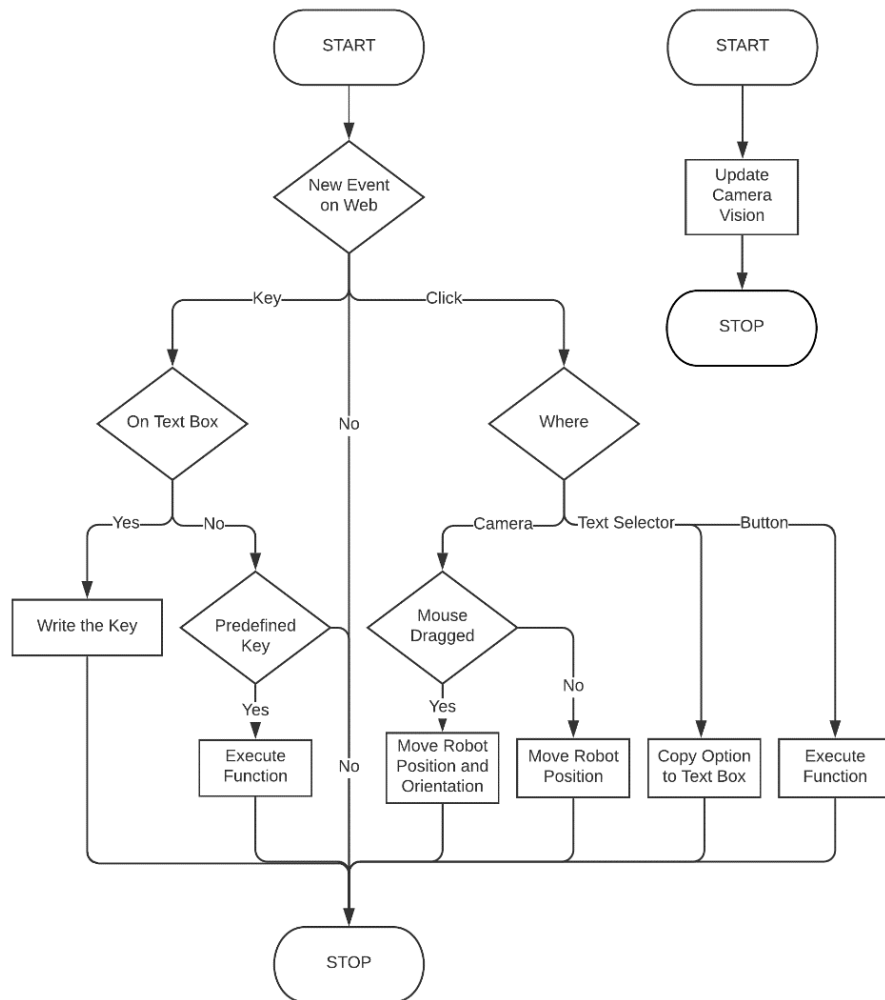


Figure 54. Wizard of Oz's Flowchart

5.4. Touch Surface

As it has been previously mentioned, the camera utilized in this project is the Kinect v2 since to simulate the desired touch screen, it is completely necessary to use a depth camera due to the extra dimension. With the depth information it is possible to develop algorithms to recognise certain objects or, in this case, human hands positions and, therefore, to detect touch events on surfaces, which would be quite challenging using only frame differences with 2D images.

The required code has been developed in Python using OpenCV in order to facilitate the integration with the robot, since it requires the same programming language, instead of using Unity's language C#. The proposed system will discern among two different events, a click and a slide. The system architecture used is shown below in Figure 55.

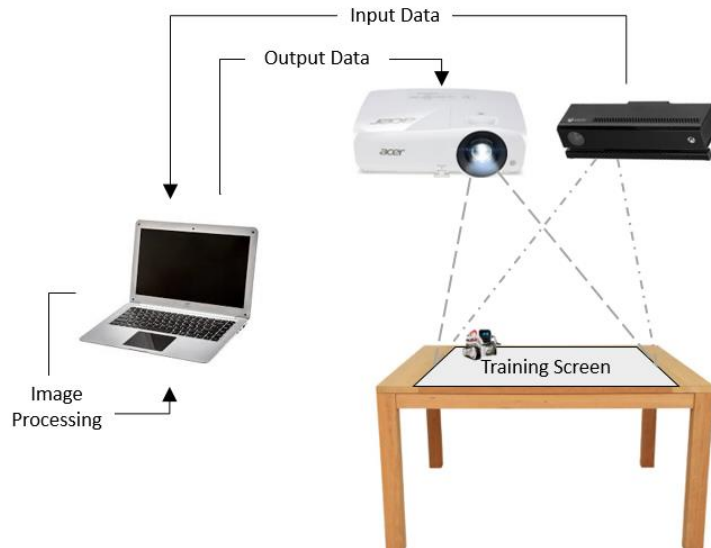


Figure 55. Touch Surface Architecture

This system requires a PC, a depth camera and a projector. The camera and the projector are located in front of the table so the camera is facing the surface where the system is projected. Then, each frame, the camera captures the images that are sent to the computer to be processed in order to create click or slides events in the application, which are visualized in the projection maintaining the learning progression.

It is worth mentioning the functioning of the depth camera. Each pixel of the image captured is converted into greyscale depending on the estimated distance measured and, therefore, every pixel at the same distance from the camera will have the same configuration. However, there are some blind areas that the camera is not able to correctly measure, such as closer to 50 cm or farther than 4 meters, where the data becomes quite unstable. Consequently, it should be placed in a suitable location, as perpendicular as possible to the surface in order to avoid significant differences in distances between parts of the projection since, as mentioned before, the information is classified in tight layers of depth.

The generic overview of the algorithm implemented is illustrated in the following Figure 56, which covers from receiving the camera images to obtaining the event's Unity pose. First of all, the images are obtained from the camera and a transformation of perspective is performed. Then the background is captured starting the moving objects recognition. Once the hand is obtained, the upper point of its contour is calculated, which is interpreted as the fingertip. Later on, the distance of that pixel is computed and compared to a threshold wall, distinguishing between click or slide events.

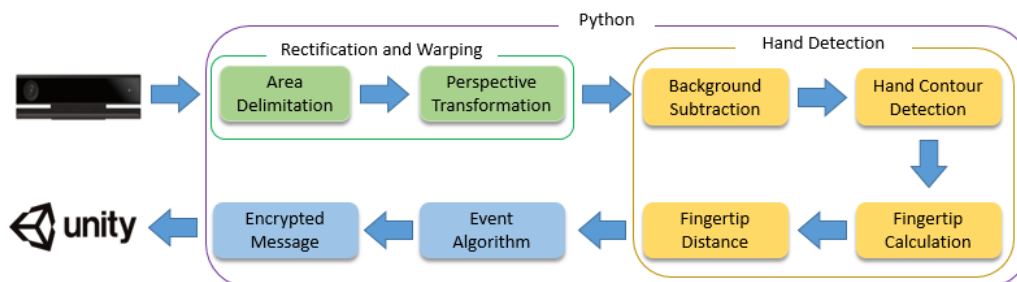


Figure 56. Hand Detection Flowchart

Then, the very first step working with the camera consists of accessing to its RGB and Depth images, which has been performed utilizing the Python Wrapper NtKinect[37]. This Dynamic Linking Library (DLL) lets the user to easily obtain the camera images through their specific functions. Examples of the camera captures are shown below.

As can be seen in Figures 57 and 58, and as expected from the camera's specifications, the RGB camera has a wider field of view than the depth one, capturing a larger part of the environment. Also, the differences in size are noticeable in both axes. Regarding the depth camera, it assigns to each pixel a tonality within the greyscale based on its distance to the camera's origin. For that reason, the pixels out of range are shown as black, which can be observed in Figure where part of the floor is not recognized. Consequently, the camera must be located within a certain distance range from the touch area, in this case a table.



Figure 57. RGB image example



Figure 58. Depth image example

Once the camera images are obtained, they must be flipped around the y-axis to rectify the horizontal mirror vision. The next step consists in detecting the fingertip and obtaining its position in pixels. This part could be accomplished with two diverse approaches, detecting the finger's position with the RGB images to be later matched with the depth camera to obtain the distance of that pixel or using only the depth camera.

While the hand recognition would be fairly easier with the RGB camera, it is proceeded to implement the second approach since the matching of both RGB and depth images is considered arguably more challenging due to the fact that the cameras aren't aligned, so they own different reference systems, the differences in both field of views and also because of the dissimilarity

in pixel resolution, 1920x1080 and 512x424px, respectively. Moreover, using only the depth camera also offers the advantage that since the image size is smaller, the implemented algorithm should be faster than on an RGB based frame. Therefore, each frame the depth information will be stored in uint8 (unsigned integer 8-bit), for visualization and algorithm purposes, and in uint16 to obtain the distance raw data.

Moreover, it is necessary to execute a perspective transformation due to the camera not being located precisely on the perpendicular of the surface, but with a horizontal angle or tilt. This method will usually transform a trapezoid shape into a rectangular area, which is more appropriate for image processing and position mapping, rectifying the existing distortion. This concept is described with the mathematical shown in the equation's system.

$$\begin{bmatrix} t_i x'_i \\ t_i y'_i \\ t_i \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \quad (1)$$

Where (x_i, y_i) represents the coordinates before the transformation and (x'_i, y'_i) after, which are related by the transformation matrix. This matrix is obtained using the OpenCV function `cv2.getPerspectiveTransform()` and it is applied to the previously delimited ROI with the function `cv2.warpPerspective()` resulting in the desired ROI with a suitable rectangular shape. A comparison of this transformation of perspective is shown in Figure 59 to illustrate the importance of this step.

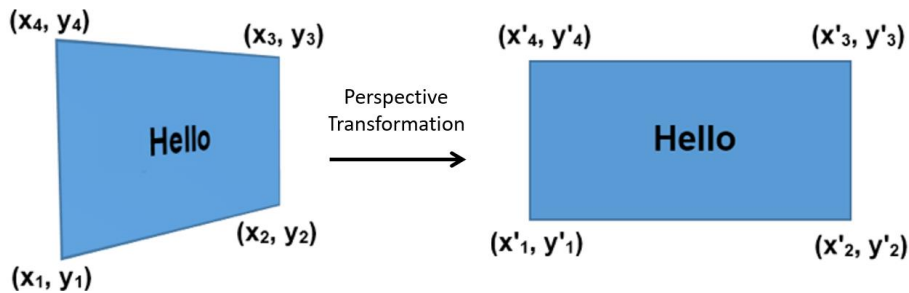


Figure 59. Perspective Transformation

Moreover, due to the camera having a field of view larger than the desired projection area, it is necessary to delimitate the ROI (region of interest). This is performed with an interface that manually lets the user select the four required vertices. This ROI will be the place where the system will be projected and, consequently, the touch screen. An example of a ROI with the corresponding perspective transformation can be seen in the subsequently Figure 60. The real ROI used for the final version of the system must match the projection of the projector, which means that the vertices of the ROI will be the vertices of the projected system screen. Therefore, these points must be carefully indicated because the accuracy of the algorithm depends on it.

