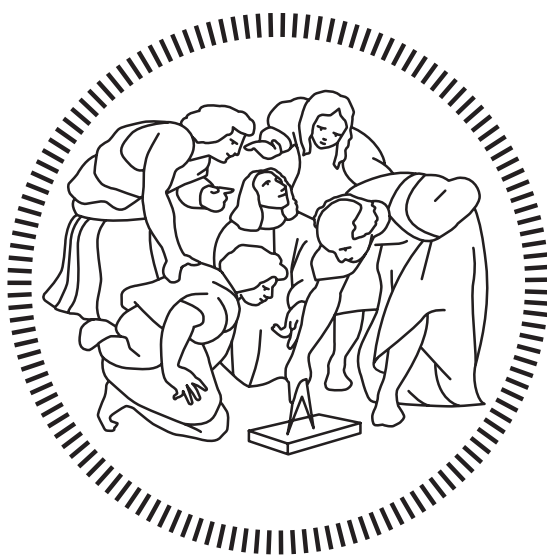


Politecnico di Milano

SCUOLA DI INGEGNERIA INDUSTRIALE E DELL'INFORMAZIONE
Master Degree – Computer Science and Engineering



Enhancing autonomy of persons with Autism through Virtual Reality

Relatore
Prof.ssa Franca GARZOTTO

Co-Relatore
Dott. Francesco VONA

Candidato
Elio Salvini – 10490058

Anno Accademico 2020 – 2021

Ringraziamenti

Voglio ringraziare la professoressa Garzotto per avermi dato la possibilità di lavorare a questo progetto, e di avermi supportato nella fase più importante del mio percorso accademico. Un grazie a Francesco, co-relatore della tesi, per avermi supportato durante tutto il progetto. Un grazie anche a Mariagiovanna, per il suo sostegno e i suoi consigli. Voglio ringraziare il team del laboratorio di I3Lab per il loro contributo. Ed un grazie anche ad Alessandra con il quale ho avuto modo di condividere molte idee durante il progetto.

Voglio ringraziare la mia famiglia per il sostegno e il supporto agli studi universitari ricevuto. Infine, voglio ringraziare i miei amici per essere stati al mio fianco in questo percorso.

Sommario

I disturbi dello spettro autistico Autism Spectrum Disorder (ASD) sono un insieme di disturbi legati al neuro-sviluppo caratterizzati da difficoltà nella comunicazione con altre persone, ristrettezza d'interessi e comportamenti ripetitivi. Persone affette da autismo sono spesso chiuse in sé stesse, vivono in un mondo isolato dove hanno limitata capacità di comunicare ed interagire con gli altri. Le caratteristiche di questo disturbo rendono difficile per le persone che ne soffrono portare a termine compiti quotidiani. Fare uso dei trasporti pubblici, fare la spesa o andare al museo non sono attività semplici per persone autistiche.

Autonomie per l'Autismo Attraverso realtà virtuale, realtà Aumentata e Agenti conversazionali (5A) è un progetto finanziato da Fondazione TIM il cui obiettivo è migliorare l'autonomia di persone autistiche nella vita quotidiana. Al fine di raggiungere questo scopo, 5A fa uso dell'effetto combinato di tre differenti tecnologie, la realtà virtuale, la realtà aumentata e gli agenti conversazionali. 5A opera in svariati contesti di vita quotidiana che possono essere mappati in quattro macro-scenari: trasporti pubblici, servizi culturali e commerciali, sanità e pubblica amministrazione. Questa tesi parla dello sviluppo di un'applicazione per lo scenario della spesa al supermercato. L'applicazione sviluppata è chiamata 5A Supermarket, il suo obiettivo è allenare persone affette da autismo tramite l'utilizzo della realtà virtuale. L'applicazione offre un'esperienza completamente configurabile per adeguarsi al meglio alle necessità dei suoi utenti. Le sessioni di allenamento virtuale proposte da 5A Supermarket sono tipiche attività che prendono luogo nello scenario di un supermercato, dalla raccolta dei prodotti indicati su una lista della spesa, alla fase di pagamento.

5A Supermarket è stata sviluppata per due differenti dispositivi di realtà virtuale, in questa tesi parleremo anche delle differenze tra le due versioni dell'applicazione, sia da un punto di vista funzionale che tecnico.

L'estensione di 5A Supermarket per includere la realtà aumentata e gli agenti conversazionali non è trattata in questo documento. Tuttavia, il processo di sviluppo ha preso in considerazione il bisogno futuro di supportare anche queste tecnologie.

Abstract

Autism Spectrum Disorders (ASD) are a group of neurodevelopmental disorders characterized by difficulty with communication with other people, restricted interests and repetitive behaviors. People affected by ASD are often self-absorbed and seem to exist in a private world in which they have limited ability to successfully communicate and interact with others. The characteristics of this disorder make difficult for people affected by it to perform daily task activities. Making use of public transport, doing shopping or going to a museum are not easy tasks for autistic people.

5A is a project funded by TIM Foundations that aims to improve autistic people's autonomy in their daily life. In order to achieve its purpose, 5A makes use of the combined effect of three different technologies, Virtual Reality, Augmented Reality and Conversational Agents. 5A operates in several different daily contexts that are mapped into four macro scenarios: public transports, commercial and cultural services, health services and public administration. This thesis will describe the development of a 5A application for supermarket shopping scenario.

The developed application is called 5A Supermarket, its goal is to train people affected by ASD through the use of Virtual Reality. It offers a completely configurable experience to best suit every user need. 5A Supermarket training consists of typical activities that take place in a supermarket scenario, from the collection of products signed on a list, to the payment phase.

5A Supermarket has being developed for two different Virtual Reality devices, in this thesis we will also discuss the differences between the two versions of the application, both from a functional and a technical point of view.

5A Supermarket's extension to integrate it with Augmented Realty and Conversational Agents technologies are out of the scope of this document. However, the design and development process took into account these future needs to support also these technologies.

Contents

Ringraziamenti	iii
Sommario	v
Abstract	vii
Contents	x
List of Figures	xii
List of Tables	xiii
Listing	xv
1 Introduction	1
1.1 Context and Rationale	1
1.1.1 ASD and VR	1
1.2 5A Purpose	3
1.3 Goal of the thesis	4
1.4 Thesis structure	4
2 State of the Art	7
2.1 Virtual Reality for ASD	7
2.2 Shopping training for ASD in virtual supermarkets	9
2.3 Characteristics of HDMs and their effect on UX	11
2.3.1 HDM and Degrees of freedom	11
2.3.2 Comparison of different HMDs User Experience	12
3 5A - NGR Analysis and Development Process	15
3.1 Needs	15
3.2 Goals	16
3.3 Requirements	18
3.4 Development Process	19
4 Design	21
4.1 VR Experience	21
4.1.1 Assumptions	21
4.1.2 Main Tasks	22
4.1.3 Take Products Tasks' Subset	23

4.1.4	Prevent errors with warnings	24
4.2	Configurable Experience	26
4.2.1	Interactions	26
4.2.2	Supermarket Related	29
4.2.3	Distractors	30
4.2.4	Configuration Parameters Recap Tables	32
4.2.5	Difficulty Levels	32
4.2.6	Scene Configuration	33
4.2.7	Localization	33
4.2.8	Supermarket Scene Design	34
4.3	Virtual Assistant	36
4.3.1	Virtual Assistant Design	36
4.3.2	Static Conversation	39
4.4	Oculus Quest Version	40
4.4.1	Devices comparison	40
4.4.2	Main Configuration Changes	41
4.4.3	Main Interactions Changes	41
4.4.4	Reasons behind laser pointer	44
4.4.5	Grab mechanic	45
4.4.6	Task changes	45
5	Implementation	47
5.1	Unity3D	47
5.2	Libraries & Plugins	48
5.2.1	Unity Technologies	49
5.2.2	Other	50
5.3	Architecture	51
5.4	Configurable Scene	53
5.4.1	Data Model	53
5.4.2	Run-time scene generation	56
5.4.3	Limitations	58
5.5	Cardboard Interaction System	59
5.6	Oculus Quest Interaction System	61
5.7	Comparison Between Interaction Systems	62
6	Conclusions	63
6.1	Limitations due to Covid-19	63
6.2	Contribution	63
6.2.1	5A Supermarket	63
6.2.2	Cardboard and Oculus Quest versions discussion	65
6.3	Application limitations	67
6.4	Future improvements	68
A	Acronyms	69

List of Figures

Figure 1.1	Cardboard headset closed.	2
Figure 1.2	Oculus Quest	3
Figure 2.1	VR4VR Practising social skills.	8
Figure 2.2	VR4VR Money Management task.	8
Figure 2.3	Comparison between real and virtual supermarket of Aalborg University study.	9
Figure 2.4	HMDs average prices 2015-2020 [9]	11
Figure 2.5	Titan of the Space, application used for University of Ioannina' study.	12
Figure 3.1	Association stakeholders, needs and goals	17
Figure 4.1	Cart menu screen.	24
Figure 4.2	Warning message in task <i>Get Cart</i>	25
Figure 4.3	Warning message in task <i>Produce Products sub-task</i>	25
Figure 4.4	VR movement anchor with teleport PCS.	27
Figure 4.5	VR movement anchor with continuous PCS.	27
Figure 4.6	V2 I am Cardboard - example of cardboard with gear button .	27
Figure 4.7	POP3.0 Cardboard - example of cardboard without gear button	27
Figure 4.8	Hint example with visual and verbal channels active	28
Figure 4.9	Hint example with just the verbal channels active	28
Figure 4.10	Hint screen example when auditory channel is active	28
Figure 4.11	Card payment	30
Figure 4.12	Cash payment	30
Figure 4.13	Distractors User interface (UI) screen.	31
Figure 4.14	Supermarket first version	34
Figure 4.15	Supermarket first version (interior)	34
Figure 4.16	Supermarket second version (interior scene)	35
Figure 4.17	Supermarket final version	35
Figure 4.18	Supermarket final version (interior)	36
Figure 4.19	Virtual Assistant Sketches	37
Figure 4.20	Virtual Assistant position with respect to player	38
Figure 4.21	Virtual assistant asset	38
Figure 4.22	Virtual Assistant conversation flow diagram.	39
Figure 4.23	Oculus Quest Controllers	40
Figure 4.24	Pointer at the centre of the view - cardboard VR version . . .	42
Figure 4.25	Pointer when looking at interactive objects - Cardboard VR version	42

Figure 4.26	Controllers and lasers - Oculus Quest version	42
Figure 4.27	Laser when pointing interactive objects - Oculus Quest version	43
Figure 4.28	Teleport parabolic ray - Oculus Quest version	44
Figure 4.29	Ray changes colour when an anchor is pointed - Oculus Quest version	44
Figure 4.30	User holding a product with the right controller - Oculus Quest version	45
Figure 4.31	User holding a bag with the left controller - Oculus Quest version	46
Figure 5.1	Unity Editor	48
Figure 5.2	Localization repository for text	49
Figure 5.3	Component diagram showing the general architecture of the application	52
Figure 5.4	Data model diagram. Activity related objects in green , Unity related objects in blue	53
Figure 5.5	Configurable scene generated at run-time with ambient light illumination	58
Figure 5.6	Camera Pointer Controller field of view in green . Main Camera field of view in white.	59
Figure 5.7	Camera UI pointer centered on the interactive object. It is bigger than default to inform user that they are pointing an interactive object.	60
Figure 5.8	Camera UI pointer not centered on the interactive object. It is bigger than default to inform user that they are pointing an interactive object.	60

List of Tables

Table 3.1	Stakeholders & Needs	16
Table 3.2	Goals & Needs	17
Table 3.3	Goals &Requirements	19
Table 4.1	Interaction configuration parameters	32
Table 4.2	Supermarket configuration parameters	32
Table 4.3	Distractors configuration parameters	32
Table 6.1	Main design changes in the two version of the application . . .	65
Table 6.2	5A Supermarket Cardboard suggested hardware requirements .	67

Listings

Listing 5.1	Data model for the JSON scene configuration file	54
Listing 5.2	Load scene configuration main function	56
Listing 5.3	Instantiating a list of generic objects	56
Listing 5.4	Instantiating an activity related object	57
Listing 5.5	Instantiating a generic prefab, a Unity related object	57

Chapter 1

Introduction

1.1 Context and Rationale

1.1.1 ASD and VR

Autism Spectrum Disorders (ASD) are a group of neurodevelopmental disorders characterized by deficits linked to social interactions, communication and repetitive or stereotypic behavior [1]. One of the characteristics of this class of disorders is the heterogeneity related to the deficits intensity, where in certain cases some deficits are stronger than others. For example an individual affected by ASD can attempt to make friends and have good eye contact, while another one can be completely aloof [2]. Among the various diagnosis features of ASD there are also daily tasks, in fact autism deficits could make these activities difficult to fulfill. People affected by ASD show severe difficulties in several aspects of daily life, they could feel anxious and overwhelmed in unfamiliar environments like public transports or supermarkets [3]. These difficulties force them to depend on their parents or caregivers [4]. Many autistic people need to learn how to perform activities that can be considered trivial by individuals not affected by ASD. One big challenge when dealing with these disorders is training, from more basic tasks, to more articulated ones. Among the various goals of training there is learning how to overcome daily life tasks' difficulties. These training sessions should also help people affected by autism in becoming more independent from their caregivers. During training activities there is a consistent risk that the ASD trainee loses focus due to a factor external to the activity. Also, ASD population has a decreased ability to generalize knowledge from context, good results obtained from teaching activities could lead to bad outcomes when the context is changed [3]. All these complications stress the need of nontrivial training activities in order to obtain good results on ASD targets. One known solution is simulation, that is a technique in which the learner is actively placed in a "modulated situation or context in which practice and learning can be achieved in a safe environment" [3].

An emerging technology suitable for simulation activities is Virtual Reality (VR), where immersive virtual environments are created using computer graphics. The strong immersivity of VR is due to its ability to give the illusion to the users that they are really inside the virtual environment. When talking about VR immersivity, the best results are reached when using wearable devices that can be put on user's head. These category of devices is called Head Mounted Device (HMD). HMDs reach a high

level of immersion thanks to the stereoscopic vision that replicates the perception of depth and creates realistic 3D environments. HMDs keep track of the user's head rotation and update consequently the view on the virtual environment to be coherent with it.

Considering its features, in recent years VR has been recently used as an effective new treatment for helping people affected by autism. ASD patients usually find their surroundings confusing or unpredictable, and they can also have difficulties in focusing on activities. These aspects could make it challenging to obtain results from therapy sessions [4][5]. As stated previously, ASD is characterized by a strong heterogeneity in the intensity of deficits, that could be very different from one individual to one other. For these reasons VR is producing promising results in ASD therapy field [4], since the strong immersivity provided by this technology it is possible to intensively reduce external distractions during training activities. The ability of programs to dynamically change their behaviour based on the user is another significant feature that makes VR applications suitable for ASD therapy.

We investigated the effectiveness of VR for ASD working on a project called 5A. The project we carried on focused on the use of two specific HMDs, a cardboard headset (with the support of a smartphone) and an Oculus Quest. When talking about VR, a "cardboard" visor is a device composed by a smartphone and an empty headset (the cardboard itself). The reason why cardboard headsets are very cheap is that they are just plastic/cardboard containers with two lenses (see 1.1). The acVR effect is generated by a smartphone inserted inside the cardboard headset, the screen landscape oriented splits into two parts showing at the same time both the right eye and left eye images of the virtual environment.



Figure 1.1. Cardboard headset closed.

While when using a cardboard headset there is only one button that can be pressed, an Oculus headset offers a richer interaction paradigm (see 1.2). However, even if Oculus Quest is among the cheaper HMDs, it costs some hundred euros, while a

cardboard headset can be bought for some tens of euros (assuming that the user already owns a smartphone).



