FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Development of a virtual system with a haptic device to analyze the process of learning 3D shapes

Rita Barros Neves



Mechanical Engineer Master Degree - Automation

Supervisor: Prof. António Ramos Silva

Co-supervisor: PhD. Sofia Leite

November 5, 2022

© Rita Barros Neves, 2022

Development of a virtual system with a haptic device to analyze the process of learning 3D shapes

Rita Barros Neves

Mechanical Engineer Master Degree - Automation

November 5, 2022

Abstract

Artificial intelligence systems are becoming more important in industry, as they allow the creation of intelligent automated systems that optimize production. For these systems to work in an unstructured environment is important that they can perform a good reconstruction of objects, to have high-level representations of the objects and their surroundings. The main obstacle to AI is the need for big data to develop it. This process contrasts with the human process of learning, which is done with a smaller amount of data. It is on this human ability to learn new concepts without reacquiring a big amount of data that this dissertation focuses on.

The main objective of this study is to analyze the human process of learning 3D shapes. To accomplish this, there are established the many levels a person goes through in learning 3D shapes, and the tasks she can perform at each of them. This is done based on the development theory Model of Hierarchical Complexity. After, it is created a set of questions that translate each of these stages. Finally, it is tested this model developed with users through a Virtual Environment application where they can precept virtual objects through a haptic device and answer some questions that translate the different stages of development for the recognition of 3D shapes.

The obtained results demonstrate the human ability to adapt and learn quickly to perceive objects using a haptic device. Most users appeared to be in the 7th stage of hierarchical complexity for the haptic recognition of 3D shapes. Furthermore, the test showed an increase in hierarchical complexity, and most of the users follow the concept defined by the MHC, that for a person to be able to perform a higher hierarchical task, must be able to perform all the lower ones. The main objective of this dissertation was to analyze the human process of learning and this was achieved, with the application working correctly and the questions appearing to translate the multiple stages that occur in the process of learning 3D shapes.

Resumo

Os sistemas de Inteligência Artificial (IA) são cada vez mais importantes na indústria, uma vez que permitem a criação de sistemas automatizados inteligentes que otimizam a produção. Para que estes sistemas funcionem num ambiente não estruturado é importante que sejam capazes de efetuar uma boa reconstrução de objetos, de modo a criar representações de alto nível da sua envolvente. O principal obstáculo da IA é a necessidade de uma grande quantidade de dados para a desenvolver. Este processo contrasta com o processo humano de aprendizagem, que, comparativamente com o da IA, é feito com uma menor quantidade de dados. É sobre esta habilidade humana de aprender novos conceitos que esta dissertação se centra.

O objetivo deste estudo é analisar a aprendizagem humana de formas 3D. Para tal, são estabelecidos os vários níveis pelos quais uma pessoa passa na aprendizagem de formas 3D, e as tarefas que capazes de realizar em cada um destes níveis. Estes estágios são definidos baseados na teoria de desenvolvimento Modelo de Complexidade Hierárquica. De seguida, são criadas perguntas que traduzem cada um destes níveis. Finalmente, é testado o modelo desenvolvido com uma amostra de sujeitos, através de um uma aplicação com um Ambiente Virtual. Nesta aplicação os utilizadores são capazes de sentir objetos virtuais através de um dispositivo háptico e respondem a perguntas que traduzem diferentes fases de desenvolvimento sobre o reconhecimento de formas 3D.

Os resultados obtidos demonstram a capacidade humana de adaptação e rápida aprendizagem na perceção de objetos por meio de um aparelho háptico. A maioria dos participantes aparenta estar no 7º nível de complexidade hierárquica no reconhecimento háptico de formas 3D. Além disso, o teste provou ter um aumento hierárquico de complexidade ao longo das perguntas, e com a maioria dos participantes a provarem o conceito definido pelo MHC: uma pessoa que seja capaz de realizar uma determinada tarefa de um nível hierárquico, consegue também realizar as dos níveis abaixo. O objetivo desta dissertação foi atingido, com a aplicação desenvolvida a funcionar corretamente e com as questões a serem capaz de traduzir os diferentes estágios que ocorrem no processo de aprendizagem de formas 3D.

Acknowledgements

In these past years, there were many people involved whom I need to thank. Starting with my supervisors, Prof. António Ramos Silva and Ph.D. Sofia Leite. Thank you for your constant support and all the knowledge shared. Your interest in this work was inspiring.

I would like to acknowledge the support from LAETA. This research was funded by Projects LAETA UIDB/50022/2020 and UIDP/50022/2020.

A special thank you to my extra "supervisor", Daniel, for all the support given throughout this dissertation. Your time, interest, and patience were invaluable.

To my family, who always supported and believed in me. Thank you for your patience and for always encouraging me to follow my dreams.

To my friends, you have made these last six years better than I could have ever imagined. To all the classes we missed, to all the stressful days, and the sleepless nights. But even more to all the afternoons on the grass, the endless conversations, and to all the laughter. I am very lucky for all I have experienced and for the people I have met. I take it all with me.

To my partner on the course, Pedro Santos, thank you for your support and patience. The path has been hard but memorable.

Lastly, I want to thank the excellent teachers whom I had the pleasure to be inspired by.

Rita Neves

"You never know what's around the corner. It could be everything. Or it could be nothing. You keep putting one foot in front of the other, and then one day you look back and you've climbed a mountain."

Tom Hiddleston

Contents

Acknowledgements v

1 Introduction 1
- 1.1 Framework . . . 1
- 1.2 Objective . . . 2
- 1.3 Structure . . . 2

2 Literature Review 3
- 2.1 Sense of touch in humans . . . 3
  - 2.1.1 Kinesthetic and cutaneous sense . . . 4
- 2.2 Object recognition . . . 4
- 2.3 Object shape perception on robotics . . . 5
- 2.4 Tactile sensing . . . 7
- 2.5 Object recognition on the absence of vision . . . 8
  - 2.5.1 Haptic technology . . . 9
- 2.6 The Model of Hierarchical Complexity . . . 14
- 2.7 The Van Hiele Model of Geometrical Thinking and Piaget's Theory of Spatial Development . . . 18
- 2.8 The stages of the MHC on learning 3D shapes with the sense of touch . . . 18

3 Metodology 25
- 3.1 The questions for each stage to study . . . 25
- 3.2 The components of the haptic interface . . . 29
  - 3.2.1 The haptic device . . . 31
  - 3.2.2 The virtual environment . . . 31
  - 3.2.3 The user interface . . . 54
  - 3.2.4 Communication between the UI and VE programs . . . 60

4 Data Analyses 67
- 4.1 Some considerations . . . 67
- 4.2 The participants . . . 68
- 4.3 The method followed on the data analysis of the questions . . . 68
- 4.4 The times and answers of the test . . . 70
- 4.5 Other considerations . . . 75

5 Discussion 79
- 5.1 The trial test . . . 79
- 5.2 The performances on the simple shapes and rounded cube . . . 80
  - 5.2.1 The simple shapes . . . 81

5.2.2 The rounded cube 84
5.3 The questions 89
5.4 Distribution of the users by the stages of hierarchical complexity 97
5.5 Other considerations 98

6 Conclusion **99**
6.1 Suggestions for future studies 100

A Appendix **101**
A.1 The code from the VE application 101
A.1.1 The rendering of the sphere 103
A.1.2 The serial communication 103
A.1.3 Actions performed once an object is required 105

References **109**

List of Figures

2.1 Two balls with different local shapes. On a global scale, they are both balls. At a local scale, (a) has small pits and (b) has curvilinear lines , [34] 5
2.2 Interaction between the human with the VE through the haptic device [12] 10
2.3 Four haptic devices that use cutaneous sensing (a) from [33], b) from [13], c) from[47] and d) from [8] 10
2.4 Four haptic devices that use the kinesthetic sensing (a) from [14], b) from [11], c) from [2] and d) from [36] 10
2.5 Haptic force feedback device used for the three experiments [2] 11
2.6 The shapes given on the three tests 12
2.7 The draws of a square done by a user with and without limitation of the workplace [39] 12
2.8 The vibrotactile glove from [35] 13
2.9 The exploration of a square done by a user with the vibrotactile glove [35] 13
2.10 Order of hierarchical complexity of tasks [31] 16
2.11 The representation of a part of a stair done from three cubes (objects created on Solidworks® by the author) 21

3.1 A representation of a sphere, cube, and cylinder (object created on Solidworks® by the author) 26
3.2 A representation of the rounded cube (object created on Solidworks® by the author) 27
3.3 The representation of the objects from the 8^{th} stage's questions (objects created on Solidworks® by the author) 28
3.4 The representation of the objects from the 9^{th} stage questions (objects created on Solidworks® by the author) 28
3.5 A diagram of the components of the haptic interface (composed by the author with images from [2], [41], [3], [4]) 30
3.6 The haptic device and the representations of the movements possible to perform: a) movements along the axes, duo to its digital encoders; b) movements able to perform duo to its gimbals articulations; point of contact of the device (taken and made by the author) 31
3.7 Overview of the processes that occur on a QuickHaptics, HD, and HL application ([26]) 34
3.8 Usual structure of an HDAPI application Based on [26] 36
3.9 An example of an HDAPI application on Microsoft Visual Studio® (taken from one application example of OpenHaptics®) 37
3.10 Example of a simple application on LabVIEW® (created by the author) 38
3.11 The first program created on C code on Visual Studio 40
3.12 The declaration of the function to initialize the device and its parameters 40

3.13 The declaration of the *HD_DEVICE_MODEL_TYPE* 41
3.14 The second program created on C on Visual Studio, where the position of the device is printed for the user with a 1-second interval between readings 42
3.15 Schema of the commands that happen inside the *main* function of the VE application 44
3.16 Representation of the penetration and force $\overrightarrow{F}$ performed by the top surface of a cube when the device is on the point R (draw created by the author) 45
3.17 Representation of a sphere with the force $\overrightarrow{F_1}$ it creates when the device is on the point P_1. The figure also illustrates the center C_1 of the sphere, its radius r_1, the penetration when the device is in the position C_1, and the distance d_1 from the device to the center of the sphere (draw created by the author) 46
3.18 A representation of the volumes that will define the surfaces of the cylinder (objects created on Solidworks® by the author) 47
3.19 Representation of the equation 3.7 on the Cartesian coordinate system (created by the author) 48
3.20 Representation of the equation 3.7 on the Cartesian coordinate system (created by the author) 49
3.21 Representation of the volumes that will define the surfaces of the cube (object created on Solidworks® by the author) 49
3.22 Representation of the unit normal vectors of the surfaces of a cube (created by the author) 50
3.23 Representation of a square circumscribed by a circle (created by the author) . . . 51
3.24 Representation of the two spheres that can be created with the value of x_0 with the equation 3.27 (created by the author) 52
3.25 Representation of the corners and edges for the cube (objects created on Solidworks® by the author) 53
3.26 Representation of the cube with the cylindrical hole, with all the edges curved (objects created on Solidworks® by the author) 53
3.27 Home page design of the UI application 54
3.28 Design of the page where some personal data is asked from the participants . . . 55
3.29 Design of the trial page where the user receives some information about the exploration of shapes and can try to explore a cone 56
3.30 Design of the page with the question representing the 5^{th} stage of the MHC for the haptic recognition of shapes 56
3.31 Design of the page with the question representing the 6^{th} stage of the MHC for the haptic recognition of shapes 57
3.32 Design of the page with the question representing the 7^{th} stage of the MHC for the haptic recognition of shapes 58
3.33 Design of the page with the question representing the 8^{th} stage of the MHC for the haptic recognition of shapes 58
3.34 Design of the page with the question representing the 9^{th} stage of the MHC for the haptic recognition of shapes 59
3.35 Schema of the communication between the main form and the sub-forms 60
3.36 Haptic interface components and its connections 61
3.37 Actions that happen on the VE and UI since an object is requested until the end of its rendering 63
3.38 Actions that happen on the VE and UI when an object is requested while another is being rendered 64

4.1 A representation of the face of the cube that was bad rendered (created by the author) 75

4.2 A representation of the edges of the cube with two cubic holes that were being bad rendered (created by the author) . . . 76

5.1 Average exploration time for the cone, the simple shapes, and the rounded cube . 80
5.2 Correct answers on the first three questions for each shape . . . 81
5.3 Correct answers for each shape on the first three questions . . . 82
5.4 Answers for the cylinder on the 6^{th} stage's question . . . 83
5.5 Correct answers of the properties of the cylinder on the 6^{th} stage's question . . . 83
5.6 Number of correct answers for the rounded cube, distributed according to the user's percentage of correct answers for the 8^{th} stage's question . . . 85
5.7 Correct answers of the properties of the rounded cube on the 6^{th} stage's question 85
5.8 Answers for the rounded cube on the 6^{th} stage's question . . . 86
5.9 Answers for the rounded cube on the 7^{th} stage's question . . . 87
5.10 Average exploration time of the rounded cube for the users with correct and wrong answers separately . . . 88
5.11 Exploration time of cube and rounded cube on the 7^{th} stage's question for users who answered both objects as cubes . . . 88
5.12 Exploration time of cube and rounded cube on the 7^{th} stage's question for users who answered correctly for both objects . . . 89
5.13 Percentage of users with good performance along the stages . . . 90
5.14 Percentage of users who performed badly on the first question and well on the third one . . . 91
5.15 Distribution of answers of the 8^{th} stage's questions . . . 92
5.16 The percentage of correct answers given on each of the objects of the 8^{th} stage's questions . . . 93
5.17 The percentage of answers given on the 7^{th} stage's questions for the users who on the 8^{th} stage confused the cuboid with a cube . . . 93
5.18 Percentage of the answers given on the 9^{th} stage . . . 94
5.19 Representation of the number of users who responded to each of the shapes on the 9^{th} stage's question . . . 95
5.20 Percentage of users with good and continuously good performance along the stages 95
5.21 Graphic with the differences between the percentages of users with good and continuously good performances. Representations of the 1^{st} and 2^{nd} method and 2^{nd} method without considering the first question . . . 96
5.22 Users on each stage or in transition to the next one . . . 98

A.1 Functions used on the VE to calibrate the device . . . 102
A.2 Functions used on the VE to render the sphere . . . 103
A.3 The code of the VE, inside the *main* function, for the serial communication . . . 104
A.4 The functions of the VE, outside the *main* function, for the serial communication 104
A.5 Actions performed on the VE when a message is received from the UI, requiring a shape to be rendered . . . 105
A.6 Actions performed on the VE when the user grabs the pen of the device and when he first touches the object . . . 106
A.7 Functions used on the VE to calibrate the device . . . 107

List of Tables

2.1 Steps of the process of sensing, perception, and acting. Based on [20] 7
2.2 Stages described in the Model of Hierarchical Complexity [15] 17
2.3 Table of the levels a person goes through on geometry thinking, based on Piaget's theory and the van Hiele model. Information based on the article [23] 19
2.4 Tasks developed for each stage of the MHC applied to the domain of haptic recognition of 3D shapes 22

3.1 Table with the messages sent from the UI to the VE, the commands they represent, and their ASCII code 65
3.2 Table with the messages sent from the VE to the UI, with the actions of the user they represent 65

4.1 Performance of the users who answered the 8^{th} stage's question an example . . . 68
4.2 Average time on the 8^{th} stage's question and on its objects for the users that performed this stage with and without an example 69
4.3 Distribution of the genre of the participants 69
4.4 Answers given from the users to some personal questions at the beginning of the test 69
4.5 Distribution of the age of the participants 69
4.6 Distribution of the degree of education of the participants 69
4.7 Times of exploring and of finding the objects and the percentage of correct answers, by object and stage; percentage of users with good performances on each of the stages and the average of time taken on each question 71
4.8 Percentage of correct answers for all users and users with correct answers along these shapes, using the 1^{st} method on the second question 71
4.9 Percentage of correct answers for all users and users with correct answers along these shapes, using the 2^{nd} method on the second question 72
4.10 Properties of the cylinder correctly answered 72
4.11 Properties of the rounded cube correctly answered 72
4.12 Answers from the 6^{th} stage's question for the cylinder 73
4.13 Answers from the 6^{th} stage's question for the rounded cube 73
4.14 Answers from the 7^{th} stage's question for the rounded cube 73
4.15 Number of correct answers for the rounded cube divided by the percentage of correct answers of the users on the 8^{th} stage 74
4.16 Distribution of the answers given for the cuboid on the 8^{th} stage's question 74
4.17 Distribution of answers given on the 7^{th} stage's question for the cube, from users that stated the cuboid is a cube 74
4.18 Percentage of users who correctly answered 0, 1, or 2 objects on the 9^{th} stage's question 75

4.19 Answers of the participants on the 9^{th} stage's question 76

5.1 Distribution of users by their stage or if being in transition to the next one 97
5.2 Stages where some users performed badly, although appearing to be at a higher hierarchical complexity stage 97

Symbols and Abbreviations

DLL	**Dynamic-Link library**
API	**Application Programming Interface**
HDAPI	**Haptic Device API**
HLAPI	**Haptic Library API**
PDD	**Phantom Device Driver**
SLW	**Shared Library Wizard**
CLFN	**Call Library Function Node**

Chapter 1

Introduction

The present document was composed by Rita Barros Neves in the second semester of the 2021/2022 academic year as her dissertation work for obtaining a Master's degree in Mechanical Engineering – Automation. The student was supervised by Prof. António Ramos Silva and the Ph.D. Sofia Leite.

1.1 Framework

Artificial Intelligence (AI) systems are used in industry and allow the creation of intelligent automated systems. According to some early predictions, the evolution of this area has been slower than expected, with the main reason being the huge amount of data required to develop AI technology.

Making a comparison of AI with the learning capability of humans, it is possible to understand that something is incomplete since humans need a much smaller fraction of data/experience to achieve the knowledge needed to perform a task. The solution to the AI problem can be achieved by understanding the natural process of learning used by humans.

For these AI systems to work in an unstructured environment, it is important that they can perform a good reconstruction of objects and have high-level representations of the objects and their surroundings. The two main senses used for this type of task are vision and touch. While computational vision has been intensely studied over the past years, the tactile sense in robotics is still developing, and the studies show a big potential for this area.

1.2 Objective

The main objective of this dissertation is to analyze humans' haptic process of learning shapes. The analysis will be done based on a developmental theory of human learning, the Model of Hierarchical Complexity. To accomplish this, this dissertation has three main tasks:

- Based on the cognitive model, establish a sequence of levels a person goes through when developing their recognition of shapes;
- Replicate these levels into questions;
- Create a Virtual Environment application that interacts with a haptic device, that will allow testing the levels established, to analyze if the results prove or invalidates it.

1.3 Structure

This dissertation is divided into six chapters. Chapter 1 presents an introduction to the topics addressed throughout this dissertation, contextualizing them in the field of Mechanical Engineering. It also announces the main objective of the dissertation.

In Chapter 2, a literature background about the tactile sense in both humans and robotics is presented. Also in this chapter, the developmental theory Model of Hierarchical Complexity is introduced, along with other theories that explain geometrical thinking and spatial development. At the end of this chapter, it is presented the levels defined for the learning of 3D shapes.

Chapter 3 describes the methodology, with a description of the questions designed and of the Haptic Interface components. This chapter also explains how the applications of the Haptic Interface were developed.

Chapter 4 presents the results of the statistical analysis and Chapter 5 presents a discussion of the results. To finalize, Chapter 6 summarizes the conclusions that were drawn from the developed study and suggest ideas for future research in the studied area.

Chapter 2

Literature Review

2.1 Sense of touch in humans

Humans interact and perceive with the environment through their senses. Each of these acquires unique information that shapes our behavior for different situations, according to the needs. The sense of touch is a very important tool for interactions, allowing humans to obtain properties of objects, like their stiffness, shape, temperature, etc. This information can then be used in reflexes, for automatic responses, like force regulation, and does not require a big cognitive effort. Without this sense, simple actions like grabbing an object could turn into a very hard task to perform.

On [50], individuals were asked to grab objects after their hands had been anesthetized, which results in a partial loss of their sense of touch. This experiment showed that, with this loss, the participants couldn't have a stable grasp. These results are a good example of how important the sense of touch is in the daily life of an individual.

But this sense has an even bigger role in human life than just interacting with objects; it also allows humans to control their movements. On [7], a person who lost most of his touch sensation (except temperature and pain sensation) below the neck, stopped being able to control his movements, unless when looking at the body part he wanted to move. When not looking, his arms moved uncontrollably. This shows another big importance of the tactile sense, not only on interacting with objects and persons but also on having control of our body.

Besides all the information that can be acquired through tactile exploration, in a learning environment, an active touch is a stimulus that leads to motivation, since a person is able to perform choices, control, and move objects, instead of just staring at them [28]. So it can be used as a tool to improve learning and motivate students to perform tasks and not just hear about them.

2.1.1 Kinesthetic and cutaneous sense

The sense of touch can be divided into two submodalities: kinesthetic or cutaneous/tactile. The first one is based on information received from the receptors in the muscles, tendons, and joints, while the cutaneous from the receptors embedded in the skin [21], [32].

In terms of sensory systems, most researchers agree there are three: kinesthetic, cutaneous, and haptic. The kinesthetic system is responsible for information about static and dynamic body posture, like the relative positions of the arms, head, etc. In line with the cutaneous sense, the cutaneous system uses receptors in the skin to understand when a physical interaction with another body or object occurs. The haptic system uses information about objects and events from the other two [21], [10]. A study made to test some methods for learning shapes in robots showed that both kinesthetic and cutaneous senses have an important role in object recognition since the results obtained were better when using a combination of tactile and encoder (joint angle) signals [43].

2.2 Object recognition

Object recognition is a skill used daily by any person, and touch plays an important role in this field. Tactile object recognition provides information about the objects people interact with, allowing better manipulation.

When talking about object recognition, the main two senses used for this task are vision and touch. The first one has been more studied in the last decades, but research on tactile sense has been growing and showed how haptic has a big importance on object recognition. Studies made with blind subjects show that these individuals develop geometrical representations of shapes independent of vision [28], [29]. Activation of cortical areas usually related to vision in blind individuals has been identified when performing tactile tasks [24]. For a non-blind person, haptic exploration of objects plays an important role, especially in the early ages, when children tend to bang hard objects, stroke smooth surfaces, and finger textured bodies [52], [40].

But in general terms, when identifying shapes is required, the use of both vision and haptic shows better performances than using only one of the senses [28]. Some authors suggest that our representation of objects is made of a weighted combination of cues from both senses, with the weights of each being determined according to the reliability of the cue [22]. In terms of how humans tend to combine information from these two senses, researchers have shown that there is a preference for individuals to explore the front of an object with vision and the back with touch [37]. When it comes to identifying curving shapes, haptic information may be superior then vision [29].

2.3 Object shape perception on robotics

Object shape perception in robotics consists of identifying or reconstructing the shape of objects [34]. In robotics, this perception is very important, not only for the robot to manipulate and grasp an object but also for it to have a better understanding of its surroundings and being able to make safe decisions. For example, if a robot understands it intersected something else than what it is supposed to touch, it can stop and ask the user if it is safe to continue.

As mentioned previously, object shape recognition can then be determined with either vision or touch, but the type of information obtained from these two methods can be very different. For instance, only with the vision it is possible to determine the color of the object, while the texture, hardness, and temperature of the surface can only be determined with tactile perception. Touch sensory also simulates human perception and cognitive mechanism and helps to perceive and interact with the environment [34].

Vision and tactile also differ on the scale of perception. Object shape perception can be done globally or locally. In Figure 2.1 it is possible to see, on the global scale, two balls. On a local scale, one ball with small pits areas and another with curvilinear lines. Studies have concluded that vision can be considered a shape perception of a global scale, while tactile of a local scale. This last one requires multiple gasps to obtain information from different parts of an object. When talking about information obtained on each method, being the visual perception on a global scale, the information obtained is richer, while tactile information is more local and spare [34].



Figure 2.1: Two balls with different local shapes. On a global scale, they are both balls. At a local scale, (a) has small pits and (b) has curvilinear lines , [34]

When comparing vision and touch in terms of computational analyses, visual data is not complex but requires high computation, while tactile information is complex but requires low computation [34]. Restrictions about vision perception go from illumination, rotation, and orientation to background clutter and occlusion [34]. With tactile perception, rotation and orientation can be also difficult to understand and errors may occur regarding different impressions caused by forces of various magnitudes and directions [34]. Studies show that human interactions with the

environment are based on both visual and haptic information, and these two modalities combined contribute to the perception of objects and the surroundings [10].

In robotics, to work in an unstructured environment or even just to get better performances and decrease errors, it is important to have different sensing modalities, in order to have higher-level representations of the objects and environment. With this fusion of modalities of sensing it is possible to acquire complementary information that would not be able to reach with only one of the methods. Even if the information is redundant, it is important because it reduces uncertainty and increases the reliability of the system in case of failure [34], [10]. The most used combination of sensing methods is tactile with vision. There are studies of this fusion starting from 1980 [5], where tactile sensing shows to improve performance in object recognition, in support of the vision.

With the evolution in the last decades of tactile sense, this combined with vision has shown to be very useful for several applications, like verifying contacts, where vision creates a rough model of the object and then tactile sensing obtains more detailed information from regions of uncertainty for vision like holes or boundary curves [34]. An example can be seen on [6], where after vision obtains the object contours and edges, tactile sensing is used to obtain 3D coordinates, and surface normal and tangential information of contact points. On [25], tactile and vision are also combined to better identify the content inside a container, where this object is squeezed by a robot hand, which captures pressure distributions with tactile sensors, and deformation is observed by vision.

Another important application that uses the combination of these two methods of sensing is for identifying position and orientation. Accurate information about the position and orientation of an object is essential for the right interaction of the robot with the object. An error, even if small, can compromise the success of the task, so it is of big importance to have an accurate, fast, and robust perception of these factors. Computer vision is the method most used to accomplish this task, but tactile sensing can be very important since, when the robot approaches the object, occlusion surfaces appear, which can cause errors. The robot being able to tactile sense an object can reduce errors caused by the restrictions of visual perception [34]. Regarding areas of applications using tactile perception, it has already been used in minimally invasive surgery, medical training simulations, and even for interfaces of games [34]. It is widely used in robotics for sliding detection, tactile servo, object localization, and 3D modeling [32].

Table 2.1: Steps of the process of sensing, perception, and acting. Based on [20]

	Tactile sensing system	**Hardware**	**Description**
Sensing	Transduction	Tactile sensing array	Firstly, contact parameters are measured, like the magnitude and directions of forces
	Read out signal conditioning	Analog sensors front end	A signal conditioning is required to process the analog data. The method chosen to do the transduction and the signal conditioning set the bandwidth of the data
	Linearization, compensation, etc.	Digital Core	Some low-level signal processing can be done to reduce the amount of data to be processed on the robot's processing unit
	Transmission	Transmission Bus Interface	The transmission of data is usually done using serial buses, especially the ones using a controller area network (CAN) protocol
Perception	Multiplexing	Communication Bus Hierarchy	
	Representation: data selection; sensor fusion; high level computation	Computing H/W	Some redundant data is rejected. Data from different sensors is combined. A representation of the tactile data is made, obtaining the object properties
Action	Control action: signal to actuators	Controller	Based on the information acquired, some control actions are made in order to send orders to the actuators

2.4 Tactile sensing

The action of touching an object and converting this sense into information can be divided into two processes: sensing and perception. When touching an object, the first thing that happens is the process of sensing, where the tactile sensors feel a change caused by an external stimulus. After sensing, the information obtained is interpreted and transformed into properties of the object – perception. With this, a control action can be done, and a response sent to the actuators, in order for them to act according to the environment's needs. On table 2.1, there is a description of the steps that happen between touching an object and creating a control action [34], [20].

Regarding shape recognition, vision has been the most evolved and studied area, while tactile perception only expanded in recent years, and does not have a unified theoretical framework [32], [46]. There are already different types of mechanisms for tactile sensing, but no mature method has been established for classification and collecting data [32]. There is still a need for much research about shape, texture, and deformation, so it can be used in practical applications [32].

Research about tactile sensing solutions in robotics has been done in the last decades, along with studies about how to convey tactile information. However, this last topic does not have a comprehensive study, while the solutions for tactile sensing have already extensive literature [34], [32]. There is a need to explore how humans treat the information obtained from tactile sensing, in order to create decisions. Examples of these studies are experiences with blind subjects.

2.5 Object recognition on the absence of vision

In order to better understand how and which information is acquired from tactile exploration of objects, it is important to study it in the absence of vision. For this, studies have been made with visually impaired people. The results show that some abilities are enhanced with vision loss, while others do not show as much improvement [24]. On [24], a study was made with blind and sighted subjects where they were asked to touch surfaces with different stimuli. The goal was to determine whether the loss of vision develops tactile acuity. The results showed that there is a better acuity in blind individuals and that it is independent of the degree of childhood vision, light perception level, or Braille reading. Better results were seen in women and there was a decline in performance with the age for all the groups. The blind subjects had similar results on acuity as a sighted subject of the same gender and 23 years younger. Another important conclusion to highlight was the improvement of results for bigger contact forces.

The ability to recognize 3D shapes with touch was studied on [28]. For blind and blindfolded-sighted subjects, the performances were similar in time and accuracy, but there were differences in their descriptions. The first group used geometrical terms, while the second made descriptions based on everyday objects. One more group was studied, the sighted, that showed better performances and their description was in terms of geometry, like the blind subjects. Although the results on time and accuracy were similar for blind and blindfolded subjects, this study shows how there is a difference in the way the information received is processed, with the blind subjects showing a bigger geometrical understanding than the sighted, in the absence of vision. [1] Another good way to study tactile shape recognition in the absence of vision is with haptic technology.

[1] on [28] there were made experiments with blind, sighted, and blindfolded people. The conclusions taken from the tests with blind people may not be reliable, since the number of participants was small. Also, the number of curved objects was limited, so, again, the conclusion taken from the tests with objects with curve shapes may not be reliable.

2.5.1 Haptic technology

Haptic technology consists of a human-computer interaction with touch, that allows users to feel force and motion in a virtual environment (VE) [44], [33]. Haptic is a very recent and promising technology that can change human interactions and communications. It allows people to touch virtual objects through a haptic device and sense them as real, by the use of forces, vibrations, and motions. Haptic technology allows not only this human-machine interaction but can be of great use combined with other technologies [33]. With haptic technology, users can manipulate 3D virtual objects and sense their shape, weight, texture, and even temperature. The interaction with virtual objects can go from touching a wall to more complex domains, such as performing a critical surgical procedure [44].

On this type of haptic interface, there are at least three components: the human user, the interface device, and the VE created by the computer [49]. For the human user, it is important to understand how perception and manipulation work, when interacting with the computer. Perception is based on kinesthetic and cutaneous sensors spread along the body [49]. On the other hand, manipulation happens based on motor control parameters (such as force or degrees of freedom), magnitude, and resolution [49].

The interface device performs the sensing and actuation. Firstly, the device senses the manipulation done by the user, like force or motion, and passes on this information to the VE. Then it actuates on the user, according to the signal received from the VE [49]. As for the Virtual Environment, a wide of tasks are done, with it being the component that receives the information from the interface and then evaluates what action the device should do [49]. In figure 2.2 is possible to see an explanation diagram of how the human-computer interactions occur.

This field has big potential but is still in the beginning stage, with yet unsolved limitations. A universally accepted interface does not exist, causing the need to develop one new interface for each application. Also, the haptic interface uses a limited magnitude of force, with this force being sometimes unequal in all directions. The rendering is not perfect since the user operates on continuous time, while the rendering algorithm on discrete. Some instability can appear since the haptic rendering rate must be of 100 Hz but the update rate of the graphic is 60 Hz. The haptic data is heavy and requires a big amount of bandwidth, which if not available can cause errors in the interpretation [44]. On [9] some haptic devices were studied, and the author divided them into categories: the ones that use cutaneous sensing and the ones that use kinesthetic sensing. In figure 2.3 there are some examples of the first ones, where the devices use vibration, mid-air, skin deformation, or active surfaces. As for kinesthetic sensing, the most used devices are based on manipulation, grasp, or exoskeleton (figure 2.4).