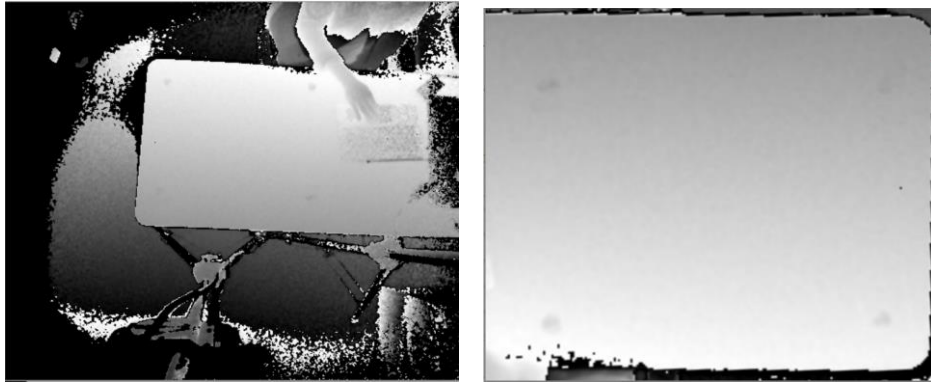


Figure 60. Example of ROI

Once the projected area has an appropriate shape for image processing, it is proceeded to identify the hand that will be carried out by a Background Subtraction Method. This technique is based on creating a forefront mask, which consists of a binary frame that contains the pixels of the moving objects in a certain region.

The modeling of this method relies on two steps. First of all, a model of the background is captured using the function `cv2.createBackgroundSubtractorMOG2()`, which contains the static parts of the ROI. Then, after computing this model, the new frames are compared to it using a defined threshold which removes the background with the initial model, resulting in the desired foreground mask highlighting the shape of the moving object, in this case the hand. It is also applied different morphological transformation using the function `cv2.erode()` in order to smooth the object shape. An example of this technique is shown subsequently in Figure 61.

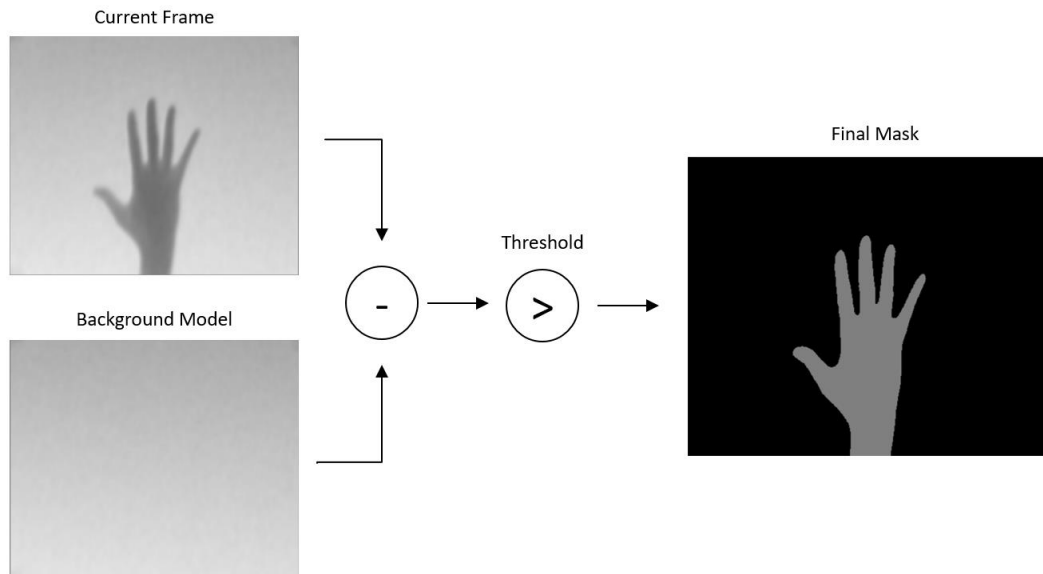


Figure 61. Background Subtractor Procedure

After that, a Gaussian blur is used along setting a certain threshold to convert the image into binary, with the functions `cv2.GaussianBlur()` and `cv2.threshold()`, respectively. Then, the method `cv2.findContours()` is applied to the obtained mask in order to only retain the contour of the hand, which is the principal interest in order to get the position of the finger. Then, it is enough to compute the indices of the pixel with the minimum value, taking into account the camera's reference system, among all the contours pixels along the forward axis to find the position on the fingertip.

The result after applying this procedure is illustrated in Figure 62 and, as expected the computed point is in the middle finger since it is the longest (left image). Consequently, this fingertip's detection requires a certain hand configuration to be effective, which simply consists in closing the hand except for the index finger such as it is illustrated on the right image, or any other configuration where the desired finger is ahead of the rest.

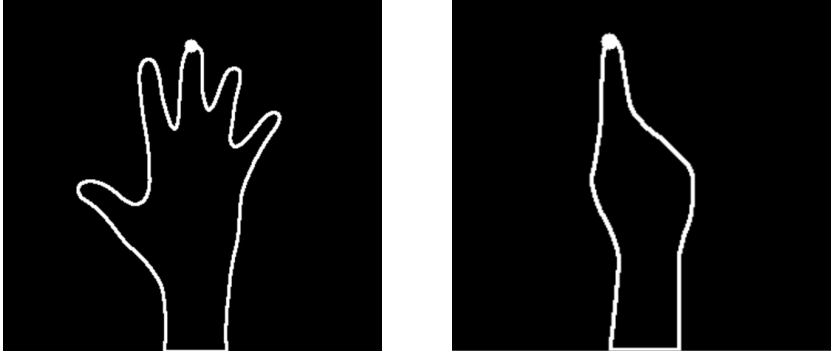


Figure 62. Detection of the fingertip

Once the coordinates of the fingertip are known, the depth value from the plane at the sensor's position is immediately obtained in millimeters through the wrapper, which relies on raw data with no extra calculations. Then, this value is compared to the same pixel's depth in the initial model resulting in the distance between the fingertip and the touch surface and that difference is compared to a threshold wall of 20 mm that triggers the start of an event, which is illustrated in Figure 63.

Increasing the wall value will reduce the distance needed to activate an event and vice versa and as it can be seen, it is not strictly necessary to touch the surface to trigger the event, but being close enough to it. Even though the optimal solution would be to only detect the touch when the finger certainly touches the surface (threshold equal to zero), it was not possible to implement it due to the camera's resolution, because the quality of the depth data falls off as you step away from the sensor and also due to the associated noise to the raw data.

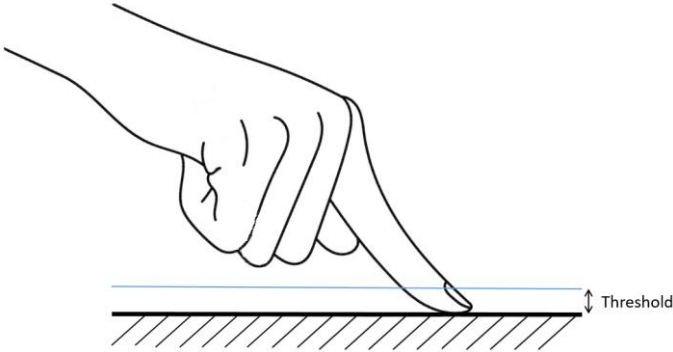


Figure 63. Fingertip's Threshold

Later on, it is proceeded to discern between the two contemplated events, a click and a slide, whose flowchart is shown in Figure 64. As it can be seen, a click is detected when a finger crosses the threshold and comes back within a certain time while the slide is indicated when the finger crosses it and stays there longer than the predefined time. The algorithm calls two functions and should not make use of any loop (arrows coming back) since it is invoked every frame and, therefore, including them would mean the loss of information from the depth camera, so only it has been decided to only involve if statements.

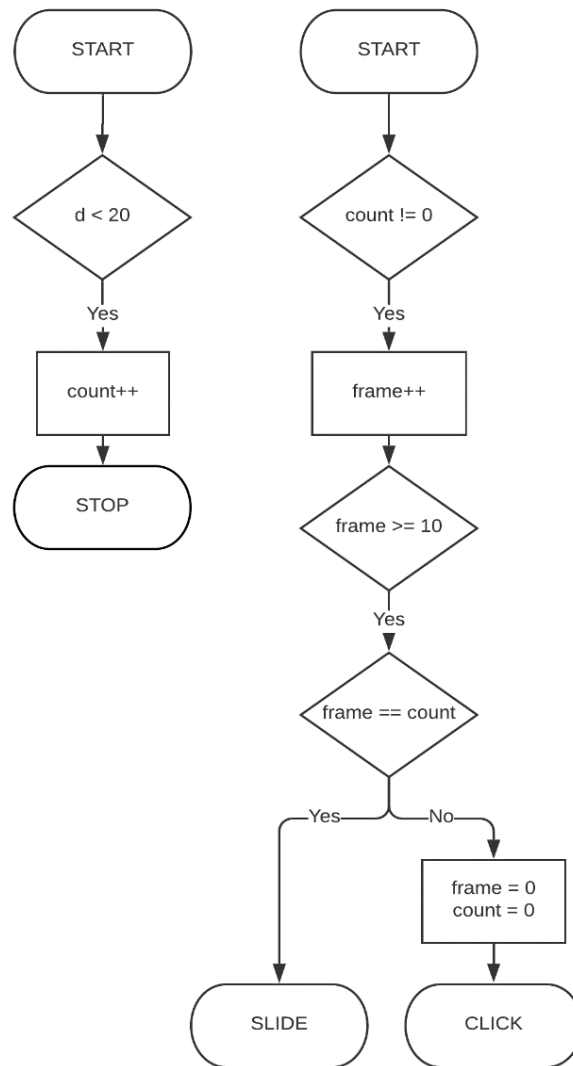


Figure 64. Algorithm to discern click or slide

Since the camera framerate is 30 FPS (Frames Per Second), the counter is compared with 10 frames in order to debounce the clicks to 3 per second, which is necessary to avoid selecting adjacent elements. It has been tested that third of a second is enough time to bring the fingertip close to the surface, touch it and lift it again since a higher threshold would unnecessarily increase the delay of the click detection, while a lower wall would increase the number of fake signals detected on each click. These fake signals would compromise the adequate functioning of the elements, since a real click would turn into multiple event triggers in the application and, therefore, must be minimized. To conclude, setting a maximum of 3 events per second offers the best performance among the previous concerns.

Once an event is detected, a message is sent to the application via socket containing the corresponding pixel position along with an identifier to discern the event triggered. The usage of this encrypted information will be further detailed in the subsequent integration sub-chapter.

5.5. Robot

In order to utilize it, an iOS or Android mobile must be connected to the robot's WiFi and run its app, enabling the SDK mode. Then, the mobile must be connected to a computer via USB while the robot is correctly placed on its charging platform. When this configuration is set, the robot can be controlled properly executing the corresponding Python commands on the computer provided by the Anki Developers and, as a result, it is able to perform the required movements, facial expressions and sounds through its speaker. This configuration has been illustrated in Figure 65.



Figure 65. Cozmo configuration

The robot code is divided into different methods that invoke several sets of actions. These actions have been carefully selected in order to increase the involvement grade of the robot in the system, reinforcing its role within it with the aim of not going unnoticed during the course of it, resulting in a more enjoyable and effective experience for the user.

The main actions utilized lies in moving the robot to a certain point of the surface, rotating about the z-axis, using its speaker to give information to the user, moving its lift and head up and down along with playing pre-set animations that combine all these possibilities. All these actions are predefined in the Cozmo API, which eases the process of controlling the robot. Then, the fundamental issue consists of the customization of these methods through the function arguments and the adequate order of invoking them in order to maintain a coherent training flow.

Moreover, some robot actions such as talking and moving around the training area have been coded to be executed in parallel, to speed up the lesson and rise its dynamism and engaging feeling. This configuration has been set to all the concurrent actions that do not conflict with each other, which includes any pair of actions that make use of the same robot characteristic such as using the speaker and also playing an animation that involves the speaker.

The tone voice of the robot is also a critical characteristic of the experience, due to the fact that it is used to communicate with the user giving information and interacting with him. However, the default voice of the robot's speaker is, in most cases, unintelligible when reciting complete sentences aside from a few simple words. For that reason, its pitch has been tuned balancing the deep and acute tones, obtaining a completely different voice from the initial and, this time, clearly understandable.

