POLITECNICO DI TORINO

Master in Computer Engineering

Master Thesis

FrejusVR

On the Use of Virtual Reality for Studying Human Behaviors During Emergencies

Supervised by
Prof. Fabrizio Lamberti
Prof. Andrea Sanna
Prof. Romano Borchiellini

Calandra Davide

October 2017

FrejusVR

On the Use of Virtual Reality for Studying Human Behaviors During Emergencies

Calandra Davide

Supervised by:

Prof. Fabrizio Lamberti
Prof. Andrea Sanna
Prof. Romano Borchiellini

Abstract

Virtual Reality (VR), since its appearance, played a big part in the development of more and more effective training tools in comparison with the past. Training simulators could provide the inexperienced user a realistic, immersive and safe Virtual Environment (VE) designed to accurately reproduce a given scenario with the purposes of information, training and evaluation. This work aims at providing a beneficial VR tool with the purpose of the communicating to the general user the emergency procedures concerning a particular scenario, a fire developing from an heavy vehicle inside the Fréjus road tunnel.
Taking advantage of the collaboration with the tunnel Authority and undergoing a tailored testing plan, the work also provides an in-depth data analysis with the purpose of assessing the developed platform.

Acknowledgements

First of all, I want to thank a lot my supervisors, Prof. Lamberti Fabrizio, Prof. Andrea Sanna and Prof. Romano Borchiellini, for the constant support, guidance and availability they provided through the whole development of the work.

Then, I wank to thank a lot SiTI, in the person of Sergio Olivero and his staff Massimo Migliorini, Alessandra Filieri, Valentina Dolci and Francesco Moretti, along with all the other people met over there, for the constant organizational, technological, technical and moral support provided during the months spent with them.

I must also express my gratitude to Michele Billi, for sharing the burden of the development of the work, along with the despair during the most difficult times and the satisfaction for the achieved results. Along with him, I would like to thank all the college colleagues and professors who helped and supported me during the last years; in particular, Fulvio Marsiglia, who collaborated with me on most of the course projects.

Lastly, I would like to thank a lot my family, my girlfriend Alessandra and my friends, for the support provided during this overlong college career, along with the nearness already proved for years.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Safety in road tunnels

The safety in road tunnels is one of the most critical aspects regarding the management of road infrastructures; tunnels are an essential features for countries where road networks have to deal with great oro-geographic obstacles such mountain ranges or water separation. Considering the peculiar tunnel characteristics such presence of confined spaces, lack of non-artificial light sources and difficulties with both intervention and evacuation, the logical consequence is a significant and continuous investment in order to improve security devices, emergency procedures and communication methods for both users and staff.

At European level, the European Directive 2004/54/EC[5] on minimum safety requirements for tunnels in the Trans-European Road Network establishes the mandatory requirements for all the galleries included in the Trans-European Road Network.

Then at national level there are countries, like Italy, with an higher percentage of road tunnels due to morphology of territory; this encourages the administrations of those countries to provide for additional projects or initiatives by their own and, in case of cross-border infrastructures, in conjunction with the neighbouring countries in order to enhance the community measures.

1.2 Motivation

This thesis is the result of the collaboration between Politecnico di Torino and SiTI[1], within the broad context of the PRODIGE[2] project.
The above-mentioned project is a partnership between Italian and French Authorities about the emergency management, realized with the contribution of the European Union and with active partners such the city of Turin, SiTI, the city of Cuneo and the French fire department (SDIS04[3]).

The PRODIGE acronym stands for to protect citizens, to defend the infrastructures, to manage major events; it's an hard task, and the Authorities in charge of the emergency management, such as Civil Defense, Fire Department and rescue teams, have to face it everyday, in a challenging environment created by climatic changes and the growing anthropic pressure over territory and infrastructures.
In order to guarantee the efficacy of the previously-mentioned Authorities, it's mandatory to assure an effective management of the operators, also ensuring the coordination of field teams at both national and cross-border level.
Within this project many Virtual Reality (VR)-based innovative tools have been developed, and now these solutions are available for facilitating the transition to a new approach based upon a mix of technology, management and training.

Four main pilot scenarios were initially developed, each of them characterized by different technical specifications, in order to show the various characteristics of excellence can be put together in VR:

- flooding among the river park Gesso e Stura in the province of Cuneo;
- flash flood and traffic accident near Saint-Paul-Sur-L'ubave;
- spill of chemicals in the area of the Maddalena Pass;
- accident inside an airport with toxic cloud formation running over a crowded area.

FrejusVR is not formally part of the list, but has been though as a possible extension of the above scenarios in the frame of a collabotion between SiTI and the GRAphics and INtelligent Systems (GRAINS) group at the Department of Control and Computer Engineering of Politecnico di Torino.

[1]Istituto Superiore sui Sistemi Territoriali per l'Innovazione, a no-profit founded in 2002 by Politecnico di Torino and Compagnia di San Paolo

[2]PROteggere i cittadini, Difendere le Infrastrutture, Gestire i grandi Eventi

[3]Service Départemental d'Incendie et de Secours

1.2.1 SiTI - Istituto Superiore sui Sistemi Territoriali per l'Innovazione

As said, SiTI is a no-profit, founded in 2002 by Politecnico di Torino and Compagnia di San Paolo, which carries out research and training activities geared towards innovation and socio-economic well-being.
Its activities are directed towards logistic and transport, environmental heritage, urban re-qualification and safety of the territory.