Figure 1.2. Oculus Quest

1.2 5A Purpose

5A is a project that aims to improve autistic people's autonomy in their daily life. Since the cognitive rigidity and focus deficit described in the previous paragraphs, for autistic people it can be hard to be autonomous in several daily contexts like public transport, shopping, relating with public administration or health services. For this reason 5A project guides autistic people through a VR training process to help them develop social autonomy. 5A support is not limited to VR training sessions, but it also helps its users using Augmented Reality (AR) and conversational agents. When user are on the field they are supported through the use of AR. While VR is used during training sessions and AR on the field, conversational agents are always present in both phases. It is important to state that VR and AR stages are not two independent parts, but they are strongly coupled thanks to the presence of trans-virtual elements. Trans-virtual elements can be every element that is equal and present in both VR and AR stages. These elements can be 2D images or 3D models, they can be sounds or voices, they can be interactive elements or feedbacks provided to users. The most important trans-virtual element is the Virtual Assistant, an avatar that communicates with the user and guides him during the whole experience. The purpose of 5A is to help users in several different daily contexts that are mapped into four macro use cases: public transports, commercial and cultural services, health services and public administration. The heterogeneity of the use cases creates the need of splitting part of the design process, however each module of the project must coordinate with others when dealing with common sections, and then specialize in the detail of each use case.

1.3 Goal of the thesis

This thesis aimed to design and implement a VR training experience to support people affected by ASD within 5A project. The solution we propose helps autistic people to become more independent in a supermarket scenario. The application, called 5A Supermarket, offers a completely configurable experience to best suit every user need. The experience's configurability is an essential feature to support the end users group of ASD people, whose deficits can be very different depending on the case.

The development process supports both cardboard and Oculus Quest headsets, by creating two versions of the same application to fully exploit the potentiality of the two devices.

The development of 5A Supermarket has been carried on during 2020 and 2021. The main development phases were a first Needs Goals and Requirements (NGR) analysis followed by an iterative phase of design, implementation and assessments of the results. At the end, we performed a final assessment of the developed applications within the laboratory team that took part in the project. Goal of this thesis isn't only the development of 5A Supermarket, but we also carried on an analysis of the differences of the two involved HMDs, and of their effect on the final experience.

1.4 Thesis structure

This thesis is formed by following chapters:

- **Chapter 1 - Introduction** the aim of this chapter is to give a general introduction and short overview of 5A project, stakeholders and main components. It touches also the thesis question.
- **Chapter 2 - State of the Art** provides an overview of VR experiences that aims to improve autistic people capability of performing daily activities without the support of a caregiver. In particular, it performs an in-depth analysis of VR trainings related to commercial services like supermarkets. Then this chapter will focus on differences between Oculus Quest and cardboard visors, and on how the examined devices diversities influence their applications.
- **Chapter 3 - 5A NGR Analysis and Development Process** explains the process of analysis and elicitation of needs, goals and requirements for 5A. Those are common between all 5A macro-areas and projects. Then, this chapter describes the development process of the application.
- **Chapter 4 - Design** reports the overall process of 5A Supermarket application design. It also focuses on the design differences between the cardboard and Oculus Quest versions.
- **Chapter 5 - Implementation** focuses on the technical details of the final developed product, analyzing the tools used, the general system architecture and the different functionalities provided by the application. This includes the description of the hardware and software resources necessary to run the app, as well as the implementation details of the main components of the system.

- **Chapter 6 - Conclusion** sums up the overall results of the thesis and provides some critical reflections on the work done. In addition, the chapter points out some possible improvements for the application to be introduced in the future, both in terms of contents and functionalities.

Chapter 2

State of the Art

Autistic subjects are characterised by some degree of difficulty with social interaction and communication, however ASD deficits are not related only to social interactions. Other problematic aspects are atypical patterns of activities and behaviours, such as difficulty with transition from one activity to another, and unusual reactions to sensations [6]. The abilities and needs of people with autism vary and can evolve over time. While some people with autism can live independently, others can have severe disabilities and require life-long care and support [6].

In recent years it has been shown how technology can overcome these problems and support the learning process of subjects with ASD [4].

2.1 Virtual Reality for ASD

At the Eurasia International Conference in 2018, Shahira Sharafeldin and Hadeel El Naggar stated that virtual environments encourage interactive learning and provide a variety of opportunities for the learner to have control over the learning process [7]. Furthermore, they list four main advantages of using virtual technologies in special needs higher education that are still valid nowadays:

- **increase motivation and engagement:** since VR generate an immersive environment in which people affected by ASD feel protagonists;
- **generate constructive learning approach:** VR experiences improve learning capabilities thanks to the fact that they allow people affected by ASD to investigate, experiment, obtain feedback and interact with virtual environment with a good degree of freedom.
- **provide cheap but effective solutions:** complex and expensive devices aren't required anymore, in fact recent technological advances ease access to VR/AR with smartphones, tablets, and video-game devices.
- **experiment environments:** the achieved degree of immersion gives the opportunity to learn more and have access to realistic objects.

5A aims to help people affected by ASD to be more independent and to learn how to perform daily tasks. To be specific, the experience we developed helps end users

to become independent in doing the shopping in a virtual supermarket. Though our project is focused on supermarkets, several different use cases can be considered for ASD training in this field.

In 2017, University of South Florida developed a VR system for training people affected by autism in several scenarios [8]. The system they used is called VR4VR, and it is based on a HMD, a 180° curved screen and a motion tracking system to detect users input. The HMD and the 180° were used separately for different scenarios. The purpose of the project was to training and teaching to ASD subjects how to perform several tasks in different contexts, from grocery stores to office (see 2.1, 2.2).



Figure 2.1. VR4VR Practising social skills.

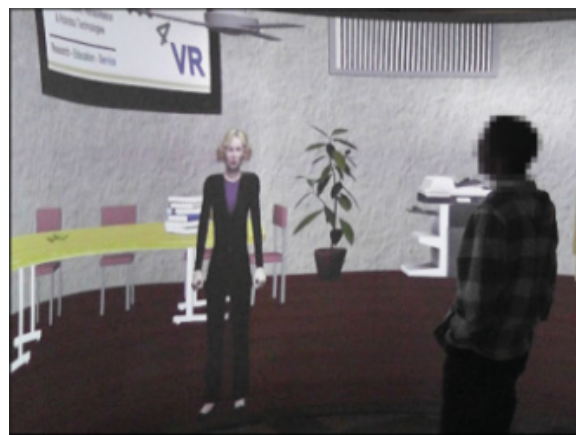


Figure 2.2. VR4VR Money Management task.

It is possible to find many common design choices between 5A and VR4VR. Both projects gave the possibility to enable elements that can distract users while performing tasks, fist all tasks can be executed with different predefined levels of difficulties. Secondly, VR4VR system was designed taking into account also therapists as stakeholders.

University of South Florida carried on a usability evaluation of their system. The results show that users improved their skills in all tasks in which they were trained. In particular, they noted that:

- Providing periodically the same hint when users was not able to complete a task was irritating them.

- When approaching complex tasks the best way to help users was splitting them in elementary steps.
- Repeating the same task more times was making users securer since any form of change may make them uncomfortable.

We took note of all this aspects when designing the 5A experience, especially the last one related to repetitive tasks. 5A aims to prepare ASD users to unpredictable events, that are something that always scares them. So we needed to project carefully how to generate this effect of unpredictability in the experience. How we faced this part of the project is more deeply explained in 4.2.

2.2 Shopping training for ASD in virtual supermarkets

An interesting study concerning training for ASD subjects in a commercial service context has been carried on by Aalborg University (Copenhagen), in 2017 [3]. In this study a virtual supermarket has been created to train people with Autism in doing shopping.

In this study a ten days training with the VR application has been conducted on a group of autistic subjects, preceded by a base-line training in a real supermarket to evaluate participant starting skills, and followed by an assessment of the result in the same real supermarket. The virtual supermarket was designed to look like the real one adopted for the study (see 2.3). The virtual supermarket was designed in this way to ease the learning process, since it is easier to apply what has been learned in the same or similar context where knowledge was acquired. 5A aims to replicate the supermarket environment as well, however since the purpose of our project is not to train users in doing shopping only in a specific supermarket, we decided not to take a specific real supermarket as a reference. Though, 5A is not referring to a real supermarket, still it remains the need of modelling a realistic and familiar supermarket that looks like a believable Italian or European one, to keep the training and the practise context more similar as possible. This familiarity between the two phase of training and assessment of what the user learned are both covered by 5A thanks to the AR stage and the presence of trans-virtual elements (see 1.2).



Figure 2.3. Comparison between real and virtual supermarket of Aalborg University study.

The shopping proposed in this study required the collection of four products contained in a shopping list. There were 9 pre-determined shopping lists, each day one of them was assigned to each participant. In 5A our goal is to provide an autonomous application that could be used without the help of therapists or caregivers (depending on the user), for a longer period than ten days. For this reason, we considered the necessity of giving users a configurable experience, not only to make the application customizable to meet the needs of each user, but also to make it effective for medium-long training periods. 5A Supermarket shopping list must change and depend on the application configuration values.

In this study it has been used an HTC Vive headset linked to a custom performing computer, built for the project. Some VR devices can be linked to a performing computer to enhance performances, while others cannot be used if not connected to a PC. 5A is using standalone low-cost VR devices, such as Google Cardboard and Oculus Quest. This devices does not depend on a PC, they are standalone and they can be used without the support of a computer. We decided develop 5A on these devices to make it more accessible to users since they belong to a low-middle price category. We also assumed many users interested in 5A don't own sufficiently performing computers to support a VR device.

Aalborg University's study confirmed, with a final assessment in a real supermarket, that ASD participants were able to complete all tasks even in a context that was more difficult than the one of the first training.

Another study concerning the same field has been conducted by a team of researchers using a desktop application called Virtual Action Planning Supermarket (VAP-S) [4]. Even if this study is not using an HMD like 5A Supermarket, the application scenario is the same. It is useful to analyse how the supermarket activity has been designed, and its results on ASD individuals.

The activity designed in VAP-S covers all the same main shopping steps of 5A Supermarkets. Users need to complete a shopping by selecting products in the virtual supermarket according to a given list, then they have to pay the collected products and to exit the supermarket.

VAP-S is able to track user's positions inside the supermarket and to register their actions. By looking at the registered information therapist can distinguish between healthy people and patients affected by mental disorders. Differently from VAP-S, 5A focus on the therapist as a stakeholder is limited to the initial configuration phase.

Even in this study ASD participants were split into two groups, an intervention group that received the treatment, and a control group that did not receive it. VAP-S training results showed that the intervention group achieved a significant improvement compared to the control group. In this study to main measures have been used to evaluate the participants:

- *efficiency*, the ability to complete the task in the shortest amount of time.
- *accuracy*, the ability to find the correct item at the lowest price.

The intervention group showed an improvement in both efficiency and accuracy in performing the shopping tasks compared to the control group.

These studies show that training in virtual environment for ASD people brings to improvements of the participant abilities of completing daily tasks related to shopping.

However, this is still an open field for research, where 5A tries to add to VR new technologies like AR and Conversational Agents to combine their potentialities and study their combined effects on ASD treatment.

2.3 Characteristics of HDMs and their effect on UX

2.3.1 HDM and Degrees of freedom

Most HDMs keep track of the user's head position to update what the user is seeing, but cheaper solutions (i.e. a mobile device with the support of a cardboard) do not keep track of it. When an HMD keeps track of both user's head rotation and position it owns six degrees of freedom (6 DoF) because both user rotational and translational movements are replicated in the virtual environment, while if a device only replicates rotational movements it owns three degrees of freedom (3 DoF) resulting in less immersivity. HMDs average price has decreased in the last 5 years (see 2.4), making 6 DoF devices more available for training purposes.

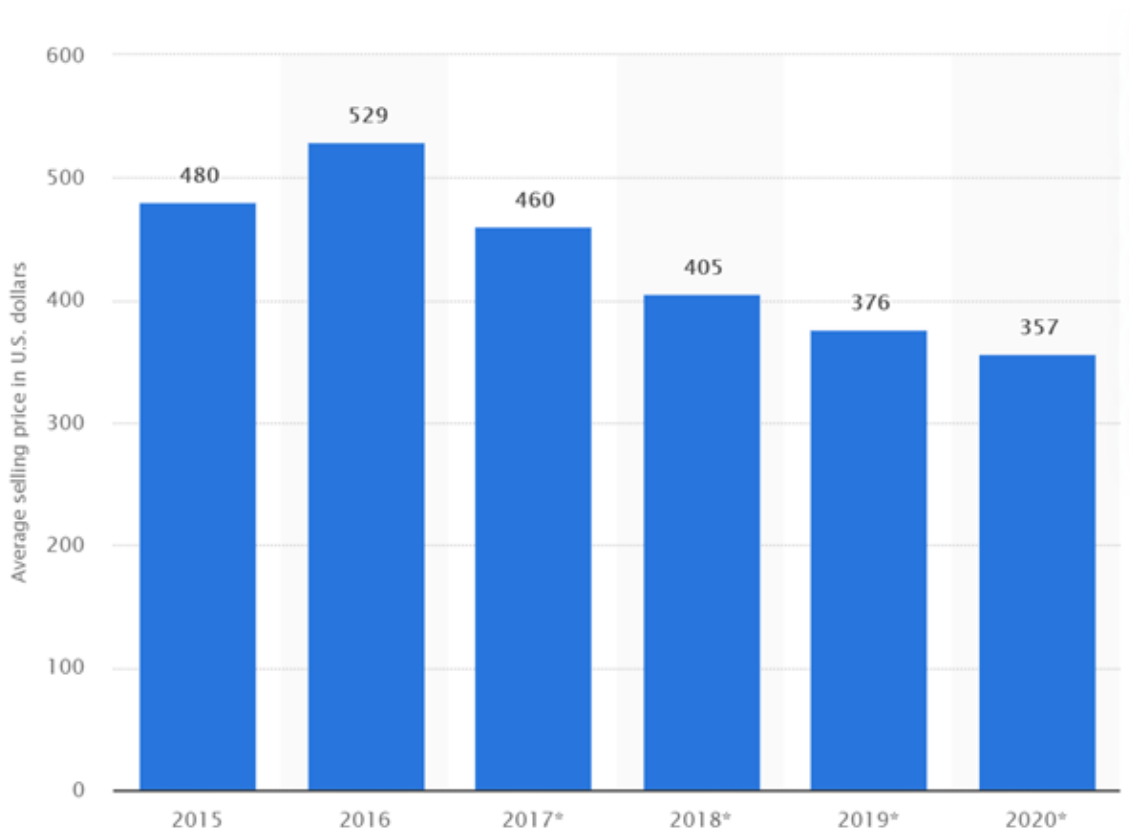


Figure 2.4. HMDs average prices 2015-2020 [9]

Cardboard headsets are 3 Degrees of Freedom (DoF) devices, they do not track head position in space, and for this reason they can be less immersive with respect to other HMDs. Also, some focusing issues could happen when using these devices, they can usually be properly solved by adjusting the phone's screen distance from the lenses. Not all cardboard headsets have the same characteristics, for example

some may allow users to adjust eye distance, while others do not. Not all cardboard headsets have a “gear button”, a button that when pressed triggers a mechanism inside the cardboard that touches the smartphone screen allowing the user to make use of one button interactions. This heterogeneity of cardboard headsets forces VR developers to implement different interaction mechanics.

Oculus Quest is a standalone 6 DoF device. Its 6 DoF do not refer only to the headset, but also to the two linked wireless controllers, this allows users to see the controllers they are holding in the virtual environment, and to use them to interact with the virtual world. The enhanced degrees of freedom, and the presence of the controllers result in a more immersive experience from an interaction point of view. While when using a cardboard headset there is only one button that can be pressed (sometimes neither that), when using an Oculus headset there are two controllers with more buttons that can be used for a richer interaction. An example of interaction that can be performed with an Oculus Quest is making user grab objects and move them in the virtual environment.

2.3.2 Comparison of different HMDs User Experience

The heterogeneous functionalities offered by different type of HMDs have an impact on the final User Experience (UX).