5.6. IPC communication

In this project, the pattern request-reply is executed to transmit information between the C# and the Python scripts. This kind of arrangement allows the application to trigger events with signals obtained from the camera and to control Cozmo adequately transforming C# scripted messages into Python commands intelligible by the robot. This pattern is illustrated in Figure 66.

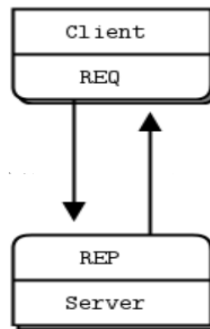


Figure 66. Socket pattern

The socket REQ-REP is in lockstep, which means that the client and the server communicate among them sending and receiving messages in a concrete order in a loop during the lifetime of the application and, therefore, other sequences such as the client sending two messages in a row will turn into an error. In this case, the application takes the role of the client and Python acts as the server.

5.6.1. Functions

It is proceeded to describe the most significant functions involved in the integration between both sides (Python and the application) and, therefore, are the socket's methods. These functions are responsible for converting camera events into events and those into robot movements. It must be noted that in order to keep the adequate order in the ZeroMQ protocol, at the end of all these functions, a message must be sent to the other side to avoid unintentional errors that could crash the application. The main ones are briefly covered subsequently.

- `MessageHandler()`: This method is defined on each side and it is called each time a new message is received from the other side of the socket. It decodes the given data and compares it with several predefined messages, triggering the corresponding set of actions.
- `ClickHandler()`: This function is called when the `MessageHandler()` discerns that the received message comes from the camera and not from the robot's thread. It is in charge of converting the fingertip's position in the camera's reference system to the Unity scale. Then, depending on these values and the lesson state, it also invokes one of the subsequent four options to activate specific actions.
- `SelectComponent()`: It is invoked if the detected click is on the selecting area, instantiating a component on the selector when the corresponding sprite is touched. Then, in order to distinguish if the component has been chosen, the distance between the click and the closest sprite is computed and compared to the threshold. After creating the component, the function also assigns a text with a certain value, accordingly to the dropdown selection.
- `Lesson()`: It manages the removal and short circuits of the components during the explanation of the robot, after looking for the closest electric element, which is approached using a distance method once again.

- `CallClosestObject()`: It is responsible for interacting with the UI elements based on the fingertip's position. First of all, it computes the closest UI component to the click, which in this case are buttons and the dropdown. Then, it checks if the click is actually on the element, which has been approached by calculating the component's corners and checking if it contains the click. To do that, the predefined function `GetWorldCorners()` has been used along with several transformations between the screen and world scale, since the UI elements have a different scale than the rest of the components. Finally, if the element is certainly clicked it is invoked, performing the associated function.
- `SlideFunction()`: This method manages the slide event of the component from the selector to a congruent location on the panel, which once again it has been accomplished using a distance approach. Then, if a slide is received from the socket it compares the distance between the fingertip and the new component with a threshold, moving its position with a linear motion until the slide event is not detected on the camera.
- `Start()`: This function is invoked when launching the application, launching the client side of the socket in order to enable the communication with the server in Python. It also establishes the required connections with the rest of the scripts instantiating objects of the corresponding classes.
- `Update()`: In this case, the update function is utilized just to smooth the slide functionality, creating a linear uniform motion instead of instantly changing the `GameObject`'s position of the dragged component each time to the received position from the camera. This has been approached using the predefined method `Vector3.Lerp()` between two consecutive fingertip's positions based on their distance and the desired movement speed.

5.6.2. Camera – Unity

It is important to note that both the camera sensor and Unity use dissimilar reference systems, so it is considered essential to map the coded messages from the camera in order to transform the camera pixels where the fingertip triggers an event into Unity coordinates to be able to interact with the corresponding elements. The messages from the camera have been coded with three parameters, a letter designing if it is a click or a slide, and two numbers representing the position of the fingertip in the camera's axes, so it must be decoded and handled by a Unity's function so the positions are coherent in the application's environment, using the following transformations.

These systems are shown in Figure 67, where X_u and Y_u represent the double of the maximum values of the axes in the interface, respectively, since its origin it is on the centre of the interface area unlike the depth camera, which is located in the upper left corner.

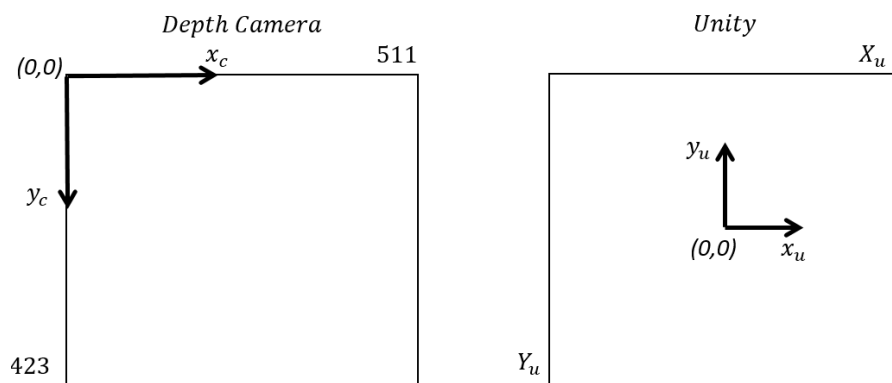


Figure 67. Depth Camera and Unity reference's systems

Then, it is necessary to perform several linear transformations in order to create events in the corresponding location to where the event was triggered in the touch area with the fingertip. These conversions are shown below, taking into account the reference systems of both parts and the required scaling. The required theoretical mathematical expression is described in the equation.

$$\begin{pmatrix} x_u \\ y_u \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x_c \\ y_c \end{pmatrix} + \begin{pmatrix} x_o \\ y_o \end{pmatrix} \quad (2)$$

As it can be seen, the resulting equations of the fingertip position conversion conform a 2x2 system since it a planar transformation, where (x_u, y_u) are the Unity coordinates, (x_c, y_c) the camera coordinates and (x_o, y_o) represents the offset values among both origins.

Regarding the matrix, there is no relation between the horizontal and vertical axis, which means that its non-diagonal values must be null, so $a_{ij} = 0$. On the other hand, the diagonal values must represent the rotation and the scaling between both systems, so the element a_{11} must be the scale factor between both x-axis and positive since the axis are aligned while the element a_{22} must be the scale factor among both y-axis and negative since the axis has contrary directions. In order to keep the scale and proportions, these factors must be the relation between the interface size in Unity, identified by X_u and Y_u respectively, and the projection area with the camera, identified by the number of pixels on each axis, width and height respectively.

Finally, the offset values represent the distance between both origins, which are directly half the interface area size in Unity on each axis. The final system that converts the fingertip's position from the camera reference to Unity is shown in the equation.

$$\begin{pmatrix} x_u \\ y_u \end{pmatrix} = \begin{pmatrix} \frac{X_u}{width} & 0 \\ 0 & -\frac{Y_u}{height} \end{pmatrix} \begin{pmatrix} x_c \\ y_c \end{pmatrix} + \begin{pmatrix} -X_u/2 \\ Y_u/2 \end{pmatrix} \quad (3)$$

Then, when the fingertip's pixel position is sent to the application, this transformation between the reference systems of the depth camera and the interface is realized. After that, the application verifies the possible event in the corresponding location and it acts accordingly.

5.6.3. Unity – Robot

The robot code is divided into several modes of action that are activated depending on the current stage of the lesson. These phases are pre-set in both Unity and Python sockets in a concrete way that lets the one and the other to know precisely in which point the lesson is. This is considered essential so as all processes run coordinated assisting a precise communication among both parts.

Therefore, when the lesson reaches certain points, the client sends specific messages to Python. Then, one method decodes the received message and invokes the corresponding robot action mode, performing the adequate set of movements the lesson requires at that moment to continue the training flow. This method first discerns between the training modes to increase the response time and, at the same time, the efficiency of the code avoiding unnecessary comparisons.

Regarding the robot position, a similar procedure to the camera integration will be carried out due to the application and the robot using different reference systems, but this time the process

will be the other way around since the Unity poses will be mapped to the robot coordinates in order to move it accordingly to the desired location in Unity, increasing the robot inclusion feeling in the system. Both reference systems are shown in Figure 68 along with a generic initial position for the robot. The relative x-axis of the robot is placed on its forward direction while the relative y-axis is situated on its left side, where X_r and Y_r delimitate the real touch surface and interface projection.

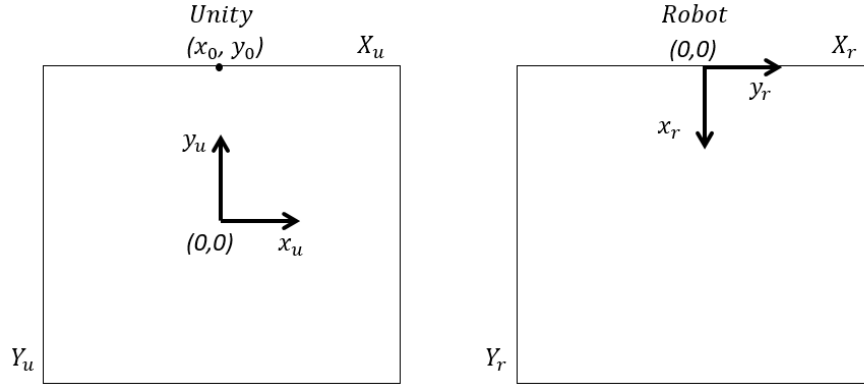


Figure 68. Unity and robot reference systems

It is important to note that the robot can move using two different coordinates systems, one is static whose origin is set to its initial position and could be modified each time the robot is picked up and moved to another surface and the other is fixed to the robot. That means that the movements made by the robot will be always incremental and, therefore, relative to the robot's position at all times and not global distances as in the first case.

In this case, the first method is chosen to due to the complexity of the code to control the robot is decreased since it is enough to indicate global positions to the robot while the second method would require the track of the robot position, increasing the difficulty of the camera's integration in order to perform incremental movements to reach the same position. This also means that the first method relies entirely on the robot odometry (dead reckoning), which results to be accurate enough to fulfil the movement requirements that the lesson demands.

$$\begin{pmatrix} x_r \\ y_r \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x_u - x_0 \\ y_u - y_0 \end{pmatrix} \quad (4)$$

Once again, the resulting equations conform a 2x2 system due to being a planar transformation, where (x_u, y_u) are the Unity coordinates, (x_r, y_r) the robot coordinates and (x_0, y_0) represents the initial position of the robot in the Unity coordinates, while the matrix describes the scaling and rotation of the axes.

In this case, and unlike during the camera integration, the diagonal values are null due to not existing relation between both horizontal nor vertical axes, so $a_{ii} = 0$. Then, the non-diagonal values must represent the required axes scaling and rotation among the two reference systems. Consequently, they represent the conversion among the Unity and robot axes, so these scale factors must be the relation between the projection area size, identified by X_r and Y_r respectively and the interface size in Unity, identified by X_u and Y_u respectively. Moreover, the element a_{12} is negative due to the associated axes have opposite directions, while the a_{21} is positive since both axes are parallel. This conversion is illustrated in the equation.

$$\begin{pmatrix} x_r \\ y_r \end{pmatrix} = \begin{pmatrix} 0 & -\frac{Y_r}{Y_u} \\ \frac{X_r}{X_u} & 0 \end{pmatrix} \begin{pmatrix} x_u - x_o \\ y_u - y_o \end{pmatrix} \quad (5)$$

Once the transformation is accomplished, the movements of the robot are controlled sending from the application the positions where the robot locates itself, decoded in Python and assigned to the corresponding method that moves Cozmo to the corresponding position in the projection area.