1.3 Objectives

The work done was focused on the realization of a serious game in a VR scenario representing an emergency inside the Fréjus tunnel (Figure 1.1); the hazardous situation in object is a massive fire, caused by a single-truck accident and spread along the tunnel by the truck.
Differently from the four main scenarios developed for PRODIGE, FrejusVR is not operator-oriented; in fact the target audience was moved to the civilians, as non-habitual users of the tunnel. The main goal of the interactive simulation is the communication of the prescribed procedures, in order to safeguard user's security in case of emergency by making them experience the emergency situation and the consequences of their actions.

Among the expected results of the work, there is the evidence of the benefits that VR technology can bring to the emergency management processes regarding the enforcement of security procedures to the civilians, with the goal to minimize the risk for people involved and to maximize the effectiveness of the operations carried out by the Authorities responsible for the intervention.

Figure 1.1: Main Menu of FrejusVR

Chapter 2

Analysis of the Case Study

2.1 Fire in tunnels

When it comes to firefighting, fires in confined spaces are not particularly hard to manage in comparison with other kind of events such large forest fires; however, this assertion loses meaning when it comes to fires inside tunnels.
This kind of events can have different causes, and they can become very hard to manage in case of lacking security measures or users' serious misbehavior.

2.1.1 Causes

The causes of fires inside tunnels, especially road tunnels, are various; in most cases heavy vehicles such trucks are involved, and this amplifies the risk due to their flammable loads. The fire can origin from a malfunctioning of the engine, from a collision between vehicles, from a fuel loss and from external factors too (e.g., a cigarette thrown out of the window).
Once started, the fire temperature starts growing as long as it has fuel and oxygen available; being the tunnel a obstacle for heat dispersion, the temperature can hits peaks of 1000° , melting the asphalt and collapsing the structure of the tunnel itself. For this reason, a tunnel fire can burn for hours and days before becoming reachable for the firefight teams.

2.1.2 Spread

As previously mentioned, the fire can quickly grow till becoming uncontrollable. Firstly, the fire spreads from the point of ignition to the whole vehicle by burning the flammable components.
Then, the high temperature of the produced gasses makes the nearest object burn through the radiation phenomenon; this is how the tunnel fire can subsequently catch one by one every other vehicles, spreading through the tunnel along the wind direction.

2.1.3 Combustion products and effects on the human body

For the whole duration of the fire, many combustion products are continuously produced by the combustion, and those products have a huge impact on the survival of the people involved. There are four main types of combustion products: heat, flames, gas and smoke.

- The heat is the major obstacle when it comes to go through the tunnel in case of fire; the human body cannot resist for more than five minutes with a temperature of 150°, and firefighting protections too cannot guarantee a complete or durable isolation[13]. This, along with the extreme temperatures reached by fires in confined spaces, can isolate whole sections of the tunnel, rendering useless every effort to rescue people inside.

- Flames are heat and light emission sources originated by burning combustible material, they can propagate the fire by contact with other fuel and they're characterized by a color useful to estimate the relative temperature. Most of firefighting protections have a very limited resistance to direct contact with flames (usually seconds).

- The gasses produced by combustion are numerous, and they depend on the kind of fuel involved in the combustion; they can also be extremely toxic and they can contribute to spreading by moving huge amount of heat through the tunnel.

- Smoke, instead, is composed by small unburnt solid particles, which nullify the visibility inside the tunnel and cause severe symptoms to whoever is exposed, such difficulty in breathing and burns of face, skin and respiratory system. Although operators cannot do anything for the visibility, the other effects of the smoke can be prevented by wearing special tools like masks or breathing apparatuses.

2.1.4 User behaviour

In case of emergency, the user if forced to make unusual decisions in condition of time pressure, panic and stress. This could led to wrong decisions, in the for of an incorrect behaviour which can endangering himself and other people.

Among the not recommended behaviours, the user should not.

- neglect the safety brochure provided at the entrance;
- travel through the tunnel with the radio off;
- waste time inside the own vehicle when asked to leave it;
- surpass a road obstacle by invading the opposite lane, especially in case of low visibility;
- surpass a vehicle on fire laying on the opposite lane, creating an airflow directed towards the fire hence providing comburent to the combustion;
- ignore the SOS devices, such buttons or telephones (this was more critical in the past when automatic detection systems were not available);
- get out from a safe area, unless requested by the rescue teams;
- stop the own car too close to other vehicles or too far from the right side;
- stop the vehicle when not specifically requested;
- forget to turn on the emergency lights or leaving the engine on before leaving the own vehicle.

Some of these wrong behaviours played a big part on some of the serious events described in the following section.

2.2 Severe accidents happened in tunnels

As previously mentioned, large fires are the most hazardous situations from the point of view of both users and rescue teams. For this reason, tunnel fire events have always been diffusely studied and analyzed in order to find the factors involved with the origin, the evolution and the exhaustion of the phenomenon.
This kind of events are very numerous, so the research has been focused on a limited set of representative cases of the near past.

2.2.1 Mont Blanc Tunnel (1999)

This event was probably the most severe in recent history in therms of loss of human life. On March 24, 1999 an heavy truck caught fire inside the Mont Blanc tunnel, and the large fire originated from the vehicle caused the death of 39 people.
The truck, loaded with flour and margarine, was travelling from France to Italy.