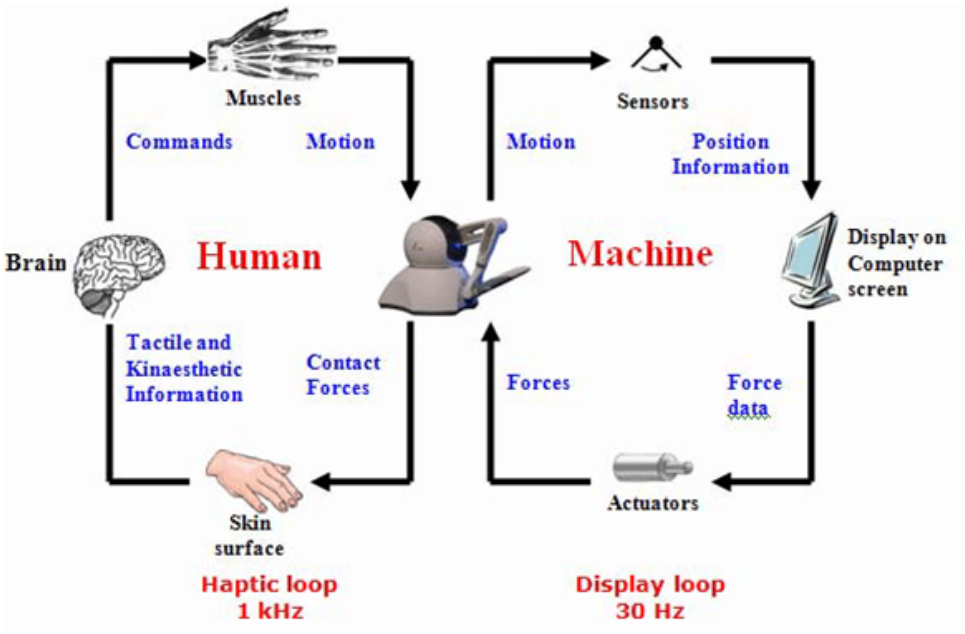


Figure 2.2: Interaction between the human with the VE through the haptic device [12]



a) LinkTouch b) FingerSight c) TextureTouch d) Ultrahaptics

Figure 2.3: Four haptic devices that use cutaneous sensing (a) from [33], b) from [13], c) from[47] and d) from [8]



a) Wolverine b) MR-Mechanic c) Phantom Omni d) Joystick

Figure 2.4: Four haptic devices that use the kinesthetic sensing (a) from [14], b) from [11], c) from [2] and d) from [36]

Haptic technology started to be used in the military, for training simulations with pilots and astronauts, but with the advances and over the years started to be an area of interest in many fields [33]. It is now being studied in multiple areas like engineering, biomechanics, computer science, neurology, and psychology [44]. Today, haptic is used for educational experiments, gaming, and virtual representation; in education it can be used on students to understand abstract and complex contents, like force effects, or even on training motor skills like surgical procedures. It is used in a wide range of areas, like special training, medicine, engineering, physics, art, sports, etc. [33].

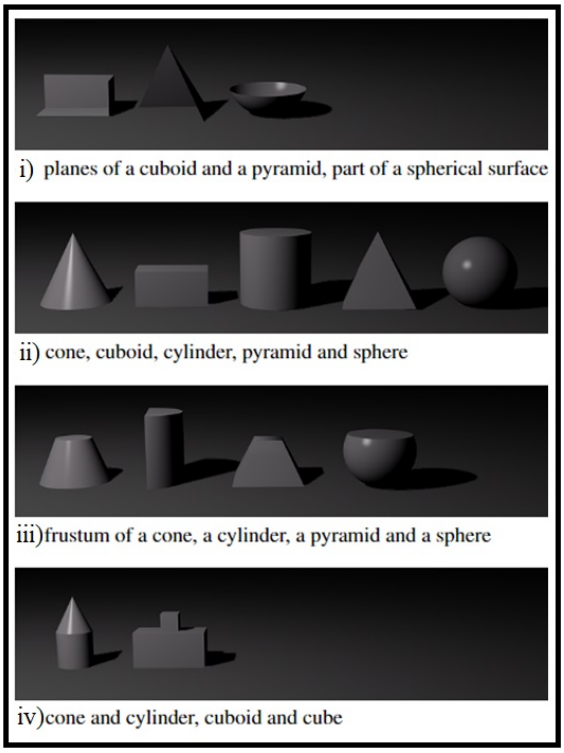
2.5.1.1 Previous tests with haptic devices

To understand the usability of haptic devices in recognizing shapes, experiments have been carried out over the years. On this sub-section, three different experiments will be analyzed and compared: [35], [39] and [45]. In the three experiments, individuals were asked to touch (without being able to use vision, using only the tactile sense) a set of shapes using a force feedback device (figure 2.5). Users were then required to identify the shapes, among other questions. These three experiments were chosen to be explored since they all use haptic devices similar to the one used in the development of this dissertation. On [35] and [39] the users felt 2D shapes, while on [45] the the collection of objects were 3D (figure 2.6).

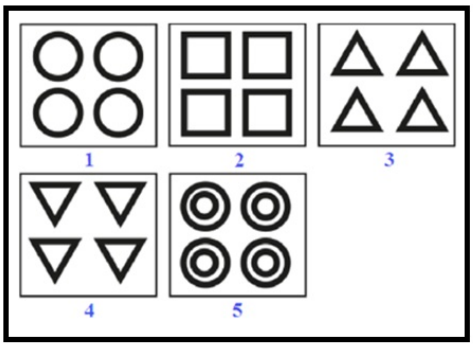


Figure 2.5: Haptic force feedback device used for the three experiments [2]

Firstly, before the tests begin, some information about the device and the procedure of the test was given to the individuals. Also, some time to explore and try the equipment was allowed, since in all the experiments the users were not familiar with the haptic device. In all three experiments, it was possible to see the importance of the device guiding the user through the contours. On [39], in one of the tests, users were asked to draw some shapes as fast and as precise as they could, first with a workplace limited by the contour of the shapes, and after without this limitation. The results of the drawing of a square can be seen in figure 2.7, and they show how the limitation of the workplace helps the user to be more precise.



a) 3D solids from the study Stamm et al. (2010)



b) 2D figures from the study of Martínez et al. (2013)



c) 2D figures from the study Rodríguez and Velàzquez (2012)

Figure 2.6: The shapes given on the three tests

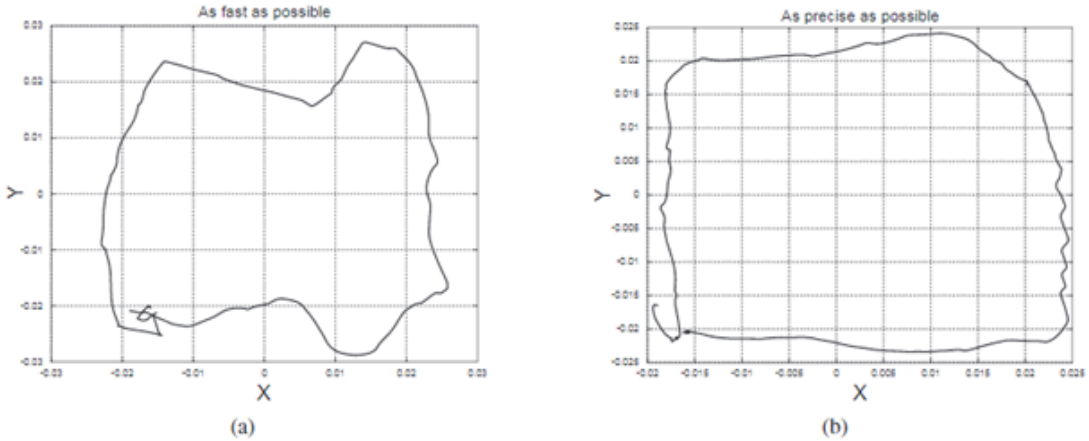


Figure 2.7: The draws of a square done by a user with and without limitation of the workplace [39]

On [35], the test done with the force-feedback device was also performed with a vibrotactile glove (figure 2.8), which did not guide the user through counters. The results showed that without the help of the device to guide the user through the counters they tended to confuse the circle with the square (figure 2.9). This confusion was not seen with the force feedback device, where users were guided through the counters. So, all the results showed that the user explores the object by following the contours of the shapes.



Figure 2.8: The vibrotactile glove from [35]

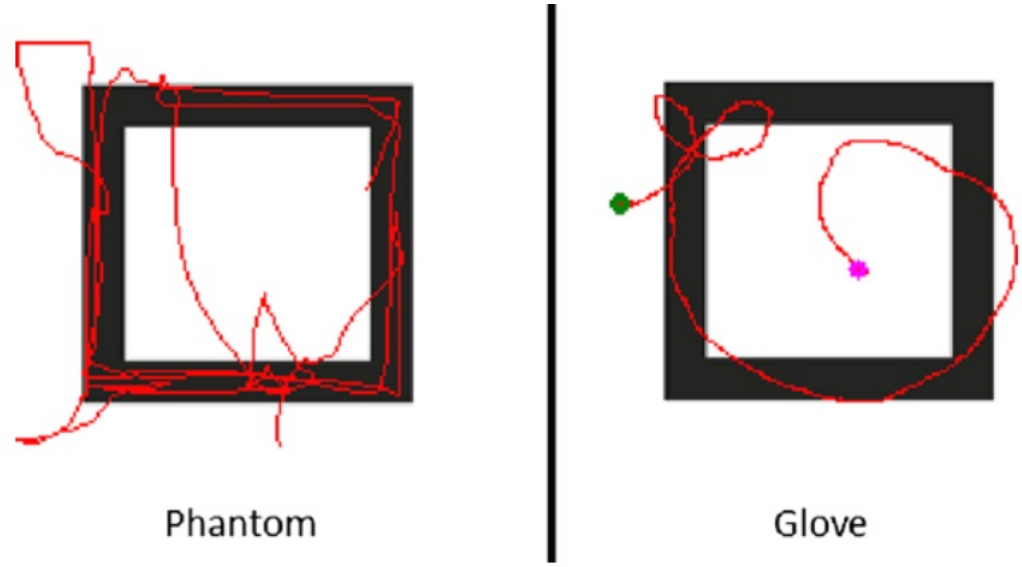


Figure 2.9: The exploration of a square done by a user with the vibrotactile glove [35]

On [45] users had more difficulty identifying figures with corners and edges since the device had a bad response for this feature of the objects; when users approached edges too fast, often they would lose track of it. The best results were for solids with no or few edges and corners, like a sphere, cylinder, and cone. Besides following corners, it can be seen that the normal procedure for users to explore objects is in horizontal and vertical directions. On [45], one of the tests was done with rotated figures, which resulted in worst performances. In this test, users were first trying to understand the posture of the different planes, and only after they started to try to recognize the shape.

Besides the device's ability to help users follow contours and the exploratory procedure chosen by users, other reasons can lead to bad performances, such as the fatigue of users when the tests are too long [35]. And, in other cases, the cause can be the insecurity of the user. From [35], the writers suggested that when an individual starts to have bad results, he can become insecure, which leads to more time on the trials. This was seen not only that when the error rate increased, the time increased too, but the contrary, that once the users start to get better performances, their time decreases. Overall, there was seen an improvement in the time taken to identify a sample along the test.

It is important to refer that the experimental procedure can have a big impact on the performance. This can be seen comparing the results obtained on the experiments done on [45] and [27]. On [27], an experiment similar to [45] was made, but using only a cube, sphere, cylinder, and cone. In the experiment from [45], the user had to explore the objects more precisely to differentiate each figure, since the list was bigger and with more degrees of complexity. This resulted in worst times for [45] when compared to [27], but in truth, this comparison can not be done, since the experimental procedure was very different, which can be the reason for the big differences obtained.

Some other conclusions to take into attention are that users have difficulty perceiving slopes since in [45] users confused cylinder with frustum of a cone. And that even when the device guides the users through corners, it can be hard to distinguish morphological similar shapes [39].

2.6 The Model of Hierarchical Complexity

This dissertation explores the cognitive process of learning shapes, based on a mathematical model that explains the stages of development of an individual according to the complexity of the hierarchical task he can perform. This theory is the Model of Hierarchical Complexity (MHC). It can be applied in multiple domains, including the learning of 3D shapes, and the stages usually go until the 16^{th} [16], [17].

Firstly, it is important to understand the definition of action and task when talking about MHC. An action is a behavior event that produces outcomes. Multiple actions can be combined in order to produce new and more complex actions [47]. A task is a set of required actions that obtain an objective. In table 2.2 it is possible to see an example of the stages of development of a person. Each stage represents a different hierarchical complexity and has tasks associated with it. For a person to be on a certain stage, that means she can perform the tasks from that level and the lower ones. This must happen because the stages have a unique sequence of hierarchical complexity, so to perform a higher-level task, one must have completed, practiced, and master the lower ones. The stage of an individual is equivalent to most hierarchical complex task solved [16], [17].

Other theories explain behavior changes along a person's age based on the development of mental structures. The MHC goes in another direction, stating that an individual changes his behavior along his age according to his changes in stages of hierarchical complexity. The stage of an individual is equivalent to the most hierarchical complex task solved in any domain [16]. Importantly, a person that performs higher in one domain can perform lower in another domain, and this is due to the experience with a given domain. Intellectual development should be measured according to the domain in which a person performed the best. This method considers that every task has two types of complexity, horizontal (classical/traditional information) and vertical (hierarchical information). The first one is related to the number of "yes/no" questions that are associated with a task, while vertical complexity evaluates the complexity of a task by splitting it into sets of lower-level sub-tasks.

This example with two tasks may be considered: A is evaluating *1+2* and B is evaluating *(1+2)+3*. In terms of traditional complexity, B is more complex than A due to the need to perform addiction more times. But this is only a repetition of actions and does not need any organization of them, not adding then any hierarchical complexity to B when compared to A [17], [18]. Consider now the task of evaluating $1 \times (2+3)$. This task requires not only to perform the actions of adding and multiplying but also an organization between them, resulting in an increase in hierarchical complexity when compared to the two-part task of evaluating *2+3* and 1×5.

The traditional complexity of a task depends on the number of times an action is repeated, while the hierarchical complexity on the non-arbitrary organization of them. Actions of a higher order of hierarchical complexity must follow some conditions:

- Are defined on two or more actions of the next lower order of hierarchical complexity;
- Organize and transform lower-order actions;
- Produce organizations of new lower order actions that are new and not arbitrary, that can not be achieved with only lower order actions.

So, for example, a task of order three operates on a task of order two, which operates on a task of order one. A task of order one is also called a simple task and it exists for all systems. For any

task, it is possible to split it into smaller tasks (figure 2.10), until the simplest one. All tasks follow a sequence of tasks, making it possible to determine a hierarchical order of task complexity. Each task can only be completed correctly or not completed at all, being the task and the performance distinct from each other [16].

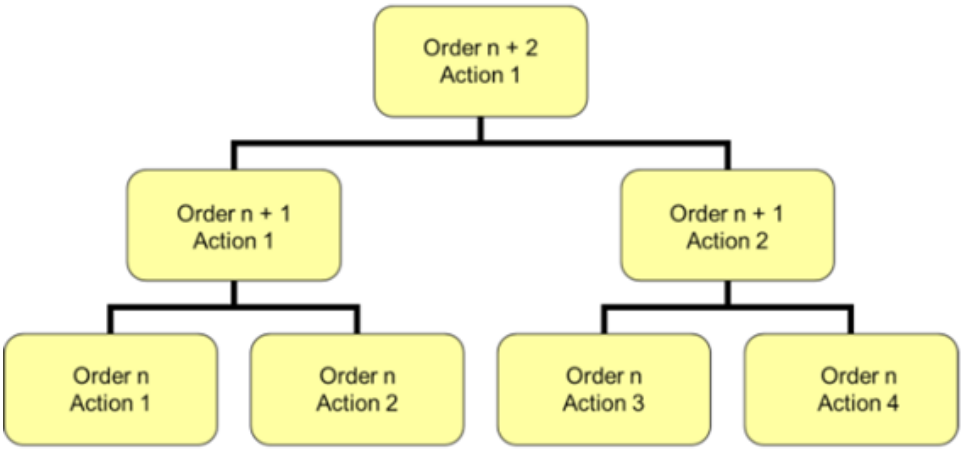


Figure 2.10: Order of hierarchical complexity of tasks [31]

This model follows some axioms that allow a correct determination of the hierarchical complexity order of an action [17]. Every action A has a hierarchical complexity *h(A)* so:

1. For a simple action A, *h(A)=0*
2. Every action A is either simple or a combination of at least two actions and a rule R that organizes these actions *A=(A1,. . . , R)*. The rule R can be a chain or coordination. A chain is only a sequence of actions in some order, while coordination requires a non-arbitrary order.
3. For a non-simple action *A=(A1,. . . ,R)*, *h(A)=maxi(Ai)* if R is chain and *h(A)=maxi(Ai)+1* if R is a coordination
4. (Discreteness) The order of hierarchical complexity of any action is a non-negative integer. In particular, there are gaps between orders.
5. (Equal Spacing) The orders of hierarchical complexity are equally spaced and all the gaps between orders are of size one.
6. (Existence) If there exists an action of order n and an action of order *n+2*, then there necessarily exists an action of order *n+1*.
7. (Comparison) For any two actions A and B, exactly one of the following holds: *h(A)> h(B), h(A) =h(B), h(A)< h(B)*. That is, the orders of hierarchical complexity of any two actions can be compared.
8. (Transitivity) For any three action, if *h(A)> h(B)* and *h(B)> h(C)*, then *h(A)> h(C)*

Table 2.2: Stages described in the Model of Hierarchical Complexity [15]

n	Stage	What they do	How they do it	End result
0	calculatory	Exact–no generalization, computations	Human made program manipulate 0, 1	None
1	sensor or motor	Discriminate in a rote fashion, stimuli generalization, move	Move limbs, lips, eyes, head; view or move objects	Discriminative and condition stimuli
2	circular sensory-motor	Form open-ended classes	Reach, touch, grab, shake objects, babble	Open-ended classes, phonemes
3	sensory-motor	Form concepts	Respond to stimuli in a class successfully	Morphemes, concepts
4	nominal	Find relations among concepts; use names	Use names and other words as successful commands	Single words: ejaculatives exclamations, verbs, nouns, number names, letter names
5	sentential	Imitate and acquire sequences; follows short sequential acts	Generalize match-dependent task actions; chain words	Pronouns: my, mine, I; yours, you; we, ours; they, them
6	preoperational	Make simple deductions Follows lists of sequential acts; tell stories	Count random events and objects; combine numbers and simple propositions	Connectives: as, when, then, why, before; products of simple operations
7	primary	Simple logical deduc tion and empirical rule involving time sequence; simple arithmetic	Adds, subtracts, multiplies, divides, counts, proves, does series task on own	Times, places, counts acts, actors, arithmetic outcome from calculation
8	concrete	Carry out full arithmetic form cliques, plan deals	Does long division, follows complex social rules, take, and coordinates perspective of other and self	Interrelations, social events, what happened among others, reasonable, deals
9	abstract	Discriminate variables such as Stereotypes; logical quantification; (none, some, all)	Form variables out of finite classes; make and quantify propositions	Variable time, place, act, actor, state, type; quantifiers (all, none, some); categorical assertions (e.g., ”We all die”)
10	formal	Argue using empirical or logical evidence; logic is linear, 1 dimensional	Solve problems with one unknown using algebra, logic and empiricism	Relationships are formed out of variables; words: linear, logical, one dimensional, if then, thus, therefore, because; correct scientific solutions

n	Stage	What they do	How they do it	End result
11	systematic	Construct multivariate; systems and matrices	Coordinates more than one variable as input; consider relationships in contexts	Events and concepts situated in a multivariate context; systems are formed out of relations; systems: legal, societal, corporate, economic, national
12	metasystematic or motor	Construct multisystems and metasystems out of disparate systems	Create metasystems out of systems; compare systems and perspectives; name properties of systems: e.g. homomorphic, isomorphic, complete, consistent, commensurable	Metasystems and supersystems are formed out of systems of relationships
13	paradigmatic	Fit metasystems together to form new paradigms	Synthesize metasystems	Paradigms are formed out of multiple metasystems
14	cross-paradigmatic	Fit paradigms together to form new fields	Form new fields by crossing paradigms	New fields are formed out of multiple paradigms

2.7 The Van Hiele Model of Geometrical Thinking and Piaget's Theory of Spatial Development

The van Hiele model of geometrical thinking is a theory about how students develop their geometric and spatial senses. It states that a person, when learning geometry, goes through a hierarchical sequence of 5 levels and that her progression is not based on her age. According to this method, the ability for one to advance to the next level is more dependent on instruction than on biology [48]. Another study made in this area was done by Jan Piaget and Barbel Inhelde, which led to the creation of Piaget's Theory of Spatial Development. This theory explains the development of spatial sense in children in stages. On [23], these two theories are compared, being concluded that some stages of Piaget's Theory are equivalent to some levels from the van Hiele model. The authors of this article joined these theories and suggested a new sequence of levels a person goes through when talking about geometry thinking (table 2.3).

2.8 The stages of the MHC on learning 3D shapes with the sense of touch

The aim of this work is to develop an application that can explore how humans learn 3D shapes through haptic exploration. This study is based on a mathematical method, the Model of Hierarchical Complexity (MHC). Rather than a theory that explains the learning of 3D shapes, this

Table 2.3: Table of the levels a person goes through on geometry thinking, based on Piaget's theory and the van Hiele model. Information based on the article [23]

Level	Explanation of the tasks a person can perform at each stage
Level 0	Is sure that a square is the same as a circle but can differentiate a ring (not filled in) from a circle (filled in). Can distinguish a ring, a two-holed surface, and a half-opened ring, but can not distinguish a circle and an ellipse. Can not distinguish a triangle from a circle or square, but correctly identifies an open ring, two intertwined rings, and an irregular surface with one hole.
Level 1	(Recognition/Visualization) The recognition of figures is made only based on their appearance and similarities with known objects, like saying a figure is a circle because it looks like a clock. Decisions are based on perception and not reasoning.
Level 2	(Analysis) The vocabulary starts to be more mathematical. A student understands the properties of a figure but does not relate them. If describing them, cannot understand which of them is necessary or redundant.
Level 3	(Ordering/Abstraction) Relations between properties and shapes are understood. A student can justify with logical arguments his reasoning. Statements like a square is a type of rectangle are done.
Level 4	Deduction) Deductions, postulates, theorems, and proofs are understood.
Level 5	(Rigor) Students compare mathematical systems to make deductions and understand the use of indirect proof and proof by contrapositive.

is a theory that explains the stages of human development, and can be applied to many domains, including the learning of 3D shapes.

Firstly, there is the need to develop the stages of the MHC for the learning of 3D shapes, since studies of this theory in this domain do not have been done yet. A concept to follow from this model is that the actions of one stage are created by joining two or more simple actions from the level before. For example, one stage is knowing the numbers 1, 2, and 3. A higher stage is knowing how to join these numbers, learning that the action of adding 1 plus 2 is equal to 3.

The definition of actions related to learning 3D shapes and their integration in existing orders of hierarchical complexity will be done based on the general actions achieved at each stage, presented in table 2.2.

The first stages occur at a very early age of the infant, so the actions that he can do are very simple. At stage zero there is no end result. The sensor or motor stage is described as the stage where the infant can move his body, like his eyes and his head. So, for learning 3D shapes, this is the stage where the infant moves his arms without purpose and touches objects unintentionally. At stage three, the circular and sensory-motor, the two actions from the stage before are joined. The infant starts to move his body with a purpose, in order to touch or grab an object. Once grabbed, the movement of the object is done on a random trajectory. Stage three is where concepts start to be formed and the infant responds to a stimulus. He can move objects with a purpose, understanding it is possible to create other actions with these movements, like banging them against each other.

The next four stages, from the 4^{th} to the 7^{th}, can be related to the first four levels described on [23], which combines Piaget's Theory of Spatial Development and the van Hiele model of geometrical thinking. The 4^{th} stage is where the infant starts to find relations among concepts and uses names. In terms of learning 3D shapes, this stage can be where characteristics of objects are recognized, e.g., their sharpness or not, their size... Next, the sentential stage is where sequences are acquired and it is possible to follow short sequential acts. This can be translated to identifying shapes based on their appearance and similarities with already known objects; like saying an object is a cube because it looks like a box. The 6^{th} stage is where simple deductions start to be made and it is possible to follow lists of sequential acts. To be at this stage, the person must be capable of understanding the properties of objects but does not relate them yet. For the 7^{th} stage, the primary, a person should understand simple logical deductions. To be on this stage, the person can logically classify families of solids, but complex figures made out of more than one shape are still described as known objects. When asked what the figure 2.11 is, the person describes it as stairs instead of saying it is three cubes.

In the 8^{th} stage, the concrete, a person can carry out arithmetic operations, so she should be

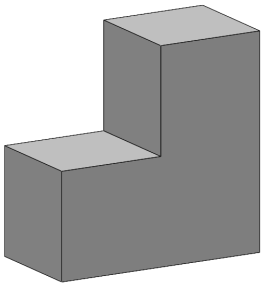


Figure 2.11: The representation of a part of a stair done from three cubes (objects created on Solidworks® by the author)

able to understand complex figures in geometrical terms; should be able to decompose and describe a complex object by geometric shapes. In the 9^{th} stage, the abstract, the person should discriminate variables and do logical quantification; understand when to use "none", "some" and "all". At this stage, the person should be able to understand a shape from a description. Then, the 10^{th} stage, the formal, is described as where the person can argue using empirical or logical evidence. For this stage, when an incomplete description is given, the person should be able to argue about the possibilities of shapes possible.

Along this process of designing the tasks, it was established that the stages to test on the application would be from the 5^{th} to the 9^{th}. To test the application, the participants must interact with a haptic device, which should create a tendency of the sample being constituted mostly by students from the *Faculdade de Engenharia da Universidade do Porto*, since it is the place where the device is. It is then expected for the sample of participants to be mostly students between 18-25 years old with good knowledge of geometry. Although having this good knowledge, this study requires the user to explore objects using only the touch. This skill to understand shapes using only this sense is not something common for the subjects. Also, this exploration of objects will be done through an unfamiliar device, where it is only possible to have one point of contact with the object. These factors add some difficulty to the task and so it was concluded that the stages to test should start on a lower one, the 5^{th}, and go until the 9^{th}. Since this work is only going to study the stages from the 5^{th} to the 9^{th}, it was decided that coming up with the definition of tasks until the 10^{th} was enough. In the table 2.4 there are represented the stages designed along this section, for the domain of the recognition of 3D shapes.

Table 2.4: Tasks developed for each stage of the MHC applied to the domain of haptic recognition of 3D shapes

n	Stage	What they do	How they do it	Tasks performed for the haptic recognition of 3D shapes
1	sensor or motor	Discriminate in a rote fashion, stimuli generalization, move	Move limbs, lips, eyes, head; view or move objects	Moves the arms without control of the direction. Touches objects unintentionally
2	circular sensory-motor	Form open ended classes	Reach, touch, grab, shake objects, babble	Moves his body to touch/grab an object. Once grabbed, moves the object on a random trajectory
3	sensory-motor	Form concepts	Respond to stimuli in a class successfully	Starts to move objects with a purpose. Bangs objects against each other
4	nominal	Find relations among concepts; use names	Use names and other words as successful commands	Starts to recognize some characteristics of objects by how they feel: it rolls, it's sharp, it's big, it's small
5	sentential	Imitate and acquire sequences; follows short sequential acts	Generalize match-dependent task actions; chain words	The recognition of shapes is made only based on their appearance and similarities with known objects, like saying an object is a cube because it looks like a box
6	preoperational	Make simple deductions; follows lists of sequential acts; tell stories	Count random events and objects; combine numbers and simple propositions	Starts to understand the properties of the objects: their faces, edges, curves. But do not relate these properties
7	primary	Simple logical deduction and empirical rule involving time sequence; simple arithmetic	Adds, subtracts, multiplies, divides, counts, proves, does series task on own	Can logically classify families of simple solids (square, sphere, cone, cylinder, ...) Complex figures are still described based on known objects, like a big cylinder with another little one on top looks like a water bottle

n	Stage	What they do	How they do it	Tasks performed for the haptic recognition of 3D shapes
8	concrete	Carry out full arithmetic form cliques, plan deals	Does long division, follows complex social rules, take, and coordinates perspective of other and self	Can understand complex objects made out of more than one geometry shape. Can decompose a complex object into simpler ones
9	abstract	Discriminate variables such as Stereotypes; logical quantification; (none, some, all)	Form variables out of finite classes; make and quantify propositions	Can understand a shape from a description
10	formal	Argue using empirical or logical evidence; logic is linear, 1 dimensional	Solve problems with one unknown using algebra, logic and empiricism	When given an incomplete description of a shape, is able to argue the possibilities

Chapter 3

Metodology

3.1 The questions for each stage to study

The first information to ask the user must be some personal data, to understand what characteristics the sample of participants has. Is also important to ask about some information that can influence their performance on the test: if they have already used a haptic device or if have any tactile or visual disability. So, the personal data that is going to be requested from the user is:

- age;
- genre;
- degree of education;
- if has ever used a haptic device;
- if has any disability that affects the sense of touch;
- if has any disability that affects the vision.

Firstly, before the stages of hierarchical complexity for the haptic recognition of 3D shapes were fully defined, the idea for the test was two create two different versions and to test them with two groups of participants. The test would consist of giving to the user some objects to explore and asking them to name/decompose them. One of the tests would start with simpler shapes and grow to unfamiliar objects that would be made out of objects that appeared before. For example, firstly the user would explore a cube, secondly a rounded cube, thirdly a cube with a cylinder on top, then a cube with a cylindrical hole, and finally a rounded cube with a cylinder on top and a cylindrical hole. On the other version of the test, the order given for the objects would be random. This would allow comparing results, to understand if with the first one, where the user goes through steps until arriving at the more complex shape, would result in better performances. After defining the stages for the task of haptic recognition of 3D shapes, it was clear that the test taught was only approaching only two stages, the 7^{th} and 8^{th}, so the idea was kept for them and more questions were thought to represent the other stages.

The idea of the MHC is that a person develops in an area by going through concepts, step by step, starting from a simple one and growing for others with more complexity. In this study, first, it will be given some simple figures to the participants, like a cube and sphere, and some questions about them. Later, the objects given will be more complex, made out of two or more simple figures. So, it is important to first define which shapes are gonna be given to the participants to explore. Taking into consideration that the participants of this test will require some time to explore each object since the method of exploration is not familiar, it seemed reasonable to start with only three main simple figures: a sphere, a cube, and a cylinder (figure 3.1).

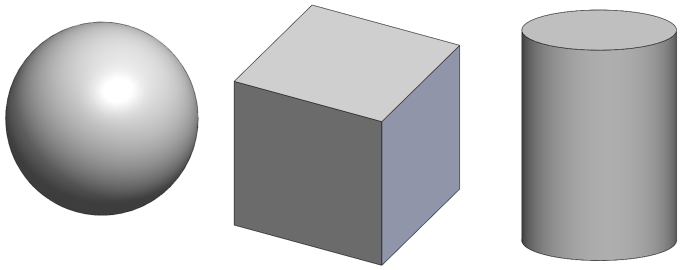


Figure 3.1: A representation of a sphere, cube, and cylinder (object created on Solidworks® by the author)

Touching and trying to recognize a virtual object through a pen is a task very undeveloped by the average person. It is then important that the participants of this test have some time to get familiar with the device and explore an object before starting the questionnaire. So, before starting with the questions it is presented to the user an object with the shape of a cone, along with information about what shape this figure has. Then, the first stage to be studied is the 5^{th}, which is defined as where the recognition of shapes is made based on appearance and similarities with known objects. Since the sample of participants expected is already familiar with geometry terms, it was decided to present to the user some objects with simple shapes to feel and just ask if they seem familiar. The objects to give will be a sphere, a cube, and a cylinder.