5.6.4. Robot – Camera

The main concern regarding this part of the integration is the discern between the hand and the robot with the camera, since both elements are identified as moving objects in the projection and, therefore, are compared to the captured background model. Then, in order to only recognise the hand, so the movement of the robot on the ROI does not affect the progression of the training experience sending fake signals, it is proceeded to implement the following procedure.

Taking into account that only two different contours will be recognised each frame, associated with the robot and the user's hand, then it will be neglected the contour of the element whose upper point has a lower index in the camera y-axis, belonging to the robot and remaining the contour of the hand. This kind of arrangement is plausible due to the fact that during the designed progression, the user's hand will never surpass the robot's position.

Moreover, another functionality endowed to the robot is to move towards the direction where the user is touching the screen. In order to manage this behaviour, it is essential to convert the camera data into robot movements, which means to find a correct transformation between both reference systems and then act accordingly to the events detected by the camera. These reference systems are shown in Figure 69.

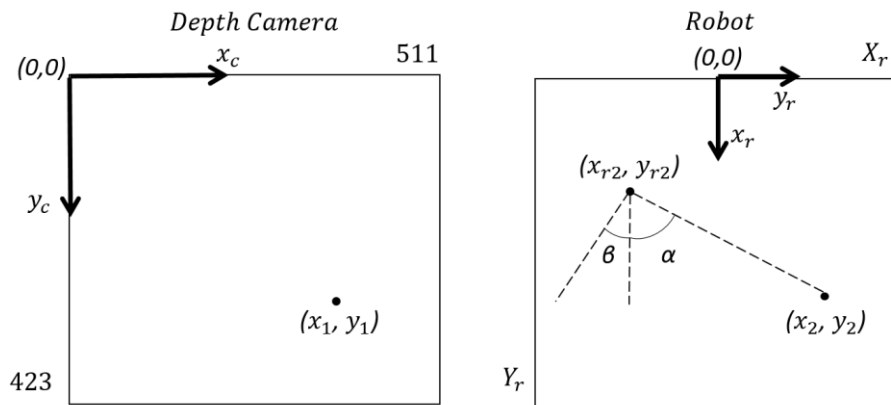


Figure 69. Depth camera and robot reference systems

Then, when this function is executed and the user touches a generic position (x_1, y_1) in pixels, it must be transformed into the corresponding position in millimetres (x_2, y_2) , legible by the robot. Therefore, and similarly to the previous transformations, a rotation with scaling will be applied followed by a translation.

The diagonal values of the matrix are null since the axes are inversely related, being parallel to the other axis, so $a_{ii} = 0$. Consequently, the non-diagonal values represent the conversion

among the camera and robot axes, so these scale factors must be the relation between the camera pixels, width and height, respectively and the screen size in millimetres, identified by X_r and Y_r respectively. After that, a final translation must be applied in the x-axis with a value of $-Y_r/2$ to amend the distance differences between the origins of both system. The matrix values are positives since the axes present the same orientation while the translation is negative because the camera origin is on the negative side of the x-axis. This conversion is mathematically illustrated in the equation 6.

$$\begin{pmatrix} x_r \\ y_r \end{pmatrix} = \begin{pmatrix} 0 & \frac{X_r}{height} \\ \frac{Y_r}{width} & 0 \end{pmatrix} \begin{pmatrix} x_c \\ y_c \end{pmatrix} + \begin{pmatrix} -Y_r/2 \\ 0 \end{pmatrix} \quad (6)$$

Once the user's click location is known in the robot domain (x_2, y_2) , the next step consists in calculating the angle between it and the current robot position (x_{r2}, y_{r2}) . This angle α is obtained using the arctangent between both points. Finally, the relative angle that the robot must perform will be the calculated value α minus its current rotation, illustrated in Figure 69 by β . This relation is shown in equation 7.

$$Angle = \tan^{-1} \left(\frac{y_2 - y_{r2}}{x_2 - x_{r2}} \right) - \beta \quad (7)$$

For example, if the robot is looking completely forward, the angle β will be null and only the angle α will be taken into account., or if the click is done in front of the robot, α will be null and the robot will only correct its current rotation.

5.6.5. Camera – Robot – Unity

With the aim of developing a solid integration that successfully combines the Python with the application, it is proceeding to implement several threads in order to run the Python parts (camera and robot) separately. It is considered essential to include threading because the ZMQ protocol locks the code execution until it receives the corresponding message from the other side, which would negatively affect the desired performance of both processes, resulting in causing an error at some point of the lesson progress. Then, it is concluded that it is necessary to implement at least three threads, one in Unity and two in Python.

On the one hand, the socket's client is defined in Unity's socket and it is responsible for the communication with the Python side. This thread is required to prevent the main Unity thread from crashing due to, as mentioned before, the ZMQ protocol locks the code until a message is received. It is implemented in a different Unity script and contains all the needed methods to decode the messages from the Python side so as to trigger the associated events such as invoking a button or dragging a component. The general flowchart of the socket script is shown in Figure 70.

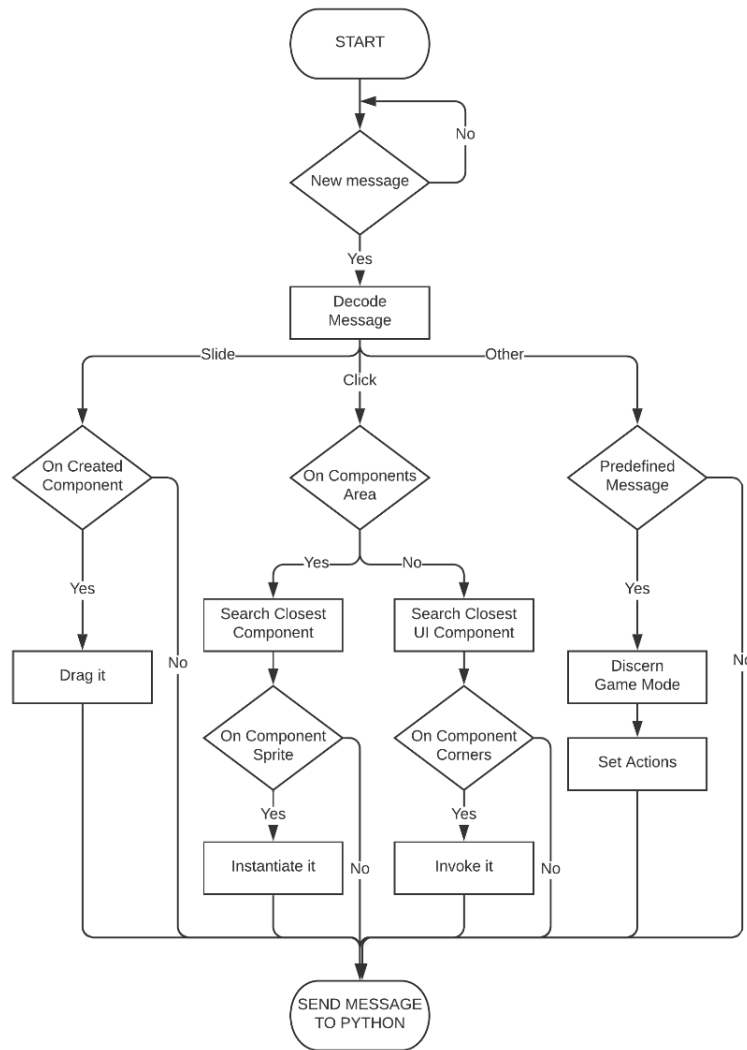


Figure 70. Unity's Socket Flowchart

On the other hand, the Python side is divided into two threads, associated to the camera and the socket's server. The camera's thread runs independently and broadcast to the socket whenever it detects a new event, either a click or a slide. Simultaneously, the socket thread waits for messages from the Unity client or the camera, checks them with pre-set actions and controls the robot sending a message back to the application, which is analysed by the Unity socket and so on. This configuration is illustrated in Figure 71.

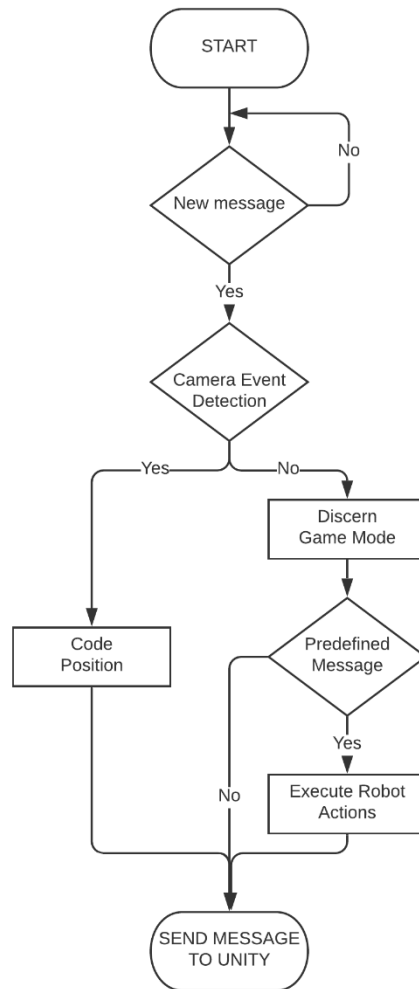


Figure 71. Python's Socket Flowchart

It is also worth mentioning that the totality of the Python side is encapsulated in one unique script. Moreover, it has been designed in a way that the robot movements and the socket are controlled in the same thread, which means that both processes are concurrent in time and, therefore, the socket does not send messages to the application until the robot has finished its actions. This configuration has been created to reduce the complexity of the Python code instead of creating a third thread, separating either of those and increasing the number of state variables needed to let every thread know what happens during the lesson at every moment, so they can act accordingly. It also affects the camera indirectly, due to the camera cannot either send a message until every processes are in the idle state.

5.7. System Deployment

As has been previously described, the hardware required to play this project consists of a projector, a camera, a computer, a smartphone, several connectors, the robot Cozmo and a planar surface. Regarding the software, the main core consists of an application developed in Unity and a Python script, as well as several libraries and dependencies that the Python script needs. As a result, its portability level could be considered very low.

Both the camera and the projector must be connected to the power and the computer, the camera through a USB 3.0 port required to fulfil the data stream requirements providing a stable connection while the projector must be directly connected through the HDMI or using a VGA adapter just to visualize its screen. Additionally, the phone must be connected to the computer

via normal USB and to the robot via Wi-Fi, and also running the Cozmo app in SDK mode to process the Python commands.

Once everything is set, the Python script must be first executed, popping up two windows showing the vision of the depth and RGB camera. The RGB is added to the WoZ vision (previously converted to greyscale) while the depth window is used to delimitate the touch area, since the camera's field of view is larger than the desired one, performing a transformation of perspective to match the touch area with the camera reference and correct the possible tilt. Then, the application developed in Unity must be executed, launching the menu screen where the training mode can be chosen. Finally, when the LxT mode is chosen, a web page pops up shutting down the IA control of the robot so the wizard is able to manually control the robot's actions. An example of the projection during the LxT mode is shown in Figure 72.

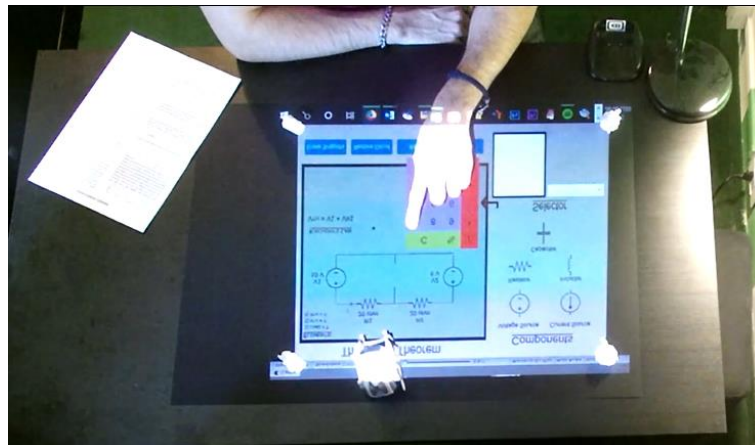


Figure 72. Deployment example

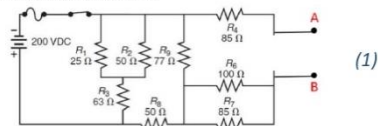
5.7.1. Manual

While in the TL the students usually attend the classes with zero or very little knowledge about the corresponding topic, during the LxT more it is the student who takes the active role and prepares the subject by himself with the available resources, prior to the evaluation. Therefore, in order to imitate that behaviour as faithfully as possible, a manual has been concisely prepared for the user, which should be revised prior to initiating the second mode.