The driver, after noticing the flames coming from the vehicle, stopped the truck and unprofitably tried to extinguish the initial fire.
The fire, fueled by the truck load and amplified by the tunnel itself, reached an extreme temperature of 1000° which melted down the asphalt and caused the collapse of the ceiling.
Human behaviour also contributed to the worsening of the situation[20]. The operator in charge for controlling the ventilation system started pumping in fresh air in order to help people inside one half of the tunnel; by doing this, he also directed the hot smoke towards the people stuck inside the other half, while at the same time fueled the massive fire with oxygen.
Furthermore, vehicles were allowed to enter the tunnel even after the ignition of the truck, and emergency shelters didn't resist to the high temperatures reached inside the tunnel. It took three years and hundreds of millions of euros to make the road tunnel again operational, and the renovation brought a huge improvement to the security measures.

2.2.2 Saint Gotthard Tunnel (2001)

October 24, 2001, another serious event extended the list of road tunnel fires.
At 9:30 AM a tragic collision between two trucks took place inside the Saint Gotthard tunnel[9], in Switzerland. The heavy vehicles were loaded with hundreds of tires, and one of the fuel tank got damaged by the collision, spreading its content on the road around the crash. Then, a spark caused by a short circuit ignited the flammable liquid, causing a huge explosion followed by a violent fire.
In this case the entrance of further vehicles was prevented and rescue teams were promptly alerted; despite this, ten people died due to intoxication and one due to burning.

2.2.3 Fréjus Tunnel (2005, 2010)

Since Fréjus tunnel has been the main focus of the scenario in question, the last fire events happened inside this tunnel have been considerably analyzed during the development of the work.

The most severe one, in terms of human lives lost, took place on June 4, 2005[8]: at 6:00 PM, a truck loaded with tires suffered from a fuel loss, which generated a fire at contact with the truck engine. Four more vehicles caught fire due to radiation, since temperatures overtook 1000°, and it took four hours for the rescue teams to reach the fire from the French side. Several hours later, two human bodies were found close to the A6 shelter, probably killed by smoke inhalation because they did not promptly leave the cabin[4].

The last fire event, which inspired most the design of the work, occurred on November 23, 2010[19], and it didn't provoke any casualty. At 8:25 AM, an heavy vehicle entered the tunnel, and six minutes later the engine started producing smoke without being noticed by the driver. At the same time, the automatic detection system notifies the starting of a fire to the control center, automatically tune in to the nearest camera.

The driver stopped the truck next to the niche number 44, and the vehicles of the opposite lane kept passing by, creating air-flows and fanning the flames. At the same time, the control room enabled the smoke extraction system and alerted the rescue teams in charge, following the bi-national emergency plan.
The driver, once got out of the truck, didn't press any SOS button, and he was rescued by another driver who got through the fire by overtaking the truck, exposing himself to a huge risk of collision. Later, every civilian involved reached or was brought inside the security shelters and later evacuated, while the firefighting worked on the extinguishing till about 8:50 AM.

2.3 The Fréjus tunnel

The Fréjus tunnel is a road tunnel which links the French city of Modane with the Italian city of Bardonecchia; it's 12,895 Km long (6,8 of them in Italy), and entered into service in 1980. It's managed by two companies, the Italian SITAF[1] and the French SFTRF[2], each with a section of competence.

2.3.1 Original structure (1980)

Initially, the tunnel was composed by a single barrel with the following characteristics[15].

- Length: 12.895 m.
- Quota at Italian entrance: 1.297 m.
- Quota at French entrance: 1.228 m.
- Slope (descent from Italy to France): 0.54%.
- Average length between sidewalks: 10.10 m.
- Traffic lanes: 2.
- Usable length for traffic: 9 m.
- Maximum height allowed: 4.30 m.
- Maximum height up to false ceiling: 4.48 m.
- Lay-bys: 5 (every 2.100 m circa).
- Lay-by length: 40,5 m.
- Lay-by width: 2 m.
- U-turns: 5 (every 2.100 m circa).

[1]Società Italiana per il Traforo Autostradale del Fréjus per Azioni
[2]Société Française du Tunnel Routier du Fréjus

- U-turn length: from 3 m to 8,5 m.
- U-turn usable length: from 6 m to 7,5 m.
- French-side outside service area: 32.000 mq.
- Italian-side outside service area: 35.000 mq.

After the Mont Blanc accident (1999), the tunnel security has been greatly improved in order to avoid similar events. In particular, the user can find the following tools.

- Yellow tracking lights on piers every 20 m.
- Blue lights on piers every 150 m indicating the safe distance.
- Semaphores every 530 m.
- SOS niches every 265 m, equipped with 2 extinguishers, 1 220 V 16 A wall socket and 1 telephone.
- SOS buttons every 20 m, corresponding to lights on piers.
- Hydrants every 130 m, connected to water pipes.
- 11 ventilated security shelters, protected by a fire-wall door and equipped with 1 medical kit, 2 5-tank racks of oxygen, 1 communication module, 1 extinguisher, 6 bottles of water, 1 bench and 1 information and localization sign (being gradually replaced by the new bypass shelters linked to the second barrel).

The traffic is continuously monitored by 241 security cameras:

- 204 fixed cameras inside the tunnel;
- 11 security cameras inside the shelters;
- 5 for the garage areas;
- 12 cameras in the Italian service area;
- 9 cameras in the French service area.

In case of alarm, the footage is streamed to the control room on various monitors depending on the type of alarm (road or DAI[3]).

- Road alarms are triggered by opening the door of a niche, by pressing an SOS button, by opening a shelter door, by opening a door to the ventilation duct or by activating of the communication module inside a shelter.