In the 6^{th} stage, the pre-operational, properties of the objects start to be understood, like their faces, edges, and curves. For this stage, it was decided to give some objects to explore and ask the user questions of yes/no: if they have vertices, edges, plane surfaces, and curvilinear surfaces. The shapes given were the same as the stage before and it was decided to introduce another one: a cube with the edges curved (figure 3.2). This new shape was decided to be studied because it can give some perspective on how the user is thinking. While on the other familiar shapes, the user could try, first to recognize the shape, and after answer the questions based on the information he knows about it. With an unfamiliar shape, the participants would have to overpass their knowledge of the usual 3D shapes and try to focus only on the properties of the object.

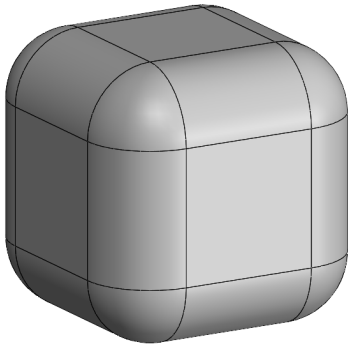


Figure 3.2: A representation of the rounded cube (object created on Solidworks® by the author)

The 7^{th} stage, the primary, is when families of simple solids can be classified. The person can identify which objects have the same shape and can also relate the shape with its name. It was then decided to ask two questions, one for each of these topics: firstly, if any of the objects given have the same shape (and if the answer is yes, which ones); and then, ask the user to name the shape of each of the objects. For this, it was decided to give two cylinders of different sizes, a cube, and the shape from the question before, the cube with round edges. With this complex object appearing again it is possible to have a better understanding of how the users perceive it and if they can distinguish it from a normal cube.

For the 8^{th} and 9^{th} stages, it is given to the user to explore complex objects made out of two or more simple figures. One object made out of three or two shapes should represent the same hierarchical complexity, just differ in the classical complexity, which is not the main focus of the study. So, it was decided to create complex objects from two shapes, the cube, and the cylinder. In the 8^{th} stage, the concrete, the user should be able to decompose a complex object into simpler ones. So, it is given to the user the option to decompose four complex objects into cubes, cylinders, and spheres. For example, the object from figure 3.3.c) can be decomposed on a cube and cylinder, so, it is a $cube + cylinder$; As for the object d), it is a cube with a cylindric hole, so it is a $cube - cylinder$. The four objects given in this stage are represented in figure 3.3.

For the last stage to be studied, the abstract, it is given to the user a description of an object and he should recognize, from four objects, if any of them goes accordingly to the description. This description is a $cube - cube - cube$, and the objects that can be described with this are represented in the figure 3.4 c) and d). The other objects given as options are also represented in this figure.

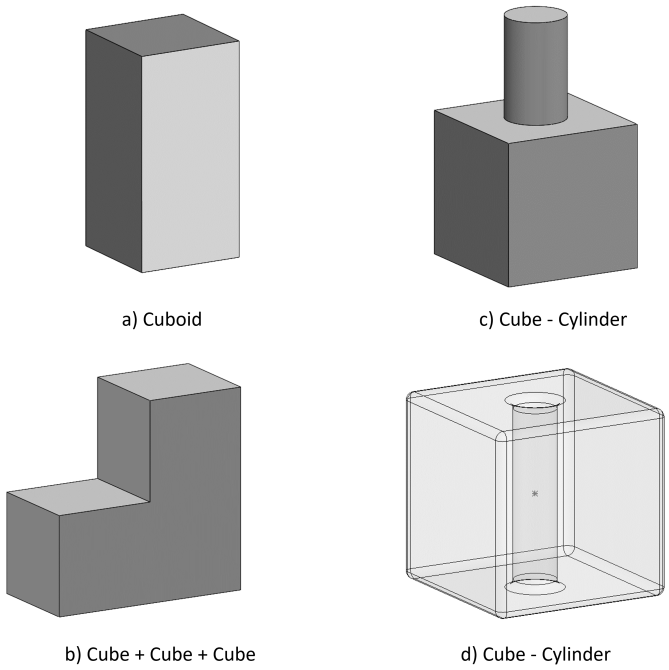


Figure 3.3: The representation of the objects from the 8^{th} stage's questions (objects created on Solidworks® by the author)

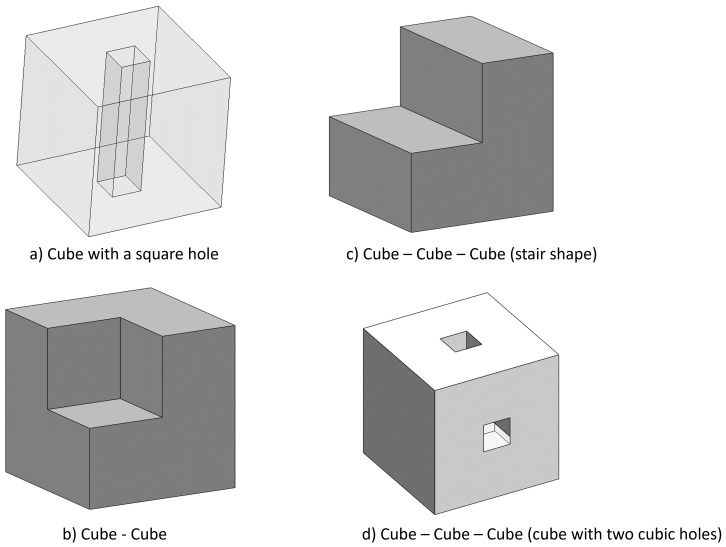


Figure 3.4: The representation of the objects from the 9^{th} stage questions (objects created on Solidworks® by the author)

Before the official test was done, a trial one was performed with a few users. Some of these users already used a haptic device and had some knowledge of the test, so their results are not important to be analyzed. This trial test was done mainly to understand if the shapes enabled a real perception of the objects, if the test had errors and if the questions were clear. From this trial test, it was possible to understand that, in the beginning, when exploring the cone, one of the users was only focusing on two directions, so it was decided to add there an indication that there are three directions of movement: up/down, left/right, front/back. Also, some participants were not understanding that on each page, a new set of objects is presented, so this information was added to every page with questions and also on the trial page. Besides this, it was possible to correct some minor errors.

The test was performed with only one device. But before, it was tried with two persons to use two devices to sense a virtual object. These individuals stated that it seemed harder to understand the object with two devices than with only one since it appears like performing two different actions at the same time. Due to the restrictions on the size of the objects, when exploring with two devices, the two pens ended up touching each other, making this process harder. It was then decided to only use one device for the test.

When first testing the application with users, some of them stated that when arriving at the 8^{th} stage's questions, one of the difficulties felt was understanding the question. Duo to this, it was decided to add to this question an example of an object and its decomposition, for the user to understand the type of answers expected for this stage's question.

3.2 The components of the haptic interface

The aim of this dissertation is to develop a haptic interface to analyze the cognitive process of humans for learning 3D shapes. For this, a user will explore 3D shapes using a haptic device and answer some questions about them. This system is constituted by a haptic device, a human user, a User Interface (UI), and a Virtual Environment (VE) (figure 3.5). The main goal is then for a user to feel objects, which are created on a VE. But for him to feel these objects, two intermediaries are needed: the haptic device, which can give the user the perception of touching an object; and a user interface, where the user can command the VE.

The user interacts with the VE through the haptic device and the UI. With the UI he can, for example, click on buttons that send commands to the VE of which shapes to display. With the haptic device, he can grab the pen and explore the workplace. When reaching the surface of an object, the user feels the contact forces rendered by the haptic device, which replicate the "feeling" of touching a real object. As for the User Interface. it is the intermediary between the user and the VE. It is where information and questions are displayed and data is saved, like the answers given and the time taken to explore the objects. The user can give commands to the VE through the UI,

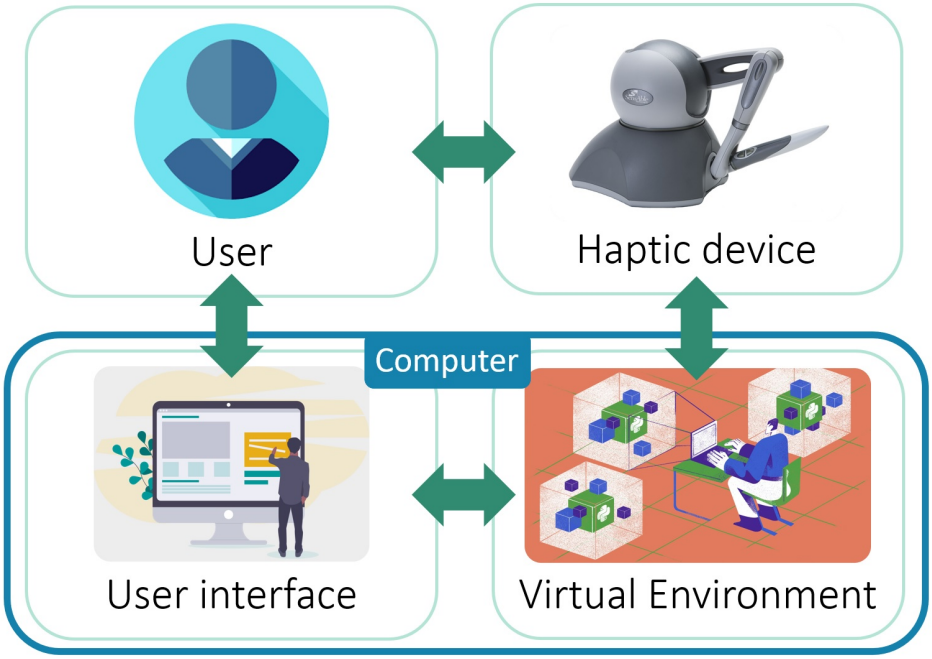


Figure 3.5: A diagram of the components of the haptic interface (composed by the author with images from [2], [41], [3], [4])

like which shape should be rendered. This application also receives information from the VE, like the time the user started to move the pen or when he encounters the object.

The Virtual Environment is the "brain" of the process. It has the code that transforms the information (received from the user interface) "Display a cube" into equations/conditions that will allow replicating the forces that occur when touching the surfaces of a cube. For this, the communication between the VE and the haptic device happens almost continuously. For example, when a cube is formed in the workplace of the device, the VE is constantly asking for the coordinates of the pen and receiving the response. The user can move freely the pen without feeling any force, as long as the pen is outside the object. But when it intersects a surface, the VE calculates the force the device should perform and sends this value to it.

Finally, the haptic device is the gadget that enables the user to interact with the objects from the virtual environment. The device used for this dissertation has a pen for the user to grab and move around the workplace.

3.2.1 The haptic device

The haptic device used for this dissertation is a SensAble PHANTOM OMNI® (figure 3.6) developed by Sensable® [1]. This device has a pen for the user to grab and move along the three axis X, Y, and Z (figure 3.6). The six Degrees of Freedom (DoF) for the position are provided by six joints/articulations: three digital encoders (encoder articulations) that contribute to the forces and allow the macro movements up-down, in-out, and left-right (figure 3.6); and three gimbal articulations to perform micro-movements and that contribute to the torque (figure 3.6). The point of contact where the device meets the virtual objects is represented in figure 3.6. Its workspace is limited to 160 W x 120 H x 70 d mm and its nominal position resolution is 0.055 mm. The maximum exertable force at the nominal position is 3.3 N. To use the device, it should be connected to the power source and to the computer, this last one is done through a FireWire connection. This information can be found on the User's Guide [42] that came with the device.

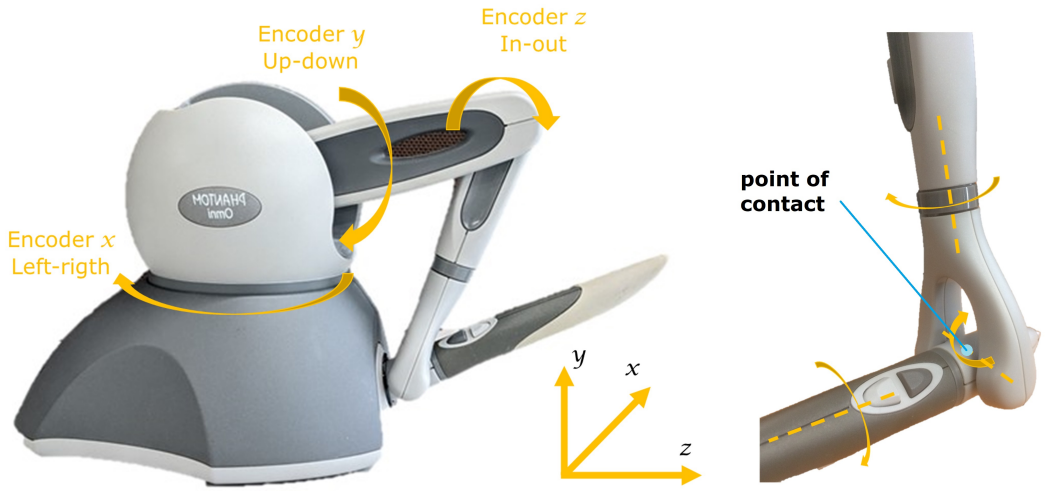


Figure 3.6: The haptic device and the representations of the movements possible to perform: a) movements along the axes, duo to its digital encoders; b) movements able to perform duo to its gimbals articulations; point of contact of the device (taken and made by the author)

3.2.2 The virtual environment

The recommendation of the 3D Systems® on how to develop applications for the Phantom is to use the OpenHaptics® toolkit on Microsoft Visual Studio®. This toolkit has a configuration and a trial application for the device, some examples of programs, and libraries of functions to use. It is used in Microsoft Visual Studio® in C or C++ language. Since the author of this dissertation had

[1] Sensable® is now part of the company 3D Systems®, which now has similar equipment to the Omni device used on this dissertation, but with a little hardware change on the type of communication with the computer

very little knowledge about programming either in C or C++ language, it was explored the possibility to instead develop the application using LabVIEW®, since it is a simple and very graphical software. From the example programs available with OpenHaptics, some simple applications were created and then tried to be replicated on LabVIEW. This process proved to be more complex than programming on Visual Studio, so the VE application was developed according to the recommendations of the manufacturers of the device: on Visual Studio, with C language. Although this, it will still be presented the process followed on LabVIEW.

3.2.2.1 Testing the haptic device

To work with the OpenHaptics toolkit, a driver is required. The Omni device has a FireWire connection, so the recommended driver is the *Phantom Device Driver v5.1.7* [1].

Before the development of the application, the device was tested with the configuration and trial programs of the toolkit. The computer used with the haptic device works with Windows 10 and has the FireWire card *2-Port PCI Express FireWire Card - PCIe FireWire 1394a Adapter*. With the Phantom connected to the computer, it was first configured and then tested, on two separate applications. The first one, *PHANToM_Configuration*, was able to recognize automatically all the information from the haptic device: the name, model, interface, and serial number. After, with *PHANToM_Test*, the device was tested with some of the functionalities of this application. Firstly, it was tried the *Read Encoder*, which was able to show the coordinates and the real-time movement of the device. Then, the *Test Forces* and *Box test* were tried. On the first one, the user can power the 3 motors on different values, which creates a movement of the device, and in the second one, a cube is rendered and possible to be touched. When trying these two tests, the device did not render any force, indicating an error between the communication of the device and the application or that the Omni may have some damage.

This specific haptic device was used before and worked correctly on a computer with Windows XP, so it was decided to also test the device on this computer. It was also sent an email to the support team of 3D Systems® to understand if maybe it is a known problem and if they have any solution. When repeating the tests on the Windows XP computer, the device worked correctly, which indicated that the problem seen was in fact in the communication between the device and the computer. After some emails were exchanged, the support team suggested it could be a problem with the firewire card or its drivers and recommended the use of the driver *OHCI Compliant IEEE 1394 Host Controller (legacy)* and the FireWire card *FireboardBlue-e™, 2-port, single chip, OHCI-compliant 1394/FireWire® host PCI Express adapter*. The driver being used on the computer was the *Texas Instruments 1394 OHCI Compliant Host Controller*, so it was decided to try to uninstall it and use the one suggested by the support team.

When proceeding to the uninstallation of the driver, this driver did not appear on the *Apps and Features* settings, the *Registry Editor* nor on the files of the computer. So it was decided to install the new driver and try to update it into the FireWire Card, skipping the step of uninstalling the other one. When trying this update, a message appeared, stating that the disk already had the more updated driver. This problem was searched on the internet and it was possible to find a solution: after installing the driver, the program of the type *inf* should also be installed [38]. With this step concluded, it was then possible to upload the recommended driver to the FireWire card. When trying the trial application, the device worked correctly for all its tests.

3.2.2.2 OpenHaptics Toolkit

OpenHaptics® is a toolkit with libraries and source code based on C and C++ programming languages. It was created to help the user to program haptic devices in a simpler way since it condensates the basic steps common to all haptic applications. It is used to develop applications on Microsoft Visual Studio® using mostly C and sometimes C++ language [26]. OpenGL® is an Application Programming Interface (API) used to create applications that produce 2D and 3D graphics, and OpenHaptics uses this API to define the geometry and implement haptic rendering algorithms.

This OpenHaptics toolkit gives the user three options to develop the Application Programming Interface (API):

- Haptic Device API (HDAPI);
- Haptic Library API (HLAPI);
- QuickHpatics micro API.

With the HDAPI, the user has low-level access to the device and the programming is done mostly in C language. The programmer can control the run time behavior of the Phantom Device Drivers (PDD) and interact with the device at a low level (e.g., read encoder or motor joint torque values). The forces are rendered directly and refreshed at a high frequency. So, to have a stable closed-loop control of the device it is important to have efficient algorithms and data structures. As for the HLAPI, it is built from the HDAPI and provides a higher level of control. It has the source code in C language and was designed for programmers with knowledge of OpenGL® API. It has a high-level haptic scene rendering and the user does not need to perform low-level actions. The interactions with the device are done in Cartesian space, unlike in HDAPI, where it is possible to obtain the encoder values([26]). The other option to develop an application with this toolkit is with QuickHaptics. This is an API implemented in C++ built from both HDAPI and HLAPI and is a fast and easy way to write haptic/graphic applications. With this, the user can easily create an application without the need to write a big amount of code, by using built-in geometries from the QuickHaptics library. Figure 3.7 has an overview of how each of these API works. Both HL and QuickHaptics API were created in an OpenGL® context and create and show objects using

OpenGL®. With HD it is possible to modify the rate of the servo loop, the control loop used to calculate and send the forces to the device.

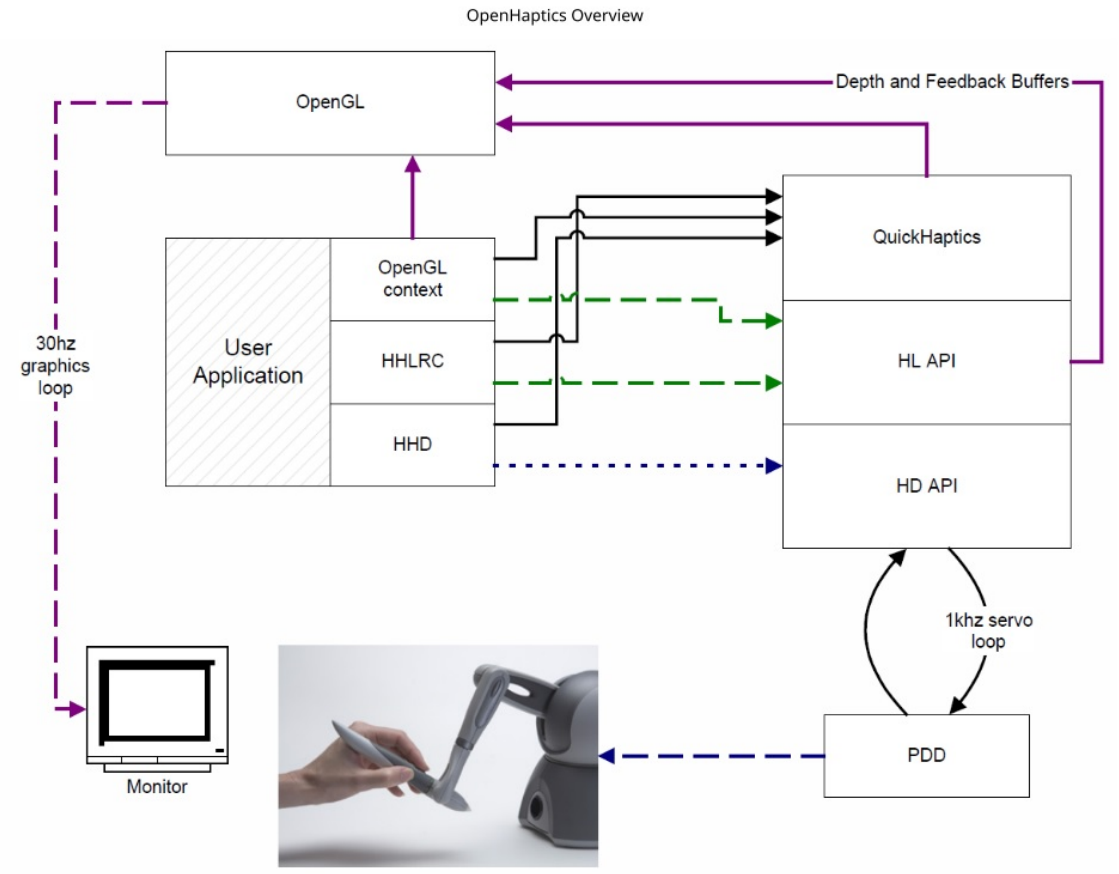


Figure 3.7: Overview of the processes that occur on a QuickHaptics, HD, and HL application ([26])

For this dissertation, the application to be developed for the VE must show multiple objects, but one at a time. The idea is for the user to explore one object, answer some questions about and then move on to the next object and questions. Besides this, is important to save information like the coordinates of the device when the user is exploring the object and some other information like the time the user touches for the first time the object or the time the user ends the exploration and puts on its initial position. To understand which API would be the better option to develop the application for this dissertation, it was consulted the catalog and examples of applications of each type of API. Observing the examples, it was possible to understand that QuickHaptics is a simple way to create an application and requires very few lines of code to create programs, but it has some restrictions. This API creates a display window with multiple objects for the user can interact with, but to have multiple objects with only one showing at a time, it would be necessary to create multiple applications, one for each object. Then, comparing the code of the HL and

HD examples, the second one appeared easiest to comprehend; the code is more "raw" and the functions used from the libraries are simple to understand. Also, the HL applications work as the QuickHaptics, by creating a display window with multiple objects. So, due to its possibility to render multiple objects one at a time, and because the code appeared to be simpler to comprehend, it was chosen to use an HDAPI for this dissertation.

The two main components of an HDAPI are the device and the scheduler. The scheduler manages a high-frequency and high-priority thread for sending forces and returning state information from the device. The scheduler is the one responsible for the communications between the application and the servo loop thread. This communication must occur at least at 1000 Hz to have stable force feedback. The user then programs the commands to give to the device on scheduler callbacks that will after be performed in the servo loop thread. These callback functions can be synchronous, which means occur one time when scheduled; or asynchronous, meaning when they are scheduled, occur on a loop until unscheduled. In figure 3.8 it is possible to see the usual structure of an HDAPI program and in figure 3.9 an example of an application following this structure. Both figures are linked through letters that indicate which part is the correspondent in the other figure. The sequence of actions is firstly the initialization of the device and enabling of the forces; then, the schedule is started and now it is possible to schedule synchronously or asynchronously callbacks. These callback functions are user-defined and are where the commands to give to the device are programmed. To do so, on these callbacks, first, a frame must be started - to guarantee a consistent device state-, and only after the interaction with the device can start. After this, the frame must be ended. When the callback is done, the scheduler may be stopped and the device disabled.

3.2.2.3 LabVIEW

LabVIEW®is a graphical control, test, and measurement environment that uses a visual programming language G. It was developed by National Instruments with the aim to help non-programmer scientists and engineers to automate their tests and measurement systems [30]. In this software, programs are built by dragging and dropping virtual representations of instruments, usually already familiar to the users. The applications created on LabVIEW®are called Virtual Instruments (VI) and have two windows: the front panel, which is the interface, where the user can drop instruments, buttons, and graphics... and the block diagram, which contains the programming code. In figure 3.10 there is an example of a simple application, where on top there is the front panel and down the block diagram.

LabVIEW®is a software designed to work with devices, so it has good support for interfacing them. It is possible to write direct bus commands (USB, Serial) or to use specific functions from a specific driver of the device. For this, Nation Instruments has a wide list of device drivers available for download. There is good documentation of the utilities, explanations on how to use

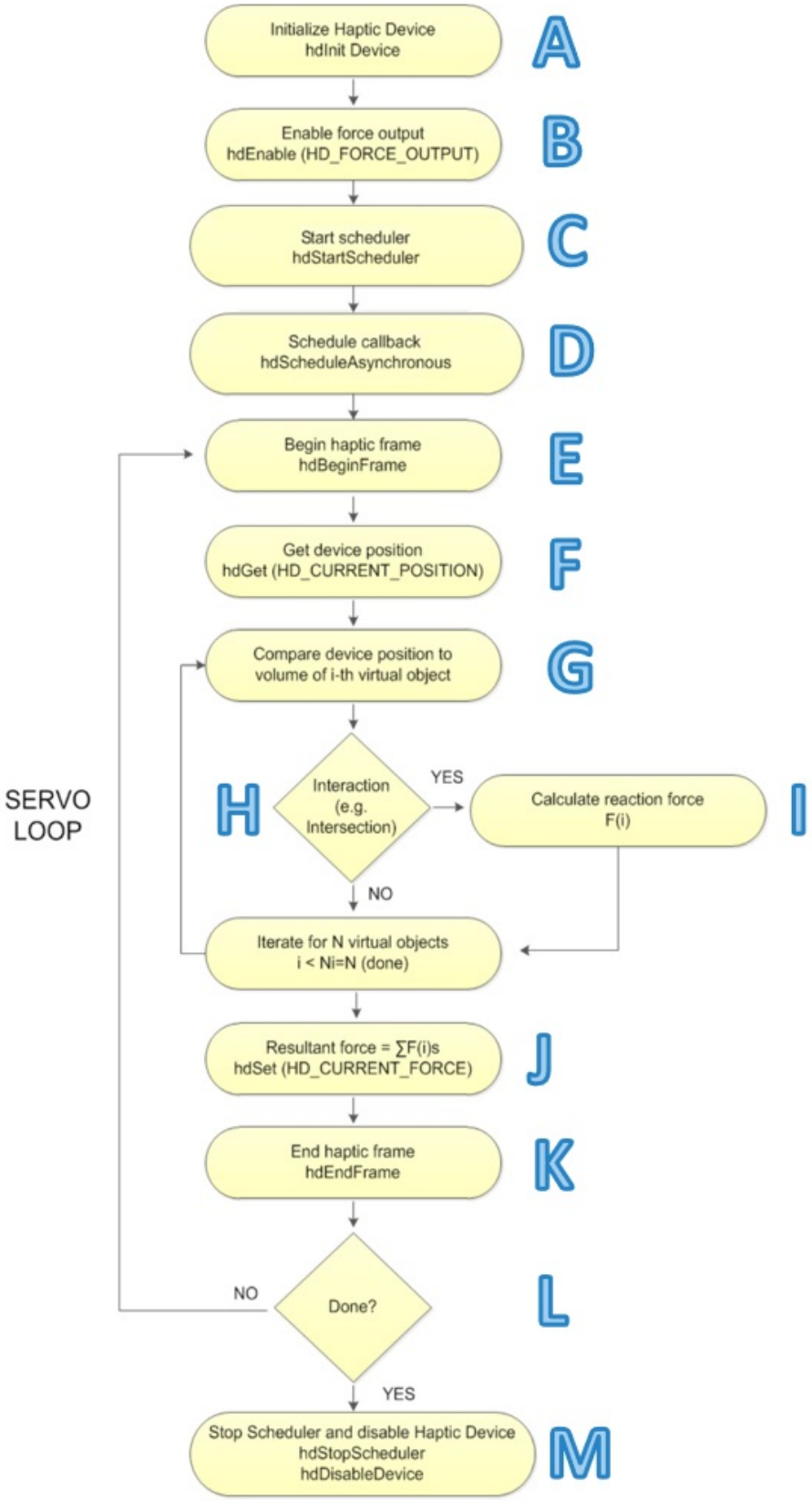


Figure 3.8: Usual structure of an HDAPI application Based on [26]



Figure 3.9: An example of an HDAPI application on Microsoft Visual Studio® (taken from one application example of OpenHaptics®)

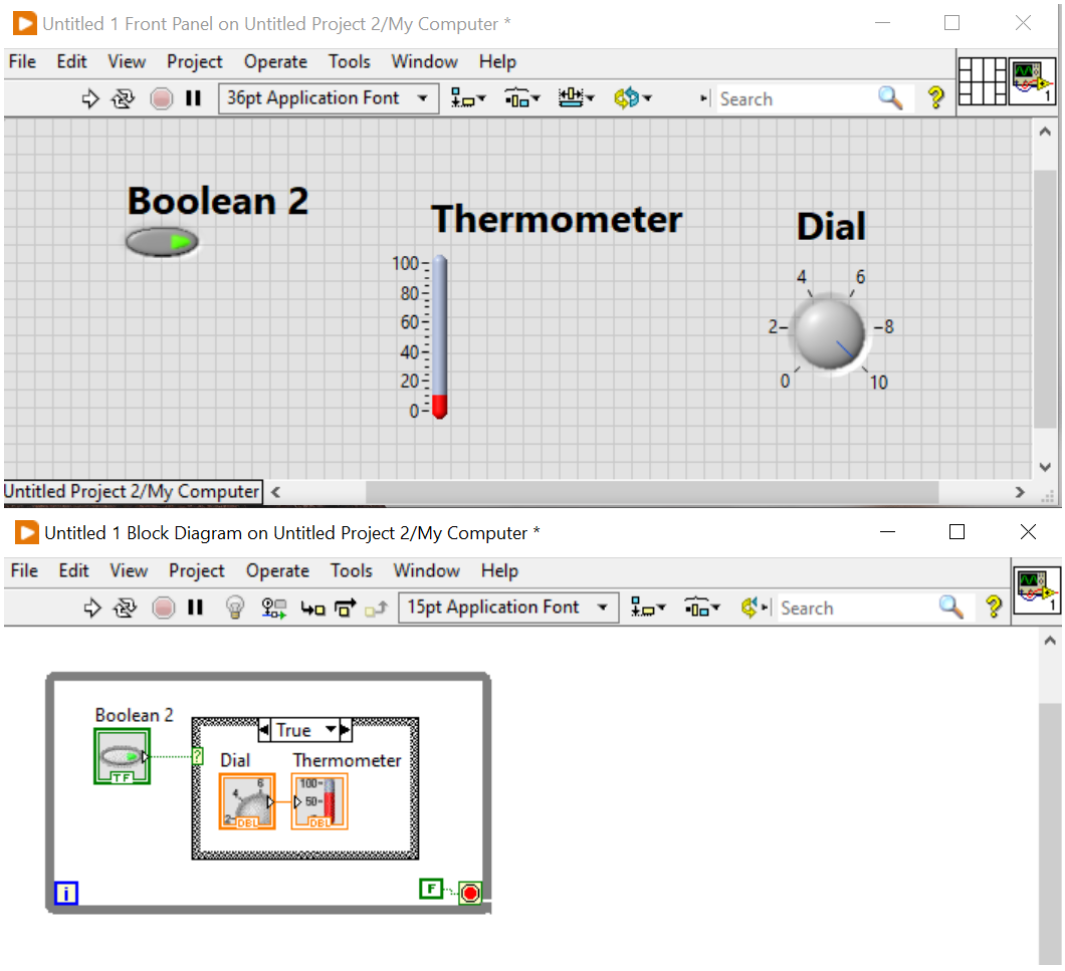


Figure 3.10: Example of a simple application on LabVIEW® (created by the author)

them, and even some examples. So, even a user without big programming skills can easily develop applications [51]. This software is very versatile, with the user being able to program in its language or even use code from another programming language. This possibility to use another programming language is done through Dynamic-link Libraries (DLL)[2] that the user can call on LabVIEW® and then use normally as a block function with input and output arguments. Due to its many functionalities and for using a simple language of programming, it was tried to develop the VE with LabVIEW®.