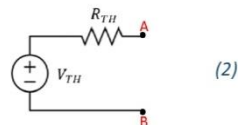
This manual contains a detailed explanation of the Thévenin Theorem, the topic of the lesson. Even though the lesson has been designed to be quite straightforward following a specific order, some points bring certain level of freedom so the user can lead the class in their own way. For that reason, it is also important to find a balance on its extension because it should be long enough to let the user adequately carry the lesson and understand the concept, but not too long to avoid loss of interest from him due to excessive information. This knowledge cheatsheet can be seen in Figure 73.

Thévenin's Theorem - Cheatsheet

Thévenin's Theorem (TH) states that "Any linear circuit containing several voltages and resistances can be replaced by just one single voltage in series with a single resistance connected across the load". It is probably one of the most interesting for linear circuit (combination of DC Current and DC Voltage generator and Resistors) in that it permits the reduction of complex networks to a simpler form for analysis and design. TH allow to reduce any linear circuit given two terminals A and B like this:



Replacing it with an equivalent - voltage generator in series with a resistor - which is far easier to manage for further uses, such as the next one:



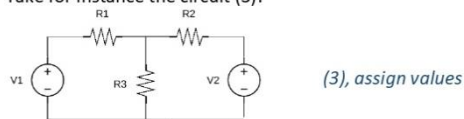
In other words, we need to find V_{th} and R_{th} in (2) thus the electric variables insisting on terminal A-B are the same to the ones in the circuit (1).

This can be done using the e.g. the so-called two-step calculation procedure, that can be summarized as follows:

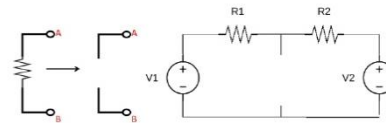
- (Remove the load between the objective A-B from the original circuit, so the equivalent is independent from it).
- Find R_{th} : by removing all power sources in the original circuit (short circuit voltage sources and open circuit for current sources) and calculating total resistance between A-B
- Find V_{th} : by calculating the voltage across the open connection points where the load resistor used to be, using circuit analysis, e.g. KCL/KVL.

The two steps can be performed in any other.

Take for instance the circuit (3):

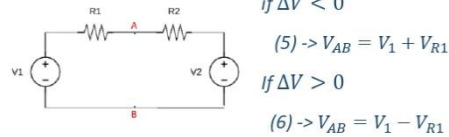


First the load must be removed, in this case it will be the middle resistor R_3 . Also, a buffer (4) can be added as a reminder.



Find V_{th} : V_{th} is the voltage drop between those terminals. Then we can calculate that using e.g. Kirchoff's Law (KVL) must be applied on the left loop along with Ohm's Law on R_1 . Alternatively, you can use KCL, Millman or Superposition method. KVL states that the directed sum of the potential differences (voltages) around any closed loop is zero. In the Example, we have simple series, hereafter calculate the current in the whole circuit then you can use just one side of it to find V_{ab} (V_{th}).

$$\Delta V = V_1 - V_2, \text{ if } \Delta V < 0$$

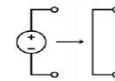


To find I simply apply the Ohm's Law to the series:

$$I = \frac{|\Delta V|}{R_1 + R_2} \quad (8) \rightarrow I = \frac{|\Delta V|}{R_1 + R_2}$$

$$V = R \cdot I \quad (7) \rightarrow V_{R1} = I \cdot R_1$$

Find R_{th} : starting from (3) without R_3 , all active elements must be neglected or, in other words, to short circuit generators and open current sources.



After that, the equivalent resistance can be calculated as the resistance seen by the terminals A-B. Compute that number. Resistors in parallel formula:

$$\frac{1}{R_{eq}} = \frac{1}{R_1} + \frac{1}{R_2} \quad (9)$$

So $R_{th} = R_1 // R_2$.

Finally, we can draw the equivalent with the values obtained selecting the components and assigning the values we found.

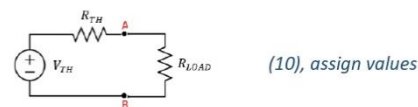


Figure 73. Domain Knowledge Cheatsheet

Chapter 6

Experimental Evaluation

In this chapter are described activities carried-out to perform a preliminary user study to validate the developed system.

6.1. Methodology

From the review of the state of the art, two observations can be made. First, although the LxT model has been extensively analysed in different contexts and disciplines (even including robots in the experience) there is still a lack of studies conducted for university subjects such as Science, Technology, Engineering and Mathematics (STEM) high-level education. Second, it has not been found any study that investigated this topic on a full MR environment, since the most similar project [1] did not create a real interaction with the digital content. Therefore, this project decided to take a step forward in this field of study and examine the impact of the LxT approach by combining the MR technology with the robot Cozmo as the teachable or teacher agent respectively, developing the required tools to carry out the experiment. Then, the research hypotheses that guided this work are:

- Is there a significant difference in the user's learning between the TL and LxT models in a MR robotic environment?
- Is the developed system able to effectively support the user's learning in both modes?

Several secondary evaluations have been performed in order to assess the usability of the system and the inclusion level of the robot.

6.1.1. Research Design and Sample

The present research used a between-subject design, which consisted of an experiment where participants are studied before and after the experimental manipulation. Therefore, the participants are divided into two groups, the TL and LxT groups. The purpose of this design is to verify if the research topic has caused a change in the participants since any changes observed across the group after the manipulation is likely associated with the training experience.

This means that the participants are tested before exposing them to the system to check their abilities/knowledge of the research topic, then they undergo the experimental manipulation to be finally tested again to corroborate any possible induced alteration with respect to the original ability/knowledge rated on the pre-test. The experimental procedure carried out is illustrated in Figure 74.

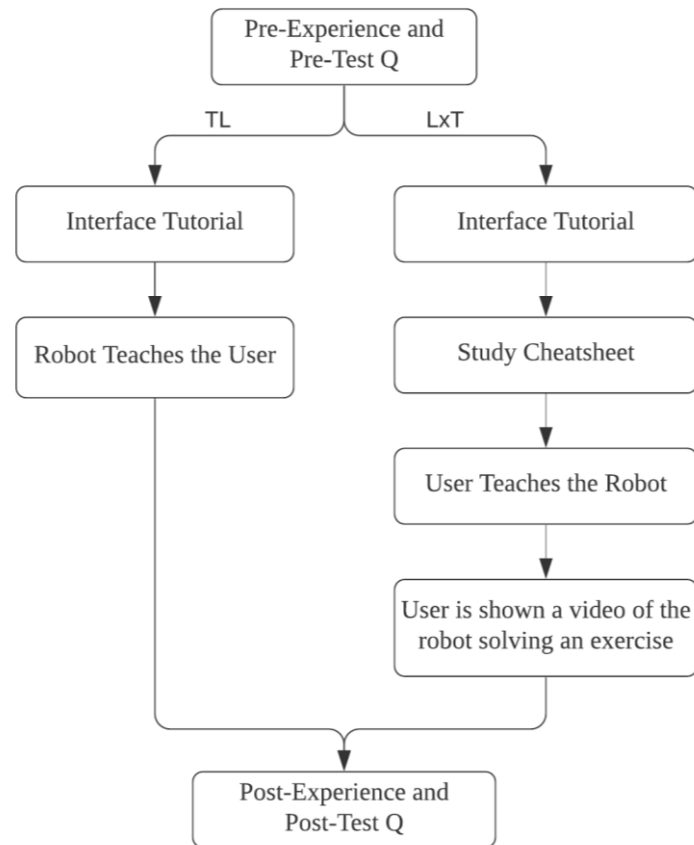


Figure 74. Experimental Procedure

In this case, a total of 6 engineering university students, aged between 22-25 years old participated in the study. The composition of the experimental group was 3 (Males = 3, Females = 0) and the experimental group was 3 (Males = 3, Females = 0). The control group followed a lesson of the Thèvenin Theorem taught by the robot, representing the TL model, while the experimental group was asked to teach the robot the same lesson, exemplifying the LxT model. Moreover, both groups followed slightly different tutorial modes, since the participants have available diverse functionalities on each learning mode, aside from were given the same pre-tests and post-tests.

Regarding the LxT group, prior to starting the lesson and after the tutorial, they were asked to study a cheat sheet about the Thèvenin's theorem, the topic of the lesson. Even though this help was available during the mode, understanding its concept and analysis was essential for teaching it fluently to the robot, avoiding undesired uncertain moments. Therefore, the users had to master the theorem before starting the lesson.

Finally, it is worth mentioning that the time spent for each participant during the crucial parts of the experimental procedure has been controlled, with the aim of adequately measuring the results extracted from the tests. Moreover, this information could also explain possible discrepancies between each participant's performance just in case their performances were dissimilar enough.

6.1.2. Instruments

The questionnaires implemented aimed to gather information about the users' perceptions about the system experience, as well as measuring their knowledge acquired after being exposed to it. These questionnaires are shown in this document following the same procedure of the experiments.

First, the pre-experience has been evaluated with the Tables 1-5. Table 1 focused on knowing the user's normal usage of the technologies involved in the project. The objective of the Tables 2-3 was to collect the learning preferences of the users in general, while the Tables 4-5 related with the project learning topic, the theorem from the electrical domain. Additionally, a pre-test must be fulfilled to collect data about the participants' prior knowledge about the topic, essential to adequately measure the possible knowledge acquisition promoted by the system exposure. The pre-test is illustrated in Table 6 and 7, along with the involved circuits (Table 8).

Please answer using numbers from 0 to 4 (Not at all: 0; Once or Few times: 1; Sometimes/Once a month: 2; Often/Once a week: 3; Every day: 4)	
1	How often do you play video games?
2	How often do you use touch screen interfaces (e.g. smartphone)
3	How often do you use/interact with toy robots (like Cozmo, Sphero, etc.)?
4	How often do you use/interact with other kinds of social robots, such as home robots (like Roomba, etc.)
5	How often do you teach to other people?