A study conducted by The University of Ioannina describes the results on the same experience of two different HMDs [10]. The tested experience operates in a didactic context. The devices involved in the study were an Oculus Rift linked to a PC, and a Google Cardboard headset with a smartphone. The "Titan of Space" application has been used for both devices to evaluate the UX. This application guides users in a virtual tour of the solar system with the goal of learning the order of planets.



Figure 2.5. Titan of the Space, application used for University of Ioannina' study.

User experience was evaluated by measuring five different user metrics:

- Spatial Presence, that refers to the sense of "being there", inside the represented virtual environment.

- Usability.
- Workload, that refers to users' mental and physical demand.
- Simulator Sickness, nausea, headache or disorientation caused by the use of the HMD.
- Satisfaction.

Rather than the clear limitations of cardboard visors compared to the Oculus Rift, the study results do not elicit the devices differences. Even if the Google Cardboard score concerning the learning gain was a bit smaller than Oculus Rift one, the study concluded that both VR headsets are capable of providing acceptable levels of immersive user experiences in educational virtual environments.

The University of Ioannina's study confirms that both 3 DoF and 6 DoF headsets can be used for educational purposes. However their focus was not on ASD users, and also the activity they used for evaluating the UX was simpler than 5A Supermarket tasks. This thesis aims to provide an analysis of the differences between cardboard visors and Oculus Quest HMDs, and how they impacted on the development of the project.

Chapter 3

5A - NGR Analysis and Development Process

5A is a long lasting, open source project conceived by *I3Lab* and financially supported by *Fondazione Tim*. Moreover, all clinical-therapeutic skills, necessary to achieve a perfect product with respect to main stakeholders, are guaranteed by neuro-psychiatrists and psychologists of *IRCCS Medea*, the only Italian scientific institute recognized for research and rehabilitation in the context of developmental age, and the autism center of *Fond Sacra Famiglia* which has a user base of about 800 subjects with ASD per year.

With the support of I3Lab team, it has been designed and developed an immersive VR experience about shopping in a supermarket.

Purpose of this chapter is to investigate needs, goals and requirements necessary to successfully design and develop 5A. The process of needs, goals and requirements elicitation has been conducted through different meetings with therapists.

At the beginning of our research, we identified two main groups of stakeholders:

- People affected by ASD, which are 5A main stakeholders.
- Therapists, tutors and caregivers of people affected by ASD. For shortness these actors will be called tutors.

3.1 Needs

To identify the needs of the users, we collected information about what it means to be in this particular kind of mental situation. For people with ASD, we identified the following needs (see Tab. 3.1):

- **N1 - Orientation difficulties:** real world is complex and convoluted, it has lot of stimuli that can be both centrally or laterally positioned with respect to the user [11]. ASD subjects can have problems in correctly directing their focus where needed. This can bring to a number of potentially critical consequences, such as difficulties in spatial orienting.

- **N2 - Reiteration tendency:** people with ASD find comfort in repeating certain behaviors and might not want to change in daily activities, therefore, they need the possibility to repeat the same activity more times to become confident with it.
- **N3 - Exaggerated reaction to external stimuli:** sensitivity to emotions is a common issue for autistic people and they can have unusually intense emotional reactions compared to the rest of the population. These exaggerated reactions need to be properly handled.

Tutors' need is related to their will of helping in the best way possible their patients. . We identified also for tutors an other set of needs:

- **N4 - Easily setup the training:** these actors may have technical difficulties and that's the reason why they must be able to easily setup the trainings.
- **N5 - Customize the training:** therapists know their patients and need to properly customize trainings on the basis of the user skills and weaknesses.

Identified Needs	
Stakeholders	Needs
People affected by \ac{ASD}	N1) Orientation difficulties
	N2) Repetition tendency of certain activities
	N3) Exaggerated reaction to external stimuli
Therapists	N4) Easily setup the training
	N5) Customize the training

Table 3.1. Stakeholders & Needs

3.2 Goals

The previous analysis created the bases to proceed and define this main goals of the application (see Tab. 3.2).

- **G1 - Find the right path and landmarks:** the first objective is to help end users to overcome their spatial orienting difficulties. One goal of the application is to make users aware of where they are, and where they have to go .
- **G2 - Make the task familiar:** The application aims to make users familiar with the task they perform and to be prepared to face the same task in real life.
- **G3 - Filter the stimuli:** real world has lots of stimuli such as people, objects, sounds that generate confusion in people with ASD. Another goal is to group all stimuli that can impact on users, and give to tutors the possibility to configure them depending on who they are working with.

- **G4 - Manage anxiety:** a goal is to help end users manage the strong anxiety that easily affect them.
- **G5 - Manage unpredictable changes:** unpredictable events can occur in real life, and they should take into account by the application. Users need to be prepared to face unpredictable situations.
- **G6 - Provide an easy set up:** since tutors may not be tech people a goal is to let them set up the VR training in an easy way.

The correlation between stakeholders, needs and goals is expressed in Fig. 3.1).

Goals		
ID	Description	Associated Need
G1	Find the right path and landmarks	N1
G2	Make the task familiar	N1, N2
G3	Filter the stimuli	N3, N5
G4	Manage anxiety	N3
G5	Manage unpredictable changes	N3, N5
G6	Provide an easy set up	N4

Table 3.2. Goals & Needs

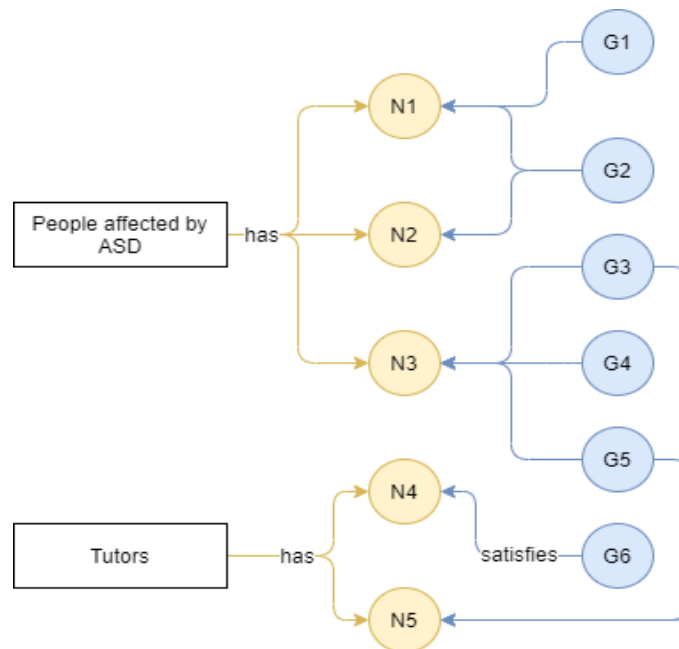


Figure 3.1. Association stakeholders, needs and goals

3.3 Requirements

After defining both *needs* and *goals* requirements were selected:

- **R1:** the application must contain all realistic landmarks necessary to complete the activity;
- **R2:** the application must implement a journey and guide users through it;
- **R3:** the game must give a certain degree of freedom in exploring the environment;
- **R4:** 5A experience must be implemented as a Final State Machine (FSM) whose states are the necessary tasks to successfully complete an activity;
- **R5:** the application must provide guidance before and during task completion;
- **R6:** the app must ensure that users follow the correct activity steps using proper communication channels;
- **R7:** the whole experience must be configurable;
- **R8:** the app configuration must give the possibility to add or remove audio and object stimuli;
- **R9:** the game must reward users with voice and static sounds when they have correctly completed a task;
- **R10:** the game must handle possible status of anxiety by implementing a static conversation system;
- **R11:** 5A must manage uncertainty between tasks and all possible interactions must trigger an outcome;
- **R12:** UI must be accessible;
- **R13:** UI must follow all best practices of Human Computer Interaction (HCI);
- **R14:** the app must provide different interaction paradigms that fully exploit each device potentialities.

Look at Tab. 3.3 to see that every goal of the application is covered by at least one requirement.

Goals and Requirements Martix						
	G1	G2	G3	G4	G5	G6
R1	X					
R2	X					
R3	X	X				
R4		X		X		
R5		X				
R6		X	X			
R7			X			
R8			X			
R9				X		
R10				X		
R11					X	
R12						X
R13						X
R14	X					

Table 3.3. Goals &Requirements

3.4 Development Process

After a global requirement and use cases analysis an iterative development process has been adopted for this project. An iteration of the development process followed these phases:

- **Design:** discussion with therapists and tutors about how design in detail the experience. At each iteration of this process we focused on few specific requirements or phases of the experience (for example the payment sequence).
- **Implementation:** it follows an implementation of what was designed in the previous phase.
- **Assessment:** what was implemented in the previous phase is evaluated by therapists and tutors. All feedbacks from testing with therapists are collected and used to correct what was designed in the first phase when needed. This step was also a starting point for new ideas in next iterations.
- **Implementation of assessment results:** in the last phase what emerged from the assessment phase is implemented.

This process has been adopted for designing the whole experience for cardboard VR. After the development of the cardboard version we dedicated to the porting for Oculus Quest. The whole porting has been carried on in one iteration since most of the experience was already designed. Oculus Quest version relies upon a totally different interaction system that allows more sophisticated interactions (compared to the cardboard version). Oculus Quest enhanced interaction paradigm results in a more immersive experience.

Chapter 4

Design

The aim of this chapter is to examine the design process and discuss the motivation behind each choice, in particular the next sections will dive deep into: the design process, UX, VR experience design, virtual assistant design, differences between Cardboard and Oculus versions design. Next paragraphs will describe the cardboard version design, then, there will be a focus on the Oculus Quest version design, and on the main differences between the two versions.

4.1 VR Experience

This section aims to analyze in detail the VR experience design. All choices were made by taking into account the stakeholders.

The user experience design follows two main principles:

- **Accessibility:** the application must be intuitive and with a smooth experience. Text attributes and position has been carefully considered and tested in order to achieve readability ease.
- **Attractiveness:** to result engaging and interesting even for users who generally have a low level of focus/attention and to promote the use in autonomy.

The aim of this training is to train end users to do shopping in a supermarket. Before starting the implementation, we analyzed with therapists the activity and defined a set of assumptions. The reason behind the definition of a set of assumptions is to clarify the experience boundaries. We decided which aspects of the real-life scenario were essential, and what was not necessary for the effectiveness of the experience. This analysis brought us to find an ordered sequence of feasible micro-tasks that must be done to complete shopping in a supermarket.

4.1.1 Assumptions

This are the assumptions:

- **A1:** the experience starts inside the supermarket parking.
- **A2:** carts are placed outside the supermarket building, in the parking.

- **A3:** to simplify payment process we rounded the final shopping list to the closest multiple of 5, in this way it is possible to pay only with the use of banknotes. Coins are not considered for the experience.
- **A4:** there are not automatic cash registers inside the supermarket.
- **A5:** cart maximum capacity is 15 products.
- **A6:** the supermarket has only one door that can be used to both enter and exit.
- **A7:** Users always have enough money to pay all the collected products.
- **A8:** the experience ends when users, after collecting the payed products in a bag, bring back the cart where they took it.

4.1.2 Main Tasks

In this section the identified main tasks are listed and analyzed:

- **Get Cart:** To begin the shopping the user has to find a cart and to collect it first.
- **Enter The Building:** After collecting the cart it is possible to enter the supermarket. If users try to enter without collecting it they cannot cross the entrance, and they will be informed that a cart is needed to enter. We decided to prevent users from entering the supermarket without a cart, even if it is possible in real life, to make them used to always search the cart at the beginning of the shopping.
- **Pass Entrance:** Once inside the building most supermarkets make a clear distinction between the entrance area and the shopping area using separators or fences. The customer is required to pass through an entrance gate to access the shopping area, this step asks user to find and pass this gate.
- **Take Products:** This is the core task of the experience. Users have to look for requested products listed inside a shopping list, and to collect them inside the cart. Products can be anywhere in the supermarket, and depending on the product they may require different interaction processes to be collected (4.1.3). Users can make use of different suggestions provided by the application to complete this task.
- **Go To Cash:** After collecting all required products it is possible to complete the shopping by going to an opened cash register.
- **Place Products On Tape:** This task asks users to collect each product inside the cart and to place it on the cash register tape.
- **Pay:** This is the payment phase. This task differs each time depending on the selected payment configuration. Users have to open a virtual wallet and select banknotes or a credit card to pay. Depending on the payment mode, users can pay with a credit card or banknotes. In the first case it is enough to select the

card and to confirm the payment. In the second case a proper set of banknotes must be selected. Users are always informed about the errors they commit during this phase.

- **Exit:** After paying users exits the shopping area crossing the cash register. The collected products are automatically put inside a bag. Users have to leave the building passing through the door.
- **Return Cart:** This is the last task. Users have to bring cart back where they have collected it. They need to interact with the carts rack to place back the cart they are using.

4.1.3 Take Products Tasks' Subset

Take Products is the core experience task and requires to be described in detail. During this phase it is possible to fully explore the supermarket, but it is not possible to exit until all products are collected. Users can consult the shopping list when they want interacting with the virtual assistant. All required products to complete the shopping are listed inside in the shopping list. The products listed in the shopping list are randomly generated at the beginning of the experience, according to the specified configuration. From an interaction point of view products can be classified in two categories:

- **Standard products:** they can be directly collected from shelves. After interacting with a product on a shelf it will be shown in detail, and a simple menu will appear. The menu shows the product name and two buttons, one to collect the product and one to put it back.
- **Produce products** they cannot be collected directly form shelves, users need to complete a sub-task to collect them.

Each product can be collected in the proper department, departments names are displayed on signs. Users are stimulated to search for the right department to find the requested products. The virtual assistant helps users in searching the right departments.

After being collected products can be removed by interacting with the cart. A menu showing each collected product appears and interacting with it will allow users to choose which product remove (see 4.1).



Figure 4.1. Cart menu screen.

Produce Products Task

It is possible to collect produce products during *Take Products* task. This task can be divided in the following sub-tasks:

- **Collect Gloves and Bag:** the first thing to do is to search for gloves and a bag in the produce department. They can be collected in any order. Each time the user has to collect a produce product they need a bag, while gloves can be collected once.
- **Collect Produce Product:** now users can collect produce products directly from shelves. Only one product can be put inside a bag.
- **Weight Product:** once the product is inside the bag it must be weighted. Users have to find a scale and they need to interact with it to weight the product. After that the product is weighted it is put automatically inside the cart.

4.1.4 Prevent errors with warnings

In order to better prepare users to real-life tasks, we decided to give them a certain level of freedom. In fact more actions can be done by users during each step of the experience, but not all of them will allow to proceed. However, the path to complete all tasks is pretty guided thanks to hints and virtual assistant support. The virtual assistant properly communicates with the user to inform them when the actions they are performing should not be done. In particular users can provide warning during the following main tasks:

- **Get Cart:** if user tries to enter the supermarket without taking the cart they cannot pass the door and they are informed that they need to take a cart before entering (see 4.2).

- **Enter The Building:** by interacting with the carts rack it is possible to return the cart, however it is not possible to this until the last task is reached.
- **Take Products:** users could try to interact with cash registers before completing the task (therefore the shopping), in this case they will be informed that they have to collect all products before paying.
- **Produce Products sub-task:** several errors can be committed in this task depending on the order of user actions (see 4.3). For example, a user could try to take before the product before gloves, or they can try to weight an empty bag. Trying to do this actions produces warning feedbacks that help user in understanding what their mistakes.
- **Pay:** depending on the payment mode chosen, users can try to pay with not enough money selected (or with too much if the payment mode is *cash without change*).



Figure 4.2. Warning message in task *Get Cart*.



Figure 4.3. Warning message in task *Produce Products sub-task*.

4.2 Configurable Experience

People's life is full of unpredictable aspects. Even when we perform a daily activity formed by several unchanging tasks, the external environment brings many different scenarios and variables in between one task and the other. This variables can be the location in which we are, how crowd the location is, the amount of background noise, the presence of more objects that may have the same function,... etc etc. We decided to make the scene completely configurable in order to

- increase smoothly training difficulty to reach a scenario similar to the real world;
- have a highly customizable experience with respect to the specific needs of every single stakeholder making the application accessible to a wider range of people affected by ASD.