First, to establish the communication between the Omni device and LabVIEW®, the better option would be to have a driver specific to this device and program. Both the support teams of LabVIEW® and 3D Systems® were contacted and research was done on the internet, but this type of driver did not appear to have been developed. So, the other option was to work with the driver that establishes the communication between the haptic device and the OpenHaptics and work with the DLL of this toolkit on LabVIEW®. LabVIEW®offers three ways to import DLL, depending on the data type it uses: if it has headers or if the library is C/C++ or .NET. All the DLLs from OpenHaptics are C/C++ libraries and have headers associated with them, so there are two methods to import them: using the Shared Library Wizard (SLW) or the Call Library Function Node (CLFN). The first one is used when the functions from the DLL use supported data types, and the CLFN is for complex or unsupported data types.

So it was first tried to use the SLW. For this, it was required to match the DLL to its header. To develop a HDAPI applications, there are four DLL: *Win32/Release*, *Win32/Debug*, *x64/Release* and *x64/Debug*. When contacting the support team of 3D Systems®, they recommended using the version Release, so the DLL used was the *Win32/Release*. As for the header that corresponds to this DLL, it is the *hd.h*. With the DLL and header defined, it was tried to upload them to LabVIEW®, but an error appeared: the functions from the DLL were being read but not being linked with the header file. When opening this header file on Visual Studio, it was possible to see it has some dependencies: at the beginning of the code, it includes some other headers. To exclude these dependencies, it was inserted the code from these other headers inside the main header *hd.h*. It was tried again to import the DLL with this new header, but another error appeared. When trying to use instead the CLFN method, the functions of the DLL were imported successfully.

With the DLL ready to be used on LabVIEW®, it was to start to develop applications to interact with the haptic device. The methodology followed was to first see some examples of applications from Visual Studio, try to understand them with the help of the API Reference Guide, and then create a simple application on LabVIEW®. So, from the example *Calibration*, it was created one simpler program that only initializes the device, gets its model, and then disables the device (figure 3.11). For this program, the functions needed are:

[2] A DLL is a dynamic-link library that contains code and data. These libraries are created in order to use their information in multiple applications at the same time

- to initialize the device: *hdInitDevice*;
- to get the model of the device: *hdGetString*;
- to disable the device: *hdDisableDevice*.

```
int main(int argc, char* argv[])
{
    HHD hHD;
    HDErrorInfo error;

    hHD = hdInitDevice(HD_DEFAULT_DEVICE);
    if (HD_DEVICE_ERROR(error = hdGetError())) { ... }

    printf("Found %s.\n\n", hdGetString(HD_DEVICE_MODEL_TYPE));

    hdDisableDevice(hHD);

    return 0;
}
```

Figure 3.11: The first program created on C code on Visual Studio

The function to initialize the device has one output and one input parameter (figure 3.12). According to the API Reference Guide, the output parameter is a *handle* and the input a *HDstring*. Both parameters have data types not usual for C, so when checking the headers it is possible to obtain more usual data types for them: the output is an *unsigned int* and the input a *const char* (figure 3.12). Again by looking at the headers, it is possible to observe that the function to get the model of the device has a *const char* as output and an *unsigned int* as input. As for the function to disable the device, it does not return any value and has as input the output from the function that initializes the device.

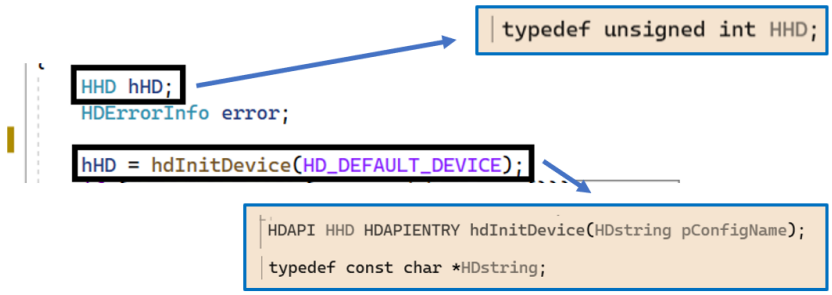


Figure 3.12: The declaration of the function to initialize the device and its parameters

On the C code, the input of the function to initialize the device is *HD_DEFAULT_DEVICE*, which goes according to its data type; but for the function to get the model of the device, the value of the input on the C code does not match its data type since it is not numeric (*HD_DEVICE_MODEL_TYPE*).

When going to the header where this parameter is defined, it is possible to see that this string is a code that corresponded to the *int* 9473 (figure 3.13).

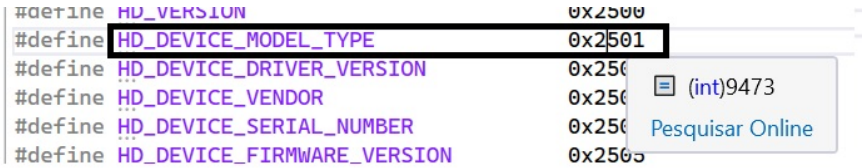


Figure 3.13: The declaration of the *HD_DEVICE_MODEL_TYPE*

With the parameters defined, the calling convention of each function was set as C, since the DLL is in this language, and the application tried. This resulted in an error, and with research, it was possible to understand that it was caused by the calling convention not being C but instead *stdcall*. With this change, the application worked and the answer "Omni" appeared as the response for the function that gets the device type.

After, another application more complex was designed, where the coordinates of the position of the device were obtained (figure 3.14). For this application, after initializing the device, it is needed to:

- start the scheduler (*hdStartScheduler*);
- create a callback where:
  - a frame is initialize (*hdBeginFrame*);
  - the position is asked (*hdGetDoublev*);
  - the frame is ended (*hdEndFrame*);
- create a structure that synchronously schedules the callback (*hdScheduleSynchronous*).

While the first four functions listed above are included in the DLL, the Callbacks are user-defined functions with complex data types. It was not possible to create them on LabVIEW®, so it was tried to create only what is inside them: to begin the frame, get the position and end the frame. The *hdScheduleSynchronous* function was also not able to be used since its first input parameter is the callback function not able to be created. With this, the application did not work, and although it was tried to understand, through Visual Studio, how this callback communicated with the device, it lead to no answer on how to replicate the application on LabVIEW®.

All this process of understanding the C code and then translating it to LabVIEW® took some time since the author is not familiar with C language, LabVIEW®, the commands from OpenHaptics nor also with headers and DLLs. The reference guide showed to be not so useful since the data type of the parameters there explained were not compatible with the ones to use in LabVIEW®. But although the development of the application on LabVIEW® was not showing improvements,



```
int main(int argc, char* argv[])
{
    HHD hHD;
    HDErrorInfo error;

    hHD = hdInitDevice(HD_DEFAULT_DEVICE);
    if (HD_DEVICE_ERROR(error = hdGetError())) { ... }

    printf("Found %s.\n\n", hdGetString(HD_DEVICE_MODEL_TYPE));

    hdStartScheduler();
    if (HD_DEVICE_ERROR(error = hdGetError())) { ... }

    printf("Press any key to quit.\n\n");

    /* Loop until key press. */
    while (!_kbhit())
    {
        PrintDevicePosition();
        Sleep(1000);
    }

    hdStopScheduler();
    hdDisableDevice(hHD);

    return 0;
}

HDCallbackCode HDCALLBACK DevicePositionCallback(void *pUserData)
{
    HDdouble *pPosition = (HDdouble *) pUserData;

    hdBeginFrame(hdGetCurrentDevice());
    hdGetDoublev(HD_CURRENT_POSITION, pPosition);
    hdEndFrame(hdGetCurrentDevice());

    return HD_CALLBACK_DONE;
}

void PrintDevicePosition()
{
    hduVector3Dd position;

    hdScheduleSynchronous(DevicePositionCallback, position,
        HD_DEFAULT_SCHEDULER_PRIORITY);

    printf("Device position: %.3f %.3f %.3f\n",
        position[0], position[1], position[2]);
}
```

Figure 3.14: The second program created on C on Visual Studio, where the position of the device is printed for the user with a 1-second interval between readings

the author was developing his understanding of the functions of the OpenHaptics and the C language. It was then decided to try to develop the VE application in Microsoft Visual Studio®.

3.2.2.4 Development of the Virtual Environment in Microsoft Visual Studio

The application to develop in Microsoft Visual Studio® is a console application without a UI, which leads to the need to have two applications communicating between them: one for the VE and another for the UI. The author of this dissertation is familiar with programming Forms applications on C# language, which are also developed on Visual Studio, and with connecting these types of applications with others through serial ports. It was then decided to create the UI on a Form application and to establish communication with the VE through serial ports.

The VE application has the normal architecture of any HDAPI program that was explained in the OpenHaptics section. Besides the components related to the haptic device, the application also has the code for serial communication with the UI. On the *main* function firstly the serial port is defined and opened. Then, the device is initialized, the forces are enabled and the scheduler started. Before entering the *while* loop that is running non-stop until the application stops, two actions are made: the device is calibrated and it is scheduled asynchronously a function that is constantly running, that checks when the user grabs the pen and finds the object being rendered. The control actions are made inside *while* loop: the callback functions that render the objects are scheduled and unscheduled, the functions that communicate with the UI application are called, etc. And before the program ends, the serial port is closed, the scheduler stopped and the device is disabled. In figure 3.15, there is schema of the *main* function of the VE application with the parts mentioned above.

The actions performed outside the *main* function are:

- functions that handle the serial communication;
- callback functions to calibrate the device;
- callback functions where the objects are rendered;
- the callback function that is constantly running, that checks when the user grabs the pen and finds the object on display;
- a function that saves the coordinates of the device pen when the user is exploring an object;
- a function that is responsible for stopping the display of an object.

The next sections will explain how the rendering of objects occurs. The other functionalities of the VE will be presented in the section 3.2.4.

3.2.2.5 The rendering of objects

Haptic rendering is the process that allows a user to feel a virtual object. For this dissertation, the objects presented to the user are frictionless since the only aim of this study is to understand

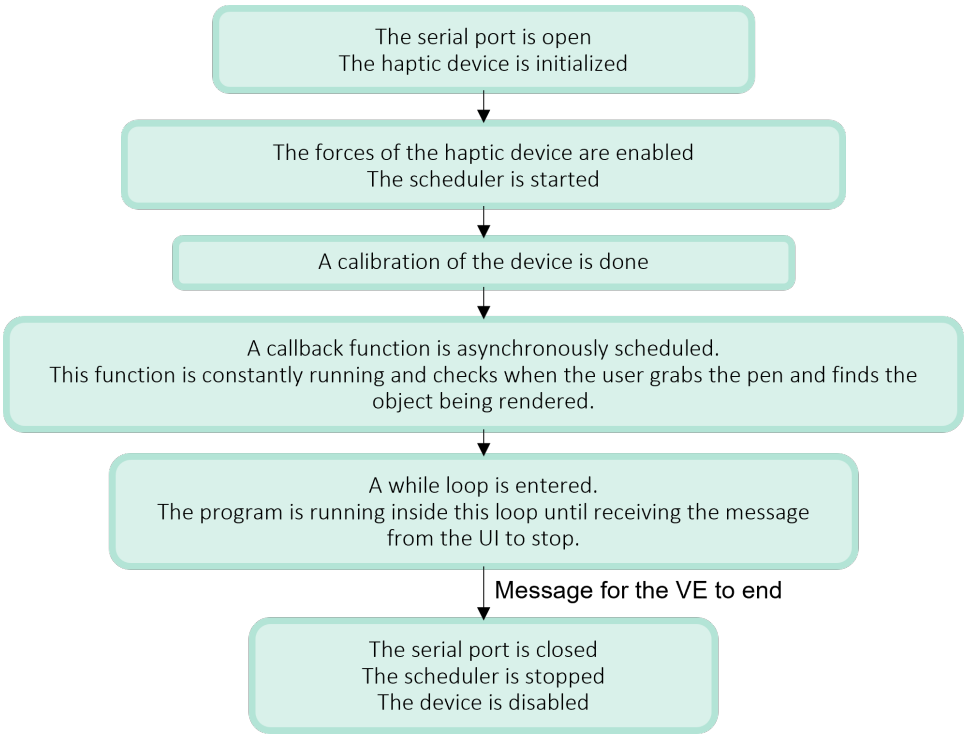


Figure 3.15: Schema of the commands that happen inside the *main* function of the VE application

the perception of shapes from the users. This facilitates the method of creating the virtual shapes since it is only needed to compute the forces normal to the surfaces. The calculation of the forces to render is going to be explained with the use of examples. Firstly it will be shown how a sphere was rendered and then two more objects.

The process of rendering the objects starts with the definition of their dimensions and their position. The workspace of the haptic device is limited by 160 W x 120 H x 70 d mm, which does not allow the rendering of objects with a big dimension, so the size of the shapes was chosen based on these limitations. With the dimensions of the object set, then its boundaries must be defined. It is when a user touches these boundaries that a force $\overrightarrow{F}$ is calculated and rendered. This force is normal to the object's surface and its magnitude is the product of the object's stiffness and the penetration depth (figure 3.16). The values for the stiffness of the object were based on the ones used on the applications given as examples in the OpenHaptics toolkit. Each object was then tested and the values of its stiffness were adjusted. The magnitude of the force F can then be defined as:

$$F = stiffness \cdot penetration \tag{3.1}$$

So, the first shape created was a sphere. It was chosen for its center C_1 to be the origin of the

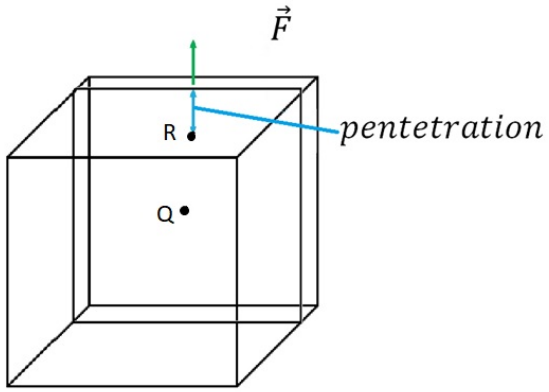


Figure 3.16: Representation of the penetration and force $\overrightarrow{F}$ performed by the top surface of a cube when the device is on the point R (draw created by the author)

Cartesian coordinate system of the device, its radius to be $r_1 = 50$ mm, and to have a stiffness of 0.25. The surface of the sphere is touched when the distance d_1 between the position of the device P_1 and the center of the sphere C_1 is smaller than the radius r_1. And when this happens, a force $\overrightarrow{F_1}$ is rendered. This can be translated in:

$$d_1 = \overline{C_1 P_1} \tag{3.2}$$

$$d_1 < r_1 \quad \Rightarrow \quad \overrightarrow{F_1} \neq (0,0,0) \tag{3.3}$$

On the figure 3.17 there is a representation of the force $\overrightarrow{F_1}$ and the other dimensions discussed above.

Once the person touches the sphere boundary a force $\overrightarrow{F_1}$ is then calculated and rendered. For this, first, it is calculated the penetration depth $penetration_1$, which is the difference between the radius of the sphere and the distance d_1 between the device and the center of the sphere. So, the penetration can be calculated by:

$$penetration_1 = r_1 - d_1 \tag{3.4}$$

This value for the penetration can be used on the equation 3.1 and obtained the value for F_1. Now the unit normal vector that defines the direction of the force $\overrightarrow{N_1}$ must be calculated. This vector is obtained by subtracting the center C_1 of the sphere from the position P_1 of the device, and then dividing this vector by its magnitude:

$$\overrightarrow{N_1} = \frac{\overrightarrow{C_1 P_1}}{||\overrightarrow{C_1 P_1}||} = \frac{\overrightarrow{C_1 P_1}}{d_1} \tag{3.5}$$

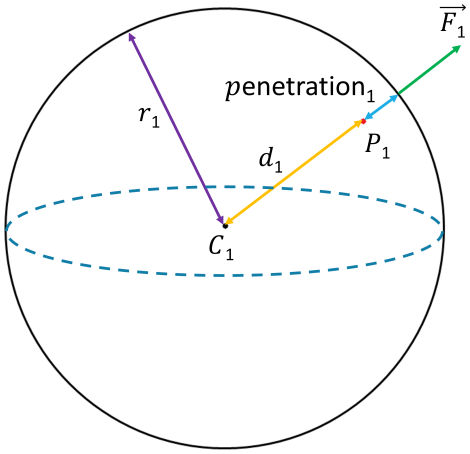


Figure 3.17: Representation of a sphere with the force $\overrightarrow{F_1}$ it creates when the device is on the point P_1. The figure also illustrates the center C_1 of the sphere, its radius r_1, the penetration when the device is in the position C_1, and the distance d_1 from the device to the center of the sphere (draw created by the author)

Finally the force $\overrightarrow{F_1}$ is calculated by multiplying its unit normal vector $\overrightarrow{N_1}$ to its magnitude F_1

$$\overrightarrow{F_1} = F_1 \cdot \overrightarrow{N_1} \tag{3.6}$$

This is the concept followed for every object. The sphere is the most basic shape to render since it has only one surface, but when the object has multiple surfaces the process to obtain the normal force is more complex. This force increases with the surface penetration, so, when defining the boundaries of a surface, what is being defined is a boundary volume. In mathematical terms, what is being defined, for example, is not a surface $x = 1$, but instead a volume $0 < x < 1$. So, when the object has multiple surfaces, multiple volumes that do not intercept each other must be created. For a better comprehension of how the shapes were created, it will be presented this process for two other objects, a cylinder and a cube with a cylindrical hole.

Starting with the cylinder, its dimensions were set for a radius of $r_2 = 50$ mm and a height of $hg_2 = 140$ mm. For his position, its center coincides with the origin of the Cartesian coordinate system.

This shape has three surfaces, one curved and two flat. These three boundary volumes were created with the help of a cone. The volumes for the two flat surfaces are two cones, and, for the curvilinear surface, it is a cylinder with the subtraction of the two cones that represent the flat surfaces. These volumes are represented in figure 3.18, on the left showing the volume for the curvilinear surface and on the right the two cones, one for each of the flat surfaces.

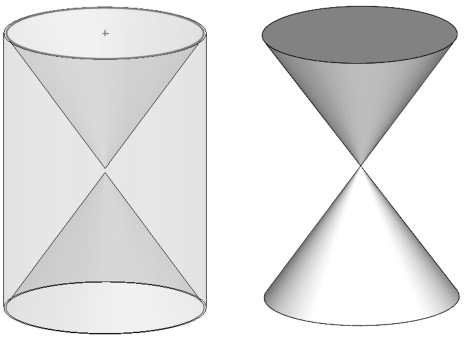


Figure 3.18: A representation of the volumes that will define the surfaces of the cylinder (objects created on Solidworks® by the author)

To define the cones, it was used the mathematical equation:

$$\frac{y^2}{a_2^2} = \frac{x^2}{b_2^2} + \frac{z^2}{c_2^2} \tag{3.7}$$

This equation defines two cones with the corner on the origin of the Cartesian coordinate system (figure 3.19), like the ones needed to render the surfaces. So it is only needed to define the values for the variables a_2, b_2 and c_2. The restrictions for these variables are:

- for the base of the cone to be a circle:

$$b_2 = c_2 \tag{3.8}$$

- for the base of the cone to coincide with the base of the cylinder:

$$\text{for} \quad y_2 = \frac{hg_2}{2} = h \quad \Rightarrow \quad x^2 + z^2 = r_2^2 \tag{3.9}$$

The two conditions from above and the equation 3.7 lead to:

$$\frac{h_2^2}{a_2^2} = \frac{r_2^2}{b_2^2} \tag{3.10}$$

Assuming for a_2 the value of 1 and to have a positive value for b_2, then:

$$b_2 = \frac{r_2}{h_2} \tag{3.11}$$

So, the equation that defines the cones is:

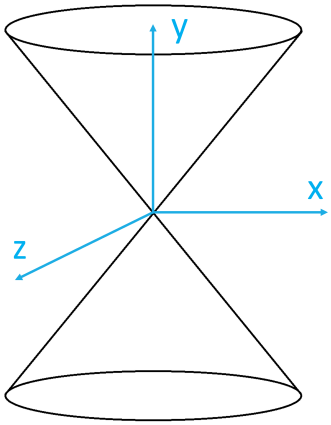


Figure 3.19: Representation of the equation 3.7 on the Cartesian coordinate system (created by the author)

$$y^2 = \frac{x^2 + z^2}{(\frac{r_2}{h_2})^2} \tag{3.12}$$

Now with the equation for the cones defined it is possible to define each of the three volumes and the forces to render. The border volume for the top flat surface of the cylinder can be defined as the volume that corresponds to these conditions:

$$y^2 > \frac{x^2 + z^2}{(\frac{r_2}{h_2})^2} \tag{3.13}$$

$$y > 0 \tag{3.14}$$

The border volume for the bottom flat surface of the cylinder can be defined as the volume that corresponds to these conditions

$$y^2 > \frac{x^2 + z^2}{(\frac{r_2}{h_2})^2} \tag{3.15}$$

$$y < 0 \tag{3.16}$$

The direction of the forces that these surfaces produce are perpendicular to the surface itself and point to the outside of the figure. The magnitude is calculated on the same basis as the sphere, with the penetration of the device on the surface. Finally, the border volume for the curvilinear surface follows these conditions:

$$y^2 < \frac{x^2 + z^2}{(\frac{r_2}{h_2})^2} \tag{3.17}$$

$$y < h_2 \tag{3.18}$$

$$y > h_2 \tag{3.19}$$

$$\overline{A_2Y'} < r_2 \tag{3.20}$$

where $\overline{A_2Y'} < r_2$ is the distance from the position A_2, where the device, is to the Y axis (figure 3.20).

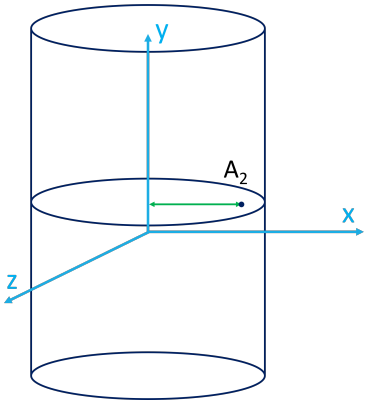


Figure 3.20: Representation of the equation 3.7 on the Cartesian coordinate system (created by the author)

The forces produced by this surface are perpendicular to the Y axis and also point to the outside of the object.

In the next paragraphs, there is going to be described how a cube with a cylindrical hole is rendered. The steps taken to create this shape were to first design the cube, create a hole - a volume that was designed for the hole where the device does not apply any forces- and then define the curvilinear surface of the cylindrical hole. The cube is centered on the point $C_3 = (c_x, c_y, c_z) = (0, 25, 0)$ and has a high of $h_3 = 70$ mm. The volumes chosen to build the cube surfaces were spheres trimmed by the cube; this shape can be seen in figure 3.21. It must be created one sphere for each of the six surfaces.

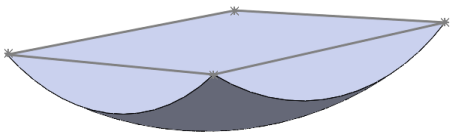


Figure 3.21: Representation of the volumes that will define the surfaces of the cube (object created on Solidworks® by the author)

Considering the unit normal vectors of the surfaces of the cube (figure 3.22), these surfaces will be described according to these vectors:

- ABCD : surface x positive (x^+);

- EFGH : surface x negative (x^-);
- ABFG: surface y positive (y^+);
- CDEH: surface y negative (y^-);
- BCGH: surface z positive (z^+);
- ADEF: surface z negative (z^-);

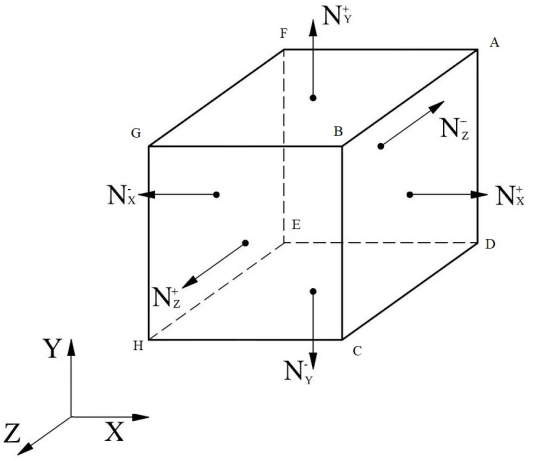


Figure 3.22: Representation of the unit normal vectors of the surfaces of a cube (created by the author)

A sphere centered on the point E=(x_0, x_1, x_2) and with a radius of r_3 can be defined on the Cartesian coordinate system as:

$$(x-x_0)^2+(y-y_0)^2+(z-z_0)^2=r_3^2 \tag{3.21}$$

To create the sphere that defines the surfaces it is needed to calculate its center and its radius. This process is similar to all the spheres so these values are going to be calculated for the surface x^+ and then generalized for the others. Firstly, it will be calculated the center of the shape. The sphere must be centered with its surface, so its center should be the point (x_0^+, c_y, c_z). Considering 3.22, the the equation that defines the sphere is:

$$(x-x_0^+)^2+(y-c_y)^2+(z-c_z)^2=r_3^2 \tag{3.22}$$

On the plane $x=c_x+\frac{h_3}{2}$ the cube and the sphere must form, on the YZ plane, a square circumscribed by a circle, which can be seen in figure 3.23. The square has a height of $h_3=70=2\cdot l_3$ mm and the circle a radius of r_4; this circle can be defined as:

$$(y-c_y)^2+(z-c_z)^2=r_4^2 \tag{3.23}$$

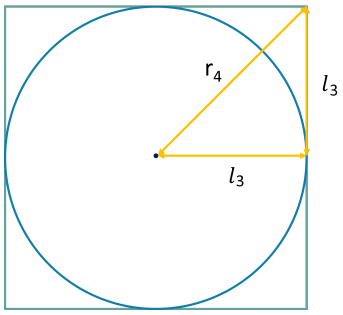


Figure 3.23: Representation of a square circumscribed by a circle (created by the author)

With the Pythagoras theorem, it is possible to obtain:

$$r_4^2 = l_3^2 + l_3^2 \quad \Leftrightarrow \quad r_4^2 = 2 \cdot l_3^2 \tag{3.24}$$

So, one of the conditions for the sphere is that for:

$$x = c_o + l_3 \quad \Rightarrow \quad (y - c_y)^2 + (z - c_z)^2 = 2 \cdot l_3^2 \tag{3.25}$$

Now, applying this conditions 3.25 to the equation 3.22, it is obtained:

$$(c_x + l_3 - x_0^+)^2 + 2 \cdot l_3^2 = r_3^2 \tag{3.26}$$

By solving this last equation:

$$x_0^+ = c_x + l_3 \mp \sqrt{r_3^2 - 2 \cdot l_3^2} \tag{3.27}$$

This equation has two values for the answer that corresponds to the two spheres from figure 3.24. The sphere desired is the one with the highest value for x_0^+, so the value for this variable is

$$x_0^+ = c_x + l_3 + \sqrt{r_3^2 - 2 \cdot l_3^2} \tag{3.28}$$

This equation 3.28 can be generalized to use for the spheres that define the other positive surfaces. As for the negative surfaces, it is obtained for the surface x^-

$$x_0^- = c_x - l_3 \mp \sqrt{r_3^2 - 2 \cdot l_3^2} \tag{3.29}$$

In this case, the sphere desired is the one with the lowest value for x_0^-, so

$$x_0^- = c_x - l_3 - \sqrt{r_3^2 - 2 \cdot l_3^2} \tag{3.30}$$

Like before, this equation 3.30 can be generalized to use for the other negative surfaces. With the center of the spheres calculated, now its radius r_3 must be defined. This must be a high

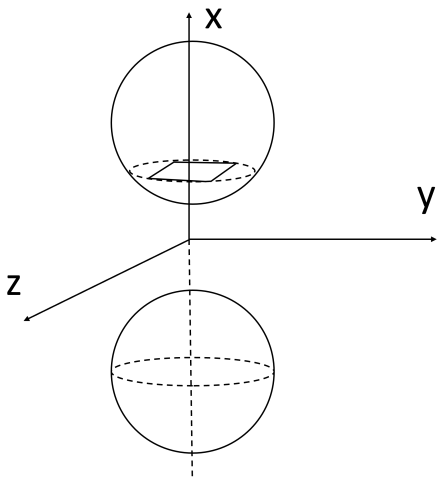


Figure 3.24: Representation of the two spheres that can be created with the value of x_0 with the equation 3.27 (created by the author)

value to reduce the intersection between volumes but must still guarantee a stable penetration of the object. After the object was created, some trials were made and it was defined as $r_3 = 300$ mm.

Now the surfaces of the cube are defined and this shape would be done, but when tested, it was noticed that the edges and corners were not rendering the right force and were not being able to be perceived, so it was decided to fillet these edges. On the code, this was done with quarters of cylinders to define the edges and a trimmed sphere to define the corners. In figure 3.25, there is a representation of these shapes. These shapes were created with the minimum dimension possible, in order to appear as a perpendicular edge. With trials, it was possible to arrive at the minimum dimension for the radius of both the sphere and cylinder to be 3 mm.

Now, the cube is completely defined. After, it was created the cylindrical hole and the cylindrical surface, that goes along the Y axis. It was designed for its radius to be $r_5 = 8$ mm and to be on the center of the x surfaces. The process for the rendering of these shapes is similar to the ones explained before. Like the cube, when going from the y surface to the cylindrical one, the edges were not giving a real perception, so they were curved with a radius $r_b = 3$ mm. The cube and the hole with the edges curved can be seen in figure 3.26.

The curved edge of the hole can be described as a a quarter of cylinder but where its center axis is a circle instead of a line. So, to define this circle, it was used the Cylindrical coordinate system (r, y, θ), where the center of the circle, for example for the up surface, is the point

$$\Big(c_x + (r_h + r_b) \cdot \cos\theta, \quad c_y + l - r_b, \quad c_z + (r_h + r_b) \cdot \sin\theta\Big)$$

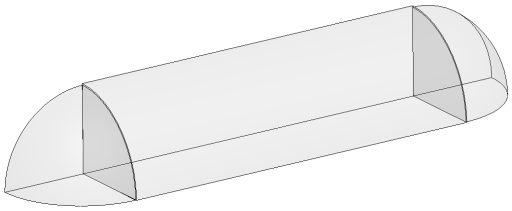


Figure 3.25: Representation of the corners and edges for the cube (objects created on Solidworks® by the author)

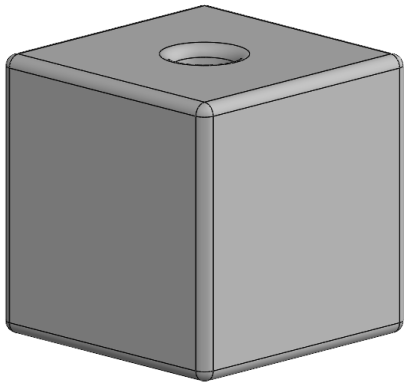


Figure 3.26: Representation of the cube with the cylindrical hole, with all the edges curved (objects created on Solidworks® by the author)

3.2.3 The user interface

The user interface is the intermediary between the user and the VE. It is where information and questions are displayed to the user and data is saved, like the answers he gives and the time taken to explore the objects. Is also where the user can give commands to the VE, like what shape should be displayed. Since the VE is being developed on a console application from Visual Studio, there is a need to create another application for the UI. The author of this dissertation is familiar with programming Forms applications on C# language, which are also developed on Visual Studio, and with connecting these types of applications with others through serial ports. So it was decided to create the UI on a Form application and try to establish communication with the VE through serial ports. Firstly, it will be present the design for the interface, and only after it is going to be explained some functionalities.

To initialize the test, there is a Home page with a brief description of the test and how it is going to be performed. It is also here where is asked the user consent to process his data. The design of this page can be seen in figure 3.27.