Table 1. Technologies related

Rate on a scale from 1 to 5 (Strongly disagree: 1; Strongly agree: 5) this statements on YOUR study's habits	
6	I study where it's quiet with little distraction
7	I study while watching TV or listening to music at home
8	I take note during class
9	I study with a friend or group
10	I put my notes in my own words to understand better
11	I put my notes copying as much as possible the teacher explaining during class
12	When I am learning a principle or definition, I try to think of at least two examples of how it might be applied or used (if not provided)
13	When I am learning material, I ask myself questions and study until I can give the answer, on two or three separate occasions, without looking at the text or my notes

Table 2. Learning preferences I

When you don't understand what you are learning in class, you usually	
14	ask to teacher during class
15	ask to teacher after class
16	ask to a colleague
17	ask "Google" (Research online)
18	ask provided Reference book

Table 3. Learning preferences II

Preferred method evaluation on electronics method (1-hate; 5-love to)	
19	Multiple Choice
20	Solving Exercise
21	Oral exam
22	Prepare a "presentation"
23	Preparing a project/assignment

Table 4. Evaluacion preference

Learning expectancy - Rate on a scale from 1 to 5 (Strongly disagree:1; Strongly agree: 5)	
24	I think I know (right now) all the theory about Thevenin's theorem
25	I think I know (right now) all the information required to simplify a circuit with the Thevenin's theorem
26	I think I can successfully pass a test on a Thevenin's theorem without further training

Table 5. Learning expectancy

General		6 Find the current in the circuit N° 1	
1	In a series circuit, which of the following parameters remain constant across all circuit elements?	A	1 A
A	Voltage	B	2 A
B	Both voltage and current	C	3 A
C	Current	D	4 A
D	Neither voltage nor current	7	The sum of the voltages over any closed loop is equal to?
2	A voltage across a series resistor circuit is proportional to?	A	0 V
A	The amount of time the circuit was on for	B	infinity
B	The value of the resistance itself	C	10 V
C	The value of the other resistances in the circuit	D	It depends on the loop's components
D	The power in the circuit	8	Calculate the voltage across the 10 ohm resistor in the circuit n° 2
3	What happens to the current in a series circuit if the total resistance is doubled?	A	12 V
A	Insufficient data provided	B	4 V
B	It becomes double its original value	C	10
C	The current does not depend on the resistance	D	Insufficient data provided
D	It becomes half its original value	9	What size resistor would produce 3 A current flow with a 12 V battery in series?
4	The equivalent resistance for two resistors in parallel is	A	3 ohm
A	$R_1 + R_2$	B	4 ohm
B	$R_1 * R_2$	C	12 ohm
C	$(R_1 * R_2) / (R_1 + R_2)$	D	36 ohm
D	$(R_1 + R_2) / (R_1 * R_2)$	10	If the current across a resistor is constant, what will happen to the voltage if the resistance is increased?
5	What is the voltage measured across a series short?	A	The voltage will decrease
A	Infinite	B	The voltage will increase
B	Zero	C	The voltage will stay the same
C	The value of the source voltage	D	Insufficient data provided
D	It cannot be calculated		
Kirchoff's and Ohm's law			

Table 6. Pre-Test 1, general and laws concept

Thèvenin Theorem			
1	The thevenin voltage ..	3	The thevenin equivalent consists in
A	Is the open circuit voltage across the load	A	A voltage source in series with a resistor
B	Is the short circuit voltage across the load	B	A voltage source in parallel with a resistor
C	Depends on the load's resistance	C	Two voltages sources in series with a resistor
D	Is always different from the circuit voltage sources	D	A current source in series with a resistor
2	The thevenin resistance is found by ...	4	Thèvenins theorem is applicable to ...
A	Shorting all voltage sources	A	Linear networks
B	Opening all current sources	B	Non-linear networks
C	Shorting all voltage sources and opening all current sources	C	Both linear and non-linear networks

D	Opening voltage sources and shorting all current sources	D	Networks with at least 5 components
5 Calculate the Thevenin resistance across the terminals AB for the circuit n° 3			
A	4,34 ohm	C	3,43 ohm
B	3,67 ohm	D	2,32 ohm

Table 7. Pre-Test II, Thévenin concept

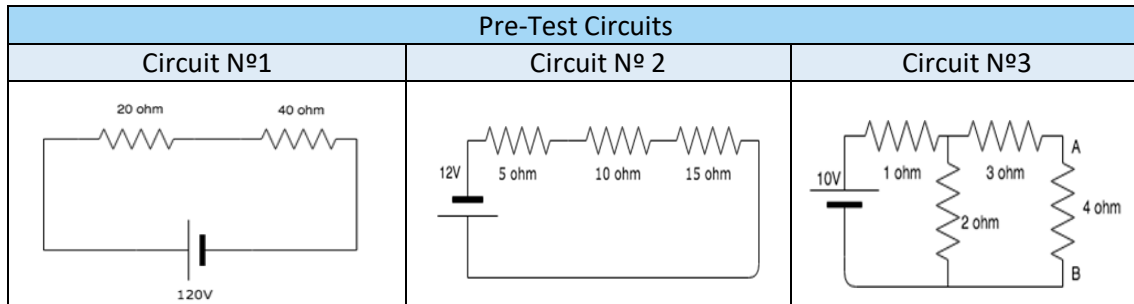


Table 8. Pre-Test Circuits

After the experiment, the post-training was also measured using several questionnaires. First, it has been collected data about the user's perception regarding the robot, following these guidelines [38]. This study proposed a standardized measurement tool for human-robot interaction focusing on five principal concepts, anthropomorphism, animacy, likeability, perceived intelligence and perceived safety. These attributes were assessed using several semantic differential scales, which are shown in Tables 9-10, along with reliability and validity indicators. Furthermore, additional statements were added in order to measure how the performance and inclusion level of the robot were perceived by the participants (Table 11).

Rate on a scale from 1 to 5 (Leftmost/Rightmost, 3 neutral)					
Anthropomorphism		Animacy		Likeability	
27	Machinelike-Humanlike	32	Artificial-Lifelike	38	Awful-Nice
28	Artificial-Lifelike	33	Dead-Alive	39	Unpleasant-Pleasant
29	Fake-Natural	34	Stagnant-Lively	40	Dislike-Like
30	Unconscious-Conscious	35	Apathetic-Responsive	41	Unfriendly-Friendly
31	Moving rigidly-Moving elegantly	36	Mechanical-organic	42	Unkind-Kind
		37	Inert-Interactive		

Table 9. User's perception of the robot I

Rate on a scale from 1 to 5 (Leftmost/Rightmost, 3 neutral)			
Perceived intelligence		Perceived safety	
43	Ignorant-Knowledgeable	49	Agitated-Calm
44	Unintelligent-Intelligent	50	Anxious-Relaxed
45	Incompetent-Competent	51	Surprised-Quiescent
46	Foolish-Sensible		
47	Irresponsible-Responsible		
48	Ignorant-Knowledgeable		

Table 10. User's perception of robot II

Rate these statements on a scale from 1 to 5 (Strongly disagree: 1; Strongly agree: 5)	
52	I perceived the robot as an intelligent part of the training system
53	I think the way the robot was acting distracted me from the main goal of the experience
54	I felt like the robot was intentionally reacting to my actions
55	I liked the way the robot was moving
56	I clearly understood the suggestion provided by the robot when needed
57	I clearly understood what the robot was saying
58	I liked the way the robot was interacting with me
59	I liked to teach/being taught by the robot
60	Successfully complete the activity was important to me
61	I felt my notes/cheatsheet distracted me from the robot

Table 11. User's perception of the robot's performance and inclusion level

Moreover, the System Usability Scale (SUS) was included in the post-experience assessment. This standard questionnaire procedure normalizes the participants' responses to effectively distinguish among unusable and usable systems. The corresponding statements are shown in Table 12 and must be scored with 0 to 4 from strongly disagree to strongly agree. Additionally, the user's learning expectancy was assessed with the Table 15.

[SUS] Rate on a scale from 0 to 4 (Strongly disagree: 0; Strongly agree: 4)	
62	I think that I would like to use this gaming system frequently
63	I found the gaming system unnecessarily complex
64	I thought the gaming system was easy to use
65	I think that I would need the support of a technical person to be able to use this gaming system
66	I found the various functions in this gaming system were well integrated
67	I thought there was too much inconsistency in this gaming system
68	I found the gaming system very cumbersome to use
69	I felt very confident using the gaming system
70	I needed to learn a lot of things before I could get going with this gaming system

Table 12. System Usability Scale (SUS)

Learning expectancy - Rate on a scale from 1 to 5 (Strongly disagree: 1; Strongly agree: 5)	
71	I think I learned all the theory about Thevenin's theorem
72	I think I learned all the information required to simplify a circuit with the Thevenin's theorem
73	I think I can successfully pass a test on a Thevenin's theorem without further training

Table 13. Learning expectancy

Then, after the post-training questionnaire, the post-test was scheduled. This test aimed to evaluate the acquired knowledge by the training experience. It consisted in repeating the same theorem questions of the pre-test (Table 7) to verify if the participants were able to correctly answer them, aside from new questions regarding concepts shown during the practice (Table 14). The circuits involved in the test are shown in Table 15.

Thèvenin Theorem			
1	Calculate the Thevenin voltage for the circuit n° 1	5	Vth is found across the _____ terminals of the network
A	5,54 V	A	Input
B	3,33 V	B	Output
C	6,67 V	C	Neither input nor output
D	3,67 V	D	Either input or output
2	Calculate the current across the load in the circuit n° 1	6	Calculate the Thevenin resistance for the circuit n° 3
A	0,86 A	A	4 ohm

B	1,23 A	B	0,66 ohm
C	2,22 A	C	2,2 ohm
D	0,67 A	D	3 ohm
3	In the circuit N° 2, V_{th} will be equal to V_1 when ...	7	Calculate the Thèvenin voltage for the circuit n° 3
A	$R_1 \gg R_2$ && $V_2 \ll V_1$	A	6 V
B	$R_2 \gg R_1$	B	10 V
C	$V_1 = V_2$	C	4 V
D	$R_1 = R_2 = R_3$	D	4,43 V
4	The thevenin theorem is especially useful for analysing ...	8	Calculate the current across the load in the circuit n° 3
A	Any kind of circuit	A	0,57 A
B	Networks with few components	B	1 A
C	Networks with resistors and capacitors	C	1,44 A
D	Networks where the load is susceptible to change	D	2 A

Table 14. Post-Test

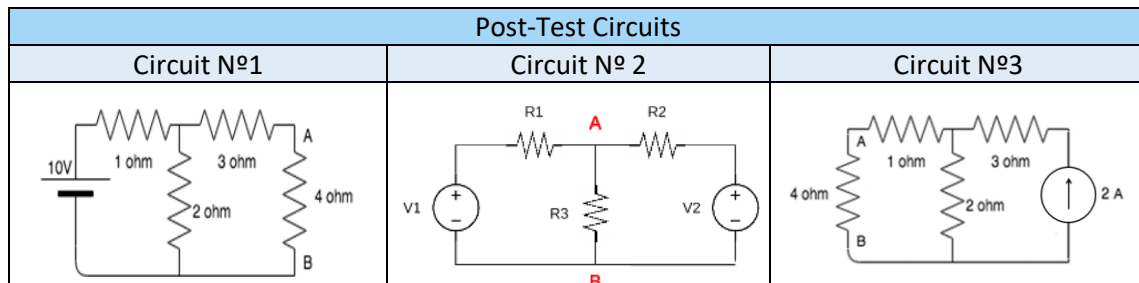


Table 15. Post-Test Circuits

Finally, the users were requested to rate several statements regarding the test (Table 16), as well as to respond to two open questions about what things did they like the most about the experience and what suggestions could improve the present system. These open questions helped to collect data about the participants' impressions that were not included in any of the questionnaires, giving them total freedom to mention any aspect of the system and/or experience.

Rate these statements on a scale from 1 to 5 (Strongly disagree: 1; Strongly agree: 5)	
74	The information provided were too complicated
75	The information were presented in a clear way
76	The quantity of the information provided was adequate to successfully pass the test
77	The quality of the information provided was adequate to successfully pass the test
78	The test was too difficult
79	Passing the test with a High mark was important to me
80	Overall I am satisfied with the training experience

Table 16. Post-Experience

Open Questions	
1	Describe 2 things you liked the most about the system/experience
2	Describe 2 things you would suggest to improve about the system/experience

Table 17. Open Questions

6.2. Analysis of Results

Once the experiments have been finished, the collected results were evaluated using the Mann-Whitney U test, which is a non-parametric statistic method for two independent samples. This algorithm is able to verify if data obtained tends to be significantly larger in one sample than the other, as well as testing the null hypothesis that two samples come from the same population. The null hypothesis in this analysis stated that there was no difference in learning gains between both groups.