[3] *Détection Automatique d'Incidents*, a type of alarm which is automatically triggered in case of incidents

- DAI alarm are automatically enabled when a vehicle stops or slows down below the minimum mandatory speed, in case of traffic jam, or when a vehicle is emitting smoke in an unusual manner. Those kind of alarms are transmitted to the control room in form of one minute video-clip over a touch console, allowing the operator to inspect the cause.

As previously mentioned, the tunnel Authority also transmit on two frequencies (103.3 FM and 107.7 FM) various recorded radio messages to guide the user in every kind of situation (Table 2.1).

Table 2.1: List of the reproduced radio messages

Message type	Message content
Standard	Welcome to the Fréjus tunnel. For your safety, the tunnel is equipped with a video-surveillance system. The compulsory distance between vehicles is 150 m, two blue lights every 150 m must separate your vehicle from the one travelling in front of you please observe speed limits, minimum 50 km/h, maximum 70. In case of emergency, move to the right side and stop your vehicle and follow the safety instructions you were given at the toll booth, always listen to the radio and have a nice trip.
Emergency	Attention please, attention please, stop at the red light, turn off the engine of your vehicle, quickly reach a safe green area indicated by flashing lights, help anyone who finds himself in difficulty.
Inside shelter	Welcome to the safe area, this area is ventilated with fresh air and you will be able to wait until a rescue team arrives. Make sure the access doors are tightly closed. Carefully follow our instructions and read the information cards. The control room operators are informed of your presence in this safe area. In case of emergency, you can directly contact an operator by pressing the SOS button. Do not go back into the tunnel, stay in the safe area, here you are safe. Try to keep the people in your group calm, a rescue team is on the way to take you out in safe conditions.

In addition, a large number of operative radio frequencies are re-transmitted, in order to guarantee the communication for the rescue teams in case of emergency.

2.3.2 Security tunnel (2009)

The doubling of the tunnel was planned in 2009, in principle as security tunnel only; the work started in 2011, and the excavation was completed in 2014. Since then, both Italian and French Authorities started the creation of a newly designed by-pass shelters, providing a connection between the traffic tunnel and the new security tunnel; the French half already divested the old shelters in favour of the new by-passes, while the Italian half is still working on the conversion.
The presence of a second tunnel greatly improves the freedom of movement of the rescue teams, giving them a way to get close to the fire and to rescue people without having to deal with smoke, heat and abandoned vehicles.

2.3.3 Conversion of the security tunnel to transit tunnel (2012-2019)

The current long term plan for the security tunnel (formalized in 2012 and provided for 2019) is a conversion of the security tunnel to road tunnel[1]. The main goal of this decision is to reduce the probability of head-on crashes to zero by splitting the two directions of travel between the two barrels, and at the same time to increase the volume of traffic bearable by the tunnel.

Chapter 3

Background

3.1 VR as training tool

Training is an indispensable tool for companies, Institutions and intervention corps, which could eventually become exorbitant, hard to manage and even harder to arrange. This is why VR has always been considered a powerful tool in order to make training easier, by developing custom solutions capable of lowering those costs while guaranteeing free replicability.
With the diffusion of new relatively cheap devices such the HTC® Vive™ and the development of new platforms (OpenVR, Windows Holographic), gaining access to VR is becoming easier everyday, rendering old solutions obsolete and overpriced.

3.2 VR devices

In the latest years the number of commercial devices has extraordinarily grown, thanks to the high number of companies which try to enter the VR market with innovative solutions before the competitors.

3.2.1 Head-mounted displays

Head-mounted displays (HDM) are the pillar of the commercial VR, and their purpose is to immerse the user inside the virtual world by providing the 360° view. There are passive viewers with little to no computation on-board (e.g. Google Cardboard and Samsung Gear VR) which exploits the sensors of smart-phones (gyroscope, accelerometer, magnetometer), along with the display and the computational power. Then, there are more evolved devices with built-in displays, dedicated sensors and external controllers in order to obtain an higher level of immersion and interaction with the virtual environment, but they usually require an external workstation. For this work two devices were initially considered, the HTC® Vive™ and the Facebook Inc. Oculus Rift, and later with the development it has been decided to focus on the first.

The HTC® Vive™[22] (Figure 3.1) was initially announced on 2014 and reached

the marked on 2016; it is characterized by a wired headset, two controllers and two base stations named "Lighthouses" used to track the position of the player inside the room.
The display has a refresh rate of 90 Hz and is composed by two 1080x1200 screens, while the headset has a gyroscope, an accelerometer and a large number of infrared sensors; the headset also contains a front camera, used by the "Chaperone" safety system along with base stations to draw a virtual wall through grids inside the virtual environment.
This safety system is necessary for the room scale locomotion mode, where the user can naturally walk through the room, and could eventually surpass this physical limit by adding game mechanics to move the "room" in a different section of the virtual environment (e.g. teleporting). Its platform is the Steam OpenVR, and was developed in collaboration with Valve.
Its controllers, being equipped with a group of sensors on the top, are continuously displayed inside the virtual environment, facilitating the interactions and enhancing the sense of presence.

Figure 3.1: The HTC® Vive™ with the bundled Controllers and Base Stations - photograph by ETC-USC, distributed under a CC-BY 2.0 license

In addition to the provided bundle, ulterior tracking device is available for purchase in order to enhance the tracking capabilities of the Vive™ system, the Vive™ Tracker (Figure 3.2). This device is is basically a Vive™ Controller deprived of handle and buttons; in addition, it is equipped with a 1/4" screw hole, guaranteeing the compatibility with most of the standard mountings used in photography.