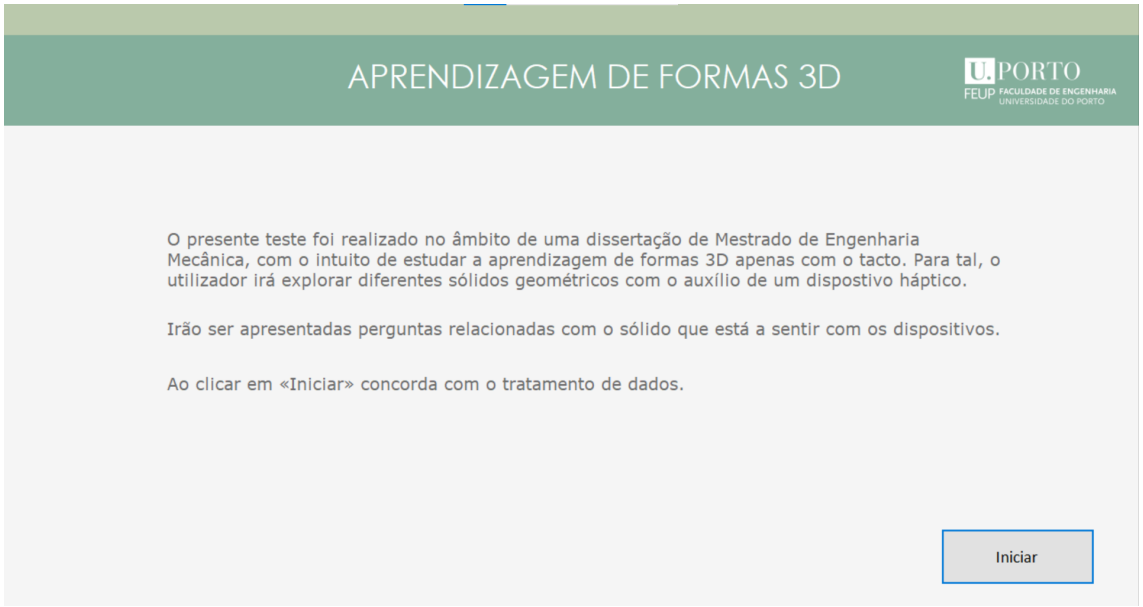


Figure 3.27: Home page design of the UI application

After, on the second page (figure 3.28) some personal data about the participant is requested:

- age;
- genre;
- degree of education;
- if has ever used a haptic device;
- if has any disability that affects the sense of touch;

- if has any disability that affects the vision.

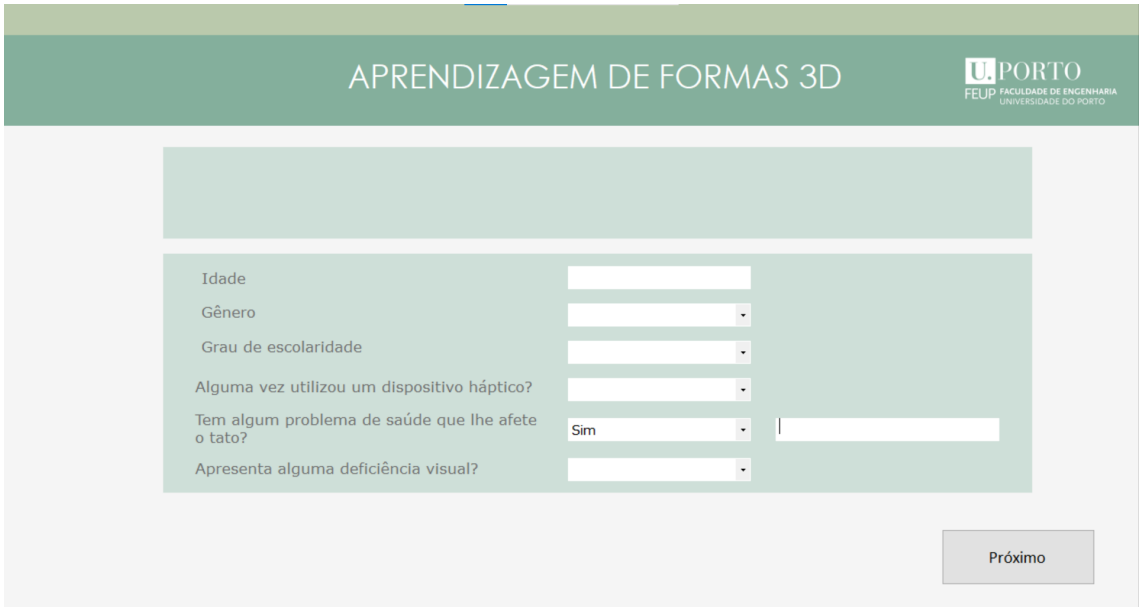


Figure 3.28: Design of the page where some personal data is asked from the participants

Lastly, before starting the questions, it is given some additional information to the participant on how to interact with the application and how to use the device to feel the objects. On this third page (figure 3.29), is also given a cone for the user to try the device. The instructions given are that before the user can start to explore an object, the pen of the device must be in the rest position, inside the inkwell. Secondly, the participant should click on the button "Cone". At this moment, the user can grab the pen and start to explore the object along the 3 axes. When ended with the exploration, the pen must be placed again in the rest position. Whenever on the same page there is more than one object for the user to feel, this process must be repeated between each object.

Finally, the test starts. On the first page with questions, it is asked to the user if the shapes of the three objects (sphere, cube, and cylinder) displayed are familiar, where the answers can be "No", "Yes", "Maybe" and "I don't know" (figure 3.30). Then, on the second page with questions it is asked for the user to say which of the objects (cube, cube with round edges, sphere, and cylinder) have edges, vertices, curvilinear surfaces, and flat surfaces (figure 3.31).

On the third page with questions, firstly it is asked the user if any of the four objects presented (two cylinders, a cube, and a cube with round edges) have the same shape. In case the user states none of the objects have the same shape (or that he does not know the answer), one more question appears, asking the name of the shapes of each object. If the participant has instead answered that there are similar shapes among the objects, besides the question to name the shapes, it is also displayed an extra question, asking which of the objects has the same shape (figure 3.32). On

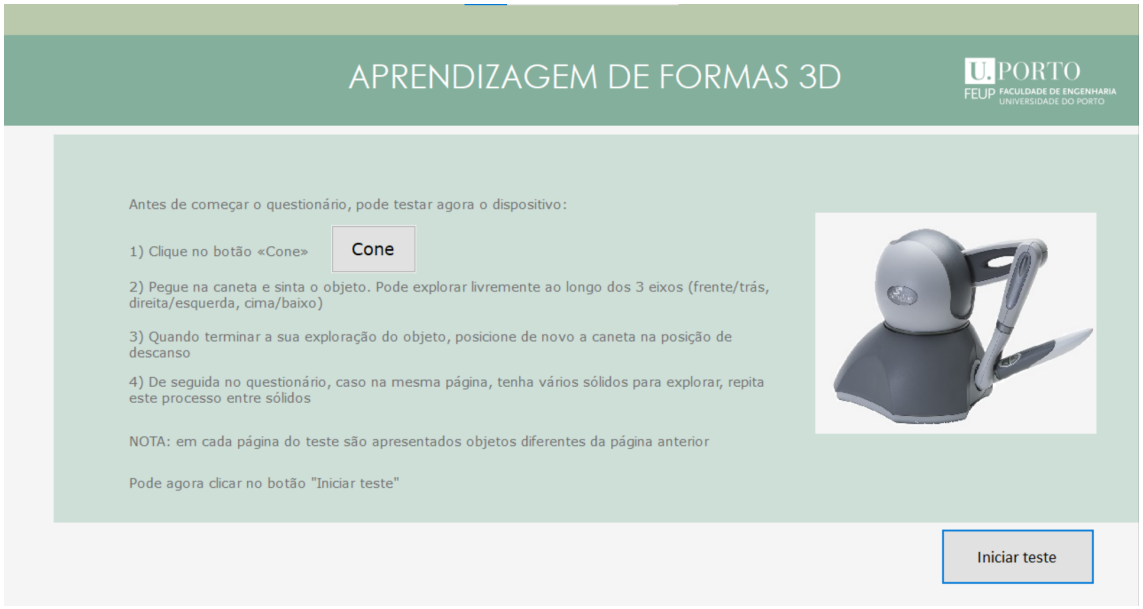


Figure 3.29: Design of the trial page where the user receives some information about the exploration of shapes and can try to explore a cone



Figure 3.30: Design of the page with the question representing the 5^{th} stage of the MHC for the haptic recognition of shapes



Figure 3.31: Design of the page with the question representing the 6^{th} stage of the MHC for the haptic recognition of shapes

the fourth page with questions, there are four complex objects for the user to precept and decomposed into simpler ones (cube, sphere, or cylinder). This decomposition is done with arithmetic: an object made out of two or more shapes is an addition; while, for example, an object with a hole is a subtraction. For a better understating of the concept from the user, an example is given with a figure: the object of the figure can be decomposed into a cylinder and a cube, so it is a $cylinder + cube$. In figure 3.33 there is a representation of this page. The objects presented can be seen in figure 3.3 at the begging of this chapter.

Finally, on the last page with questions (figure 3.34) it is used the same concept from the question before to decompose objects. It is asked to the user to find if any of the objects shown can be described as a $cube - cube - cube$. The objects for the user to explore on this page are represented in the figure 3.4 from the begging of this chapter. Then, when the user finishes the test, it is displayed a final page thanking the user for his participation.

3.2.3.1 The User Interface functionalities

With the design of the interface done, now it is going to expose the main concept used to create this application. When creating a Windows Forms Application, the main form is created. Then, to be able to have multiple pages, it was used a *panel* on the main form, which has a specific area and allows to load of other sub-forms on that space. So, the UI application is formed by the main form where multiple sub-forms are loaded, one for each page. The loading of the sub-forms is done by clicking on the button *next* from the main form. But it is on the sub-forms that the questions

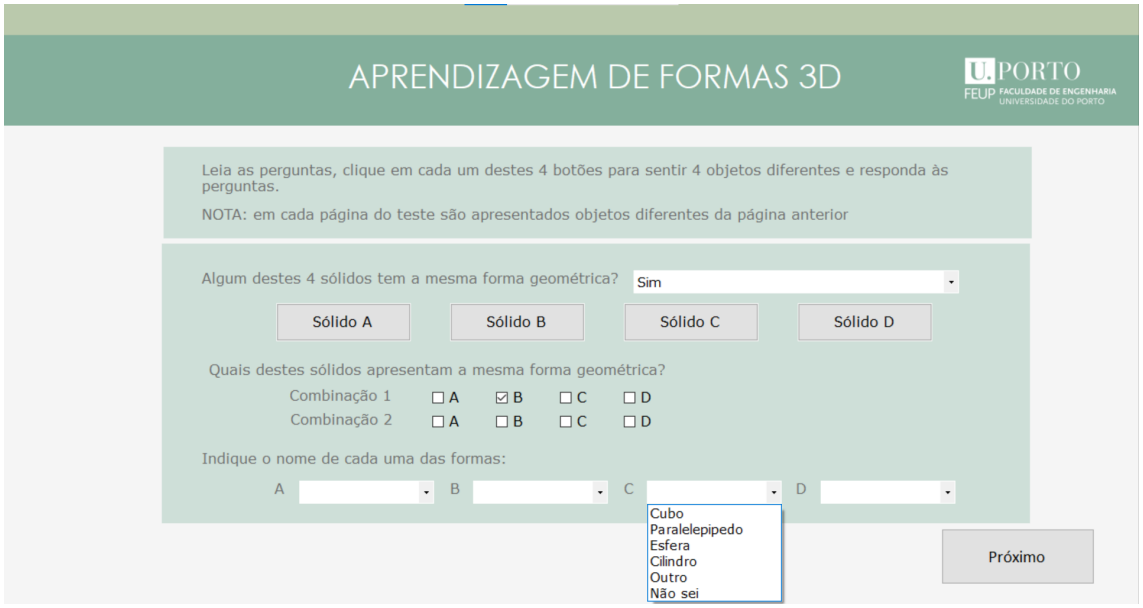


Figure 3.32: Design of the page with the question representing the 7^{th} stage of the MHC for the haptic recognition of shapes

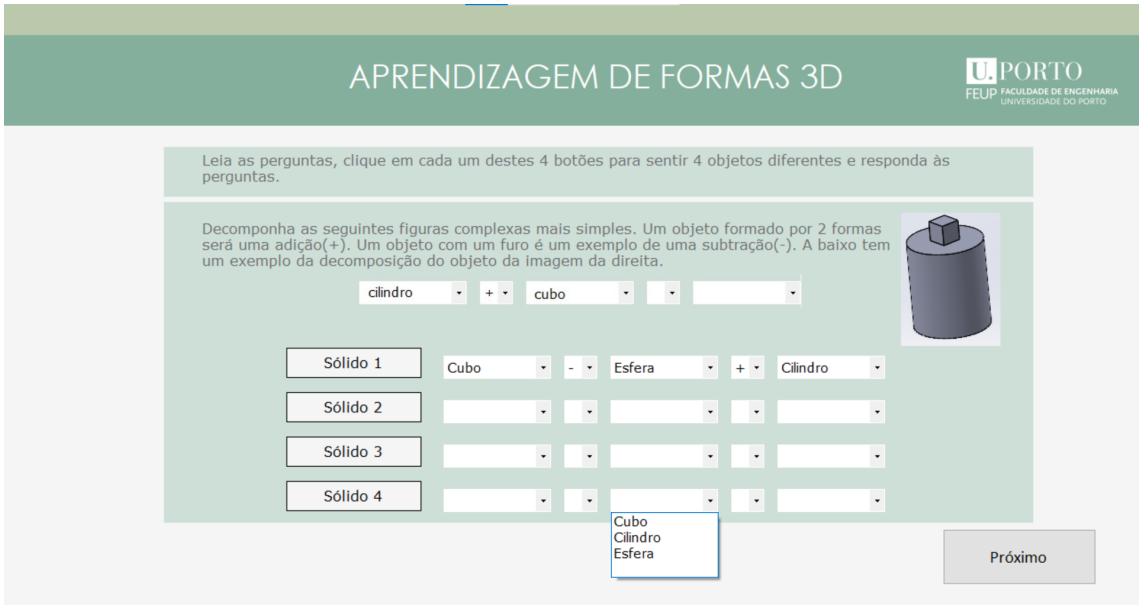


Figure 3.33: Design of the page with the question representing the 8^{th} stage of the MHC for the haptic recognition of shapes



Figure 3.34: Design of the page with the question representing the 9^{th} stage of the MHC for the haptic recognition of shapes

and answers are made and given, so it is there where the information is saved to an excel file. The moment when the information from a page should be saved to this file is before loading the next page. But the changing of pages is done on the main form, while the data saving on the sub-forms. This creates a problem with how the sub-form can understand when to save the data. To solve this, a communication using serial ports was established between the main form and the sub-forms.

Two virtual ports were linked through the application $SetUp$, the $COM10$ for the main form and the $COM11$ for the sub-forms. On Visual Studio, these ports were defined with a baud rate of 9600, timeouts for reading and writing messages of 100 ms, and 8 bytes for the data exchange. The serial port from the main form is opened once this form is loaded and closed when the application stops running. As for the serial port from each of the sub-forms, it is open once the sub-form is loaded and closed before the loading of the next sub-form, which is when the user clicks on the button from the main. So, once the person clicks on the button *next* on the main form, a message is sent to the sub-form, which saves the data, sends a message back to the main form, and closes its serial port. The loading of the next sub-form is then made once the message from the sub-form is received. A schema of these actions is presented on the figure 3.35 Some other functionalities from this program will be given further ahead when the communication between this program and the VE is explained.

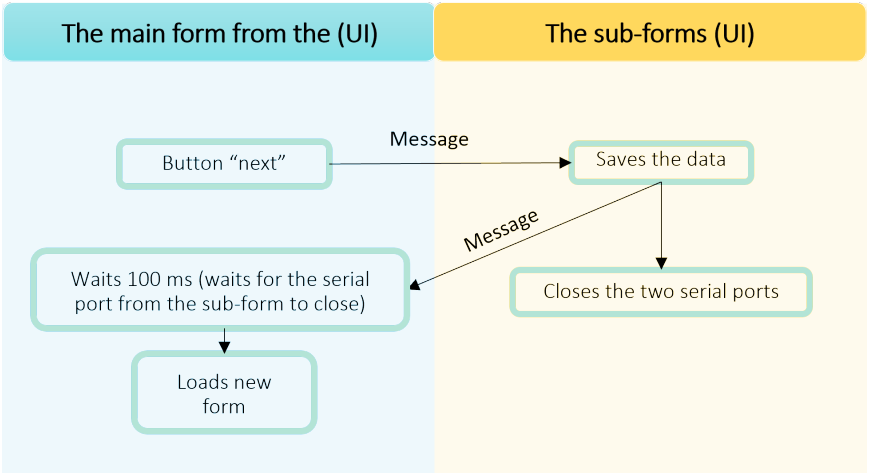


Figure 3.35: Schema of the communication between the main form and the sub-forms

3.2.4 Communication between the UI and VE programs

The communication between the programs is done by serial port. The virtual pair serial ports were linked through the application *SetUp*, and the ports used were the *COM*13 for the VE and the *COM*12 for the UI. Both have a baud rate of 9600, timeouts for reading and writing messages of 100 ms, and 8 bytes for the data exchange. The type of messages written and redden from the VE are *unsigned char* and from the UI *string*. On the UI, the creation of this port is very simple since it is a component of the Windows Forms Applications, like a *label* or a *textbox*. The VE communicates only with the sub-forms from the UI since it is there where the buttons where the user requests the shapes are. The port of the UI is opened once the sub-form is loaded and closed before the loading of another sub-form, which happens when a message from the main form is received, saying that the user has clicked on the button for the next page.

As for the VE application, the serial communications was based on an example found on the website [19]. Outside the *main* function, there are several functions regarding this communication:

- *open_serial_port* - where the configuration of the serial port is made, like timeouts and baud rate. It is also there where it is cleaned the previously read or written message;
- *write_port* - where a message is written into the serial port;
- *read_port* - where it is read the messages on the serial port;
- *get_target* - a function that checks if a new message was received. When this happens, the values are passed into *int* variables.

At the begging of the program, the values for the configuration of the port are set and it is opened. Then, it is continuously checked if a message is being received. When this happens, some command actions are triggered. As for the writing of messages, whenever this is intended, the *write_port* function is called. The port is closed before the end of the application.

Finally, all the communications between the components of the haptic interface are defined. A scheme of these connections can be seen in figure 3.36.

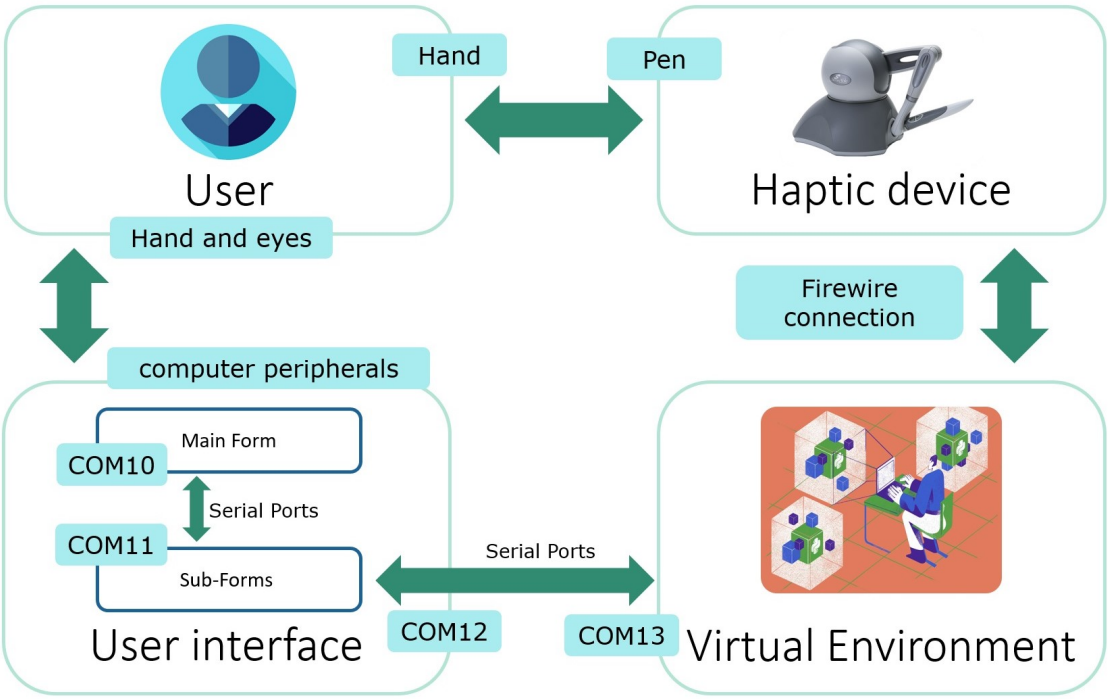


Figure 3.36: Haptic interface components and its connections

3.2.4.1 Information exchanged between the VE and UI

The data pretended to save from the test is:

- the user data;
- the answers to the questions
- time taken on each page of questions
- time taken to find the object
- time taken to explore the object
- the coordinates of the device during the exploration of the object

It is this information that dictates the messages exchanged between the programs. The coordinates of the device are obtained and saved directly on the VE application, while the others are saved on the UI application. The user data, the answers to the questions, and the time taken on each page with questions are obtained directly on the UI application, but to determine the time taken to find and explore the object, some information from the VE is needed. The UI is also responsible for the ending of the application of the VE, so, once the user clicks to exit the UI application, a message is sent to it. This results in the messages exchanged by the applications being:

- the UI requests the VE to render a specific object;
- the VE sends a message to the UI when the user:
  - starts to move the pen of the device;
  - finds the object;
  - inserts the pen in its resting position (ends the exploration).
- the UI sends a message that ends the VE application.

The actions that happen in the VE and UI from the request to the end of the rendering of an object will be now explained. Firstly, the user clicks on the button from the UI sub-form to request an object. This action sends a message to the VE and the object is then rendered. After, once the pen of the device is grabbed by the user, the VE opens the file where it will save the coordinates of the exploration of the object. It also sends a message to the UI for it to save the time and the information that the pen was grabbed. After, when the user finds the object, another message is sent to the UI for it to save the time. At this moment, the VE starts to record the coordinates of the exploration of the object. Finally, when the user ends the exploration (puts the pen inside the inkwell), another message is sent to the UI, and the VE stops the recording of the coordinates and the rendering of the object. The UI saves the time the user ended the exploration and also the information that the object is no longer being rendered. The information that the user has grabbed the pen and that he ended the exploration is used to implement a restriction in the UI. The end of the exploration of an object is defined by the user inserting the pen in its resting position. So, there was created the restriction that, once the user grabs the pen while an object is being rendered, he must then put the device on its rest position to be able to click and request another object. This condition was also created to avoid a sudden change of forces on the device and to enable its calibration between the rendering of new shapes. So, a user can request an *object A*, and then an *object B*, if the pen was not grabbed in between. Once the pen is grabbed while rendering an object, a message from the VE is sent to the UI, which will not allow the user to request another object until the pen is inside the inkwell. A representation of all the actions described above can be seen in figures 3.37 and 3.38.

The type of messages sent from the UI is *string* and is read by the VE as a *unsigned char*. To be able to use this information along the code, it is passed for a variable of type *int*, which requires decoding from normal characters to ASCII code. On the other hand, to send a message from the VE to the UI, it is directly sent as a *unsigned char* and read as a *string*, all with normal characters. On the table 3.1, there are the characters and codes passed from UI to the VE and the actions they represent. The messages sent from the VE to the UI, are listed in the table 3.2. The message sent from the UI requesting an object has two pieces of information on it: the first character represents the object to render and the second character represents the question where the request is being made. This distinction of questions is due to some objects appearing in more than one question. The information of which question the user is at will be used to save the coordinates of the user's exploration of that shape on a different file for each question.

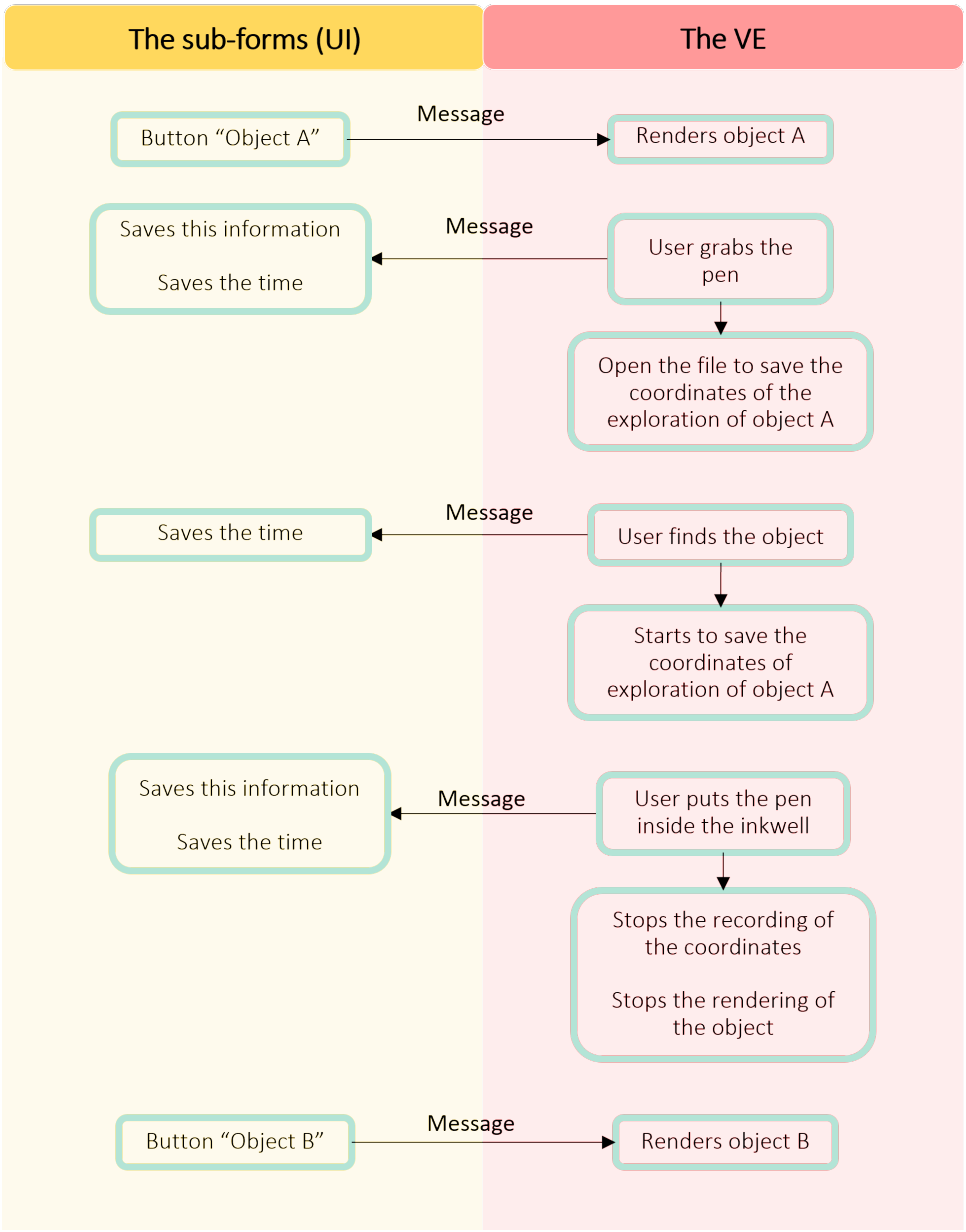


Figure 3.37: Actions that happen on the VE and UI since an object is requested until the end of its rendering

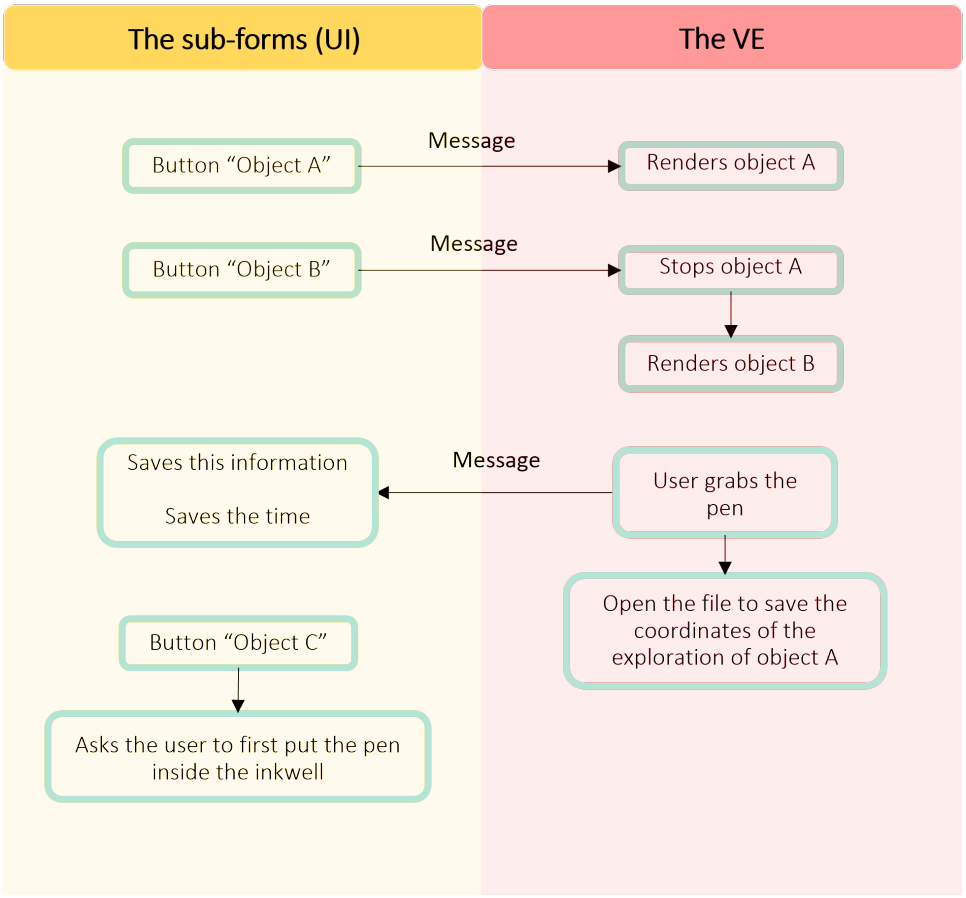


Figure 3.38: Actions that happen on the VE and UI when an object is requested while another is being rendered

Command	UI (normal characters)	VE (ASCII code)
Render sphere, exercise 5	10	[49, 48]
Render sphere, exercise 6	11	[49, 49]
Render Cube, exercise 5	20	[50, 48]
Render cube, exercise 6	21	[50, 49]
Render cube, exercise 7	22	[50, 50]
Render cylinder, exercise 5	30	[51, 48]
Render cylinder, exercise 6	31	[51, 49]
Render cylinder, exercise 7	32	[51, 50]
Render rounded cube, exercise 6	40	[52, 48]
Render rounded cube, exercise 7	41	[52, 49]
Render smaller cylinder	50	[53, 48]
Render cuboid	51	[53, 49]
Render *cube+cube+cube*	52	[53, 50]
Render cone	53	[53, 51]
Render *cube + cylinder*	54	[53, 52]
Render *cube - cylinder*	55	[53, 53]
Render *cube - cuboid*	56	[53, 54]
Render *cube - cube*	57	[53, 55]
Render *cube-cube-cube* (stair format)	58	[53, 56]
Render *cube-cube-cube* (cube with holes)	59	[53, 57]
Close the application	Z	[90, . . .]

Table 3.1: Table with the messages sent from the UI to the VE, the commands they represent, and their ASCII code

Table 3.2: Table with the messages sent from the VE to the UI, with the actions of the user they represent

Action of the user	Message sent from the VE to the UI
User grabs the pen	100
User finds the object	111
User ends the exploration	110

Chapter 4

Data Analyses

This chapter presents the results obtained from the experiment with the haptic device. The data was analyzed using descriptive statistical analysis. Firstly, there is a summary of the participant's characteristics. Then, it is analyzed the times and answers for the shapes and the questions.