The configurable parameters can be divided in three groups.

4.2.1 Interactions

The first set of configurable parameters is related to basic interactions:

- **Movement Options:** some users can suffer from motion sickness when playing with VR applications. For this reason we added the possibility to select between two different type of locomotion to let them move around the VR environment:
 - **Teleport:** when this mode is activated, the player points at a location they'd like to move to and then it is instantaneously teleported to the new selected position (see 4.4).
 - **Continuous movement:** in this case, once the player points at a location, a linear trajectory is built and the player is smoothly moved on it with a certain speed (see 4.5). This is done to give to users the perception of moving through the VR world as if they are walking.

While moving during the experience users can recognize the selected movement option from the PCS that appears by interacting with the anchors. To make this point more clear look at the following figures.

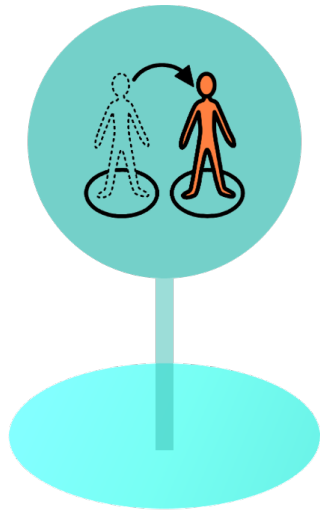


Figure 4.4. VR movement anchor with teleport PCS.



Figure 4.5. VR movement anchor with continuous PCS.

- Interaction Options:** 5A experience has been first designed for Cardboard headsets. Nowadays there exists many types of Cardboard that mainly differs for the presence or not of a gear button (see 4.6, 4.7). In order to support a wider range of low-cost headset, it is possible to configure if the interaction will be done through *Gaze & commit* or *Gaze dwell*. In the first case to interact with an object it is necessary to look at it and to press the gear button. In the case of *Gaze dwell* the interaction with an object is triggered when a user stares at it for a given amount of time. Users can see a pointer at the center of their view, this pointer shows effects to help them in understanding if the object that they are looking at is interactable or not.



Figure 4.6. V2 I am Cardboard - example of cardboard with gear button



Figure 4.7. POP3.0 Cardboard - example of cardboard without gear button

- **Active Communications Mode:** This parameter is strictly related to hints and warnings provided during the experience. The application uses three different elements to communicate: Picture Communication System (PCS), texts and the voice of a virtual assistant. It is possible to chose which of them should be used by selecting among 3 different modes:
 - **PCS and Text:** PCS, texts and voice are active (see 4.8).



Figure 4.8. Hint example with visual and verbal channels active

- **Text:** texts and voice are active (see 4.9).



Figure 4.9. Hint example with just the verbal channels active

- **Speech Only:** only voice is active (see 4.10).

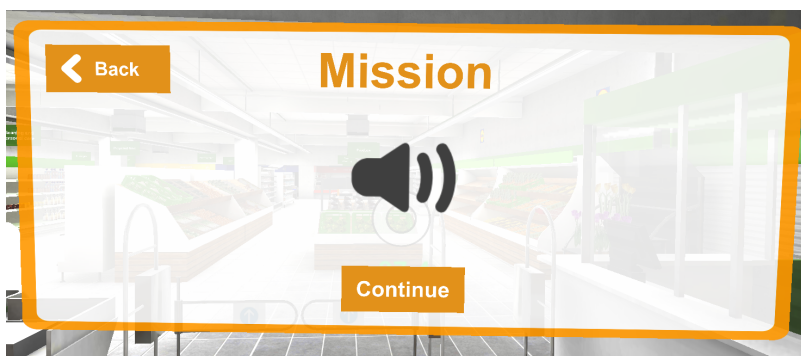


Figure 4.10. Hint screen example when auditory channel is active

4.2.2 Supermarket Related

The supermarket is the environment in which tasks are performed and, for this reason, there are many configurable variables specific for this area. In real life scenarios, supermarkets can be very various, for this reason we designed a deep customization level for the supermarket scene (see 4.2.6), but in this paragraph we describe the customization parameters that can be customized from the settings panel. It is possible to configure these aspects:

- **supermarket map:** since supermarkets have different planimetries it is possible to pick between 3 different pre-generated maps. Other deeper customization settings related to the planimetry will be discussed later.
- **quantity of products:** this variable can be set to *very low*, *low*, *medium*, *high* or *very high*. Depending on the specified value, the number of products that must be collected to complete the shopping can vary from 3 to 15. This parameter has a strong impact on the time required to end the experience.
- **supermarket brand:** it is possible to choose among the most known Italian supermarket brands, then, it will be shown inside the virtual supermarket logos. The aim of this variable is to make the supermarket more believable since ASD users can have less difficulties in orientating in a familiar environment.
- **all cash registers opened:** if this variable is set to true all cash registers inside the supermarket will be always opened. Otherwise, if set to false, the number of opened cash registers will be random, in this case at least one cash register will be opened. Closed cash registers cannot be used to pay the collected products.
- **vegetables/fruit shelf enabled:** collecting fruit or vegetables requires a more complex interaction since it is not possible to directly put these products inside the cart, but they must be put inside a bag, and then weighted. Since this process can be too complex for some users it is possible to enable/disable the presence of these kind of products in the shopping list.

Another important configurable aspect related to the supermarket environment is the payment method. Specific users may always pay by credit card, some others may prefer to pay by cash or the selected method may depend from time to time. An extra division has been applied to the cash payment method, in fact it is possible to:

- **pay using credit card:** no computation are needed for this mode, users just need to select the credit card in their wallet (see 4.11).
- **pay with change:** If the selected amount of money exceeds the shopping cost the cashier will give back the change (see 4.12). This modality covers the real life scenario in which end users affected by ASD are not really strong in doing math calculations and may not be able to recognize a possible right combination of banknotes to pay the shopping.

- **pay without change:** this modality is aimed to force the end users to do math calculation to pay. They are required to provide a specific amount of money to pay, the cashier cannot give change back. This choice add a secondary value to the experience by helping users to improve their computing skills.

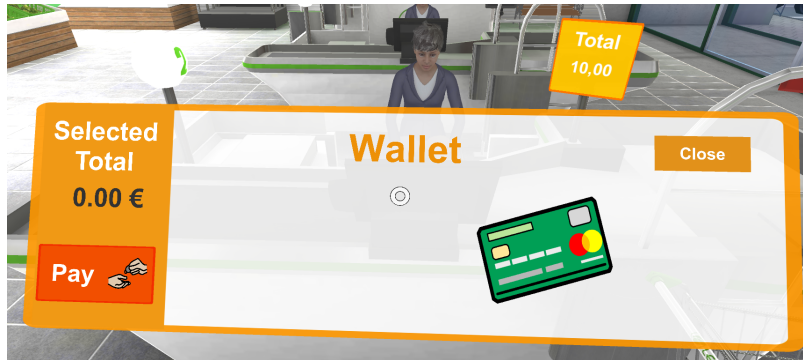


Figure 4.11. Card payment



Figure 4.12. Cash payment

4.2.3 Distractors

This group of configurations gives the opportunity to add or hide 3D assets and sounds that can divert the end user attention from the tasks that they have to perform (see 4.13). In particular it is possible to add or remove from the VR scene the following assets:

- **animated human characters:** the crowd quantity inside the supermarket can be customized. If it is set to *none* the supermarket will be empty, the only people that will be present are the cashiers (that cannot be removed since they are required during the payment phase). This variable can be set also to *low*, *medium* or *high*, to increase or reduce the quantity of people inside the supermarket. Every time the the experience is started the number of people that will visit the supermarket is determined by this parameter, but which character is activated and which not is determined randomly, so it is possible to meet

different avatars repeating the experience more times while keeping the same crowd quantity configuration.

- **audio tracks:** there are 5 different city and supermarket ambience sounds that can be played or not during the experience. It is important to highlight that only ambient sounds can be deactivated, it is not possible to turn off interaction sounds since they are feedbacks whose purpose is to help the user, they are not distractors.

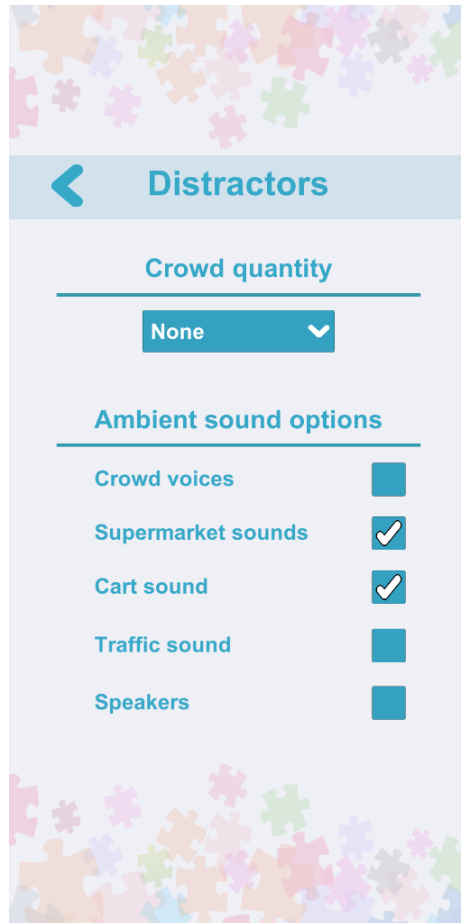


Figure 4.13. Distractors UI screen.

4.2.4 Configuration Parameters Recap Tables

Table 4.1. **Interaction** configuration parameters

Parameter	Values
Movement Options	Teleport, Continuous Movement
Interaction Options	Gear Button, Gaze Mode
Active Learning Channels	PCS and Text, Text, Speech Only

Table 4.2. **Supermarket** configuration parameters

Parameter	Values
Supermarket Map	SupermarketA, SupermarketB, SupermerketC, Configurable
Quantity of Products	Very Low, Low, Medium, High, Very High
Supermarket Brand	Carrefour, Conad, Coop, Esselunga, Eurospin, Lidl, Pam
All Cash Registers Opened	true, false
Vegetables/Fruit Shelf Enabled	true, false
Payment Mode	Card, Cash with Change, Cash without Change

Table 4.3. **Distractors** configuration parameters

Parameter	Values
Crowd Quantity	None, Low, Medium, High
Crowd Voices	true, false
Supermarket Sounds	true, false
Cart Sound	true, false
Traffic Sound	true, false
Speakers	true, false

4.2.5 Difficulty Levels

Since the VR experience is highly configurable, we decided with the help of the therapists to give users the possibility to select between three pre-configured levels with increasingly difficulty and a completely customizable one. These levels of difficulty influence every parameter described in the previous paragraphs. Regardless of the selected level, users can always change configuration parameters before starting the training.

4.2.6 Scene Configuration

In real life supermarkets can have very different planimetries, for this reason we concluded that three pre-computed supermarket maps were not enough for an highly configurable experience. The experience is designed to be run more times and this implies that users will learn well the planimetry of the virtual supermarket. Once that users know by heart the location of every department they will not search for them and the experience will no longer be able to train this searching skill. Moreover, 5A aims to prepare users to unpredictable situations, one of this situations is the re-arrangement of the departments of a supermarket.

The best solution we found is to make the planimetry of the supermarket fully configurable. In the starting configuration menu it is possible to choose the custom map. In this case user's experience will take place in a customized supermarket scene.

The level of detail of the scene/map configuration file allows to custom number, positions and rotations of departments, cash registers and movement anchors. It is also possible to custom single departments, the number and type of shelves, signs and their writings. Also other information about the state of the listed objects can be customized using this file.

This feature provides highly configurable maps that can be built ad hoc for each user. This, and the other configuration experience parameters meet the need of the application to adapt to the heterogeneous target of ASD.

While designing this part, we were searching for a way to always stimulate users search skill and to prepare them to unpredictable changes, but we avoided randomly generated maps. Using random procedural map generation would have raised an issue, therapists or caregivers could want to make user try the same map more times. The necessity of trying the same map more times could be given by the fact that user fails one task and needs an external help to teach them how to proceed. Also, repeating the experience in the same place is useful to increase user's familiarity with the surrounding environment. Individuals affected by ASD can have several difficulties in generalising what was told to them (and therefore also to put it in practise), so to have better chances that the user will complete the same task a second time it is necessary to fully recreate the context when they failed. If the supermarket map changes randomly at each run it is not possible to recreate the same task context twice.

4.2.7 Localization

5A supports both Italian and English languages. Italian was a mandatory language to support since 5A use cases are all based on italian context, also most of the users for testing will be italian and they could not know english. However we decided to support also english to make the application accessible also to people that are not italian mother tongue. The localization has been designed to be scalable to easily add new languages if needed in the future.

4.2.8 Supermarket Scene Design

The design of the supermarket scene, from an aesthetic point of view, followed the same iterative process adopted for the whole project. The scene has been re-designed three times since the first results did not satisfy us. The versions that were created are:

- **Supermarket 1:** the beginning idea was to build the scene using free asset gathered from online stores. Unfortunately, it was not possible to find a complete free supermarket asset bundle, so we needed to search for each single shelf, product and supermarket object model. This solution main issue was the risk of inconsistency among the various models, since they were created by different artists they sometimes did not seem to belong to the same environment. We tried to overcome this issue by collecting models from the same sources/artists. However, the final result was not so satisfying, the building seemed a generic supermarket, but it was not resembling a real supermarket we can see everyday (see 4.14, 4.15). Also, several constraints related to the building model made it not suitable for a configurable experience.



Figure 4.14. Supermarket first version



Figure 4.15. Supermarket first version (interior)

- **Supermarket 2:** at this point of the development process we found the necessity of making also the supermarket planimetry configurable. This forced us to re-design the scene in order to make it more configurable, to categorize shelves inside departments and to divide the internal building from the external using two different scene. In this way the intern scene was more suitable to be configured at run-time (see 4.16).

However, the supermarket was still not immersive. Many models were correct representation of real life objects, but the resulting environment was not resembling an european supermarket.

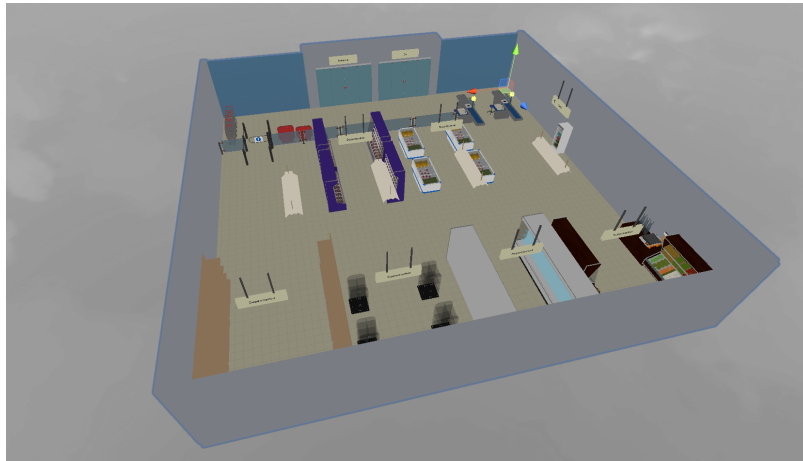


Figure 4.16. Supermarket second version (interior scene)

- **Supermarket final version:** the last version of the supermarket has been created using a 3D asset bundle purchased on Unity Asset Store. The bundle we used has been designed to work with Unity (main development tool used for implementation). The internal of the building was already divided from the external part of the model, and it was wide enough to allow a custom disposition of departments. From an aesthetic point of view the supermarket models were designed on the basis of real european supermarkets. The whole result is a more immersive and believable environment (see 4.17, 4.18).