This test was run setting a p-value of 10% which, in statistical hypothesis testing, discern significant differences when the Mann-Whitney algorithm shows lower values than the predefined threshold. Therefore, if the corresponding p-value obtained through the Mann-Whitney exact method is equal or lower to 0.1, a significant difference can be assumed, rejecting the initial null hypothesis.

From the pre-experience questionnaire, the statements that showed significant differences between both groups are illustrated in Table 18, calculating the mean and standard deviation of the collected responses for each group. As it can be seen, the statement 8 “*I take note during class*” presented a relevant dissimilarity between both groups, since the participants of the TL mode were less used to do it. Moreover, the LxT users had a higher tendency to “*put their notes on their own words to understand better*” and to “*copy as much as possible while the teacher is explaining during class*” since the statements 10 and 11 also showed a meaningful value. Finally, the TL users were more reluctant to “*ask doubts to the teacher during the class*” (Q14). These questions confirmed a small behaviour variation between both groups during the lessons they normally take.

Q	Mean		SD		p-value
	LxT	TL	LxT	TL	
8	4,00	2,00	0,00	1,00	0,10
10	4,67	2,67	0,58	1,53	0,10
11	2,67	1,33	0,58	0,58	0,10
14	3,67	1,00	1,53	0,00	0,10

Table 18. Pre-experience significant differences

The same procedure has been executed on the post-training questionnaire, whose results are presented in Table 19. In this case, the LxT participants were more likely to “*think that there was too much inconsistency in this gaming system*” and that they had to “*learn a lot of things before they could get going with this gaming system*”, statements 67 and 70, respectively, compared to the TL responses. This result was associated with the higher complexity level of the LxT, inherent to the input tool and due to the responsibility lied almost entirely on them, while the TL mode could be considered more simple from the user’s point of view, with less and easier interactions with the system. Regarding the expectancy factor, the TL users were less confident about “*successfully passing a test on a thèvenin's theorem without further training*” after being trained with the system (Q73). This observation could demonstrate that the user feels his knowledge reinforced when explaining to other agents, while his learning confidence could be deteriorated when receiving a lesson.

Q	Mean		SD		p-value
	LxT	TL	LxT	TL	
67	2,33	1,00	0,58	0,00	0,10
70	3,33	1,33	1,15	0,58	0,10
73	4,00	2,33	0,00	0,58	0,10

Table 19. Post-training significant differences

On the other hand, and as mentioned before, the knowledge acquired from the experience was measured using the between-subject methodology, with the aim of evaluating the prior users' knowledge about the present topic and their knowledge after being trained with the system, possibly observing a relevant evolution between both phases. Then, the obtained pre-test results are shown in Table 20, whose questions were divided into three groups by the topic. The objective of this questionnaire was to validate the users' competencies so as to confirm they were adequate to take the experience. The limitation imposed relied on having enough base of circuits, concerning general concepts and basic laws such as Ohm's and Kirchoff's ones and, at the same time, have little to inexistent knowledge about the Thèvenin theorem.

		Max	Mean		SD		p-value
			LxT	TL	LxT	TL	
Pre-Test	General (Q1-5)	5,00	3,67	4,33	0,58	1,15	0,40
	Laws (Q6-10)	5,00	5,00	4,33	0,00	0,58	0,20
	Theorem (Q11-15)	5,00	2,00	1,33	0,00	0,58	0,20
	Total	15,00	10,67	10,00	0,58	1,00	0,40

Table 20. Pre-Test results

As it can be observed in the table, all the participants passed the parts related to general and laws topics, while they were not able to pass the theorem-based questions. Moreover, the Mann-Whitney analysis did not differentiate significant dissimilarities between both groups base knowledge, since the p-value was in every section higher than the predefined threshold, 0.1. These two reasons confirmed the suitability of the participants to take the experiment.

Regarding the post-test, the same procedure analysing the data was carried out. In this case, the questions were also divided into three sections; repeated questions from the pre-test about the theorem to verify if the students were able to correctly answer them after the training experience, a new set of questions about it and, finally, a problem regarding circuits with current sources. This last section was included to verify the attention capability of the users since during the lesson a circuit with generators was solved and, therefore, the procedure with current sources was not explicitly shown but mentioned.

		Max	Mean		SD		p-value
			LxT	TL	LxT	TL	
Post-Test	Repeated (Q1-5)	5,00	4,67	4,33	0,58	0,58	0,70
	New (Q6-10)	5,00	4,00	3,33	1,00	1,15	0,70
	Current Source (Q11-13)	3,00	2,00	1,33	1,00	0,58	0,40
	Total	13,00	10,67	9,00	1,53	2,00	0,40

Table 21. Post-Test results

As it can be seen in the table, even though the LxT group did show higher overall performance, the statistical analysis did not show significant differences between both groups. However, both groups did pass the full test (including the repeated questions that they were unable to correctly answer during the pre-test) confirming the capability of the developed system to effectively support both learning models. This hypothesis is also confirmed by performing the analysis

focusing on the repeated questions, whose results are shown in Table 22. As can be seen, relevant differences were observed within the same groups before and after being exposed to the training system.

		Mean		
		Max	LxT	TL
Learning Gains (repeated questions)	Pre-Test	5	40%	26,6%
	Post-Test	5	93,4%	86,6%
	p-value		0,10	0,10

Table 22. Observed learning gains

With respect to the times spent by each group on each part of the experiment, meaningful differences have been observed. First, the LxT group required a longer period (whether guided or alone) in order to get to know the interface, variation associated with the necessity of comprehending the input tool's functionality. Moreover, the time spent for the LxT users during the mode was significantly higher due to the responsibility to carry out the lesson they were not expert in a system they did not master. However, solving the pre and post-tests did not show relevant differences but, together with no variations observed in the scores, resulted in similar analysis capabilities demonstrated by the users.

			Mean		
			LxT	TL	p-value
Times	Interface	Guided	0:11:10	0:07:12	0,10
		Alone	0:13:30	0:05:21	0,10
	Test	Pre	0:10:54	0:11:35	0,70
		Post	0:14:44	0:15:27	1,00
	Experiment	Study	0:07:30	0:00:00	N/A
		Unity	0:17:15	0:06:54	0,10
	Total		1:15:03	0:46:30	0,10

Table 23. Time distribution

Regarding the open questions, the main suggestions from the participants were related to the input tool and the system calibration, along with the overall performance. The principal points of the collected feedback are mentioned in Table 24. This feedback could be taken into consideration for future system refinement.

	Enjoyable Characteristics	Possible Improvements
Traditional Learning	<ul style="list-style-type: none"> Transforming a surface into a touch screen Be taught by a robot 	<ul style="list-style-type: none"> Smoother robot movements Robot responsiveness and speech
Learning by Teaching	<ul style="list-style-type: none"> Interactions with the robot Input tool's potential Transforming a surface into a touch screen Robot's voice and face detection result in a funnier and more personal procedure 	<ul style="list-style-type: none"> Higher precision of the touch screen (calibration) Interactions with figures and equations Fixed input tool rather than appearing when clicking Add parenthesis for the input tool

Table 24. Open Questions Feedback

Chapter 7

Conclusions

In this final chapter, a recapitulation of the project will be carried out highlighting different aspects such as if the initial objectives have been achieved, what problems have appeared during the system's development and the possible future investigation lines that could improve the proposed system.

7.1. Final Considerations

Even though there are plenty of studies that compare the TL with the LxT model in different environments, following diverse experimental procedures and involving teaching agents such as robots, there is still lack of research combining these ideas with a MR technology with the aim of teaching an engineering level topic like the electrical circuits.

Therefore, this motivated the present master's thesis with the aim of investigating the capabilities of a robotic MR platform to effectively support the learning of the students regarding the mentioned learning models and if there are any significant differences between them using the developed system. Based on the characteristics of each learning approach, two main modes have been developed aside from an interface tutorial in order to master it.

Consequently, an application called Mirelab has been developed in a Unity environment, responsible for the interface and integrating all the events related to every involved part, and Python, in charge of controlling the robot actions and the processing of the camera images for the hand's track. On the one hand, the chosen robot was Cozmo from the company Anki and it was endowed with several functionalities such as face detection, clicking following and animations with the objective of creating a more personal and interactive experience. On the other hand, the camera used was the depth sensor Kinect v2 from Microsoft and the algorithm implemented for the fingertip detection was based on the background subtraction method, which was able to track moving elements in a static environment.

One of the biggest concerns during the implementation of the project was its usability since the users should be confident making use of the system. For that reason, a good calibration of the finger detection algorithm was required to let the user effectively click in the desired place triggering the corresponding events, without the need of unnecessarily repeating the same action over and over again. Moreover, the system's complexity should find a balance between having enough functionalities to perform all the needed actions and not being excessively difficult to manage by the users, especially during the LxT mode. Consequently, spent enough time during the interface tutorial was essential to know the capabilities of the system and practise to master the interactions between starting the real modes.

Finally, an experimental evaluation was carried out with 6 participants (3 for each mode) using a between-subject methodology, so their experience and knowledge of the topic were assessed before and after being exposed to the training system by several questionnaires. The objective of this external evaluation was to possibly confirm the main hypotheses; the capability of the system to support the user's learning and the potential differences in learning gains between both approaches, aside from secondary assessment such as the robot influence and/or inclusion or the system performance.

The collected data did not show significant differences between the learning of both groups, but all participants were able to pass the post-test about confirming the capability of the system to effectively support the student's learning in both models. Moreover, the analysis showed a significant complexity difference between both groups, since the LxT users perceived the system more complicated to use, observation associated with the complexity of the input tool and the responsibility to carry out the lesson.

Therefore, one of the two main hypotheses was confirmed, leading to the conclusion that the present project managed to successfully design and implement a MR robotic training platform able to support multiple learning models. Moreover, regarding its usability, it was achieved to a certain extent since several users found difficulties to fluently interact with the interface at some points.

7.2. Lesson Learned

After noting the objectives accomplished, it is also important to know the lessons learned during the implementation of the system and perform a critical assessment about what could have been done more efficiently, as well as to recapitulate the obstacles that delayed any part of the programmed project.

7.2.1. System Concept

The conception of the system was probably the biggest bottleneck of the project, even if it is well-known that it is an iterative process, it consumed more time than initially expected. This resulted in the delay of the rest of the parts, since it was the first task and could not be performed in parallel with any other.

The difficulty encountered during the brainstorming lies in the necessity to keep taking steps forward in the right direction and exploring areas that have not been fully developed. The concept of the system had to combine several characteristics that have not been well investigated yet, and also engage the user being an educational but also fun activity. For that reason, the state of art takes a big role when it comes to the system design since an excellent understanding of the current technology is essential to know what works, what does not, what could be enriched and what areas have room for improvement.

The impact of this issue, and therefore the time spent on it, could have been possibly reduced expanding the conception of the game to other social circles, such as writing in a specialized forum for ideas, writing to other recognized authors about their articles related to this topic or simply conducting surveys about gaming and technologies preferred. As a result, perhaps the concept process would have been expedited enough to find a suitable idea spending less time and resources.

7.2.2. Redesign

The implementation of a system is considered an iterative process and, therefore, the redesign is inevitable in most cases, since it is inherent to this type of projects where preliminary versions must be critically tested on the target platform in order to obtain the best possible outcome. With that said, several concerns stood out during the realization of the project.

On the one hand, one of the main challenges of the project consisted of actually making the robot interact with the user and correctly synchronise it with the application events. However, the first versions could not accomplish this goal. In most cases, it was just used mostly as a moving speaker that could be probably easily replaced by a voice coming from the pc or directors. Even if that was not the role designed for it, its resulting movements and actions were

not that engaging or performing simple but simultaneous tasks such as moving and talking was hard to accomplish.