4.1 Some considerations

Before proceeding to the evaluation of the data, it is important to analyze one aspect. When the application started to be tested with users, on the 8^{th} stage's question there was no example of how the users should answer. This was added after some of the first 7 participants stated that they were finding it hard to understand this question. So, it is first important to understand if the analysis of the test can be done considering these participants. In the table 4.1 there is the number of correct answers given to each stage's question by these users. The values show that users E, F, and G performed badly on all the questions from the test, so, the fact that on the 8^{th} stage's question there was not an example of how to answer it, does not seem to be the reason for their bad responses there. Of the other four users, two of them decomposed correctly more than two objects and another one was able to understand one of the shapes. So, these values do not indicate that these users did not have better results on this question due to not understanding the question.

Although some participants stated that at first, they were not understanding what to do on the 8^{th} stage's question, it appears that they were able to overcome this obstacle. Looking at the average of the times taken on this question and each of its shapes, for these 7 users and the rest of the participants, they are similar (table 4.2). This indicates that, although some of the participants who performed the 8^{th} stage's question without the example stated that it was hard to understand what the question was asking, this factor did not seem to affect their results. It is possible that the users who stated this problem were the ones who performed badly on the test, so, their difficulty in not understanding the 8^{th} stage's question may be from not being at that stage of complexity.

Table 4.1: Performance of the users who answered the 8^{th} stage's question an example

User	Number of correct answers on each stage				
	5^{th}	6^{th}	7^{th}	8^{th}	9^{th}
A	3	2	3	1	1
B	3	2	3	0	1
C	3	2	3	2	1
D	2	2	3	3	0
E	1	1	0	0	1
F	2	0	1	1	0
G	1	1	0	0	1

4.2 The participants

The study for this dissertation had 35 participants and their characteristics are presented on the tables 4.3, 4.4, 4.5 and 4.6. Most of these users are between 19 and 25 years old and are male. Their degree of education is almost equally distributed between the high school, bachelor, and master and none of the users tried a haptic device nor had any tactile or visual disability.

4.3 The method followed on the data analysis of the questions

According to the MHC, a person at an order *n* of hierarchical complexity should be able to perform at least one task from this *n* order and all the tasks from the stages before. It also defines substages or transition steps necessary for a person to change from one order to the next of hierarchical complexity. In all the questions of the test, the participants have a limited set of options as answers, which influences the way they answer and also the data analysis. With the questions having a restricted set of answers, the users can answer a question with a guess when in fact not know the response. Also, when not sure of the answer, the participant can choose one by excluding the ones that seem less likely to be correct. Duo to this, for the first four questions, where there are three or four objects, it will be considered:

- None or 1 correct answer (0 - 34%): inconclusive;
- 2 correct answers (35 - 67%): transition to the stage;
- 3 or 4 correct answers (> 68%): the user can perfume correctly the tasks for the stage. This will be what is considered a good performance on a stage.

Table 4.2: Average time on the 8^{th} stage's question and on its objects for the users that performed this stage with and without an example

		Users without example	Users with example
Average time	Question 8^{th}	00:07:16	00:07:01
	Cuboid	00:01:25	00:01:53
	Cube+Cube+Cube	00:01:36	00:01:33
	Cube+Cylinder	00:01:06	00:00:58
	Cube-Cylinder	00:01:12	00:01:12

Table 4.3: Distribution of the genre of the participants

Genre	Distribution	
Male	27	77%
Female	8	23%

Table 4.4: Answers given from the users to some personal questions at the beginning of the test

Question	Participants who answered "Yes"	Participants who answered "No"
Any visual disability?	0	35
Any tactile disability?	0	35
Ever used a haptic device?	0	35

Table 4.5: Distribution of the age of the participants

Age	Number of participants	Distribution
17-18	2	6 %
19-25	28	80%
27-29	4	11%
57	1	3%

Table 4.6: Distribution of the degree of education of the participants

Degree of Education	Distribution
High School	10
Bachelor	13
Master	12

While on the first, third, and fourth questions each object has a unique answer, the third question asks the user to identify 4 properties for each of the 4 objects. This question will then be evaluated by two methods separately:

- 1^{st} method - considering the whole answer by each object. Each answer represents it the user understood or not all the properties of the object

- 2^{nd} method - by properties, making the sum of all the properties the user understood.

Finally, on the last question, although presented four objects to the user, only two are the correct answer. The data relevant to analysis is if the user has identified none, one, or two of the correct objects as answers. So, for this stage it will be considered: if the user has performed well on the stage before, one correct answer will be considered as an indicator that he is in transition to the last stage; in other cases, one correct answer will be considered inconclusive, so, ignored; as for two correct answers, then the user is capable of correctly performing the tasks for this last stage.

4.4 The times and answers of the test

In table 4.7 there are presented the percentage of correct answers and the times of exploring and finding each object, along with the time taken on each question. From these values, is possible to understand that the shape later found by the users and where they spent more time was the cone, with 6 seconds for the average time to find it and 1 minute and 14 seconds for the average time of exploration. The average time on the questions increased along the test, but from the 8^{th} to the 9^{th} stage's questions, there was seen a decrease. The shapes with better results were the sphere and the cube. From the objects that appear on the first three questions, it is possible to see that the rounded cube is the one with the worst results. The analysis of these questions will be done with and without considering this object.

To understand if there is a hierarchical complexity along the test, It will be analyzed the good performances along the questions. This analysis will be done for all the users and then for the users with continuously good performances. The values are presented in table 4.8 and 4.9, for the two different methods used on the second question, and also with considering and not the responses for the rounded cube. The difference between the users with good and continuously good performances is also presented in these tables.

Table 4.7: Times of exploring and of finding the objects and the percentage of correct answers, by object and stage; percentage of users with good performances on each of the stages and the average of time taken on each question

Question	Shape	Average time to find (s)	Average exploration time (s)	% of correct answers	Average time for questions (s)
Trial	Cone	00:06	01:14	-	-
5th stage	Sphere	00:01	00:34	89%	02:41
	Cube	00:01	00:44	91%	
	Cylinder	00:00	00:39	77%	
6th stage	Cube	00:01	00:48	91%	03:54
	Rounded cube	00:02	01:05	31%	
	Sphere	00:01	00:24	89%	
	Cylinder	00:00	00:42	51%	
7th stage	Small cylinder	00:01	00:51	80%	04:06
	Cylinder	00:00	00:35	83%	
	Cube	00:02	00:44	80%	
	Rounded cube	00:01	00:42	14%	
8th stage	Cuboid	00:00	01:47	31%	07:04
	Cube+cube+cube	00:01	01:34	20%	
	Cube+cylinder	00:01	00:59	71%	
	Cube-cylinder	00:01	01:12	49%	
9th stage	Cube-cuboid	00:01	00:58	-	03:51
	Cube-cube	00:01	00:50	-	
	Cube-cube-cube (stair)	00:01	00:48	-	
	Cube-cube-cube(holes)	00:01	00:37	-	

Table 4.8: Percentage of correct answers for all users and users with correct answers along these shapes, using the 1^{st} method on the second question

	With rounded cube			Without rounded cube		
Stage	% of users with good performance	% of users with continuasly good performance	Difference	% of users with good performance	% of users with continuasly good performance	Difference
5^{th}	71%	71%	0%	71%	71%	0%
6^{th}	60%	51%	9%	51%	49%	3%
7^{th}	71%	49%	23%	71%	46%	26%
8^{th}	29%	23%	6%	29%	23%	6%
9^{th}	3%	0%	3%	3%	0%	3%

Table 4.9: Percentage of correct answers for all users and users with correct answers along these shapes, using the 2^{nd} method on the second question

	With rounded cube			**Without rounded cube**		
Stage	**% of users with good performance**	**% of users with continuasly good performance**	**Difference**	**% of users with good performance**	**% of users with continuasly good performance**	**Difference**
5^{th}	71%	71%	0%	71%	71%	0%
6^{th}	86%	69%	17%	91%	71%	20%
7^{th}	71%	60%	11%	69%	57%	11%
8^{th}	29%	23%	6%	29%	23%	6%
9^{th}	3%	0%	3%	3%	0%	3%

Since the rounded cube and the cylinder showed results not so good as the cube and sphere, it will be analyzed the responses regarding these objects on the 6^{th} stage's questions. In the tables 4.10 and 4.11, there is shown the percentage of correct answers for each of the properties of these shapes. The results show that each property of the cylinder and the rounded cube was understood at least by 74% and 51% of the users, respectively. It was then compared the answers given for these objects with the properties of other familiar shapes, with the results being presented in the tables 4.12 and 4.13. It is possible to see that the cylinder was misunderstood with other shapes by 31% of the users. As for the rounded cube, this value is higher, with it being confused with other shapes by more than half of the users. When looking at the answers from the 7^{th} stage question, the values for the cylinder improve, but for the rounded cube are worst (table 4.7). On the 4.14 there are presented the responses from the 7^{th} stage's question for the rounded cube. It is observed that 60% of the users stated it has the shape of a cube or cuboid.

Table 4.10: Properties of the cylinder correctly answered

Properties of the cylinder	**Correct answers**
Edges	83%
No corners	74%
Curvilinear Surfaces	86%
Flat Surfaces	89%

Table 4.11: Properties of the rounded cube correctly answered

Properties of the rounded cube	**Correct answers**
No Edges	51%
No corners	74%
Curvilinear Surfaces	89%
Flat Surfaces	74%

Table 4.12: Answers from the 6^{th} stage's question for the cylinder

Answers	Distribution
Correct	51%
Cube	11%
Cone	14%
Sphere	6%
Others	17%

Table 4.13: Answers from the 6^{th} stage's question for the rounded cube

Answers	Distribution
Sphere	17%
Correct	31%
Cylinder	17%
Cube	11%
Cone	11%
Others	11%

Table 4.14: Answers from the 7^{th} stage's question for the rounded cube

Answers	Distribution
Cube	46%
Cuboid	14%
Sphere	14%
Correct	14%
Do not know	11%

To explore if the rounded cube had bad results because it is a complex shape, it was seen the results of the users for this shape, divided by their percentage of correct answers on the 8^{th} stage questions (table 4.15). Although the users who have more correct answers about this shape answered correctly to more than half of the questions from the 8^{th} stage, the values are still low.

Table 4.15: Number of correct answers for the rounded cube divided by the percentage of correct answers of the users on the 8^{th} stage

% of correct answers o the 8^{th} stage	Number of correct answers for the rounded cube
0%	1
25%	5
50%	7
75-100%	3

With the simple objects and the rounded cube explored, it will be analyzed the results for the 8^{th} and 9^{th} questions. From the table 4.7, it is possible to see that many users were not able to decompose the cuboid correctly. When looking at the answers given for this object, it is observed that many users confused it with a cube (table 4.16). But when looking at the answers of these sample of participants on the question before, 75% of them were able to distinguish them (table 4.17).

Table 4.16: Distribution of the answers given for the cuboid on the 8^{th} stage's question

Answer	**Distribution**
Cube + Cube	33%
Cube	33%
Cube - Cube	9%
Other	24%

Table 4.17: Distribution of answers given on the 7^{th} stage's question for the cube, from users that stated the cuboid is a cube

Answers	**Distribution**
Cube	75,00%
Cuboid	16,67%
Cylinder	8,33%

As for the 9^{th} stage, only 3% of the users were able to give the full correct answer, but still, more than half of the participants identified one of the correct objects (table 4.18). Looking at the responses given, it is possible to see that the cube with two cubic holes was understood by a big percentage of the users (table 4.19).

Table 4.18: Percentage of users who correctly answered 0, 1, or 2 objects on the 9^{th} stage's question

Number of correct answer	% users
0	43%
1	54%
2	3%

4.5 Other considerations

While rendering the shapes, it was seen some bad responses from the device to the algorithm. This was first experienced with the cube. All its surfaces are designed with similar conditions. Although this, the surface that is perpendicular to the Z axis and that is on the negative part of this axis showed some bad response to the rendering of forces (figure 4.1). When touching this surface, the sensation felt was that it has a big roughness, and if applying a big strength, the pen would vibrate. This was solved by lowering the value of the stiffness for this shape, but if the user applies too much strength to the pen, it is still possible to be felt some roughness. This perception of roughness is small and did not appear to represent an obstacle to the exploration of the object.

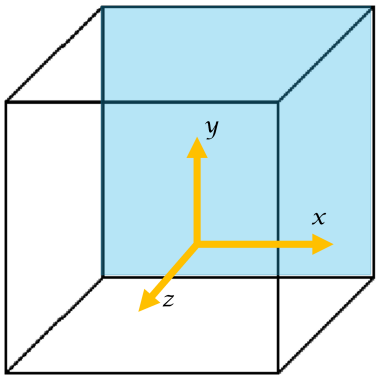


Figure 4.1: A representation of the face of the cube that was bad rendered (created by the author)

This problem with surfaces perpendicular to the Z axis and that are on the negative part of this axis appeared on another object, the cube with two cuboid holes. On this object, the problem of the pen vibrating appeared in its holes, on their bottom edges (figure 4.2). These edges are rounded, so it was tried to fix the problem by changing their stiffness, and the stiffness of the surfaces that join on these edges, but the vibration was still happening. The code was checked and compared with the other edges of the holes multiple times, so it does not seem to be a problem from the algorithm. The author considered this problem as not an obstacle to identifying the holes of the shape, since the user could still feel all its surfaces and other edges. During the test, this vibration was felt sometimes by the participants, and they were notified it was a mistake from the device.

Table 4.19: Answers of the participants on the 9^{th} stage's question

Answers	Participants
Cube-cuboid	1
Cube-cube	4
Cube-cube-cube (stair)	3
Cube with 2 cuboid holes	19
None	4
Do not know	6

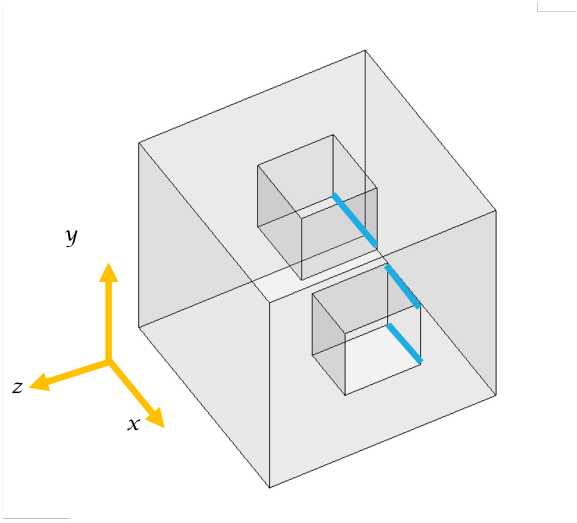


Figure 4.2: A representation of the edges of the cube with two cubic holes that were being bad rendered (created by the author)

Another object that the device had some problems with rendering was the cone. When exploring this shape along its height, it was possible to understand its narrowness. But when exploring along the XZ plane, the rounded shape was not very evident. In this type of exploration, its surface had a big roughness that sometimes felt like there may be an edge on the object. This problem did not disappear with the changing of values for its roughness. This bad response from the equipment seemed to be caused by a limitation of precision of the device. The cone only appears on the trial test and the users are informed about its shape. So, despite the perception of this object not being the best, it was still able to give a good perception of it, especially for a user that knows the shape that is touching. Also, the appearance of this shape had the main goal for the user to try the device and to understand how to perform the exploration of a virtual object with the haptic device.

The Omni device used is an older version the one produced now by 3D Systems®, with the main difference being its connection to the computer. The one used for this dissertation uses a firewire connection, while the most recent devices use USB ports. During the tests, it happen a

few little times that the device suddenly disconnected. This only requires disconnecting and connecting again the firewire cables and restarting the test. This may had happen from the equipment being an older version.

Chapter 5

Discussion

This chapter discusses the obtained results. Firstly, it will be seen whether the trial test was helpful for the users to understand how to explore a virtual object. Secondly, it will be studied the values obtained for the simple shapes and the rounded cube, in order to elaborate on why there were good performances for some of them, and bad for others. Then, the test reliability will be analyzed and the stage of each user in this domain will be identified. All of these analyzes will allow conclusions to be taken about whether the questions were well structured and if they were able to translate the tasks that represent the stages of the MHC for haptic recognition of shapes.

5.1 The trial test

Firstly, to understand if the trial page was important for the users to get familiar with the device, it is going to be compared to the time taken to find and explore the cone and the other simple objects, and the rounded cube. The average time taken to find the objects, for the cone was 6 seconds, while for the rest of the objects it dropped to only 1 second. As for the average exploration time of the cone and the simple shapes, they are represented in the graphic of figure 5.1. From observing this graphic, is possible to understand that the object with the highest value is the cone. These highest values for the cone, on both exploring and finding it, indicate that in fact, it was important to have a trial page for the user to get familiarized with the device. With this, the participants were able to try the device, understand how to make a good exploration of a virtual object, and then spend less time on the next objects.

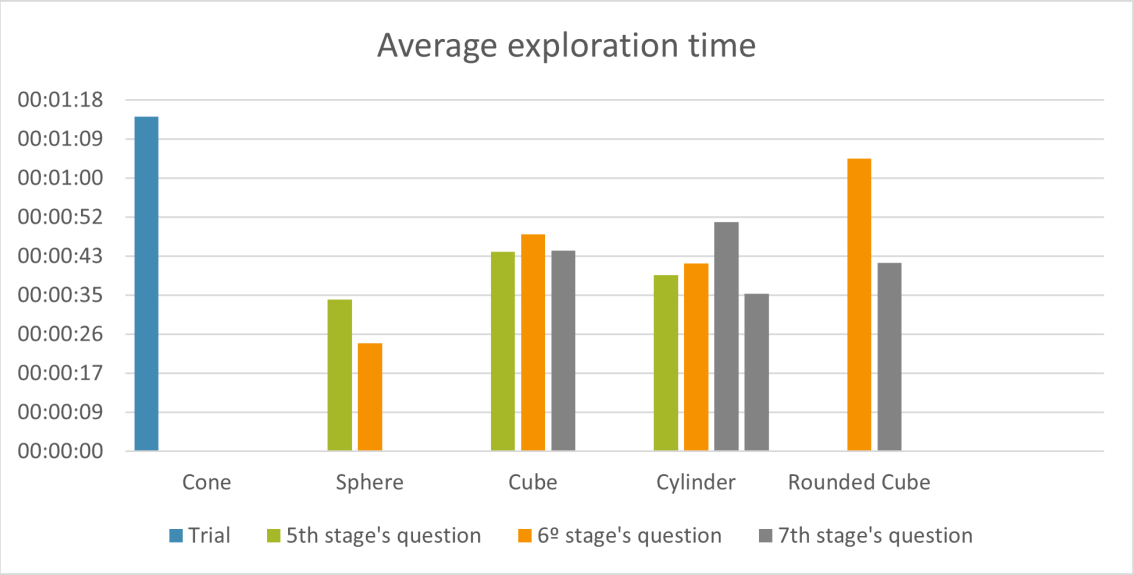


Figure 5.1: Average exploration time for the cone, the simple shapes, and the rounded cube

5.2 The performances on the simple shapes and rounded cube

Before evaluating the results of the questions, the simple shapes and the rounded cube will be analyzed, since they appear more than one time during the test. The sphere appears on the first two questions, the cube and the cylinder on the first three, and the rounded cube on the second and third questions. Figure 5.2 represents the average of correct answers for each of these shapes. The sphere and the cube have similar values, both near 90%, which indicates the users were able to understand these shapes. The cylinder has a value a little lower, but still, most of the users were able to identify it. As for the rounded cube, the values are very low, with only 22% of the time the users being able to answer correctly to this shape.

After this global evaluation of these four objects, it is going to be analyzed the results for each shape by question. But since the values obtained for the rounded cube are so different from the ones from the simple shapes, firstly it will be discussed the results of these three shapes and then separately the ones of the rounded cube.

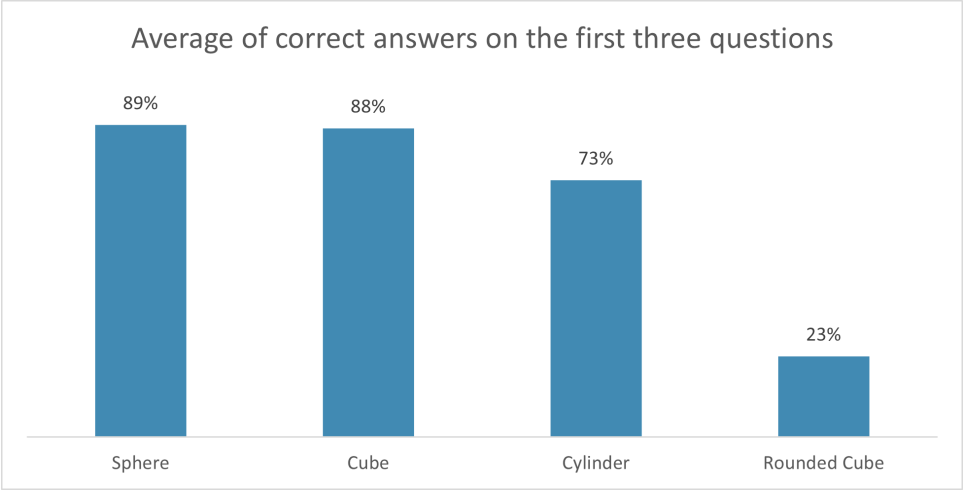


Figure 5.2: Correct answers on the first three questions for each shape

5.2.1 The simple shapes

According to the MHC, for a user to be able to correctly perform a task from a n stage of hierarchical complexity, he should also be able to perform the tasks from the stages below. So, along with the questions, the percentage of users answering correctly should maintain or decrease. In figure 5.3 there is a graphic that represents the average of users who correctly answered the first three questions for each of the simple shapes. It is possible to see that the sphere and cube show similar results, with around 90% of the users being able to identify these shapes on the first two questions. Then, from the second question to the third question, there is a small decrease for the cube, which only indicates that some users may not be able to perform the complexity tasks from the 7^{th} stage of hierarchical complexity.

The results for each of the simple shapes should be similar for the same stage's question, but some variability was seen for the cylinder, especially on the second question. On the first question, the percentage of users who answered correctly for this shape was smaller than the ones for the cube and sphere. Then, on the second question, this difference of values is even more evident, with only 51% of the users being able to correctly describe all the properties of the cylinder. These results show that the users had more difficulty perceiving this shape than the other normal shapes.

From the first to the second question there is a decrease in the correct answers given for the cylinder. The answers to the first question do not translate into any information that enables the deconstructing of the users thinking when answering it, but what may have happened is that the participants felt some properties of the shape, but did not have a clear understanding of it. Then, due to the recognition of familiar properties considered the shape as being also familiar. Also, some users may have confused this object with another familiar shape.

There is then an increase in correct answers from the second to the third question. This shows that whit repeating the task of haptic exploration, the users were able to improve their ability to recognize shapes through touch and reached the same value of correct answers as the ones obtained for the cube. This indicates that the users who answered wrong for this shape on the second question and then correct on the third, are able to precept the properties of the cylinder, just did not make a good exploration of this shape on the second question. When comparing the answers given for the cylinder on the second question with properties of other familiar objects, the values show that more than half of users who responded wrong for this shape were misunderstanding it with other familiar shapes (figure 5.4). This shows that the users may have answered wrong for the cylinder due to a bad exploration: they had a perception of some properties of the cylinder but then, instead of continuing a detailed exploration of the rest of its characteristics, assumed it was another familiar shape, influencing them to assume they were feeling corners or even flat surfaces. Then, by repeating the task on the next question, were able to improve their haptic exploration objects, performing a better exploration that lead them to correctly understand this shape.

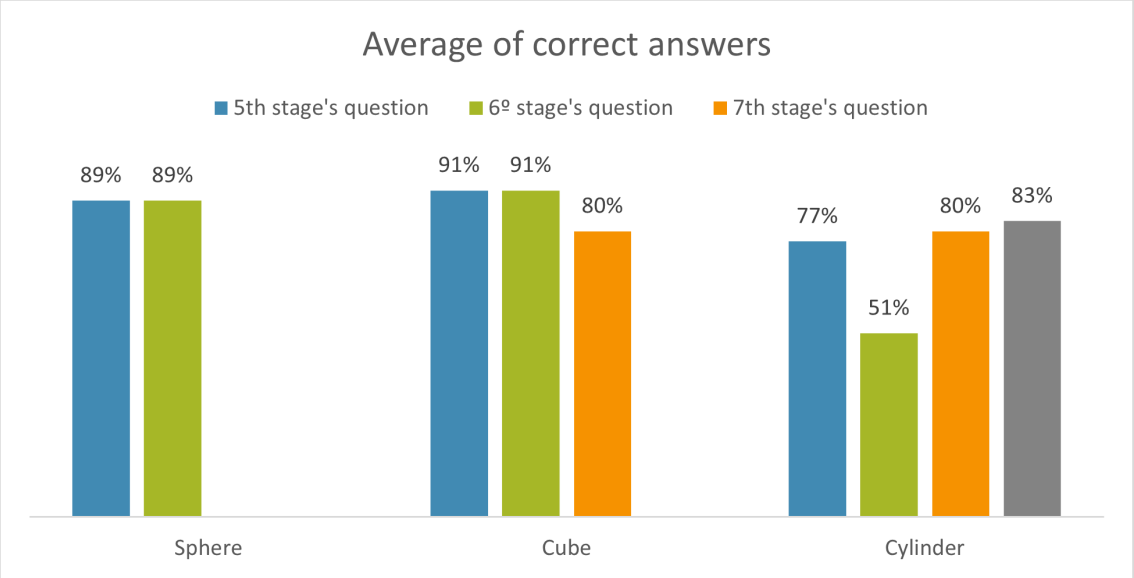


Figure 5.3: Correct answers for each shape on the first three questions

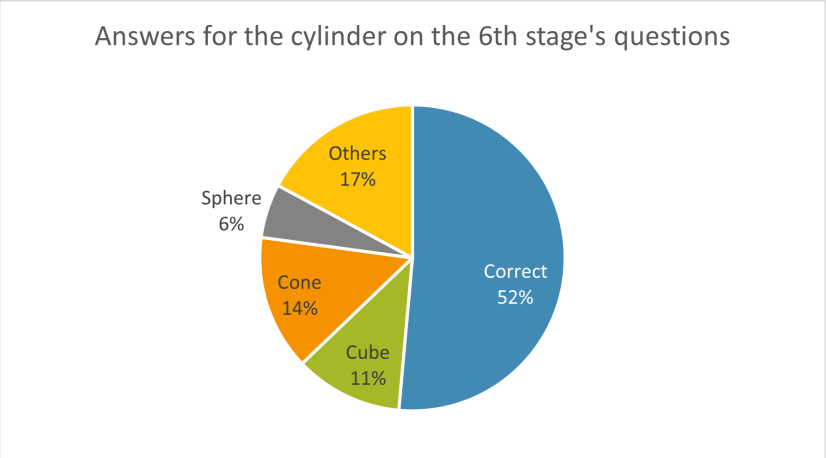


Figure 5.4: Answers for the cylinder on the 6^{th} stage's question

The values for the correct answer of each of the properties of the cylinder are represented in figure 5.5. Each property was perceived at least by 74% users, which shows that the rendering of this object was able to give a good perception of the real shape of a cylinder. This also goes accordingly with the assumptions taken before, that the users understood some properties of the cylinder, but that made them think of another familiar shape, leading them to perform a bad exploration of the rest of the properties.

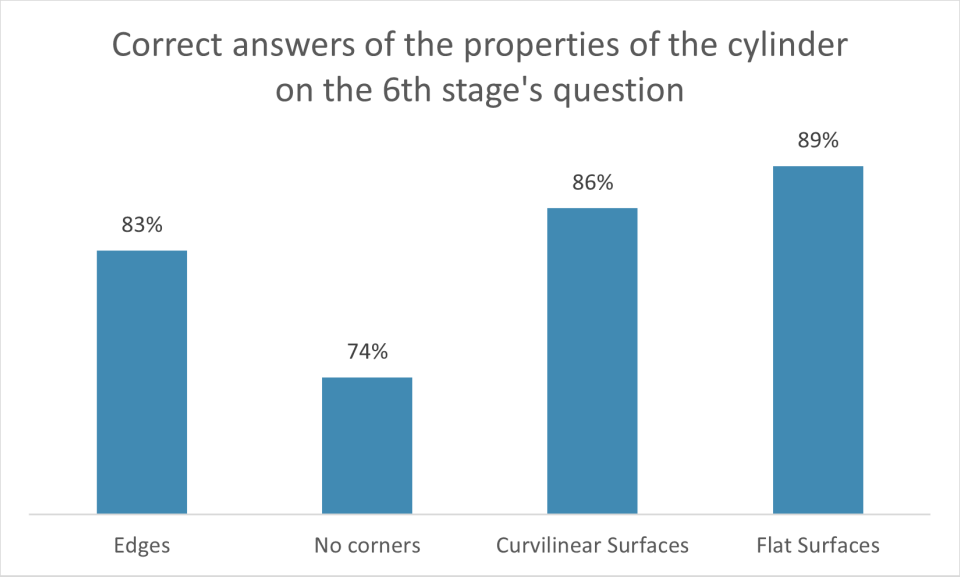


Figure 5.5: Correct answers of the properties of the cylinder on the 6^{th} stage's question

5.2.2 The rounded cube

The rounded cube is the only object made out of more than one simple shape that appears in the early questions. It was included in the second and third questions but the fact that it is not a simple shape was not expected to represent any obstacle to answering the questions of these stages. On the 6^{th} stage of hierarchical complexity, the user should be able to understand the properties of the objects, without creating a connection between them. The introduction of a rounded cube in this stage was done to try to lead the users to focus on the properties of the objects instead of matching the objects with a familiar shape and then answering the questions based on their knowledge of the shape. So, its appearance at this stage would only test if the user can feel corners, edges, and curvilinear and flat surfaces. The fact that this shape is complex should not represent any obstacle to identifying its properties.

Then, a user at the 7^{th} stage of hierarchical complexity should be able to identify families of shapes. So the goal with the rounded cube appearance on this level was only to understand if the users could distinguish from its properties that it does not belong to any of the cube, cuboid, sphere, and cylindric family of shapes. It was also chosen to insert this solid at this stage to have better data to analyze for it because otherwise, it would only appear in one question. Its appearance on two different questions allows the comparison between data and to take more consistent conclusions.

So, although its appearance on both the 6^{th} and 7^{th} stage question was not expected to cause bad results, the performance of the users for this shape was indeed bad. To understand if these bad results were due to this object representing a higher hierarchically stage, it is going to be analyzed the answers for this shape by groups of users. These groups will be formed based on the participant's average accuracy on the 8^{th} stage's question. On the figure 5.6 there is a graphic with the number of correct answers for the 6^{th} and 7^{th} stage's question. The results are better for the users who understand at least two of the complex objects, but are still low, which indicates that the bad answers for the rounded cube are not due to higher hierarchical complexity.

It will then be analyzed the answers from the 6^{th} stage question for this shape, in order to understand if the users were able to identify each property of this shape. In the graphic of the figure 5.7 it is possible to see the percentage of users that understood each of these properties. The results are good, with a big percentage of the users understanding correctly that the rounded cube has flat and curvilinear surfaces and no corners. Almost half of the sample was still able to understand that this shape does not have edges. This indicates that the rendering of this shape was done correctly, so this was not the cause for the overall bad results. Still from looking at the results for each property of the rounded cube, it is possible to see that 76% of the users felt a curvilinear surface and 49% an edge. The rounded cube is a simple cube where the edges were filleted to create a rounded surface. So, the statement that this shape has curvilinear surfaces and

edges is a contradiction. Nevertheless, 37% of the users answered that the rounded cube has edges and curvilinear surfaces. This value indicates that the users were performing a bad exploration of the curvilinear surface, with sometimes being able to follow the contours and feeling a rounded surface, but other times feeling an edge instead.