Figure 3.2: HTC® Vive™ Tracker mounted on leg support

The Oculus Rift[23] (Figure 3.3), the device produced by Oculus VR[1], was initially bundled with a regular game-pad, therefore it was not characterized by the same level of interaction offered by the Vive™; this is no longer true since Oculus developed and released a dedicated controller solution for its HMD, capable of representing the fingers' position due to its shape and button layout.
It mounts two 90 Hz 1080x1200 OLED[2] panels, and differently from the Vive™ it has headphones integrated which provide real-time 3D audio effects, and its bundled with desk sensors with USB connectivity, resulting much handier than the wall-mounted Lighthouse of the competitor.

Figure 3.3: The Oculus Rift - photograph by Evan-Amos, distributed under a public domain license

[1]A division of Facebook Inc.
[2]Organic Light Emitting Diode, a screen manufacturing technology

3.2.2 Glove Controllers

Although most of the VR devices are bundled with dedicated hand controllers, a more natural interaction method could be required in case of specific applications. VR goves are interaction devices designed to give a provide a completely natural interaction, along with providing haptic feedback through actuators directly on fingers.

3.2.3 Locomotion

Locomotion is one of the most critical aspects in VR applications, in particular when it comes to virtual environment than are wider than the real world space available for the user. Many solutions have been proposed or developed, but every method have different advantages and drawbacks.

Room scale with teleporting, as previously mentioned, is the default method offered by HTC® Vive™ in its SDK[3], and it is widely used for all the applications where realism is not the main focus. The room scale walking is surely the best way to move inside the environment, but if the environment is wider than the room or developed through length it could defeat the purpose of the simulation.

A software alternative to this system is the arm swinging, a locomotion mode based on the movement of arms captured through controller, that permits to move the room inside the virtual environment while the user is not moving in the real world. Alternatively, additional sensors can be used to track the legs of the user and capture the movement while in place, and than moving the player with the same method of the arm swinging.

Lastly, a dedicated hardware solution could be purchased and interfaced with the VR system, the VR treadmills. These devices are less affordable than the VR headset, more bulky, and harder to get hold of, but they ideally offer a an higher level of naturalness while moving inside the virtual environment.

[3]Software Development Kit, a kit composed by development tools and software documentation provided for a given product

3.2.4 Cyberith Virtualizer

The Cyberith Virtualizer (Figure 3.4) is an example of omni-directional treadmill; it has a flat slippery base (special overshoes are supplied) with sensors to track the walk of the user and a directional ring which can track user orientation and chest position (standing, crouched).

Figure 3.4: Cyberith Virtualizer, in combination with the HTC® Vive™

3.3 Previous Works

When it comes to VR, the evolution of the devices along with the natural growth in computing power makes hard to find similar works using the same kind of approach. Nevertheless, a research for previous works was performed anyway, with the aim to find useful starting points for the work.

In particular, the authors of A virtual reality based fire training simulator integrated with fire dynamics data[3] proposed an interesting framework for interfacing fire dynamics data obtained with an off-line CFD[4] computation into a Virtual Reality based fire training simulation system, including techniques to process data in order to obtain a 3D visualization of toxic gasses and hazard levels.
The logic of the technique is subdivided in three blocks: numerical fire simulation, pre-processing and real-time processing.

- In the numerical fire simulation, an FDS-based[5] simulation is performed on the basis of a given scenario and geometrical data for the calculation space. The output is then passed to the second block.
- The raw data coming as output of the numerical simulation are then converted in ASCII, re-sampled, transformed and octree-partitioned[6].
- In the real-time processing, the octree database is used along with a LOD[7] selection system and a frame interpolation logic to provide the data used inside the simulation.

Exploiting this technique, a physically precise fire simulation could be integrated in a VR environment.

A different approach was followed by the authors of A virtual reality based fire training simulator with smoke hazard assessment capacity[24]. In this paper, the focus is moved on the smoke hazard, being recognized as one of the principal cause of death in fire. For the purpose, a visualization technique based on fire dynamics data and volume rendering has been designed in order to create a simulator characterized by a realistic and accurate smoke environment, ideal for training. The investigated scenarios included a subway station and a primary school.

[4]Computational Fluid Dynamic, a numerical analysis method for fluid dynamic simulation through computer

[5]Fire Dynamic Simulation, a CFD model of fire-driven fluid flow

[6]Tree data structure used as method of recursive space partitioning in 3D graphics

[7]Level Of Detail, an optimization method used in computer graphics

Chapter 4

Methodology

4.1 Requirement definition

Requirement definition took place in SiTI, where along with the supervisors the following required activities were identified.

- Scenario definition.
- Storyboard definition.
- 3D modeling of the setting.
- Modeling of the physics effects.
- Audio effects implementation.
- Modeling of the fire sources.
- Unity Build for HTC® Vive™.

Along with the following discretionary ones.

- Interaction with objects.
- Interaction with Non-Playable Characters (NPC).
- Implementation of additional tools.

Then, the workload had been subdivided with the colleague Michele Billi, who took charge of the aspects more coherent with his particular skill set, such graphic rendering, animation and sound design.

4.2 Required activities

The following activities were considered compulsory for the delivery of the work, thus they have been all completed before moving on the optional ones.

4.2.1 Scenario definition

The scenario definition required an initial research about the topic, and has been discussed for a significant amount of time before being considered definitive.