On the other hand, even though when the concept of the system was defined, including the idea and technology involved, the lessons' progressions did not result as expected. The lesson was very simple, did not caught the attention of the user, and the interaction with the robot was minimal. Therefore, several redesigns were needed during the implementation, delaying again the end of the project.

7.2.3. Hand Recognition

Even though the developed algorithm that converts any planar surface to a touchable one is functional and performs the required actions, the user must be careful with certain movements. The main issue lies in that the depth camera cannot discern the fingers with the surface when they are close enough, due to the camera working principle which classifies each pixel in a certain depth layer. This resulted in the incapability of the camera to effectively recognise the hand contour as soon as it was close enough to the surface, since both were categorised in the same layer of depth and, therefore, losing track of the hand. The presented issue was solved using estimated corrections to the obtained raw data.

Another issue regarding the camera consists of the discerning algorithm between the robot and the hand, due to the implemented method does not recognise hands or fingers but moving objects from the initial captured background. Therefore, it always tracks the hand and the robot positions because they are the only moving objects. In this case, and due to the hand never surpass the robot position, the method works since it associates the hand with the lower contours within the scene. However, this algorithm cannot be considered robust enough for future uses due to this concern without minimal modifications.

It was also proceeded to recognise the hand using the colour camera, that was more robust, and later find the depth of the associated pixel in the depth camera but, after numerous tries it was quite hard to match both images due to the differences in resolution and reference systems.

Moreover, it is worth mentioning the dispersion of the camera raw data, since the same pixel showed different distance values at each frame, being significant when the pixel was close to an edge such as, for example, a hand contour. This negatively affected the performance of the detection of touch events, which was solved by modifying the fingertip algorithm.

These issues could have been possibly ameliorated by the use of an already tested algorithm for hand recognition or applying a more robust recognition method than the background subtractor.

7.3. Future Work

The achievements and the objectives fulfilled, along with the main limitations and problems encountered will serve as a basis to guide possible future lines of research.

7.3.1. Hand Gestures

In the present project, it has not been introduced any hand gestures due to the training progression did not require to discern between several actions, except the differentiation among a click and a slide which was approached using time.

Consequently, the development of a robust algorithm that, aside from recognising moving objects, could identify diverse predefined hand patterns, could definitely bring new opportunities to create systems where the user has a higher capability of interaction with the

touch area. Examples of these patterns could be from resizing objects separating or joining the fingers to associate the number of fingers shown to concrete actions.

7.3.2. Comparisons

Even though the main comparison carried out in this project was the contrast between two diverse learning models such as the lecture-based against the LxT, other analysis could have been performed to stress the differences between two or three technologies in a given condition. So, these types of comparisons are always done under a common denominator, which results to be the same platform.

Therefore, another possible comparison could stress the differences and capabilities between the present table projection and a holographic environment created by the hololens when it comes to interact with a touch area and with robots. A mixed mode could also integrate both approaches and could assess the digital content impact on the system modifying the number of holographic objects. The progression could cover from being all the objects interactable with the camera, to being all digital content created by the hololens.

Bibliography

- [1] E. Walker, V. Giroto, Y. Kim, and K. Muldner, ‘The Effects of Physical Form and Embodied Action in a Teachable Robot for Geometry Learning’, *2016 IEEE 16th Int. Conf. Adv. Learn. Technol.*, 2016, doi: 10.1109/ICALT.2016.129.
- [2] D. Martinoia, D. Calandriello, and A. Bonarini, ‘Physically Interactive Robogames: Definition and design guidelines’, *Rob. Auton. Syst.*, vol. 61, no. 8, pp. 739–748, 2013, doi: 10.1016/j.robot.2013.04.017.
- [3] D. Robert, R. Wistort, J. Gray, and C. Breazeal, ‘Exploring mixed reality robot gaming’, *Proc. 5th Int. Conf. Tangible Embed. Interact.*, 2011, doi: 10.1145/1935701.1935726.
- [4] P. Alves-Oliveira, P. Sequeira, F. S. Melo, G. Castellano, and A. Paiva, ‘Empathic Robot for Group Learning’, *ACM Trans. Human-Robot Interact.*, vol. 8, no. 1, pp. 1–34, Mar. 2019, doi: 10.1145/3300188.
- [5] D. Leyzberg, S. Spaulding, and B. Scassellati, ‘Personalizing Robot Tutors to Individuals’ Learning Differences’, Yale University.
- [6] E. Hansson, ‘Investigating Augmented Reality for Improving Child-Robot Interaction’, KTH Royal Institute of Technology, Sweden, 2019.
- [7] M. J. Kintu, C. Zhu, and E. Kagambe, ‘Blended learning effectiveness: the relationship between student characteristics, design features and outcomes’, *Int. J. Educ. Technol. High. Educ.*, 2017, doi: 10.1186/s41239-017-0043-4.
- [8] J. Shabatura, ‘Using Bloom’s Taxonomy to Write Effective Learning Objectives’, *Assignments and Measuring Student Learning*, 2013. [Online]. Available: <https://tips.uark.edu/using-blooms-taxonomy/>.
- [9] N. Thuy Thi Thai, B. De Wever, and M. Valcke, ‘The impact of a flipped classroom design on learning performance in higher education: Looking for the best “blend” of lectures and guiding questions with feedback’, *Comput. Educ.*, vol. 107, pp. 113–126, 2017, doi: 10.1016/j.compedu.2017.01.003.
- [10] K. K. Bhagat, C. N. Chang, and C. Y. Chang, ‘The Impact of the Flipped Classroom on Mathematics Concept Learning in High School’, *Educ. Technol. Soc.*, vol. 19, pp. 124–132, 2016.
- [11] J. M. Fautch, ‘The Flipped classroom for teaching organic chemistry in small classes: Is it effective?’, *Chem. Educ. Res. Pract.*, vol. 16 (1), pp. 179–186, 2015, doi: 10.1039/c4rp00230j.
- [12] P. Baepler, J.D.Walker, and M. Driessenb, ‘It’s not about seat time: Blending, flipping, and efficiency in active learning classrooms’, *Comput. Educ.*, vol. 78, pp. 227–236, 2014, doi: 10.1016/j.compedu.2014.06.006.
- [13] S. G. Wilson, ‘The Flipped class: A Method to address the challenges of an undergraduate statistics course’, *Teach. Psychol.*, vol. 40(3), pp. 193–199, 2013, doi: 10.1177/0098628313487461.
- [14] J. A. Gómez-Tejedor, ‘Effectiveness of flip teaching on engineering students’ performance in the physics lab’, *Comput. Educ.*, 2020, doi: 10.1016/j.compedu.2019.103708.
- [15] D. S. McNamara, ‘Self-Explanation and Reading Strategy Training (SERT) Improves Low-Knowledge Students’ Science Course Performance’, *Discourse Process.*, vol. 54:7, pp. 479–492, 2017, doi: 10.1080/0163853X.2015.1101328.
- [16] Y. Bargh, J. A., & Schul, ‘On the cognitive benefits of teaching’, *J. Educ. Psychol.*, vol. 72, pp. 593–604, 1980, doi: 10.1037/0022-0663.72.5.593.
- [17] S. Y. Okita and D. L. Schwartz, ‘Learning by Teaching Human Pupils and Teachable Agents: The Importance of Recursive Feedback’, *J. Learn. Sci.*, vol. 22, pp. 375–412, 2013, doi: 10.1080/10508406.2013.807263.
- [18] J. Werfel, ‘Embodied Teachable Agents: Learning by Teaching Robots’, Wyss Institute for Biologically Inspired Engineering, Harvard University.
- [19] A. L. Baylor, ‘Three Research Directions for Affective Learning Technologies’, *ICLS 2018 Proc.*, 2018.
- [20] D. Duran, ‘Learning-by-teaching. Evidence and implications as a pedagogical mechanism’, *Innov. Educ. Teach. Int.*, 2016, doi: 10.1080/14703297.2016.1156011.
- [21] K. Muldner, V. Giroto, C. Lozano, W. Burleson, and E. Walker, ‘The Impact of a Social Robot’s

- Attributions for Success and Failure in a Teachable Agent Framework’, *ICLS 2014 Proc.*, 2014.
- [22] F. Tanaka and S. Matsuzoe, ‘Children Teach a Care-Receiving Robot to Promote Their Learning: Field Experiments in a Classroom for Vocabulary Learning’, *J. Human-Robot Interact.*, vol. 1, pp. 78–95, 2012, doi: 10.5898/JHRI.1.1.Tanaka.
- [23] D. Hood, S. Lemaignan, and P. Dillenbourg, ‘When Children Teach a Robot to Write: An Autonomous Teachable Humanoid Which Uses Simulated Handwriting’, *École Polytechnique Fédérale de Lausanne (EPFL)*.
- [24] L. D. Riek, ‘Wizard of Oz Studies in HRI: A Systematic Review and New Reporting Guidelines’, *J. Human-Robot Interact.*, vol. 1, pp. 119–136, 2012, doi: 10.5898/JHRI.1.1.Riek.
- [25] N. Tran, J. Rands, and T. Williams, ‘A Hands-Free Virtual-Reality Teleoperation Interface for Wizard-of-Oz Control’, *Int. Work. Virtual, Augment. Mix. Real. Human-Robot Interact.*, 2018.
- [26] A. D. Wilson, ‘Using a Depth Camera as a Touch Sensor’, *Interact. Tabletops Surfaces, ITS*, 2010.
- [27] P. Sharma, R. P. Joshi, R. A. Boby, S. K. Saha, and T. Matsumaru, ‘Projectable Interactive Surface Using Microsoft Kinect V2: Recovering Information from Coarse Data to Detect Touch’, *IEEE Int. Symp. Syst. Integr.*, 2015.
- [28] C. Zhu, ‘Making a Touch Screen out of a Projector’, 2019. [Online]. Available: <https://www.seas.upenn.edu/~zchuning/blog/2019-07-06-interactive-projector-system/>.
- [29] P. Pirone, F. Lamberti, and A. Cannavò, ‘Designing Interactive Robotic Games based on Mixed Reality Technology’, *Int. Conf. Consum. Electron.*, 2019.
- [30] P. Baldo and F. Lamberti, ‘Design and Implementation of a Mixed-Reality Robotic Game’, *Politecnico di Torino*, 2019.
- [31] F. LAMBERTI, F. G. PRATTICO, D. CALANDRA, G. PIUMATTI, and F. BAZZANO, ‘Robotic gaming and user interaction: Impact of autonomous behaviors and emotional features’, *IEEE Games, Entertain. Media (GEM 2018)*, pp. 441–448, 2018.
- [32] J. Chen, A. Cannavo, and F. Lamberti, ‘User Perception of Robot’s Role in Floor Projection-based Mixed-Reality Robotic Games’, *Politecnico di Torino*, 2017.
- [33] Unity Technologies, ‘Unity’, 2019. [Online]. Available: <https://unity.com>.
- [34] Intel, ‘OpenCV’. [Online]. Available: <https://docs.opencv.org/>.
- [35] P. Hintjens, ‘ZeroMQ’. [Online]. Available: <http://zguide.zeromq.org/page:all>.
- [36] Anki, ‘Cozmo SDK Examples’, 2018. [Online]. Available: <https://github.com/anki/cozmo-python-sdk/tree/master/examples>.
- [37] Y. Nitta, ‘Kinect V2 C++ Programming with OpenCV on Windows10’, 2017. [Online]. Available: <http://nw.tsuda.ac.jp/lec/kinect2/>.
- [38] C. Bartneck, D. Kulic, E. Croft, and S. Zoghbi, ‘Measurement Instruments for the Anthropomorphism, Animacy, Likeability, Perceived Intelligence, and Perceived Safety of Robots’, *Int J Soc Robo*, vol. 1, pp. 71–81, 2009, doi: 10.1007/s12369-008-0001-3.