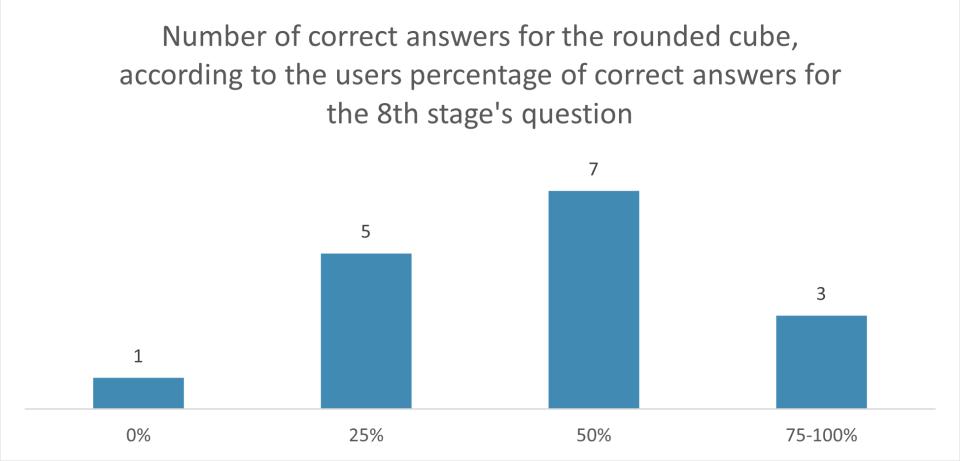


Figure 5.6: Number of correct answers for the rounded cube, distributed according to the user's percentage of correct answers for the 8^{th} stage's question

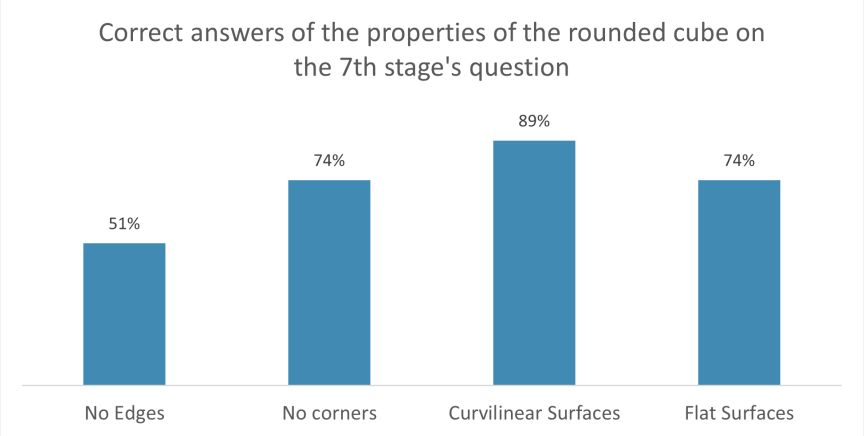


Figure 5.7: Correct answers of the properties of the rounded cube on the 6^{th} stage's question

Although the percentages of correct answers for each property are mainly good, they contrast with the whole answer, with only 31% of the users being able to identify correctly all the features of the object (figure 5.8). When comparing these responses given with the properties of other shapes, it appears that 51% of the participants were misunderstanding this object with other familiar shapes. Then, when looking at the responses to the third question, they validate this assumption. On this question, more than half of the participants stated that this shape is a cube or

a cuboid and 14% confused it with a sphere (figure 5.9). This results in 86% failing to accomplish the fact that the rounded cube fits neither on the cube, cuboid, sphere, nor cylinder family of shapes. This big percentage of users misunderstanding the rounded cube with other familiar shapes indicates that, either they were doing a poor exploration of this shape, or that, when unable to identify its shape, were trying to find similarities with familiar shapes, and then assuming these shapes for it. This fact also explains the decrease in performance for this object from the second to the third question, 39% to 11%. On the 7^{th} stage, before requesting the user to name the shapes, it is asked if any of the objects have the same shape. This may have led the users to force similarities between the rounded cube and other objects that appear on this level, more specifically, with the cube.

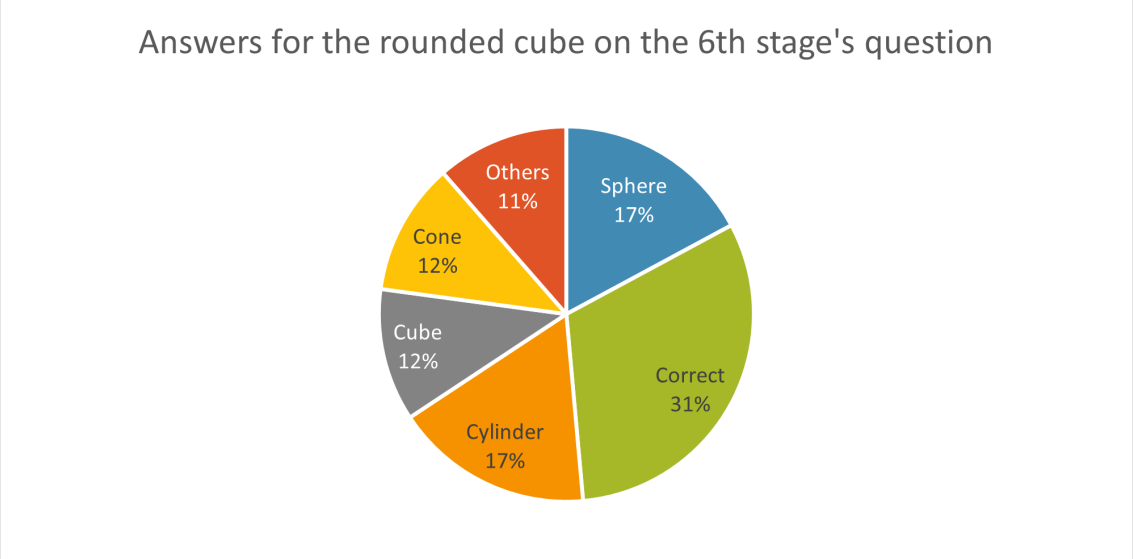


Figure 5.8: Answers for the rounded cube on the 6^{th} stage's question

Now, it is going to be analyzed the times for both questions on the rounded cube for different groups of participants: the users who answered correctly and the ones that mistook the rounded cube for other familiar shapes. The results are represented in the graphic of figure 5.10. It is possible to understand that the users who understood this shape took more time in their exploration. This difference in times is especially notable for the 7^{th} question, which indicates that both assumptions from before may be correct: the users were trying to find similarities between this object and a familiar shape, leading them to perform a bad exploration.

Another indicator of the bad exploration of the users can be again seen when comparing the times taken to explore the cube and the rounded cube on the 7^{th} question. Since this last shape is not a familiar shape to the user, it is expected that he needs more time to explore it than the other simple ones. So, the time taken to explore this shape and the cube are going to be analyzed for the third question for two groups of participants: the ones that answered correctly about both

objects and the ones that said they are both cubes. These values are represented on the graphic of the figures 5.11 and 5.12. It is possible to observe that the users who understood correctly both shapes took more time on the rounded cube than on the cube. On the other hand, the users that confused them took a smaller time exploring the rounded cube, in comparison to the cube. Again, this indicates that the users were not making a good exploration of this shape, and only trying to find similarities with familiar shapes. It is important to notice that the sample of users who, on the 7^{th} level question, correctly answered the shape of the cube and the rounded cube is small, so the assumptions being taken from this sample are not very reliable.

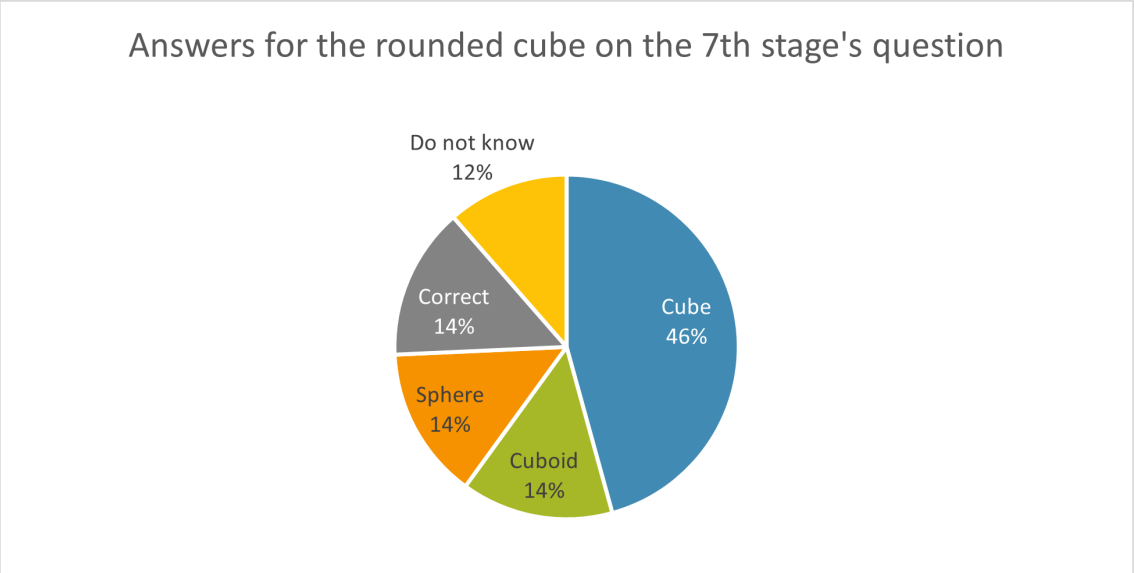


Figure 5.9: Answers for the rounded cube on the 7^{th} stage's question

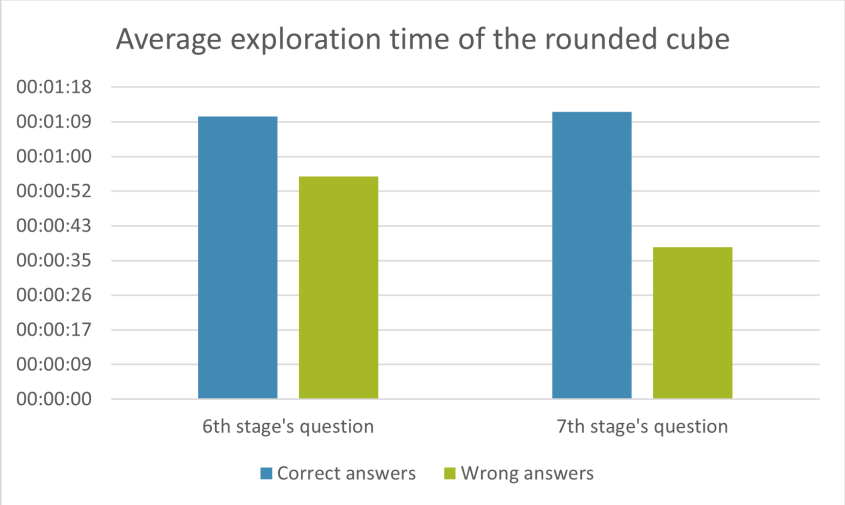


Figure 5.10: Average exploration time of the rounded cube for the users with correct and wrong answers separately

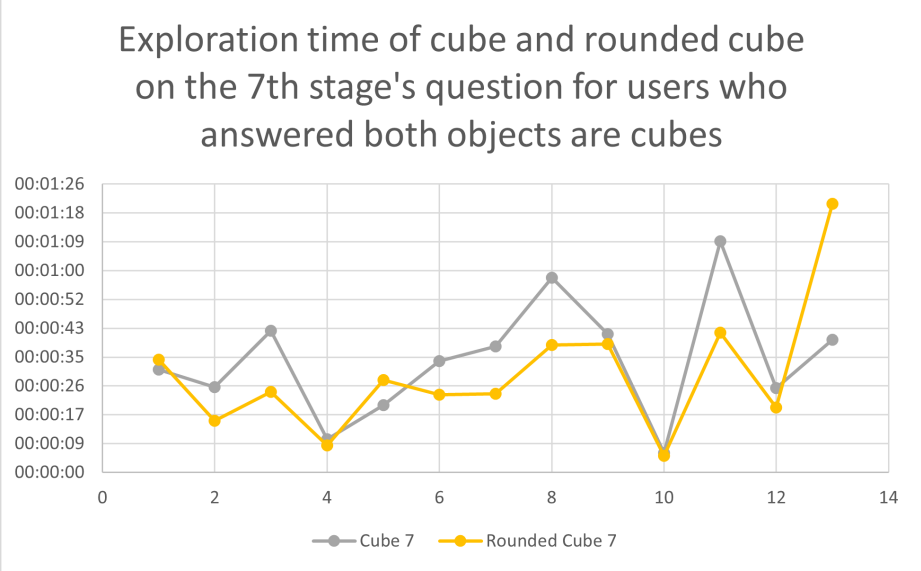


Figure 5.11: Exploration time of cube and rounded cube on the 7^{th} stage's question for users who answered both objects as cubes

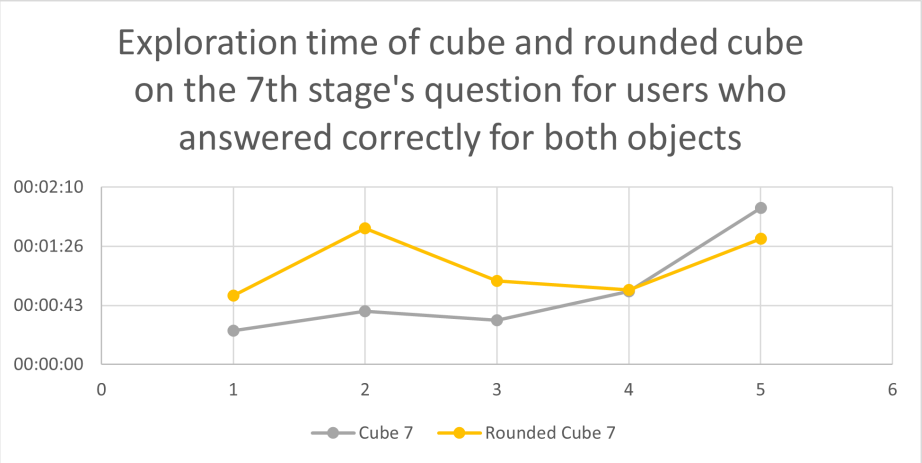


Figure 5.12: Exploration time of cube and rounded cube on the 7^{th} stage's question for users who answered correctly for both objects

In conclusion, all the analyses made for the rounded cube indicate that the rendering of this shape was done correctly, with the users being able to feel the properties of the virtual object. The bad performances appear to be caused by its unfamiliarity, resulting in the participants trying to match this object with a familiar shape. For this study, this shape appearing on the early questions did not show good results, and the performance of the users for it does not appear to translate their capability of recognizing shapes through haptic exploration. Due to this, this shape will not be considered in further analysis.

5.3 The questions

Before starting the analysis of the questions, it is important to indicate that, due to the bad results obtained for the rounded cube, this shape will not be considered, nor in the analysis of this section, nor for the distribution of the users along the stages. As said in the last chapter, the second question was analyzed through two different methods, the 1^{st} one evaluating the correct answers by object and the 2^{nd} by all the properties. In figure 5.13 there are the graphics that represent the percentage of users who performed well for each question, using both methods. The results obtained from each translate different information so both will be discussed.

When looking at the results from the 1^{st} method, it appears as if 29% of the users were not able to haptically explore objects. On the other hand, the results from the second method indicate that, although 29% of the users had difficulties with the first question, on the next one 91% were able to understand most of the properties of the objects. This percentage still shows that some users were not able to develop their skill of haptic exploring objects, with this value dropping to only 9% of the participants.

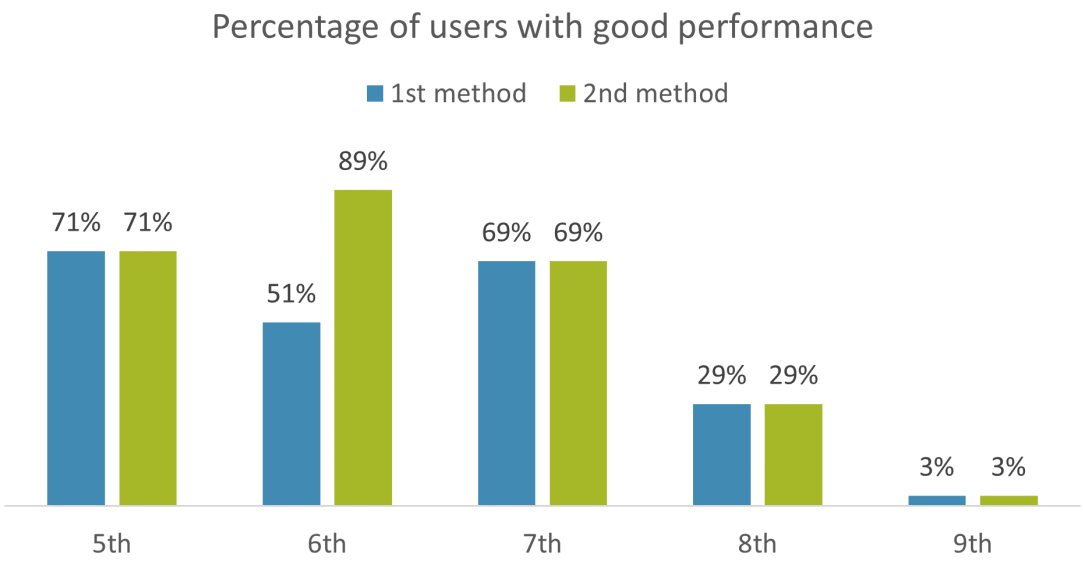


Figure 5.13: Percentage of users with good performance along the stages

From the 1^{st} method, It is possible to observe a decrease and then an increase of correct answers from the 5th to the 6th and 6th to 7^{th}, which are a result of the decrease and increase in the results of the cylinder. This decrease appears to be from the participants performing a bad exploration of this shape and confusing it with other objects. On the second question, only half of the users appeared to be mature enough to perform well on the next one, however, 69% succeed. This shows that by repeating the task of haptic exploring objects, some users were able to improve then improve their perception of the cylinder on the 3^{rd} question.

As for the results regarding the 2^{nd} method, the increase from the first to the second question may be, or from the users improving their skill of haptic exploring objects or from a bad understanding of the first question. In this question, it was only asked the users if the object seemed familiar, but it is possible that some tried to match the objects with their shapes. When looking at the results on the third question for the users that performed badly on the first (figure 5.14), it is possible to see that half of them did not show enough maturity in haptic exploration to identify the shapes of the objects, while the other half was able did. So, it is possible that in fact, some users were trying to answer the first question by performing a task hierarchically more complex than the one requested, resulting in bad results. On the other hand, half of the users only needed to get familiar with the task of haptically exploring objects. Then, the decrease from the 2^{nd} to the 3^{rd} question only indicates that some users are at the 6th stage of hierarchical complexity.

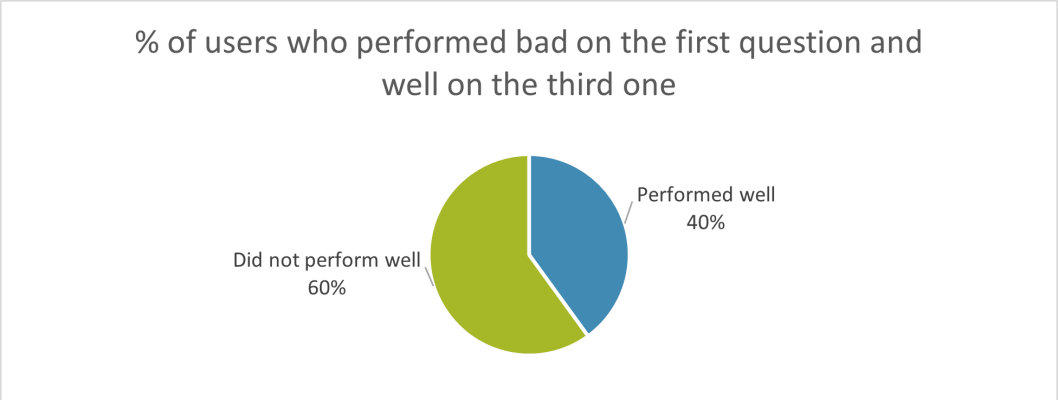


Figure 5.14: Percentage of users who performed badly on the first question and well on the third one

Looking at the overall of the first three questions, it is then possible to say that only a small number of users were not able to learn how to haptic explore objects. There is also seen an improvement from the first to the second question, which can be from this question leading the users to improve their exploration method. On the second question, only half of the users appeared to be mature enough to perform well on the next one, however, 69% did. This demonstrates an improvement in the participants in their haptic exploration.

The biggest decrease of values is from the 7^{th} to the 8^{th} stage's question, which indicates that the stage where more individuals are at is the 7^{th}. This decrease continues until the last question, which indicates that there was a higher hierarchical complexity along the questions of the test. In the 8^{th} stage, only 28% of the participants decomposed correctly at least three of the four objects (figure 5.15. Nevertheless, despite not being completely mature with decomposing complex figures, 26% of the users showed to be in transition to this stage of hierarchical complexity with correctly decomposing two objects.

In figure 5.16 there is a graphic with the percentage of correct answers for each shape of the 8^{th} stage's question. The objects with the worst results are the cuboid and $cube + cube + cube$. Most of these wrong answers happened because of a misunderstanding of dimensions, with the users describing these objects as cube and $cube + cube$, respectively. These results were not expected, especially for the cuboid, which is a familiar shape. Although on this question some users confused a cuboid with a cube, most of these participants, on the question before, identified correctly the cube as being a cube instead of a cuboid (figure 5.17). This confusion on the definition of cuboid and cube on the 8^{th} stage's question may had happen because the participants were so focused on finding an unfamiliar object, that did not pay attention to the dimensions of the object and only its other properties (corners, edges, and surfaces). This may also be the cause of the errors on the $cube + cube + cube$. Another possibility is that the users were doing a bad exploration of the cuboid, only exploring two dimensions, which can lead them to assume the object is a cube.

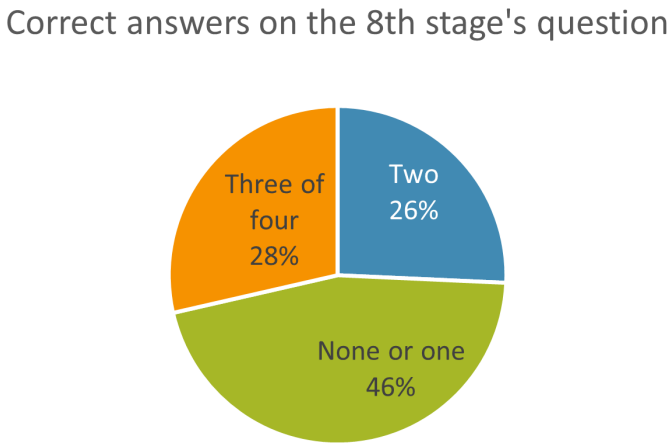


Figure 5.15: Distribution of answers of the 8^{th} stage's questions

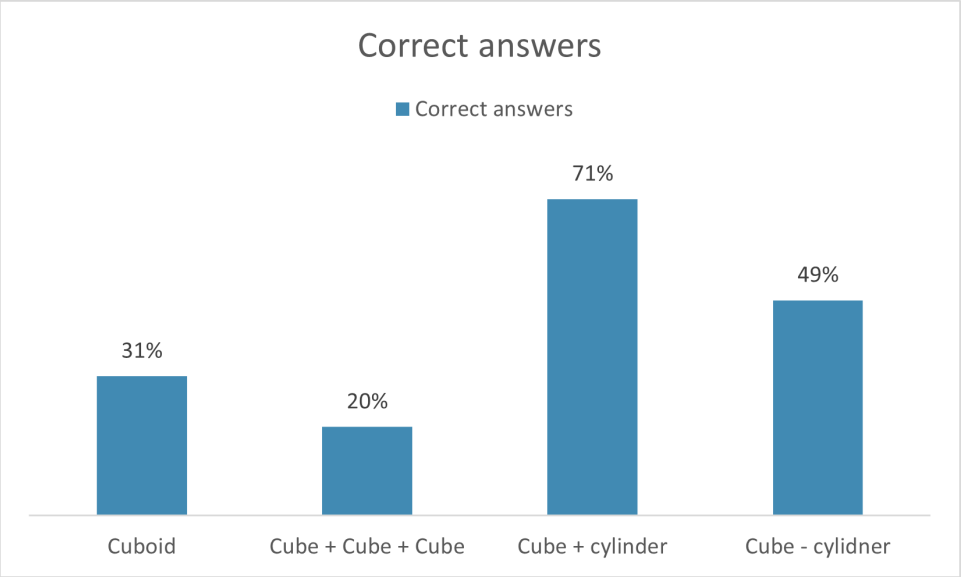


Figure 5.16: The percentage of correct answers given on each of the objects of the 8^{th} stage's questions

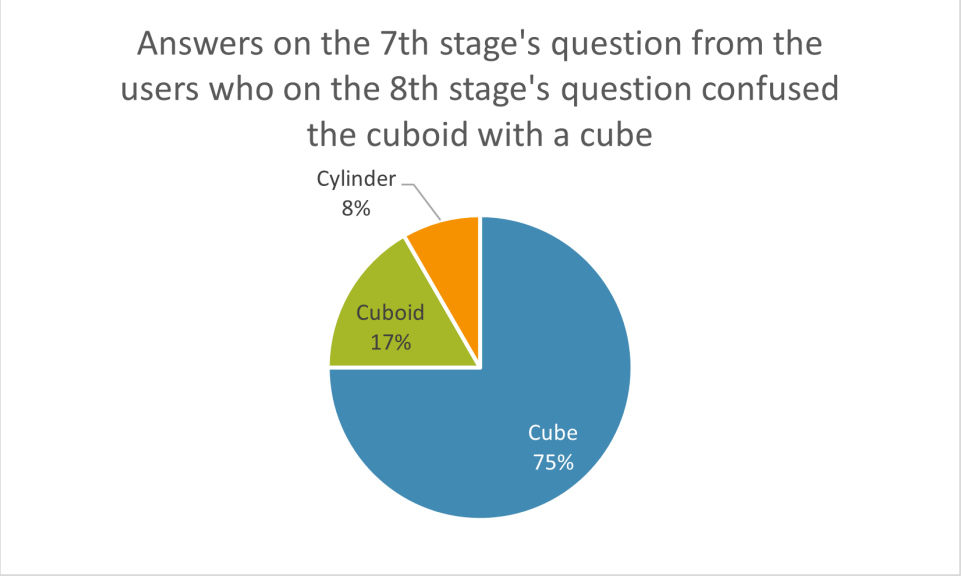


Figure 5.17: The percentage of answers given on the 7^{th} stage's questions for the users who on the 8^{th} stage confused the cuboid with a cube

On the 9^{th} stage's question, only 3% of the participant could identify the two correct objects, with the rest of the users divided between being able to identify one object or none (figure 5.18). Of the two correct objects, the one that most users identified was the cube with two cubic holes (figure 5.19), which indicates that the problem seen with its rendering was not an obstacle for the users to precept this shape. As for the other correct object, only 3 users understood it. On the cube with two cuboid holes, the user can feel the 6 square surfaces of the bigger cube, and 5 square surfaces of each of the cubic holes. As for the other object, the one with a stair shape, the user can only percept 2 square surfaces of the big cube, and then 2 of each of the holes. So, the three cubes are less obvious in this shape when compared to the first one, which justifies the very few users who realize this shape is correct. The data from this last question is not enough to establish solid conclusions, since, as assumed before, when a participant gives only one correct answer for a stage, it may not translate his knowledge. This correct answer can be a guess or chosen by exclusion. On the last question, a participant, although not realizing which of the objects are the correct ones, could just choose one as his answer just because the others seemed less likely to be described as $cube - cube - cube$. Although, what the results indicate is that this stage was more complex than the one before, with fewer participants being able to perform well.

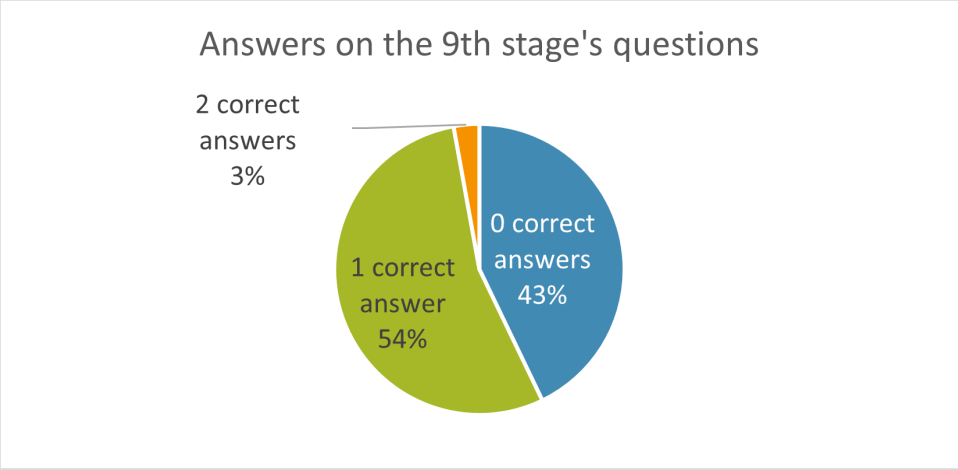


Figure 5.18: Percentage of the answers given on the 9^{th} stage

With the results for each question analyzed, now it will be made an overall analysis of the performance of the users along them. The method used to evaluate the second question will be the 2^{nd}, since it is the one that better represents this stage, according to the method designed. The graphic of the figure 5.20 represents the users with good and continuously good performances along the questions. These values are represented in two graphics. According to the MHC, for a user to be at an n stage of hierarchical complexity, he should be able to complete all the tasks from the stages before. So, this should result in the two graphics from the figure having the same values. This does not confirm, and the difference between them indicates the percentage of the results from this test that invalidates the model developed for the haptic recognition of 3D shapes.

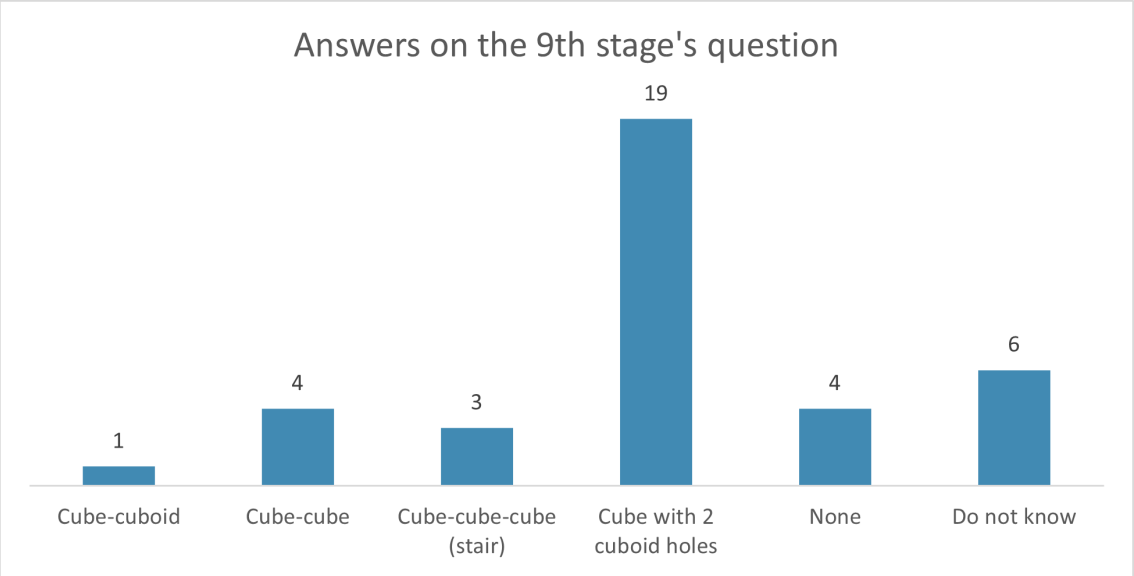


Figure 5.19: Representation of the number of users who responded to each of the shapes on the 9^{th} stage's question

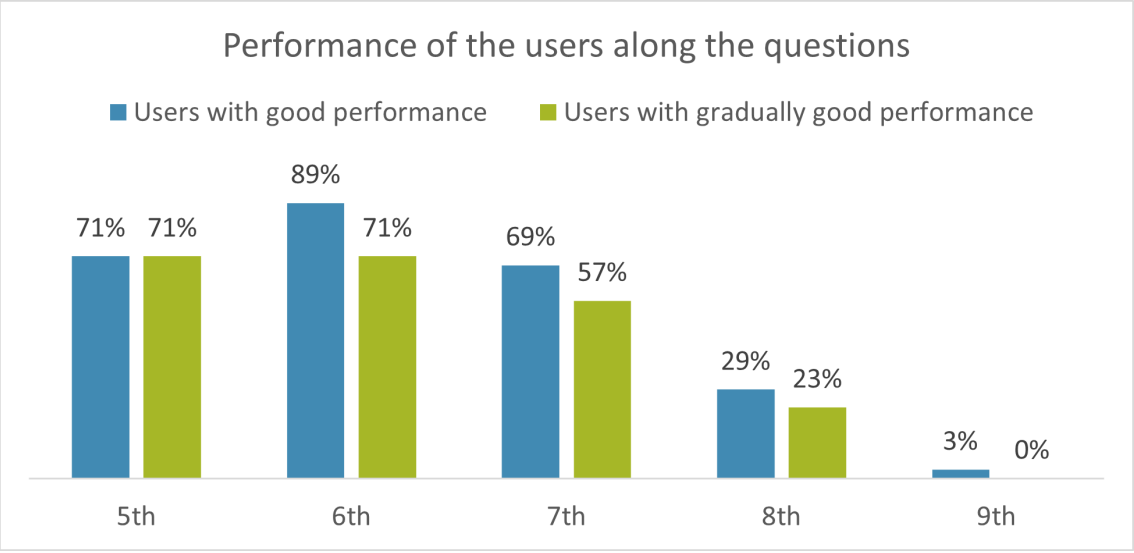


Figure 5.20: Percentage of users with good and continuously good performance along the stages

In figure 5.21 there is a graphic with the representation of these differences between the two methods. The results for the 1^{st} method most invalidate the model developed when compared with the 2^{nd} method. This goes accordingly to expected since the 2^{nd} method of analysis is the one that better translates the 6th stage. Also in this figure, there is a representation of these values for the 2^{nd} method when not considering the first question. From here, it is possible to understand that, if assuming that in the first question the users were still getting familiar with the device, the results get even closer to the conditions of the MHC. But overall, the percentages that invalidate the method on each question are small and indicate that in fact, the test goes according to the MHC. The user's performances decrease from the 2^{nd} to the last question, indicating also an increase in the hierarchical complexity along the questions.