Figure 4.17. Supermarket final version



Figure 4.18. Supermarket final version (interior)

4.3 Virtual Assistant

Virtual assistant is one of the most important elements of 5A experience and for my thesis. In this section it will be analyzed Virtual Assistant design choice and the static conversation. The reason why the conversation is divided in static and dynamic is because a portion of it is procedural and follows the tasks' FSM and the other one needs to be context aware.

The graphic design of the Virtual Assistant has not been carried on for this project. The assistant graphic aspect was designed by a group of students that participated to 5A.

4.3.1 Virtual Assistant Design

The avatar's asset design was driven by the idea of being simple, easily recognizable, adaptable, trustable, loyal, precise and wise. Before selecting the virtual assistant shape, 2 different types were highlighted:

- **Human character:** this option has been discarded for two main reasons:
 - people affected by ASD have difficulties in interpreting both verbal and non-verbal human language.
 - a human character may be confused with the other customers that simulate a crowded environment.
- **Non-human character:** two options were detected:
 - an avatar shaped as a dog since this species is easily recognizable by everyone and has a well known body language. Moreover, it is a common thought that dogs are loyal, trustable and good friends of humans.
 - an avatar shaped as a robot since users affected by ASD often prefer to interact with devices and computers which are considered a a source of precise and exact knowledge.

A robot character was selected mainly for four reasons:

- it is considered as loyal and trustable as a pet;
- it adds a sense of high precision in delivering information thanks to the fact that it reminds a machine;
- it allows to achieve a simpler, good looking and more real shape with brilliant colors and cleaner textures;
- it is a very popular shape for mascots and companions in video-games.

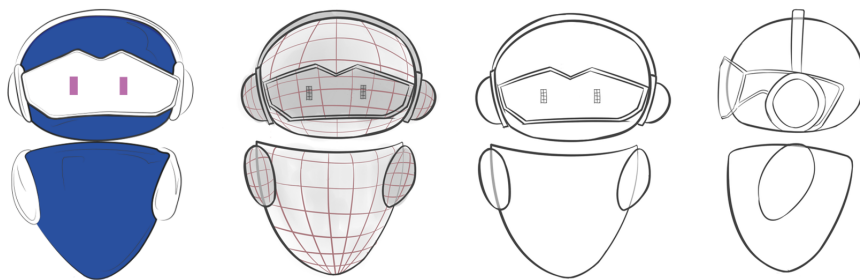


Figure 4.19. Virtual Assistant Sketches

Final avatar design has stumps as arms and the whole body is a floating oval (see 4.19). Limbs and appendages are reduced to the minimum in order to avoid complex animations and to avoid creating pointless extra visual noise on screen. To give a more friendly look body components are slightly asymmetric.

The head is a floating oval and is treated as a visual continuation of the body. The most distinguishing features are the headphones and the screen board. Together they form a sort of “headset” that is mimicking a VR headset. This is a symbol of the fact that the robot is communicating from the virtual world to the real world with end users, it aims to create a connection and a strong channel with players. In fact, while the assistant is talking from VR to real world, end users are doing the exact opposite. The chosen color for body is blue since it calls to mind feelings of calmness or serenity. Blue is often described as peaceful, tranquil, secure, and orderly. It is also frequently seen as a sign of stability and reliability. Pink has been chosen for avatar’s eyes since it creates a great contrast with blue and because it calls to mind feelings of optimism, creativity and happiness.



Figure 4.20. Virtual Assistant position with respect to player

Virtual assistant is located at the bottom right of players and it is always looking at them (see 4.20). This position was selected because it makes the assistant reachable without blocking end users visibility. The avatar asset has been designed also to be an interactive object, by clicking on it the dynamic conversation analyzed will be triggered (see 4.21).



Figure 4.21. Virtual assistant asset

4.3.2 Static Conversation

Virtual Assistant conversation is designed to be both static and dynamic depending on the need level of communication. For simple communication the Virtual Assistant works as a static FSM, giving hints about the current task, warning users when they commit errors. However, 5A aims also to provide dynamic conversation based on a Natural Language Understanding (NLU) conversational agent, to fully support users when they demand for help. For this second conversational modality users can directly speak to the Virtual Assistant using their voices.

Currently 5A Supermarket does not support dynamic conversations. Anyways, we took into account in the design process the future need to integrate this feature.

We decided to split Virtual Assistant conversation into static and dynamic parts to avoid the excessive use of external expensive services. The dynamic conversation depends on services to translate user voices into text, to process the user request into an NLU conversational agent, and to translate the textual agent answer in the Virtual Assistant voice. These three external services can be very expensive, and should not be used when not needed. Static conversation has been used every time that the communication pattern could be known a priori.

Static conversation flows is described by diagram 4.22

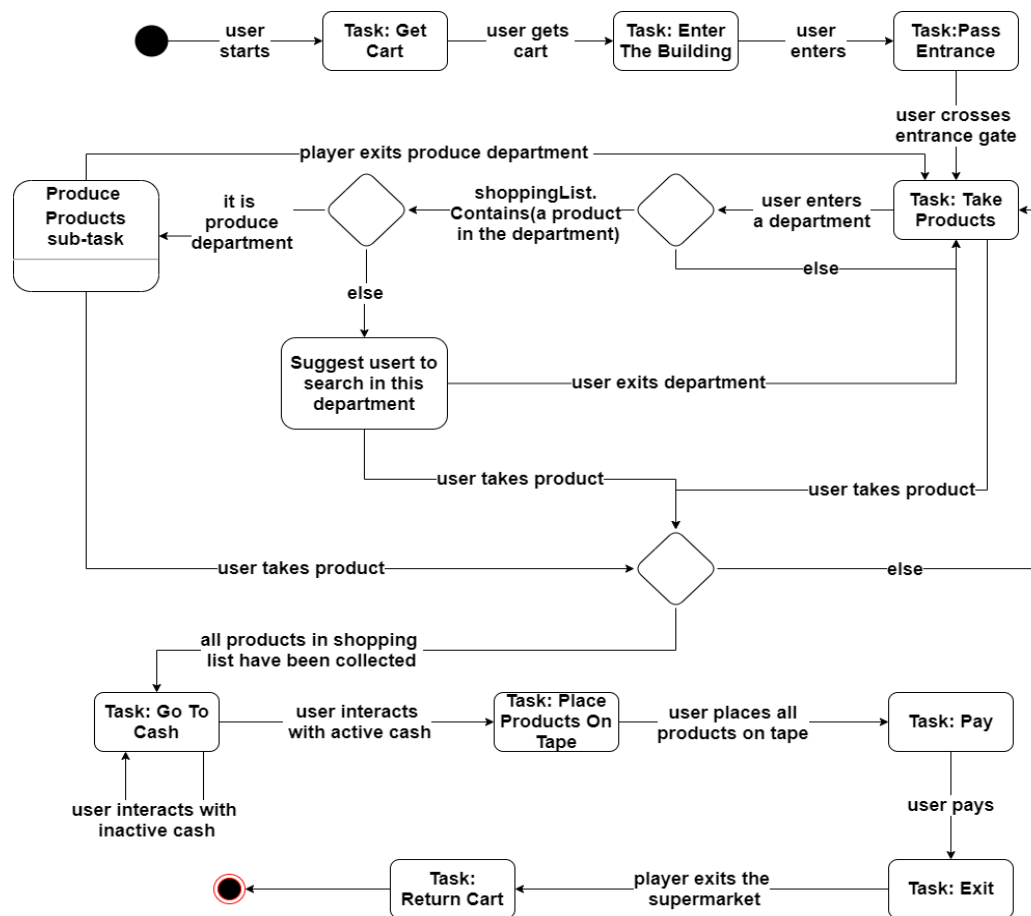


Figure 4.22. Virtual Assistant conversation flow diagram.

4.4 Oculus Quest Version

This part is focused on 5A - Supermarket Oculus Quest design, and on the differences between the two versions, with a particular attention to user's interactions.

4.4.1 Devices comparison

As introduced in 2.3, Oculus Quest is a 6 DoF HMD. The main difference between cardboard VR is that not only user head's rotation movements are replicated in the virtual environment, but also its translation movements. Moving head right, to maybe have a better view of something inside the virtual environment, does not result in any movement inside the scene when talking about cardboard VR, while instead it will result into an head translation when using the Oculus Quest. This means that the cardboard version of the application could miss some user's input related to their head movements. Users could expect that when they translate their head the application provides a feedback accordingly moving the view (as this happens when they rotate their heads). The absence of a proper feedback as answer to a user input reduces the immersivity of the application and reduce the quality of the UX.

Degrees of freedom aren't the only difference that Oculus Quest and its "cardboard counterpart" have, in fact Oculus is linked to two controllers that allows users to bring their hands inside the virtual environment. Both controllers translations and rotations are replicated inside the virtual environment. Controllers opens the way to a more variety of interactions instead of just looking at the object to interact with it. The variety of buttons, and the presence of analog stick (see 4.23) give to users a more detailed control of their movements inside the virtual environment (making a comparison with the cardboard version).



Figure 4.23. Oculus Quest Controllers

As already explained, one of the primary reason why VR is being used in ASD therapies is its immersiveness, whose ability of capturing user's focus and to isolate them from external factors is a key element to let them properly work on a training

activity. It is clear that part of the VR immersiveness is mitigated by the limitation of cardboard VR. It is also true that cardboard VR can be easily used for therapy, while the use of an Oculus Quest could not be affordable for everyone, even if 6 DoF HMD prices are decreasing, an Oculus Quest still costs some hundreds of euros. The main subject of this section will be a comparison between the two devices. This comparison will not only focus on interactions changes, but also on the effectiveness of ASD therapy using the two devices, with a particular attention to understand if the stronger immersiveness of Oculus Quest has a considerable effect in the results of training sessions.

4.4.2 Main Configuration Changes

The different device also comported some changes of the Interaction configuration described at 4.2.1. The two changes about this configuration part are:

- **Movement options changes:** in Oculus Quest version there is not the possibility of choosing movement mode, it will always be teleport based. We removed the continuous movement mode because of the 6 DoF of the device. If users can move inside the virtual environment while moving is head an issue raises if they do it while they are already being moved by the application inside the virtual environment, as it happens in continuous movement mode of the cardboard version. This combination of movement would probably cause motion sickness in many users.
- **Interaction options changes:** in this version is no longer possible to choose between *Gaze & commit* and *Gaze dwell*. This distinction is not necessary now since all Oculus Quest have the same controls, and there is no more the need of a *Gaze dwell* mode to support users when they do not have any control.

4.4.3 Main Interactions Changes

The most important change concerning interaction mechanics is that users no longer move their head to choose with which object interact, now they can use controllers. Both controllers now are the origin of lasers whose function is to help user understand which object they are pointing. When moving/rotating the controllers, users are also moving lasers' origin and consequently the lasers themselves as well.

The screen view is slightly changed with respect to the cardboard version, since the pointer at the centre of it now is not present. The pointer of the cardboard version is helping users in understanding with which object they are interacting; it gives feedback about the interaction using a simple animation to show when an object is interactable or not (4.24, 4.25). In Oculus Quest version the pointer feedbacks about possible interactions are lost since it is no longer used. These feedbacks are substituted by lasers colour change, in fact when an interactable object is pointed the laser default colour is changed to inform users that they can interact with the pointed object (4.26, 4.27).



Figure 4.24. Pointer at the centre of the view - cardboard VR version



Figure 4.25. Pointer when looking at interactive objects - Cardboard VR version



Figure 4.26. Controllers and lasers - Oculus Quest version



Figure 4.27. Laser when pointing interactive objects - Oculus Quest version

In the Oculus Quest version, interactions with objects are still triggered pressing the corresponding button when pointing an interactive object (but in this case the button is located on the controller, not on the headset).

In the cardboard version there were not differences between interacting with movement anchors or with other interactive objects, while in the Oculus Quest version there is a variation of the laser mechanic described before. In fact, to interact with movement anchors it is necessary to switch from standard laser to a parabolic ray pressing the teleport button. While holding the teleport button the laser is no longer a straight line but a parabolic line, like the trajectory of a projectile, where the point of impact determines which object the user is pointing (4.28). When a movement anchor is pointed and user is holding teleport button the parabolic laser changes its colour to inform that it is possible to interact and then move to the selected anchor (4.29). The main reasons why we decided to apply these change are two:

- Distinguish between movement interaction and other interactions. Since this type of interactions are the only ones that comports a consistent movement of the user inside the scene, we thought that it was necessary to make it clearer with the parabolic line visual feedback. By just looking at the type of laser that the controllers are casting it possible for the users to understand if their actions will result in a movement inside the scene or not.
- Uniform with the visual standard for teleport in VR applications. It is widely spread the use of parabolic ray to make users choose where to teleport inside a scene. This visual choice is also used by Unity developers as standard for teleport in their VR demos [12].

It is important to highlight that independently from this change the movement systems remains the same as the teleport based of the cardboard version.

Another change with respect to the cardboard version is the presence of two controllers i.e., two different and concurrent instruments to interact with the environment. In this version of the application it is possible to interact with two different objects at the same. An example of task that requires the use of both controllers is the sub-task related to produce products, it will be explained in detail in the next section.



Figure 4.28. Teleport parabolic ray - Oculus Quest version



Figure 4.29. Ray changes colour when an anchor is pointed - Oculus Quest version

4.4.4 Reasons behind laser pointer

We opted for the laser pointer to select and interact with objects to ensure a good UX. Instead of the laser pointer we could let users interact with objects by directly touching them. This second interaction paradigm is more realistic, then immersive w.r.t the laser pointer, but it could let to some issues. The supermarket scene is designed to be realistic, as object proportion and distances are lifelike. This means that to directly touch objects inside the scene the user needs to move for a considerable space in the real life. This brings some limitations, in fact the application would need a sufficiently open space to be used, preventing it to be suitable for small rooms. Another issue related to this kind of interaction is the impossibility for some users to reach products in the highest shelves lines, in fact it is not easy for users to touch them while they are sit.

The laser pointer interaction is a good compromise between immersivity and a smooth UX. Users are not forced to excessively move to interact with the surrounding objects, and they can easily enjoy the application without the need of standing for the whole experience duration.

4.4.5 Grab mechanic

In particular one feature that changed in all tasks related to products it is the way users can collect them. In the cardboard version to collect product it was necessary to select them by looking and interacting with them, then after choosing on a menu if taking or leaving the product the action was performed automatically by the application that was moving the product inside the cart (or back to the shelf, depending on the user choice). In the Oculus Quest version instead pointing with the laser to a product and interacting with it will make the user grab the selected product (4.30). When the user is grabbing a product the controller that selected it is no longer visible, it is substituted by the product, and it is not possible to use it to interact with other objects until the product is left. Now, when users move the controller that is holding the products its rotations and translations will be replicated by the product. This gives users the sensation they are really moving the object that they are holding inside the virtual environment. While holding the product pressing again the interaction button will leave it where it is, making it fall to ground if nothing stops it. After leaving a product that was grabbed the controller that became not visible returns to be visible, the laser pointer is enabled again, and it can be used to interact again with other objects.

An important value added by this mechanic is a more immersive products interaction, the way users handle products is more similar to real life compared to the cardboard version. Also, now products have a believable physical behavior, since they can fall to the ground or in the cart, they can bounce on each other while together inside the cart.