The starting subject was the representation of a fire inside a tunnel, developing from a car accident involving an heavy vehicle; close attention should have been payed on the fire and smoke representation, with the goal of developing a dynamic system while preserving a good level of graphic fidelity.
Being the SiTI a remarkable interface towards the various security corps, and being the other PRODIGE scenarios developed for operators' training, the initial target audience of FrejusVR was the same. However, after further analysis and considerations, the focus has been moved to a different kind of user, the civilian.

The reasons behind this choice were numerous; in the first place, the various interactions with firefighters, civil defense and tunnel Authority brought out a huge problem with the behavior of the random user of the tunnel in case of emergency. The cause of this misbehaviour could be identified in a cultural issue (people are not inclined to learn or pay attention to safety instructions when communicated in a non-emergency situation), or in a communication issue (the tools provided for learning those instructions are improperly designed or unusable); therefore, investigating the cause of this problem while developing a tool useful to mitigate it was acknowledged as main goal. This choice also simplified the subsequent testing activity, since common people would have been much easier to involve than firemen, at least in terms of bureaucracy and coordination.

Another point of emphasis was put on the graphic rendering of the tunnel; after identifying the Frèjus tunnel as scenario subject, it has been decided that the fidelity of the representation was a priority, therefore arrangements should have been made with the tunnel Authority, in order to gather useful information and to organize an on-site expedition.

Given this, the scenario was also supposed to be designed to support a future conversion from civilian usage to operator usage, guaranteeing a reasonable level of flexibility.

4.2.2 Storyboard definition

Activities started with the realization of a storyboard representing the upcoming simulation; for its definition, numerous tunnel fire events footage and reports were looked for and meticulously analyzed. The result of the analysis displayed that the fire usually develops from heavy vehicles, and not necessarily after a car crash.

The prescribed behaviors were also investigated in order to identify a set of interactions useful for our implementation; in particular, the official security brochure has been the starting point, provided by the tunnel Authorities of Frèjus and Mont Blanc tunnels at the entrance. The tool is characterized by a green side (Figure **??**) and a red side (Figure 4.2).

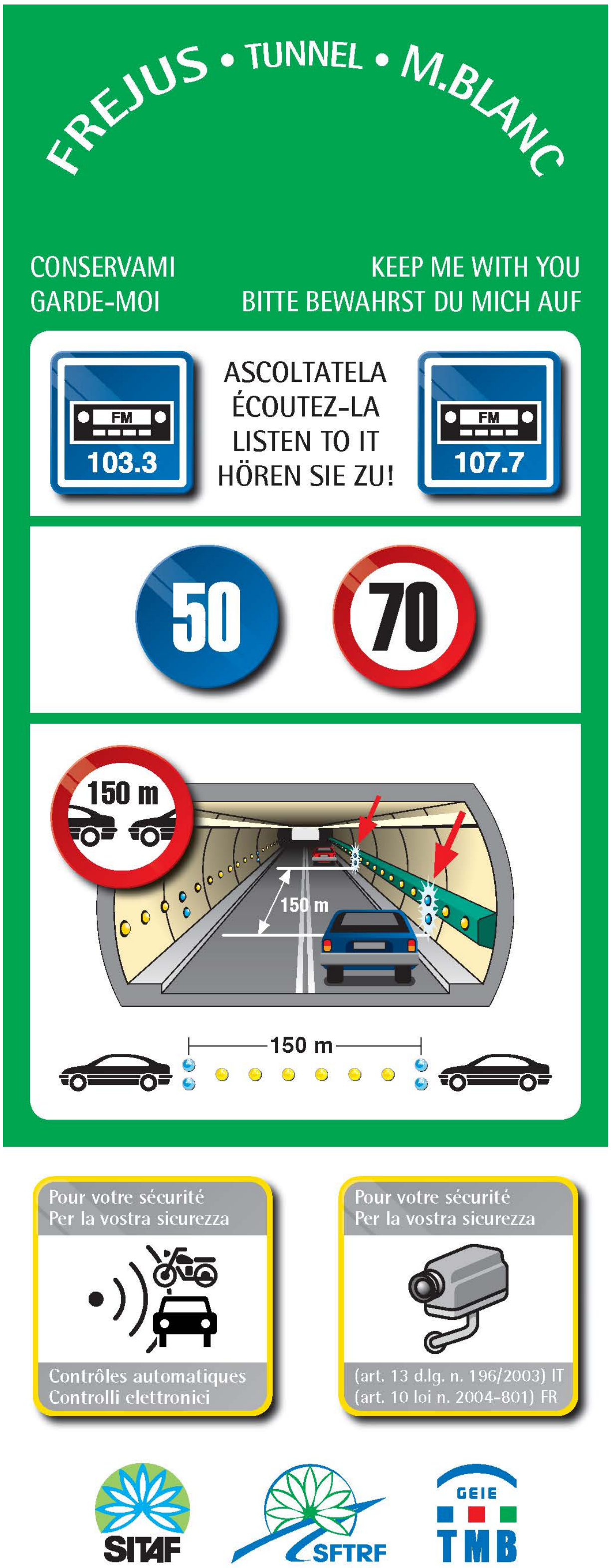

Figure 4.1: Green side of the official brochure, describing the rules of conduct in case of regular situations.

Figure 4.2: Red side of the official brochure, describing the rules of conduct in case of emergencies.

The green side invites user to:

- listen to the given radio frequency;
- keep a minimum speed of 50 km/h and maximum speed of 70 km/h;
- while traveling , keep a 150 m distance between you and the preceding vehicle, using the blue lights on the side of the road as a reference.