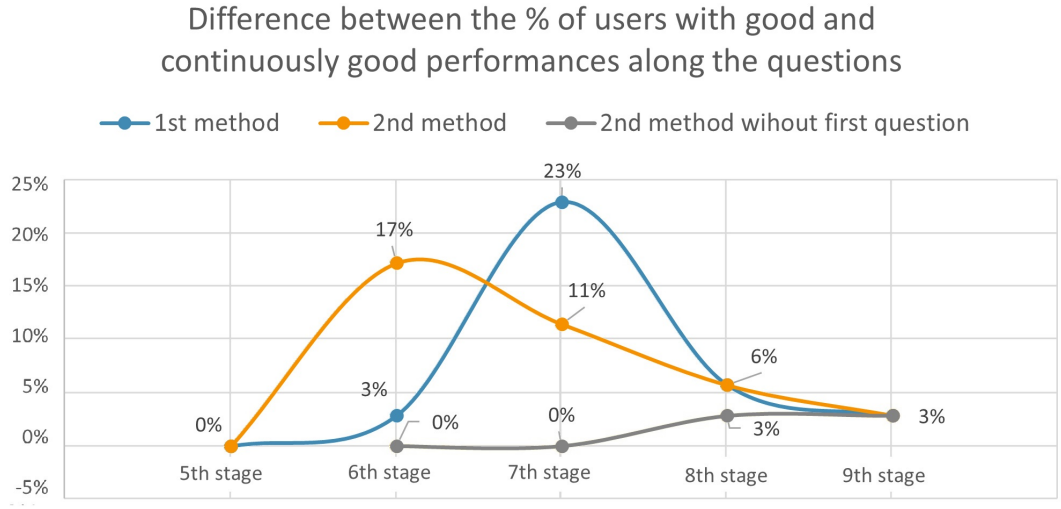


Figure 5.21: Graphic with the differences between the percentages of users with good and continuously good performances. Representations of the 1^{st} and 2^{nd} method and 2^{nd} method without considering the first question

5.4 Distribution of the users by the stages of hierarchical complexity

The distribution of the users was made according to the method explained in the last chapter, where, for a user to be at a *n* order of hierarchical complexity, he must have answered correctly to three or four of the questions that represent that stage. The second question was evaluated using the 2^{nd} method, with the sum of the properties correctly answered. In table 5.1 there is presented the distribution of the users by stage or if being in transition to the next one. When analyzing the results, it is possible to see that some users appear to be in the *n* stage but performed badly for a stage with lower hierarchical complexity. The number of users and the stages where that happens are presented in the table 5.2.

Table 5.1: Distribution of users by their stage or if being in transition to the next one

Stage	Users on the stage	Users in transition to next one	Total
5th	0%	11%	11%
6th	6%	11%	17%
7th	23%	17%	40%
8th	9%	20%	29%
9th	3%	-	2%

Table 5.2: Stages where some users performed badly, although appearing to be at a higher hierarchical complexity stage

Stage	Number of users
5^{th}	6
7^{th}	1
8^{th}	1

From all the users, only 8 appear to be at a *n* stage and were not able to perform a task from a lower stage. This bad performance happened four times on the first question, which indicates that it is due to the need of the user to get familiar with the haptic device. On the graphics from figure 5.22, there are represented two graphics that shows the percentage of users that are in each stage and transition to the next one. As expected, few users are at the first two orders of hierarchical complexity, with most of them being in transition to the next level. The stage with fewer participants is the last one, and the one where more users are at is the 7^{th}. Although 40% of the participants appear to be on the 7^{th} stage, almost half of them are transitioning to the next one. Also, of the users on the 8^{th} order of hierarchical complexity, more than half are in transition to the next stage. This indicates in fact that the sample of participants has a good capability of imaging complex objects. If these users had more practice with the task of exploring objects haptically, maybe they would be able to reach a higher-order stage.

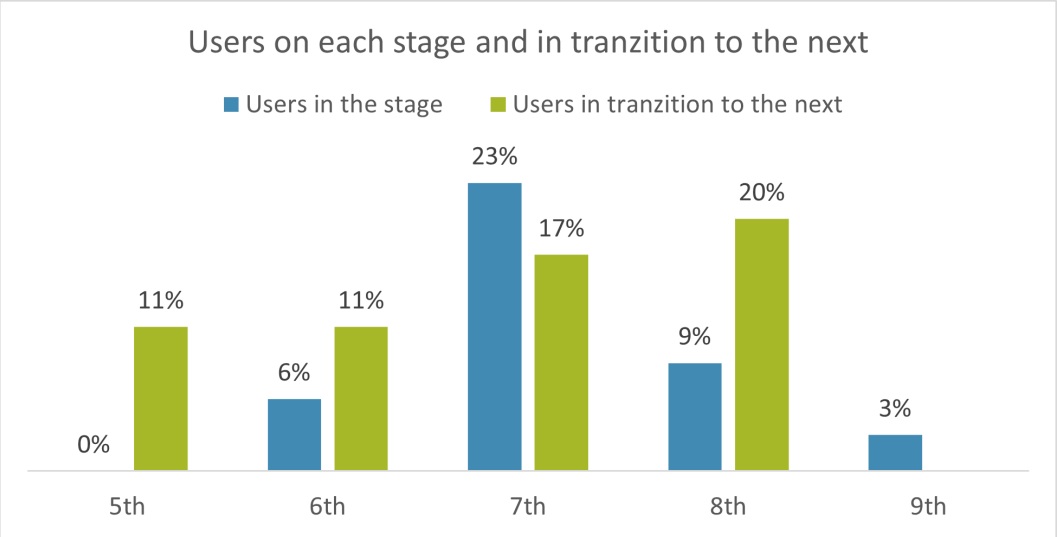


Figure 5.22: Users on each stage or in transition to the next one

5.5 Other considerations

Besides the sample discussed before, the test was also performed with two persons between 41 and 44 years old: a female user from the Faculty of Letters and a male with almost total vision disability. The results from the visually impaired participant indicated that this user may be in the 8^{th} stage of hierarchical complexity. The other participant showed difficulty in performing the test, without having results that can indicate which stage she is at. Although this sample is small, joining it with the other user from the early sample that is above 40 years old, there is possible to see a difficulty to adapt to this new haptic exploration of objects with the increase of age. Although expected a better performance from the visually impaired participant, his results go according to the average of the users from the sample before. This user had some difficulties with the cylinder shape and perceiving the dimensions of the objects.

These two tests were watched more closely by the author of this dissertation, and with this, it was possible to understand that the users were performing a bad exploration of the objects, mainly moving the pen of the device along one axe at a time, which would result on bad performances for curvilinear surfaces. The exploration was also being done with fast movements of the pen, making it harder to understand the edges of the objects. Also, when understanding some surfaces and properties of an object, the participants would stop their exploration there, without exploring all the surfaces of the object. This bad exploration can also explain the results from the sample of participants discussed before.

Chapter 6

Conclusion

The first conclusion from the test is that the application created was working correctly, and the objects rendered gave a good perception of their shape. The trial test showed the importance of firstly giving the users an object to explore before the test. The participants spent a great period exploring the first object afterward and were able to decrease their exploration time for all the simple shapes and the rounded cube. This shows the importance of having a trial object for the user to get familiar with the unfamiliar method of exploring an object using a haptic device. It also showed the ability of the users to fast improve on a task never performed. This improvement in their ability to explore objects using a haptic device occurs along the first three questions, where the same shapes appeared more than once.

Accordingly, to the test results, most users are on the 7^{th} stage of hierarchical complexity. These are capable to identify the shape of objects using a haptic device. Only a few reached the highest level, and another small part did not appear to be able to haptically percept the properties of the objects.

On the first question, some users were unable to perform well, however, along the test improved their results. A more extensive trial test could improve these first results. On the other hand, these bad performances can be from a possible misunderstanding of the first question, with the users trying to understand the object's shape instead of just finding if the current object is familiar to any they have experienced. This might lead some users to try to perform a hierarchical task higher than the one requested.

The rounded cube appearing on the early questions did not show good results, and the users' performance does not appear to translate their capability of recognizing shapes through haptic exploration. The bad results for this shape appeared to be caused by its unfamiliarity, resulting in the participants trying to match this object with a familiar shape.

Another conclusion is that the data from the last question is not enough to create a solid analysis. This is because of the small amount of data possible to take from only two answers.

Overall, the test showed an increase in hierarchical complexity, and most users follow the concept defined by the MHC, that for a person to perform a higher hierarchical task, must be able to perform all the lower ones. These values allow the conclusion that the model designed appears to translate the MHC applied to the haptic learning of 3D shapes.

6.1 Suggestions for future studies

It would be interesting to continue this work, by improving the test and trying it with a bigger sample of participants, with bigger differences in geometry knowledge. For instance, this test could be performed with primary school students, but only a version with the first three stages tested. The results from this type of experience could allow taking conclusions on how humans evolved their method of solving haptic tasks with the increase of geometry knowledge. A suggestion to improve the questions of the test performed would be:

- On the first question to ask instead if the object appears familiar to any object from the user's daily life;
- As a transition step, create a question before the 8th stage, where some familiar objects are given and asked to be described according to their names in our daily life, like a bottle of water, some stairs, a cup, etc. And only then, asking the user to decompose them into geometrical shapes;
- On the last question, give multiple objects, each one with a description. The user should then answer whether the object and description match.

It could be also studied if the rounded cube appearing only on the 8^{th} stage's question would show improvements in the perception of this shape.

Appendix A

Appendix

A.1 The code from the VE application

A.1.0.1 The calibration of the program

Regarding the calibration of the device, there are two occasions when this occurs, at the beginning of the application and whenever the pen is put inside the inkwell (rest position). This is done through synchronous scheduler callbacks (figure A.1). This type of callback is done with the function *hdScheduleSynchronous* that does not return any variable and has as input parameters: the name of the function to call, the information to pass to the function, and the scheduler priority. These callbacks functions occur one time, when called, and return an *unsigned int HD_CALLBACK_DONE* (figure A.1).

Inside the *while* loop, the function *GetCalibrationStatus* is continuously called in order to check if a calibration is needed. What this function does is schedule the callback *CalibrationStatusCallback* that communicates with the device to obtain the information if calibration is needed. Then, when the device requires calibration, the callback function *UpdateCalibrationCallback* is scheduled and the device calibrated. On the figure A.1 there is the code for this process.

Inside the *main* function and the *while* loop

```
if (GetCalibrationStatus() == HD_CALIBRATION_NEEDS_UPDATE )
{
    hdScheduleSynchronous(UpdateCalibrationCallback, &calibrationStyle,
        HD_DEFAULT_SCHEDULER_PRIORITY);
```

Outside the *main* function

```
HDCallbackCode HDCALLBACK UpdateCalibrationCallback(void *pUserData)
{
    HDenum *calibrationStyle = (HDenum *)pUserData;

    if (hdCheckCalibration() == HD_CALIBRATION_NEEDS_UPDATE)
    {
        hdUpdateCalibration(*calibrationStyle);
    }

    return HD_CALLBACK_DONE;
}

HDCallbackCode HDCALLBACK CalibrationStatusCallback(void *pUserData)
{
    HDenum *pStatus = (HDenum *)pUserData;

    hdBeginFrame(hdGetCurrentDevice());
    *pStatus = hdCheckCalibration();
    hdEndFrame(hdGetCurrentDevice());

    return HD_CALLBACK_DONE;
}

HDenum GetCalibrationStatus()
{
    HDenum status;
    hdScheduleSynchronous(CalibrationStatusCallback, &status,
        HD_DEFAULT_SCHEDULER_PRIORITY);
    return status;
}
```

Figure A.1: Functions used on the VE to calibrate the device

A.1.1 The rendering of the sphere

In figure A.2 there is the code created to render the sphere according to the explanations given in chapter 3.

```
HDCallbackCode HDCALLBACK FrictionlessSphereCallback2(void *data)
{
    HHD hHD = hdGetCurrentDevice();
    const double r = 50.0;
    const hduVector3Dd C = { 0,0,0 };
    const double stifness = .25;

    hdBeginFrame(hHD);

    hduVector3Dd P;
    hdGetDoublev(HD_CURRENT_POSITION, P);
    double d = (P - C).magnitude();

    if (d < r)
    {
        double penetration = r - d;
        hduVector3Dd N = (P - C) / d;
        hduVector3Dd F = stifness * penetration;
        hduVector3Dd Force = F * N;
        hdSetDoublev(HD_CURRENT_FORCE, Force);

    }

    hdEndFrame(hHD);
    return HD_CALLBACK_CONTINUE;

}
```

Figure A.2: Functions used on the VE to render the sphere

A.1.2 The serial communication

At the beginning of the *main* function of the VE, the serial port is initialized. After, inside the *while* loop that is continuously running until the program ends, it is checked if a message was received. A representation of the actions regarding the serial communication inside the *main* function is represented in A.3. Since the code from the program developed is long, this figure shows a similar application with only the functionalities for the serial communication. The functions used for the serial communication are presented in figure A.4.

```c
int main()
{
    buffer0 = 0;
    buffer1 = 0;

    //open serial port
    const char* device = "\\\\.\\COM13";
    uint32_t baud_rate = 9600;

    //Define a porta série e abere-a
    port = open_serial_port(device, baud_rate);


    while (true)
    {
        target = get_target();

        if (buffer0 != 90) { ... }

    }

    CloseHandle(port);

    return 0;
}
```

Figure A.3: The code of the VE, inside the *main* function, for the serial communication

```c
HANDLE open_serial_port(const char* device, uint32_t baud_rate)
{
    HANDLE port = CreateFileA(device, GENERIC_READ | GENERIC_WRITE, 0, NULL,
        OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, NULL);
    if (port == INVALID_HANDLE_VALUE) { ... }

    // Flush away any bytes previously read or written.
    BOOL success = FlushFileBuffers(port);
    if (!success) { ... }

    // Configure read and write operations to time out after 100 ms.
    COMMTIMEOUTS timeouts = { 0 };
    timeouts.ReadIntervalTimeout = 0;
    timeouts.ReadTotalTimeoutConstant = 100;
    timeouts.ReadTotalTimeoutMultiplier = 0;
    timeouts.WriteTotalTimeoutConstant = 100;
    timeouts.WriteTotalTimeoutMultiplier = 0;

    success = SetCommTimeouts(port, &timeouts);
    if (!success) { ... }

    // Set the baud rate and other options.
    DCB state = { 0 };
    state.DCBlength = sizeof(DCB);
    state.BaudRate = baud_rate;
    state.ByteSize = 8;
    state.Parity = NOPARITY;
    state.StopBits = ONESTOPBIT;
    success = SetCommState(port, &state);
    if (!success) { ... }

    return port;
}
```

```c
int write_port(HANDLE port, uint8_t* buffer, size_t size)
{
    DWORD written;
    BOOL success = WriteFile(port, buffer, size, &written, NULL);
    if (!success) { ... }
    if (written != size) { ... }
    return 0;
}

SSIZE_T read_port(HANDLE port, uint8_t* buffer, size_t size)
{
    DWORD received;
    BOOL success = ReadFile(port, buffer, size, &received, NULL);

    return received;
}


int get_target()
{
    //a mensagem vai ter 8 caracteres
    uint8_t buffer[8];
    DWORD result =  read_port(port, buffer, sizeof(buffer));

    //if any new message
    if (result != 0)
    {
        buffer0 = buffer[0];
        buffer1 = buffer[1];
    }
    return buffer[0];
}
```

Figure A.4: The functions of the VE, outside the *main* function, for the serial communication

A.1.3 Actions performed once an object is required

For better comprehension, first, some variables that will command actions are going to be presented, and after, it will be explained all the process from the start to the end of rendering an object. One variable was defined to translate when a new object is being rendered - *mexe*- and another to establish when the user finds the object for the first time - *cont*. To start, the VE receives a message that tells him which shape to render. This message is saved on another variable and cleaned, the variables *cont* and *mexe* are set to $false$ and the object rendered (figure A.5).

Inside the *main* function and the *while* loop

```
if (buffer0 == 49)
{
    //if another figure is being rendered already
    if (qual > 0){ ... }

    //saves the shape
    qual = 1;

    //cleans variables
    zerar();

    //renders the shape
    sp[qual - 1] = hdScheduleAsynchronous(
        FrictionlessSphereCallback2, 0, HD_DEFAULT_SCHEDULER_PRIORITY);
}
```

Outside the *main* function

```
void zerar()
{
    cont = false;
    mexe = false;
    cod = buffer1;
    buffer0 = 0;
    buffer1 = 0;
}
```

Figure A.5: Actions performed on the VE when a message is received from the UI, requiring a shape to be rendered

Then, once the person grabs the pen, a message is sent to the UI and the excel file that is going to save the coordinates of the device is open. After, once the person finds the object, meaning, once the force rendered is different from $(0,0,0)$, another message is sent to the UI and the variable *cont* is set as $true$. While this variable has this value, at each iteration of the program, the coordinates of the device are saved. All these commands can be seen in figure A.6.

Finally, once the user inserts the pen in its resting position, which is when the device's calibration status is changed to *HD_CALIBRATION_NEEDS _UPDATE*, the device is calibrated, the variable *cont* is set as $false$ and the function wr_end is called. This function sends a message to the UI, closes the excel file, and stops the rendering of the object. It is also called at the beginning of the rendering of an object, in case another object is already being rendered. This happens when the user clicks on a button on the VE, does not grab the pen to explore it, and then clicks on another button. All these commands can be seen in figure A.7.

Outside the *main* function and inside the Callback function that is always running

```
//If a new object is being rendered and the user grabs the pen
if (mexe == false && position[1] > -63.000)
{
    //envia uma mensagem ao c# (100)
    uint8_t col[3];
    col[0] = 49;
    col[1] = 48;
    col[2] = 48;
    write_port(port, col, sizeof(col));

    //if (shape = sphere and question = first)
    if (qual == 1 && cod == 48)
    {
        //opens the excel document
        fp = fopen("esfera5.csv", "a");
    }
    mexe = true;
```

Outside the *main* function and inside the Callback function that is always running

```
//if (new object and the force is not (0,0,0))
if (cont == false && (f[0] != 0 || f[1] != 0 || f[2] != 0))
{
    cont = true;

    //send message
    uint8_t coma[3];
    coma[0] = 49;
    coma[1] = 49;
    coma[2] = 49;
    write_port(port, coma, sizeof(coma));
}
```

Inside the *main* function and the *while* loop

```
//since the user finds the object, until stops the exploration
if (cont == true)
{
    //save the coordinates
    write_e();
}
```

Outside the *main* function

```
void write_e()
{
    hduVector3Dd position;
    hdScheduleSynchronous(DevicePositionCallback, position,
        HD_DEFAULT_SCHEDULER_PRIORITY);
    fprintf(fp, "%f3,%f3,%f3\n", position[0], position[1], position[2]);
}
```

Outside the *main* function

```
HDCallbackCode HDCALLBACK DevicePositionCallback(void *pUserData)
{
    HDdouble *pPosition = (HDdouble *)pUserData;

    hdBeginFrame(hdGetCurrentDevice());
    hdGetDoublev(HD_CURRENT_POSITION, pPosition);
    hdEndFrame(hdGetCurrentDevice());

    return HD_CALLBACK_DONE;
}
```

Figure A.6: Actions performed on the VE when the user grabs the pen of the device and when he first touches the object

Inside the *main* function and the *while* loop

```
if (GetCalibrationStatus() == HD_CALIBRATION_NEEDS_UPDATE)
{
    hdScheduleSynchronous(UpdateCalibrationCallback, &calibrationStyle,
        HD_DEFAULT_SCHEDULER_PRIORITY);

    //stops wrtiting the coordinates
    cont = false;

    wr_end();

}
```

Outside the *main* function

```
void wr_end()
{
    //sends a message
    uint8_t col[3];
    col[0] = 49;
    col[1] = 49;
    col[2] = 48;
    write_port(port, col, sizeof(col));

    if (mexe = true)
    {
        //closes the document
        fclose(fp);
    }

    //unschedule the object
    if (qual > 0)
    {
        hdUnschedule(sp[qual - 1]);

        qual = 0;
    }
}
```

Figure A.7: Functions used on the VE to calibrate the device

References

[1] Haptic device drivers for openhaptics.

[2] "phantom omni" - 6 dof master device.

[3] User interface.

[4] Avatar, social, profile, people, user, Aug 2016.

[5] P. Allen. Surface descriptions from vision and touch. In *Proceedings. 1984 IEEE International Conference on Robotics and Automation*, volume 1, pages 394–397, 1984.

[6] P. Allen. Surface descriptions from vision and touch. In *Proceedings. 1984 IEEE International Conference on Robotics and Automation*, volume 1, pages 394–397, 1984.

[7] Keith Andrews. Book reviews : Cole j 1991: Pride and a daily marathon. london: Duckworth. 194pp; figs. £14.99 (hb). isbn 0 7156 2385 0. *Clinical Rehabilitation*, 6(2):179–179, 1992.

[8] Hrvoje Benko, Christian Holz, Mike Sinclair, and Eyal Ofek. NormalTouch and texturetouch: High-fidelity 3D haptic shape rendering on handheld virtual reality controllers. In *UIST 2016 - Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 717–728. Association for Computing Machinery, Inc, oct 2016.

[9] Carlos Bermejo and Pan Hui. A Survey on Haptic Technologies for Mobile Augmented Reality. *ACM Computing Surveys*, 54(9), dec 2022.

[10] Marten Bjorkman, Yasemin Bekiroglu, Virgile Hogman, and Danica Kragic. Enhancing visual perception of shape through tactile glances. In *IEEE International Conference on Intelligent Robots and Systems*, pages 3180–3186, 2013.

[11] Jonathan Blake and Hakan B. Gurocak. Haptic glove with MR brakes for virtual reality. *IEEE/ASME Transactions on Mechatronics*, 14(5):606–615, 2009.

[12] Mustafa Calis and M.P.Y. Desmulliez. Haptic sensing for mems with application for cantilever and casimir effect. *CoRR*, abs/0805.0929, 04 2008.

[13] Tom Carter, Sue Ann Seah, Benjamin Long, Bruce Drinkwater, and Sriram Subramanian. Ultrahaptics: Multi-point mid-air haptic feedback for touch surfaces. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, UIST '13, page 505–514, New York, NY, USA, 2013. Association for Computing Machinery.

[14] Inrak Choi, Elliot W. Hawkes, David L. Christensen, Christopher J. Ploch, and Sean Follmer. Wolverine: A wearable haptic interface for grasping in virtual reality. In *IEEE International Conference on Intelligent Robots and Systems*, volume 2016-November, pages 986–993. Institute of Electrical and Electronics Engineers Inc., nov 2016.

[15] Michael Lamport Commons. Introduction to the model of hierarchical complexity. *Behavioral Development Bulletin*, 13(1):1–6, 2007.

[16] Michael Lamport Commons. Introduction to the Model of Hierarchical Complexity and Its Relationship to Postformal Action. *World Futures*, 64(5-7):305–320, oct 2008.

[17] Michael Lamport Commons, Robin Gane-Mccalla, Cory David Barker, and Eva Yujia Li. The model of hierarchical complexity as a measurement system. *BEHAVIORAL DEVELOPMENT BULLETIN*, 19(3), sep 2014.

[18] Michael Lamport Commons and Alexander Pekker. Hierarchical Complexity: A Formal Theory. Technical report, aug 2004.

[19] Pololu Corporation, 2001.

[20] Ravinder S. Dahiya, Giorgio Metta, Maurizio Valle, and Giulio Sandini. Tactile sensing-from humans to humanoids. *IEEE Transactions on Robotics*, 26(1):1–20, feb 2010.

[21] Ravinder S. Dahiya and Maurizio Valle. *Robotic tactile sensing: Technologies and system*, volume 9789400705791. Springer Netherlands, jan 2014.

[22] Marc Ernst and Martin Banks. Humans integrate visual and haptic information in a statistically optimal fashion. *Nature*, 415:429–33, 02 2002.

[23] Whitney George. Bringing van Hiele and Piaget Together: A Case for Topology in Early Mathematics Learning. *Journal of Humanistic Mathematics*, 7(1):105–116, jan 2017.

[24] Daniel Goldreich and Ingrid M Kanics. Tactile Acuity is Enhanced in Blindness. *The Journal of Neuroscience*, apr 2003.

[25] Püren Güler, Yasemin Bekiroglu, Xavi Gratal, Karl Pauwels, and Danica Kragic. What's in the container? classifying object contents from vision and touch. *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3961–3968, 2014.

[26] Original Instructions. OpenHaptics - Programming Toolkit. page 155, 2018.

[27] G Jansson. Basic issues concerning visually impaired people's use of haptic displays. 2000.

[28] M Gail Jones, Alexandra Bokinsky, Thomas Tretter, Atsuko Negishi, M Jones, Alexandra Gail, Thomas Bokinsky, and Atsuko Negishi Tretter. A comparison of learning with haptic and visual modalities. *The Electronic Journal of Haptics Research*, 3(6), may 2005.

[29] M.G. Jones, A. Bokinsky, T. Andre, D. Kubasko, A. Negishi, R. Taylor, and R. Superfine. Nanomanipulator applications in education: the impact of haptic experiences on students' attitudes and concepts. In *Proceedings 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. HAPTICS 2002*, pages 279–282, 2002.

[30] Jeffrey Kodosky and National Instruments. Jeffrey kodosky. *Proc. ACM Program. Lang*, 4(June), 2020.

[31] Eva Yujia Li, Michael Lamport Commons, Jonas Gensaku Miller, Terri Lee Robbinet, Helena Marchand, Carrie Melissa Ost, and Sara Nora Ross. Relationship among measures within the social and moral development domain. Technical Report 3, 2014.

[32] Huaping Liu, Yupei Wu, Fuchun Sun, and Guo Di. Recent progress on tactile object recognition. *International Journal of Advanced Robotic Systems*, 14(4):1–12, jul 2017.

[33] Lu-miao Liu, Wei Li, and Jia-jia Dai. Haptic technology and its application in education and learning. pages 1–6, 08 2017.

[34] Shan Luo, Joao Bimbo, Ravinder Dahiya, and Hongbin Liu. Robotic tactile perception of object properties: A review, dec 2017.

[35] Jonatan Martínez, Arturo S. García, José P. Molina, Diego Martínez, and Pascual González. An empirical evaluation of different haptic feedback for shape and texture recognition. *Visual Computer*, 29(2):111–121, feb 2013.

[36] C. S. Meera, Pinisetti Swami Sairam, Vineeth Veeramalla, Adarsh Kumar, and Mukul Kumar Gupta. Design and analysis of new haptic joysticks for enhancing operational skills in excavator control. *Journal of Mechanical Design, Transactions of the ASME*, 142(12), dec 2020.

[37] Fiona N. Newell, Marc O. Ernst, Bosco S. Tjan, and Heinrich H. Bülthoff. Viewpoint dependence in visual and haptic object recognition. *Psychological Science*, 12(1):37–42, 2001.

[38] Studio 1 Productions. Fixing firewire (ieee 1394) problems with windows 11, 10, 8 and 7. installation and download instructions for the legacy firewire (ieee 1394) for use with windows 10, 8 and 7.nbsp; special instruction for windows 11 information on pci-e firewire cards, video capture, audio devices and scanners., Mar 2022.

[39] José-Luis Rodríguez and Ramiro Velàzquez. Haptic Rendering of Virtual Shapes with the Novint Falcon. *Procedia Technology*, 3:132–138, 2012.

[40] H. A. Ruff. Infants' manipulative exploration of objects: Effects of age and object characteristics. *Developmental Psychology*, (20), 1984.

[41] Jose Salvatierra. Working with python virtual environments: The complete guide, Jul 2022.

[42] SemsAble, 2003.

[43] Harold Soh and Yiannis Demiris. Incrementally learning objects by touch: Online discriminative and generative models for tactile-based recognition. *IEEE Transactions on Haptics*, 7(4):512–525, oct 2014.

[44] M. Sreelakshmi and T.D. Subash. Haptic technology: A comprehensive review on its applications and future prospects. *Materials Today: Proceedings*, 4(2, Part B):4182–4187, 2017. International Conference on Computing, Communication, Nanophotonics, Nanoscience, Nanomaterials and Nanotechnology.

[45] M Stamm, M E Altinsoy, and S Merchel. Identification accuracy and efficiency of haptic virtual objects using force-feedback. Technical report, Dresden, 2010.

[46] Sepideh Tabrik, Mehdi Behroozi, Lara Schlaffke, Stefanie Heba, Melanie Lenz, Silke Lissek, Onur Güntürkün, Hubert R. Dinse, and Martin Tegenthoff. Visual and tactile sensory systems share common features in object recognition. *eNeuro*, 8(5), sep 2021.

[47] Dzmitry Tsetserukou, Shotaro Hosokawa, and Kazuhiko Terashima. Linktouch: A wearable haptic device with five-bar linkage mechanism for presentation of two-dof force feedback at the fingerpad. In *2014 IEEE Haptics Symposium (HAPTICS)*, pages 307–312, 2014.

[48] Vassilios S. Argyropoulos. Tactual shape perception in relation to the understanding. *The British Journal of Visual Impairment*, 2002.

[49] Dangxiao WANG, Yuan GUO, Shiyi LIU, Yuru ZHANG, Weiliang XU, and Jing XIAO. Haptic display for virtual reality: progress and challenges. *Virtual Reality and Intelligent Hardware*, 1(2):136–162, apr 2019.

[50] G. Westling and R. S. Johansson. Factors influencing the force control during precision grip. *Experimental Brain Research*, 53(2):277–284, 1984.

[51] Wikipedia. Labview, Jul 2022.

[52] Mimi Hebert Wright and Jeffrey J Lockman. *A longitudinal study of banging during infancy: The effects of object properties on manipulation*. PhD thesis, Tulane University, 1988.