Figure 4.30. User holding a product with the right controller - Oculus Quest version

4.4.6 Task changes

The Oculus Quest version provides a more immersive and deeper way to interact with the virtual environment, this consequently brought to a re-design of some tasks to

fully exploit the device potentialities. The tasks that changed with respect to the cardboard version are the followings:

- **Take Products:** the main change comparing this task with the respective one in the cardboard version is that now products cannot be put automatically by the application inside the cart, users have to grab products, move the grabbed product to the cart position and leave it inside the cart. This time users can fail the task since if they leave the product out of the cart it will be automatically re-positioned on the shelf. Once a product is put inside the cart it is considered collected, and it will remain inside the cart until the *Place Products On Tape* task. Actually, if users wants to remove a collected product it is possible to do it using the same menu described for the cardboard version. This task is been re-designed to reduce the application intervention and to make it more immersive exploiting Oculus Quest features.
- **Produce Products sub-task:** this task uses the grab mechanic for both products and bags. The bag and gloves must be collected to start the task, gloves can be collected as in the cardboard version, but the bag now exploits the grab mechanics. Once the bag it is grabbed, it substitutes the controller that grabbed it, then the other controller can be used to select the desired product. After selecting the product, it will be put inside the bag(4.31). This process emulates the real-life shopping in produce department, when a customer is holding a bag in one hand, and they take the desired product using the other free hand. Once the product is in the bag it must be placed on the scale to be weighted. If the bag is placed outside the scale plate it can be grabbed again to then place it correctly on the scale. After weighting the bag it can be put inside the cart.
- **Place Products On Tape:** also this task now uses the grab mechanic to move product from cart to the tape. If users fail to place the products on the cash register tape, they will be automatically put again inside the cart. The task will end when all products have been put on the tape.



Figure 4.31. User holding a bag with the left controller - Oculus Quest version

Chapter 5

Implementation

Once both requirements elicitation and design were over, we started to work at the implementation of the solution.

The result consists of two version of the same VR application that support two devices:

- Android devices coupled with Cardboard headsets.
- Oculus Quest devices.

The developed application provides users with a VR experience to help them to become more independent in doing shopping in a supermarket.

This chapter dive deep into the used tools, libraries and plugins, the architecture and the implementation of some peculiar functionalities.

5.1 Unity3D

5A Supermarket experience has been created using Unity3D, a popular and powerful game engine developed by Unity Technologies (see Fig. 5.1). Unity support functionalities of 2D and 3D game development and offers a primary scripting API in C#, for both the Unity editor in the form of plugins, and the games themselves. Additionally, the engine supports specification of texture compression, resolution settings and mipmaps for each platform supported and allows functionalities of reflection, bump and parallax mapping, as well as dynamic shadows, Screen Space Ambient Occlusion (SSAO) and full-screen post-processing effects.

The reasons behind the choice of Unity as game engine are the following:

- **Unity wide adoption:** According to Unity Report 2021 [13], Unity remains the game engine of choice with 61% of developers surveyed using it, and the second most popular choice as a mobile ad network. Moreover, 94 of the top 100 development studios by global revenue are Unity customers and there are around 2.8B monthly active end users who consume content created or operated with Unity solutions. This numbers makes clear that Unity Technologies are highly supported and with a wide range of libraries and useful plugins.
- **Free tier plan:** it offers the Core Unity real-time development platform and it is eligible for companies or organizations with revenue or funding lower than

\$100K in the last 12 month.

It provides:

- all core features to create and optimize high-end games and interactive experiences;
 - build once and deploy across all supported platforms with Unity’s industry-leading multi-platform support;
 - continuous access to tech updates, improved performance, and new platforms.
- **VR support:** Unity supports multiple platforms for VR development and recommends to use its built-in XR Management to load and manage the target platform SDKs. Unity supports Oculus, Windows Mixed Reality and third parties ones as Google VR and OpenVR.

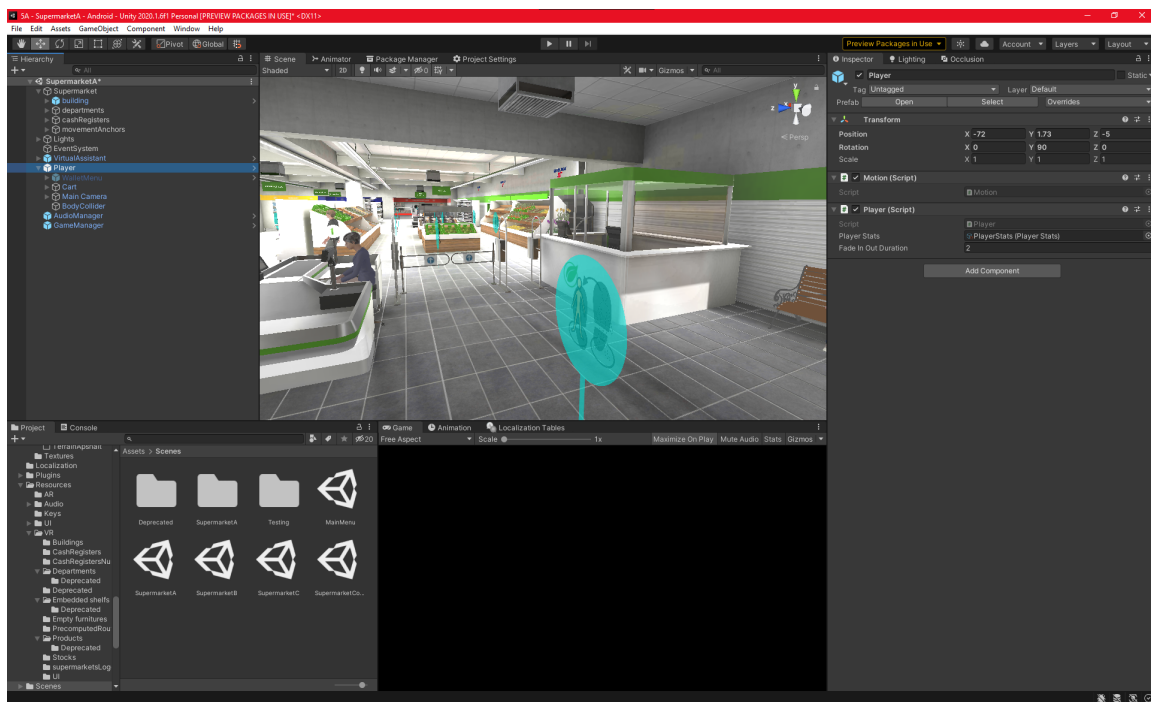


Figure 5.1. Unity Editor

5A Supermarket Cardboard has been build with Unity version 2020.1.6f1, while for 5A Supermarket Oculus Quest we used a more recent version, 2020.3.8f1.

5.2 Libraries & Plugins

The aim of this section is to give a brief explanation of most important libraries added to Unity project.

5.2.1 Unity Technologies

Localization

Localization is a Unity preview package that provides tools for adding support for multiple languages and regional variants to applications [14]. It doesn't just support multi-languages text(see Fig. 5.2), but also culture-specific Assets of type Texture and Audio. It is compatible with *2018.3* and later versions of Unity Editor. Even if 5A is an Italian project that will first run in Milan, we decided to lay the foundation for a polyglot application with a centralized localization repository.

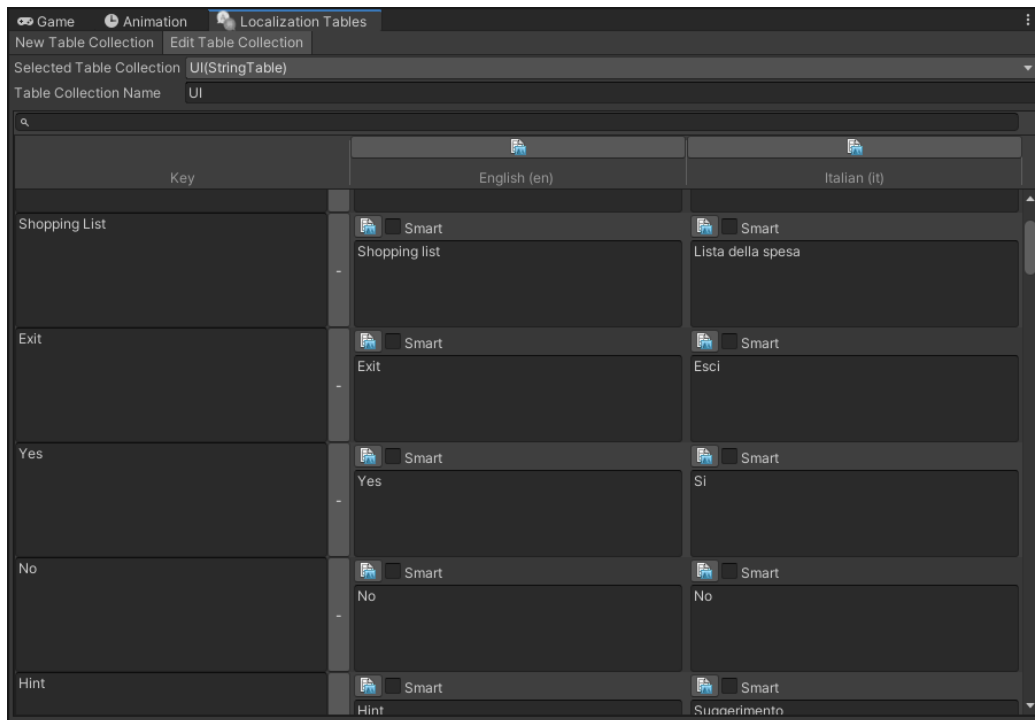


Figure 5.2. Localization repository for text

Used version: 0.10.0-preview - January 26, 2021

XR Plugin Management

Unity package dedicated to management of other XR plugins [15]. This package has been used to retrieve other XR packages compatible to used Unity version.

Used version: 4.0.5 - May 7, 2021

XR Interaction Toolkit

Unity preview package that implements a generic VR interaction system, independently by the target VR device [16]. This interaction system provides a variety of possible ways to make user interact with the virtual environment. This package is based on common mechanics employed in VR applications.

This package has been used only for Oculus Quest version development.

Used version: 1.9.1 - March 19, 2021

5.2.2 Other

Google Cardboard XR Plugin for Unity

This SDK provides everything is needed to create a VR experience for Google Cardboard in Unity *2019.4.25f1* or later [17]. It allows to turn a smartphone into a VR platform by making it displays 3D scenes and supports the features VR essential features:

- motion tracking;
- stereoscopic rendering;
- User interaction via the viewer button.

This package has been used only for cardboard VR version development. **Used version:** 1.4.0

XR Oculus Plugin

Package that provides a communication interface between Oculus devices and Unity [18]. This package allow to collect user's input from the Oculus device. Together with XR Interaction Toolkit it allows user to interact with the virtual environment through the Oculus device.

This package has been used only for Oculus Quest version development.

Used version: 1.0.0-pre.3 - May 11, 2021

Microsoft Speech SDK

The Microsoft Speech Software Development Kit (SDK) exposes Speech service capabilities to empower the development of speech-enabled applications. The .NET SDK, based on .NET Standard 2.0, is used since 5A Supermarket is written in C#. To be specific, we used it to enable Speech-to-text (also known as speech recognition) capability which transcribes audio streams into text.

AWS SDK

AWS SDK for .NET simplifies use of AWS Services by providing a set of libraries that are consistent and familiar for .NET developers. All AWS SDKs provide support for API life-cycle consideration and higher level abstractions. In terms of AWS services, I used the following:

- **Amazon Cognito** to enable access control to web and mobile apps quickly and easily. To be specific, an Amazon Cognito identity pool was created to provide temporary AWS credentials for users who are guests (unauthenticated) and need to use AWS services.
- **Amazon Polly** Text-to-speech capability which text into lifelike speech.

To let AWS SDK for .NET work with scripting back-end Intermediate Language To C++ (IL2CPP), I added to the project the following xml:

```
<linker>
  <assembly fullname="AWSSDK.Core" preserve="all"/>
  <assembly fullname="AWSSDK.Polly" preserve="all"/>
    <assembly fullname="AWSSDK.CognitoIdentity" preserve="all"/>
    <assembly fullname="AWSSDK.SecurityToken" preserve="all"/>
</linker>
```

5.3 Architecture

As said at the beginning of this chapter, 5A Supermarket architecture is formed by a fat client. The logical components can be found in the client (see 5.3) and in this section we list the most important ones:

- **UI Manager:** singleton component for 2D starting menu. It is responsible for managing the level selection and for handling the generation of a one shot custom level. There exist specific functions for each configurable parameter is associate to a *UI Manager* function which stores the new data into a temporary data model and triggers related UI changes.
- **Audio Manager:** general components for sounds and audio effects, it is responsible for loading the audio files from the source folder of the project and providing other components with an interface to play music and sound effects when needed
- **VR Game Manager:** singleton component in which the following VR state parameters are stored:
 - *current FSM step:* keep track of the current task that has to be performed (see 4.1.2)
 - *current experience configuration:* every configuration parameter about the experience is stored in this component (see 4.2).
 - *required products:* products that user need to collect to complete the *Take Products* task (see 4.1.3).

It manages the activation of other components when a new main task or a warning arise.

- **Virtual Assistant Manager:** component that keeps track and manages most active and interactive UI components into VR experience and loads from localization and plays the audio resources of static virtual assistant conversations. This component also manages some Virtual Assistant behaviors.
- **UI Object Managers** some UI objects are not managed by Virtual Assistant Manager:

- *HintMenu*: it's manager handles the load of current step PCS and text and the activation of interactive buttons in it.
 - *WarningMenu*: it's manager handles the load of current warning and the activation of interactive buttons in it.
 - *WalletMenu*: it's manager handles the virtual wallet on the basis of the configured payment mode.
- **Player**: component responsible for managing the object that represents user in the virtual environment. It contains the current state of the user.
 - **Interaction components**: group of components that handles user interaction with other virtual objects. Depending on the application version, these components can be very different since they are linked to the device inputs (see 5.5, 5.6).
 - **Input Managers**: set of components dedicated to process user input. Some of these components differ in the two application versions because they are related to the device.
 - **Game Objects**: generic category that includes all objects with in-game functionalities.

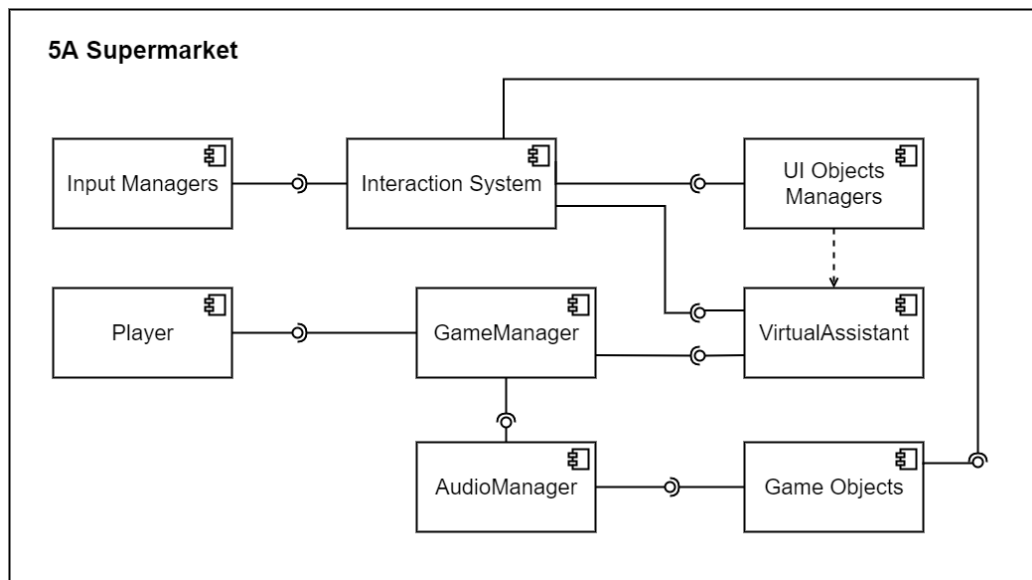


Figure 5.3. Component diagram showing the general architecture of the application

5.4 Configurable Scene

As already described in the design chapter, scenes planimetry can be generated at run-time reading a configuration file. The configuration file is a JSON file that can be retrieved remotely by the application (see 6.3). This configuration file currently can be generated using a utility script inside the project, but this is a temporary solution to test this application feature. In this section some scripts will be shown to explain how this feature has been implemented.