The red side invites the user to:

- stop the vehicle, keeping a minimum distance of 100 meters from the burning vehicle;
- turn off the engine;
- turn on the hazard lights;
- reach the nearest emergency shelter following the indications;
- if possible, ask for help using the SOS telephone in a niche or by pressing an SOS button;
- feel free to use the extinguishers available in niches and shelters.

As result of the investigation, the following storyboard concept was developed.

1. once the simulation starts (figure 4.3) the user finds himself or herself on his vehicle travelling inside the tunnel while the radio is broadcasting the usual instructions for the drivers;
2. after a while he or she will spot a stopped vehicle on fire (figure 4.4) in the opposite lane, and he or she will have to brake. After gaining the controls over the car brake system, the user can manually operate the brakes to keep the safe distance of 100 meters from the danger, thus contributing at preventing the spread of the fire. Stopping the car at lower distances will result in an error; the same goes for not braking at all (the vehicle will brake autonomously leaving the user very close to the danger);
3. after stopping the car, the user will be able to freely interact with the scenario (figure from 4.5 to 4.10) in order to complete the simulation by following the indications reported in the brochure. At the conclusion of the simulation, there will be a summary of the errors and of the correct actions made by the user.

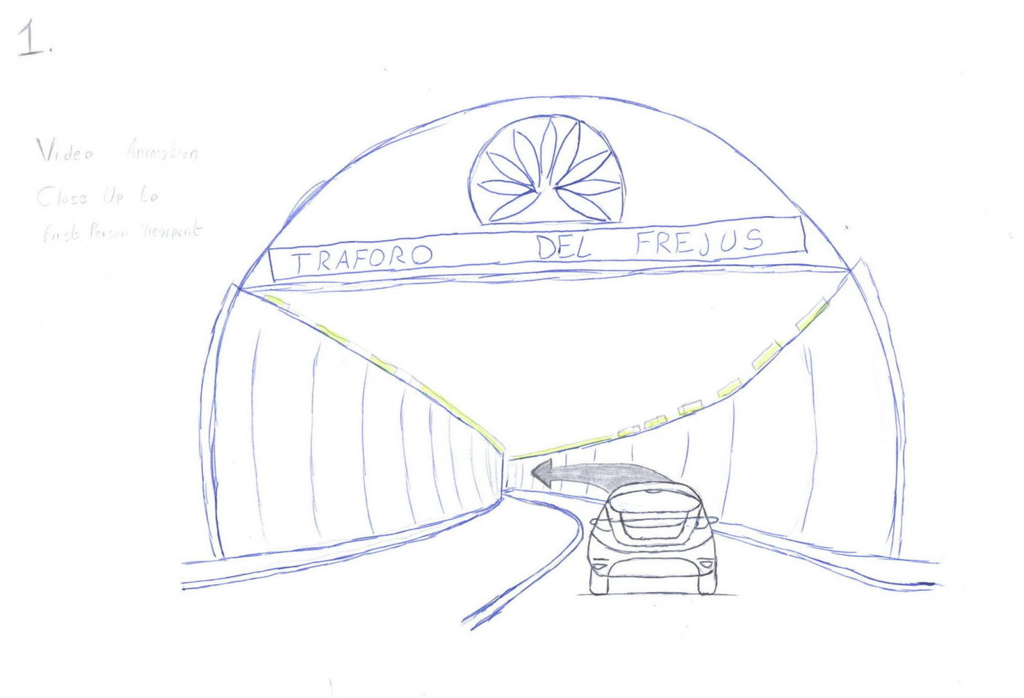

Figure 4.3: Page 1 of the storyboard.

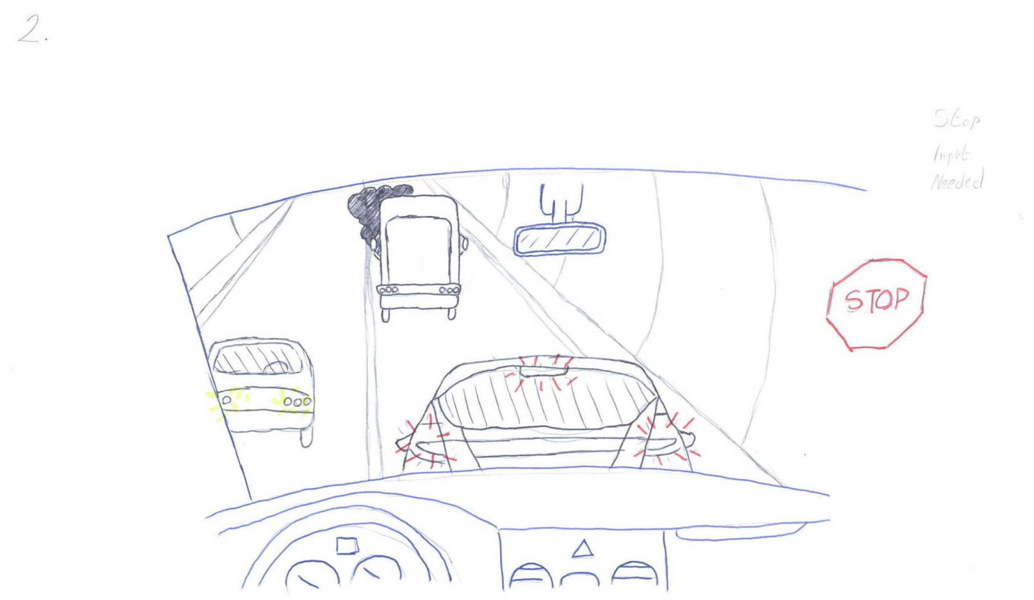

Figure 4.4: Page 2 of the storyboard.