5.4.1 Data Model

The JSON file is based on the data model shown at 5.1. The root of the data structure is the class *SupermarketScene*, from it, it is possible to reach every other class in the data model. It is possible to distinguish between two groups of objects inside the data model, supermarket objects, i.e. *activity related objects*, and *Unity related objects*. Examples of classes representing the first group of objects are *Department*, *Shelf* and *CashRegister*. These are all custom objects created ad hoc for the activity. The other group of classes represents base objects defined by Unity to be part of the scene structure. Examples of Unity related objects in the data model are *CustomVector3* and *CustomTransform* classes, that respectively represent the *Vector3* and *Transform* Unity objects, together used to represent the position, rotation and scale of objects in the scene space. *Obj* class is a Unity related object that represents a prefab instance. All together this classes represents the configurable part of the scene. Every field in the relative JSON file can be edited to modify position, quantity, and state of the listed objects. The diagram 5.4 shows how this classes are linked in the data model, stressing the difference between *activity related objects* and *Unity related objects*, objects that appear as single field and list of objects.

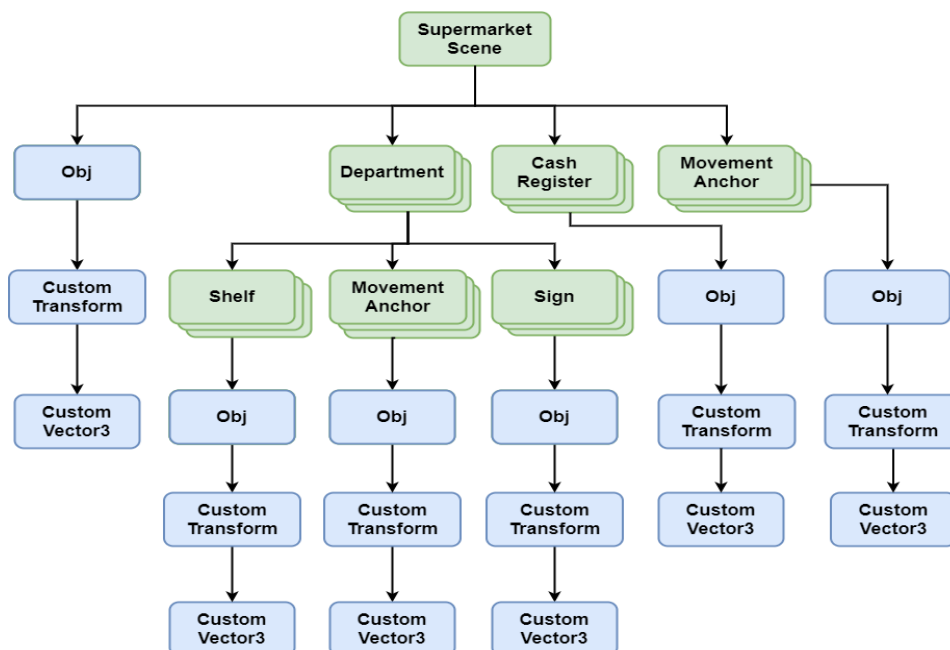


Figure 5.4. Data model diagram. Activity related objects in green, Unity related objects in blue

```
public class SceneDataModel
{
    /// <summary>
    /// Encodes a standard Vector3.
    /// </summary>
    [Serializable]
    public class CustomVector3
    {
        public float x;
        public float y;
        public float z;
    }

    /// <summary>
    /// Encodes a standard Transform.
    /// </summary>
    [Serializable]
    public class CustomTransform
    {
        public CustomVector3 position;
        public CustomVector3 rotation;
        public CustomVector3 scale;
    }

    /// <summary>
    /// Contains the info of a scene object.
    /// </summary>
    [Serializable]
    public class Obj
    {
        public string name;
        public string prefabPath;
        public CustomTransform customTransform;
    }

    /// <summary>
    /// Info about a movementAnchor
    /// </summary>
    [Serializable]
    public class MovementAnchor
    {
        public Obj instance;
        public float cartRotation;
    }

    /// <summary>
    /// Info about a shelf
    /// </summary>
    [Serializable]
    public class Sign
    {
        public Obj instance;
```

```

    public string localizationReference;
}

/// <summary>
/// Info about a shelf
/// </summary>
[Serializable]
public class Shelf
{
    public Obj instance;
    public bool isEmbedded;
}

/// <summary>
/// Info about a supermarket department
/// </summary>
[Serializable]
public class Department
{
    public string name;
    public CustomTransform customTransform;
    public List<Shelf> shelves;
    public List<MovementAnchor> movementAnchors;
    public List<Sign> signs;
}

/// <summary>
/// Info about a cash register
/// </summary>
[Serializable]
public class CashRegister
{
    public Obj instance;
    public int num;
}

/// <summary>
/// Info about the scene of the supermarket interior
/// </summary>
[Serializable]
public class SupermarketScene
{
    public Obj building;
    public List<Department> departments;
    public List<CashRegister> cashRegisters;
    public List<MovementAnchor> movementAnchors;
}
}

```

Listing 5.1. Data model for the JSON scene configuration file

5.4.2 Run-time scene generation

The class responsible for the run-time scene generation is the VRSceneConfiguration-Manager. This class takes as input the scene configuration in as JSON string format, it parses it creating an object containing the whole data model, then it passes each portion of the data model to functions dedicated to instantiate their content (see 5.2).

```

/// <summary>
/// Reading the configuration json file of the scene. Configurable gameObjects in the
/// scene are ///instantiated, initialized and positioned.
/// </summary>
/// <param name="jsonText">scene configuration json</param>
private void LoadSceneConfiguration(string jsonText)
{
    var supermarketScene = JsonUtility.FromJson<SupermarketScene>(jsonText);
    InstantiateObj(supermarketScene.building, configurationRoot);
    InstantiateList(supermarketScene.departments, configurationRoot, "departments",
        InstantiateDepartment);
    InstantiateList(supermarketScene.cashRegisters, configurationRoot, "cashRegisters",
        InstantiateCashRegister);
    InstantiateList(supermarketScene.movementAnchors, configurationRoot, "
        movementAnchors", InstantiateMovementAnchor);
}

```

Listing 5.2. Load scene configuration main function

Each "instantiate" function creates one or more gameObjects in the scene. The hierarchical structure shown in 5.4 is preserved passing a parameter "parent" to each "instantiate" function; as soon as the object is instantiated it will be set as child of the specified parent.

An interesting function is the one dedicate to create generic list of objects. To properly work this function needs both a lsit of objects and a callback function that will be called to instatiate each element in the list (see 5.3).

```

/// <summary>
/// Instantiate a generic list of gameObject contained in the json
/// </summary>
/// <param name="list">list from json obj</param>
/// <param name="parent">parent Transform for the list</param>
/// <param name="listName">name to give to the list</param>
/// <param name="instantiateListEl">callback to instantiate each list elem</param>
private void InstantiateList(IEnumerable<object> list, Transform parent, string listName,
    InstantiateListEl instantiateListEl)
{
    var pos = Vector3.zero;
    var rot = Quaternion.identity;
    var instance = Instantiate(new GameObject(),pos,rot);
    if(parent!=null) instance.transform.SetParent(parent);
    instance.name = listName;

    foreach (var obj in list)

```



```

    {
        instantiateListEl(obj, instance.transform);
    }
}

```

Listing 5.3. Instantiating a list of generic objects

Functions dedicated to instance activity related objects often set some values to the initial state of the created object. For example the cash register number is configurable, and it is set after that the object is instantiated (see 5.4).

```

private GameObject InstantiateCashRegister(object genericCashRegister, Transform
parent)
{
    var cashRegister = (SceneDataModel.CashRegister) genericCashRegister;
    var instance = InstantiateObj(cashRegister.instance,parent);
    instance.GetComponent<CashRegister>().cashNumber = cashRegister.num;
    return instance;
}

```

Listing 5.4. Instantiating an activity related object

The last function that is important to see in detail is the one dedicated to the instance of prefab objects. As shown in 5.5, the prefab path inside the project folder is retrieved from the JSON file, and it is used to choose which prefab to instantiate. If several variants of the same prefab are present in the project they can be referenced in this way in the configuration file. This feature allows to create at run-time a large variety of different supermarket, with variations of the key objects. For example introducing a new type of automatic cash register does not need any modification to the data model, since it will be enough to adjust the path to the new cash register prefab. Currently, in the project there is only the base prefab for each activity related object.

```

/// <summary>
/// Instantiate a generic prefab
/// </summary>
/// <param name="genericObj">obj to instantiate from json obj</param>
/// <param name="parent">parent transform to be set for the prefab that this function
    is going to instantiate</param>
/// <returns></returns>
private GameObject InstantiateObj(object genericObj, Transform parent)
{
    var obj = (SceneDataModel.Obj) genericObj;

    if(string.IsNullOrEmpty(obj.prefabPath)) return new GameObject();

    var path = obj.prefabPath.Substring(17,obj.prefabPath.Length - 24);
    var pos = CustomVectorToVector3(obj.customTransform.position);
    var rot = Quaternion.Euler(CustomVectorToVector3(obj.customTransform.rotation));
    var scale = CustomVectorToVector3(obj.customTransform.scale);
}

```

```
var instance = Instantiate(Resources.Load(path) as GameObject, pos, rot);
if(parent != null) instance.transform.SetParent(parent);
instance.transform.localScale = scale;
instance.gameObject.name = obj.name;

return instance;
}
```

Listing 5.5. Instantiating a generic prefab, a Unity related object

5.4.3 Limitations

The main limitation linked to run-time scene generation is related to lighting system. For performance reasons, scene with many 3D models use pre-computed lighting, where the effect of light on materials, shadows cast by objects and reflections are computed before building the application, and then saved in files linked to the scene. When the scene is loaded at run-time lighting effects are loaded the scene related files. This approach is not possible for scenes that are generated at run-time since the scene planimetry and objects are not known before running the application.

The solution adopted to overcome this issue is simplifying the lighting system, that substantially consists of a fixed ambient light. There are not self-shadows, shadow cast or reflections inside the configurable scene (see 5.5).



Figure 5.5. Configurable scene generated at run-time with ambient light illumination

5.5 Cardboard Interaction System

The package [17] used for cardboard VR does not add specific components designed for building an interaction system (as instead XR Interaction Toolkit does [16]). This forced me to implement components to realise an interaction system based on cardboard VR inputs. This system consists of two main components:

- **Camera Pointer Controller:** main component that manages interactions. It is linked to user's camera, and it comprehends user's field of view. This field of view is a Unity collider with a cone shape, orientated as shown in 5.6; it is linked to main camera movements and rotations, hence it rotates together with user's head. Camera Pointer Controller detects each object that enters and exits the field of view in order to understand at which objects user is looking. It is able to find the closest interactive object (not covered by an obstacle) among the object inside the field of view, and to trigger an interaction with it depending on the selected interaction mode among *Gaze & commit* and *Gaze dwell*. It also detects when user's presses the cardboard gear button from the Google Cardboard XR Plugin.

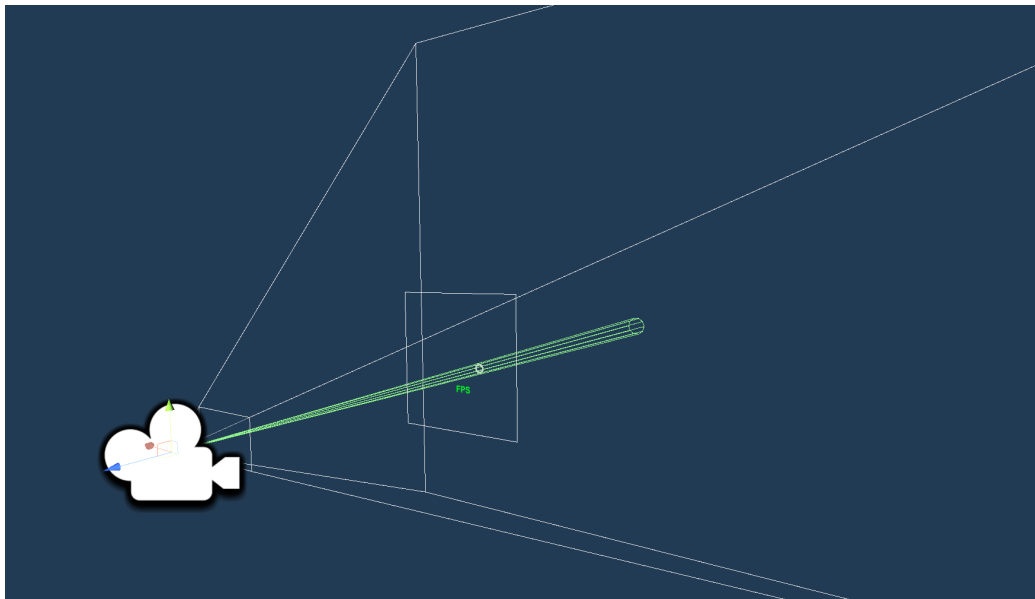


Figure 5.6. Camera Pointer Controller field of view in green. Main Camera field of view in white.

It is important to stress that Main Camera field of view is not Camera Pointer controller field of view. The first one represent the area that user can see, the latter is a smaller area, orientated to fit the UI pointer, whose purpose is to detect which objects user is considering for an interaction.

Camera Pointer Controller can properly detect interactive object and trigger an interaction with them only if the interactive objects posses a Gaze Controller component.

- **Gaze Controller:** this component allows Camera Pointer Controller to recognize interactive objects and to interact with them. When an interaction

is triggered by Camera Pointer Controller, it calls the pointed object's Gaze Controller and a Unity event function stored in it is called. Gaze Controller's Unity event function is used to activate different components depending on the interaction result.

Gaze Controller component also stores a tag to group interactive objects in categories. Interactive objects are not enabled for the whole duration of the experience, but when they are interactable or not depends on the task that user is carrying on, and on the Player component state (see 5.3). These tag groups for interactive objects have been implemented to focus user's attention only on objects that are relevant in the current phase of the experience.

Other components with minor tasks are involved in the system and dialogue with the two main ones. An example of this components is Camera Pointer UI, a class that handles the UI pointer shown at the centre of the view, its color and shape, to produce interaction feedbacks when an interactive object is pointed.

Since this interaction system is not based only on a simple ray cast in the direction user is looking at, it is possible to interact with objects even if they are not fully centered with the UI pointer, as show in 5.7 and 5.8). The interaction system has been implemented in this way to help users with difficulties in centering the object inside the pointer.



Figure 5.7. Camera UI pointer centered on the interactive object. It is bigger than default to inform user that they are pointing an interactive object.

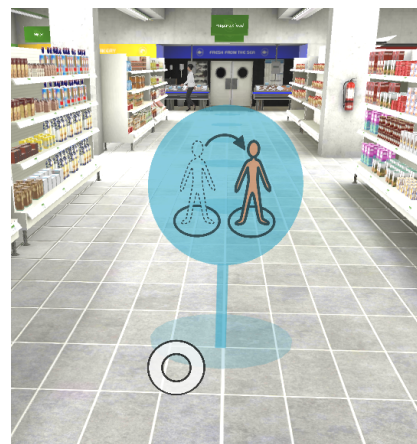


Figure 5.8. Camera UI pointer not centered on the interactive object. It is bigger than default to inform user that they are pointing an interactive object.

5.6 Oculus Quest Interaction System

For 5A Supermarket Oculus Quest most of the interaction system has not been implemented since it is based on XR Interaction Toolkit [16] components. This section is focused on the XR Interaction Toolkit components used in the application, and on the main differences with the cardboard version interaction system.

Main Components

XR Interaction Toolkit component is still in development, however it is already accessible to Unity users as a preview package, and, even if it is still not officially released, it offers a big variety of components to implement many common VR applications features. Its main components are:

- **XR Input Manager:** it detects all user's input from the device. Others components consequently react when the XR Input Manager detects a specific input they are waiting for.
- **Interactor components:** components dedicated to trigger interactions. Depending on the input configuration, when an input action related to a specific interactor is detected, the relative interactor will trigger the interaction. For example there is an interactor dedicated for laser interactions, when an interactive object is pointed by the laser and the input key associated to the laser interactor is detected, it is possible to trigger the object interaction with the help of other components.
- **Interactable components:** components that can be interacted by interactors. Different type of interactables components allow different behaviors for the interacted object.
- **XR Interaction Manager:** it handles all interactions. It is the core component responsible for interactions since it mediates the communications between interactor and interactable components.
- **XR Rig:** one of the biggest component in the system. It comprehends several sub components. It represents user's inside the scene, it handles user's camera, controllers, and movements. Some of its sub-components are:
 - *Track Pose Driver:* it detects device/user's head rotations and translations, and it correctly adjust the Main Camera orientation inside the scene.
 - *Teleportation Provider:* component responsible for user's teleport movements.
 - *Snap-Turn Provider:* component responsible for XR Rig rotation over Y axis, in other words it allows user to turn right/left using the controllers sticks. To avoid motion sickness this rotation is not continue, but the minimum rotation that can be performed is of a 45° range.

5.7 Comparison Between Interaction Systems

The two interaction systems described for 5A Cardboard and Oculus Quest versions are deeply different, even if some design concepts are shared between them. Cardboard interaction system is based on two main components, that can be compared to other XR Interaction Toolkit components:

- Camera Pointer Controller assumes a role similar to XR Interaction Toolkit interactors.
- Gaze Controller assumes a role similar to XR Interaction Toolkit interactables.

However this parallel isn't correct, since XR Interaction Toolkit interactors and interactables do not communicate directly as it happens for Camera Pointer Controller and Gaze Controller, but they depend on the Input Action Manager as intermediary. In fact, the proper way to trigger an interaction from a script, using XR Interaction Toolkit, is calling a proper function from XR Interaction Manager passing it a reference of the interactor and the interactable involved in the interaction. The logic handled by the XR Interaction Manager for the Oculus Quest version is distributed among the Camera Pointer Controller and Gaze Controller components, that directly communicate to handle user's interactions.

XR Interaction Toolkit provides a richer interaction system compared to the one implemented for 5A Cardboard, it takes into account the need of different type of interactions, and it is based on a Model-Controller logic, as described before. 5A Cardboard interaction system is simpler, as the device that it supports is simpler compared to Oculus Quest (from an interaction point of view). The restricted input actions available make easier the creation of a custom interaction system for the application when talking about cardboard VR, while when dealing with VR devices able to detect more sophisticated user's input a third party input system is the best choice in most cases.

Chapter 6

Conclusions

6.1 Limitations due to Covid-19

Unfortunately, due to Covid-19, it has been impossible to test the application with end users. The testing phase has been pursued thanks to the laboratory. The absence of testing with the target users forced us to relate only on therapist feedbacks.

These limitations also prevented us from conducting a proper usability evaluation of the two developed applications. We have not been able to appropriately assess the impact of our design choices on the target users, especially we cannot determine if there is a preferred device between the cardboard VR headset and Oculus Quest, or which are the peculiarities of the two devices that have a strong impact on the end user experience.

In this chapter we will focus on the overall result of the project from a qualitative point of view.

6.2 Contribution

The main contribution of this project consists of two applications devoted to help ASD users in acquiring daily life autonomy. These applications take part to 5A's big group of projects, and covers one of the most important commercial services scenarios.

Another important contribution of this thesis is a deep comparison between the characteristics of two different HMD, Cardboard VR headsets and Oculus Quest devices. This comparison can be split into two main sections:

- The impact that the different potentialities and limitations of the two devices had on the design of the experience.
- the difference between the interaction systems of the two devices from an implementation point of view.

6.2.1 5A Supermarket

5A Supermarket offers a variety of features to its end users, from a completely configurable experience, to localized UI and dialogues. The main results achieved by the application are:

- A complete experience that guides users from the beginning of the shopping to the end leaving them a certain level of freedom and to mistake. The experience covers the shopping from the products selection to the payment, to the cart return. The experience takes into account the existence of different types of products, providing more complex interactions to collect products from the produce department.
- Several aspects of the experience are configurable. To meet the needs of the end users affected by ASD, characterised from a strong heterogeneity, several application functionalities depends on variables that can be freely customized by the end users, by therapists or their caregivers.
- Configurable planimetry for the activity environment. To let user access an endless variety of possible supermarkets, the scene were the activity takes place can be remotely configured with a deep level of customization. This feature also makes the application easily extendable to variation of the already existent activity objects, there is no need to create a new scene with the new objects, they can be simply added to the project and then loaded at run-time.
- A fully localized experiences, that supports two different languages, Italian and English, for both UI and dialogues with the Virtual Assistant. The localization not only affects texts and audio, it also influences the PCS shown when listening to the Virtual Assistant.
- Virtual Assistant able to perform a static-conversation on the basis of user's actions. The Virtual Assistant exploits three learning channels, auditory, visual and verbal, to communicate with users. This is a key feature linked to typical lack of focus of ASD users, we cannot know which of these three learning channel is the preferred by users, so we let them choose which enable and which not. The choice to let all three channels active by defaults when grants the major possibility that users will listen and understand what the application wants to communicate.

6.2.2 Cardboard and Oculus Quest versions discussion

From a design view point the two version present several changes in the activity tasks. This changes were necessary to fully exploit the potentiality of both devices. The next table (see 6.1) shows a recap of the main differences of the two applications from a design point of view.

Table 6.1. Main design changes in the two version of the application

5A Supermarket Cardboard	5A Supermarket Oculus Quest
Configurable movement, two supported modes, teleport and continuous	Movement is not configurable, the only mode available is teleport (to avoid motion sickness)
Configurable interactions, Gaze & Commit and Gaze Dwell modes to support all cardboard headsets	Interaction is not configurable
Interaction mechanic based on gaze, users mostly move their head to interact with objects	Interaction mechanic based on controller laser pointer, users interact with objects using their hands
Same interaction for movement anchors and other interactive objects	Two different interaction laser/ray based for movement anchors and other interactive objects
Rough grab mechanic limited to some activity tasks to move objects from a starting point to an ending point	Grab mechanic used in most tasks that requires to handle/move products
Simple products physical behavior	Realistic products physical behavior
Products are automatically moved inside cart when user decides to collect them	User need to move products from shelves to cart using the grab mechanic
Produce products can be collected with a standard sequence of base interactions	Produce products require the use of both controllers to be collected
Products are automatically positioned to their destinations when user needs to move them, it is not possible to fail this action	Users can fail while positioning products into the cart or on the cash register tape
Users can observe the scene only from pre-defined points defined by movement anchors	Users can have a wider vision of the scene, not limited to what can be seen from movement anchors positions because they can also translate their head in the virtual environment

After testing the two application versions we concluded that:

- Oculus Quest version offers a more immersive experience thanks to the major number of user inputs and movements that can be tracked. It is able to offer more realistic and sophisticated interactions, to better reproduce the experience

of a real shopping in a real supermarket. The better immersivity provided by Oculus Quest can focus well user attention on activity tasks, and isolate them from external interference.

- Oculus Quest version interactions, though more immersive, are also more difficult to perform, and they can fail. Users are required to move the grabbed objects with a certain accuracy to correctly put them inside the cart, to place them on the cash tape or a scale plate. If users are not able to place the object to the destination they have to grab them and try again. This could raise some issues depending on the users abilities. Cardboard version still moves itself objects in most cases, and when users need to move them they cannot fail, the grabbed object can only be placed on the destination area. This difference could make more attractive for some users the Cardboard version, whose tasks could be easier to complete.
- The way Oculus Quest experience is designed forces users to move more compared to the cardboard version. In Oculus Quest version users not only move their heads, but also their hands to interact and move objects in the virtual environment. Even if both applications can be used while sit (according to have a swivel chair), Oculus Quest requires users to move frequently their arms, and this could get them tired if training sessions are long, or maybe because users wants to dedicate time to VR training at the end of the day. This is a known issue, especially in VR gaming where developers need to correctly balance between immersive mechanics and weary mechanics.

From this comparison it is clear that there is not a winner between the two devices. Both devices showed their strengths and weaknesses, and which device can produce better results in ASD treatment depends on the user abilities. A user with a very strong difficulty in focusing, that gets easily distracted would probably benefit from the Oculus Quest strong immersivity, while users that find more difficult to handle objects in a virtual environment would probably prefer the cardboard headset. How many users belong to the first case, and how many to the latter, cannot be inferred, it needs a proper usability evaluation to be studied.

It important to highlight that the Oculus Quest version could implement the same interaction system of the cardboard version (the opposite is not true). However, it would be a waste of Oculus Quest potentialities, since, as stated before, an Oculus Quest is far more expensive then a cardboard headset, it is crucial to chose the right device on the basis of the desired interaction mechanics.

6.3 Application limitations

At the end of the development we noted that the 5A Supermarkets has still some limitations, some linked to design choices, some others to technical reasons. In this section we will list the main limitations that affect the application:

- 5A Supermarket Cardboard supports Android devices, this means that the target set of devices is very heterogeneous, especially when talking about performance. We registered very different performances depending on the used device. Table 6.2 shows the suggested hardware requirements to run the application.

Table 6.2. 5A Supermarket Cardboard suggested hardware requirements

Hardware Requirements	
Processor	4 x 2.8 GHz Kryo 385 + 4 x 1.8 GHz Kryo 385
GPU	Adreno 630

It is still possible to improve the application performances, but it is a long task since it would require to work with several 3D models. The number of triangles of several 3D models is high, it could be reduced using an appropriate tool. To do this improvement it is necessary to export each model with an excessively high number of triangles, import it in a 3D graphic tool to reduce the number of triangles, export it from the tool and then import it again in Unity. This procedure is easy for just one 3D model, but it could be very long when the project has hundreds of models.

- Scene configuration allows to customize the position and type of shelves, but not what there is inside, it is not possible to customize products positions inside shelves. Products will always be at the same level of the shelf they belong. For example, if a product is at the top of a shelf it will be always at the top of it, even if the shelf changes position in the scene.
- When the scene is generated at run-time from a configuration file it is not possible to pre-compute lighting effects since the position of each object in the scene is not known before running the application (see 5.4). This results in a scene whose lighting is based only on ambient light, with less realistic impact.
- Both experience and scene configurations can be downloaded remotely. A service dedicated to register and provide configurations 5A applications has not been developed yet. To test the application some default configuration files have been uploaded to an external service [19] that works as a simple JSON files online repository.
- Currently, Localization package does not work on Oculus Quest version. Unity Localization is a preview package, so it is still in development, and it is still not able to support Oculus Quest device.

6.4 Future improvements

5A Supermarket can still be improved in several ways to make the experience richer and more various. Furthermore, 5A Supermarket is developed to fit a shared environment of services among all 5A applications. Some of these services are already set up, like Microsoft Azure Speech To Text and Amazon Polly Text To Speech, but, other services can still be added to enhance the experience.

Some of the possible future improvements can be:

- A web app to easily design planimetries for the supermarket scene, and create scene configuration files to send to the application. One good way for generating the scene configuration file could be creating a dedicated web app, whose stakeholders will only be therapists or caregivers. This app could let them configure the planimetry of the supermarket through a simple UI.
- The overall experience of a shopping can be too long for some users, while others may desire to focus only to a specific task of the application. For these reasons an other possible parameter that can be added to the experience configuration can be the starting phase of the experience. This would avoid to force users to execute every time all tasks.
- To support dynamic and context-aware conversation with the Virtual Assistant also in 5A Supermarket, Rasa server will be properly trained.
- Currently, 5A Supermarket Oculus Quest does not support Speech To Text service and it is not linked to Rasa server. To align the Oculus Quest version with other 5A applications these services should be implemented.
- Other type of products can be added to the supermarket, with their related tasks subsets to be collect.
- Shopping list is randomly generated at the beginning of the experience. Through the experience configuration menu it is possible to partially control this random generation (produce products shelves can be enabled/disabled). It could be a useful feature to let user customize the content of the shopping list by choosing each single product.

Appendix A

Acronyms

5A	Autonomie per l'Autismo Attraverso realtà virtuale, realtà Aumentata e Agenti conversazionali
AR	Augmented Reality
ASD	Autism Spectrum Disorder
DoF	Degrees of Freedom
FSM	Final State Machine
HCI	Human Computer Interaction
HMD	Head Mounted Device
IL2CPP	Intermediate Language To C++
NLU	Natural Language Understanding
NGR	Needs Goals and Requirements
PCS	Picture Communication System
SDK	Software Development Kit
SSAO	Screen Space Ambient Occlusion
UI	User interface
UX	User Experience
VAP-S	Virtual Action Planning Supermarket
VR	Virtual Reality

Bibliography

- [1] American Psychiatric Association. *Diagnostic and Statistical Manual of Mental Disorders (DSM-5®)*. en. American Psychiatric Pub, May 2013.
- [2] Eric Courchesne et al. “The ASD Living Biology: from cell proliferation to clinical phenotype”. en. In: *Mol. Psychiatry* 24.1 (Jan. 2019), pp. 88–107.
- [3] Ali Adjorlu et al. “Daily Living Skills Training in Virtual Reality to Help Children with Autism Spectrum Disorder in a Real Shopping Scenario”. In: *2017 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*. Oct. 2017, pp. 294–302.
- [4] Liron Lamash, Evelyne Klinger, and Naomi Josman. “Using a virtual supermarket to promote independent functioning among adolescents with Autism Spectrum Disorder”. In: *2017 International Conference on Virtual Rehabilitation (ICVR)*. June 2017, pp. 1–7.
- [5] M Moore and S Calvert. “Brief report: vocabulary acquisition for children with autism: teacher or computer instruction”. en. In: *J. Autism Dev. Disord.* 30.4 (Aug. 2000), pp. 359–362.
- [6] World Health Organization. “Autism spectrum disorders”. In: ().
- [7] Shahira Sharafeldin and Hadeel El Naggar. “Potential of Virtual reality in special needs higher education”. In: July 2018.
- [8] Lal Bozgeyikli et al. “Vocational Rehabilitation of Individuals with Autism Spectrum Disorder with Virtual Reality”. In: *ACM Trans. Access. Comput.* 10.2 (Apr. 2017), pp. 1–25.
- [9] *VR standalone HMD average selling price 2015-2020*. <https://www.statista.com/statistics/682804/vr-standalone-hmd-average-selling-price>. Accessed: 2021-6-24.
- [10] Nikiforos M Papachristos, Ioannis Vrellis, and Tassos A Mikropoulos. “A Comparison between Oculus Rift and a Low-Cost Smartphone VR Headset: Immersive User Experience and Learning”. In: *2017 IEEE 17th International Conference on Advanced Learning Technologies (ICALT)*. July 2017, pp. 477–481.
- [11] J A Wainwright and S E Bryson. “Visual-spatial orienting in autism”. en. In: *J. Autism Dev. Disord.* 26.4 (Aug. 1996), pp. 423–438.
- [12] *XR-Interaction-Toolkit-Examples*. URL: <https://github.com/Unity-Technologies/XR-Interaction-Toolkit-Examples>.
- [13] Arjun Gohil, Emory Irpan, and Nathan Tenboer. *2021 Gaming Report - Unity insights from 2020 and predicted trends for 2021*. Tech. rep. Unity Technology.

- [14] Unity Technologies. *Localization Manual*. URL: <https://docs.unity3d.com/Packages/com.unity.localization@0.4/manual/index.html>.
- [15] *About the XR Plug-in Management package*. <https://docs.unity3d.com/Packages/com.unity.xr.management@4.0/manual/index.html>. Accessed: 2021-7-2.
- [16] *XR-Interaction-Toolkit-Examples*.
- [17] Google LLC. *Cardboard XR Plugin*. URL: <https://github.com/googlevr/cardboard-xr-plugin>.
- [18] *About the Oculus XR Plugin*. <https://docs.unity3d.com/Packages/com.unity.xr.oculus@1.9/manual/index.html>. Accessed: 2021-7-2.
- [19] *JSON Storage & JSON Hosting Service - JSONBin.io*. <https://jsonbin.io/>. Accessed: 2021-7-4.