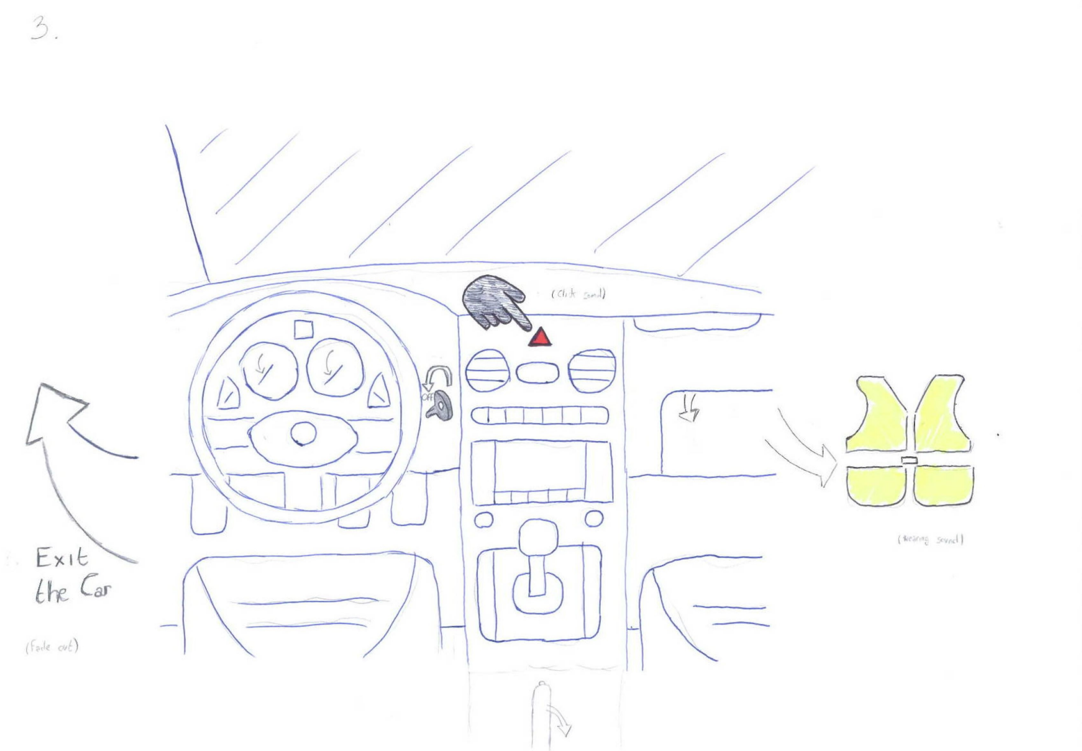

Figure 4.5: Page 3 of the storyboard.

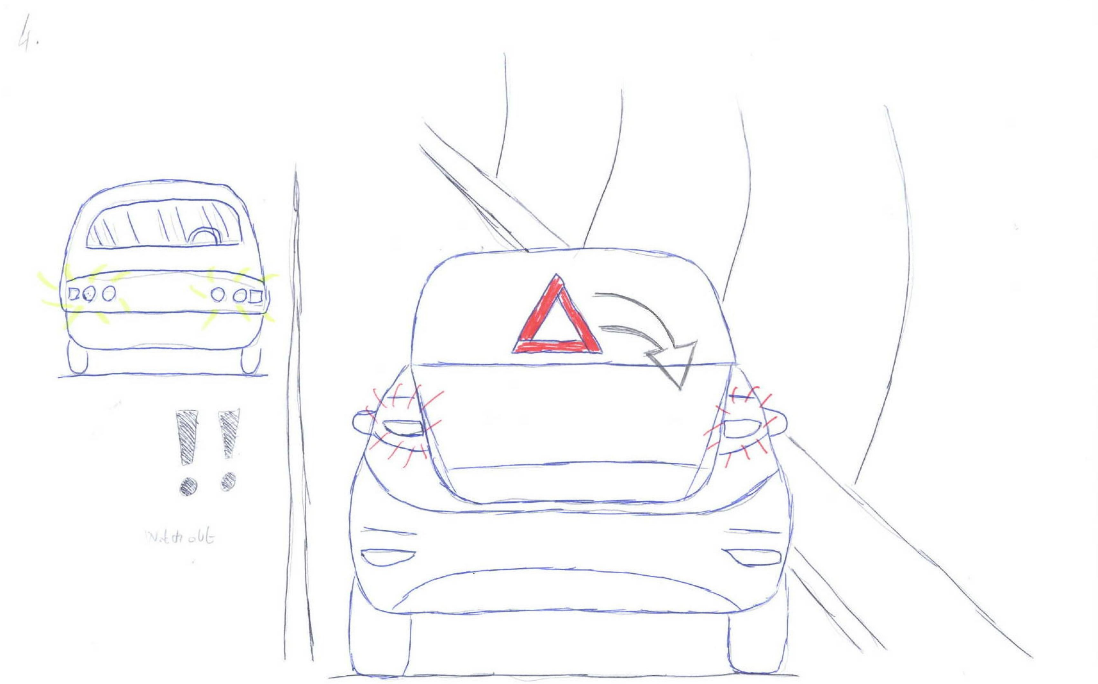

Figure 4.6: Page 4 of the storyboard.

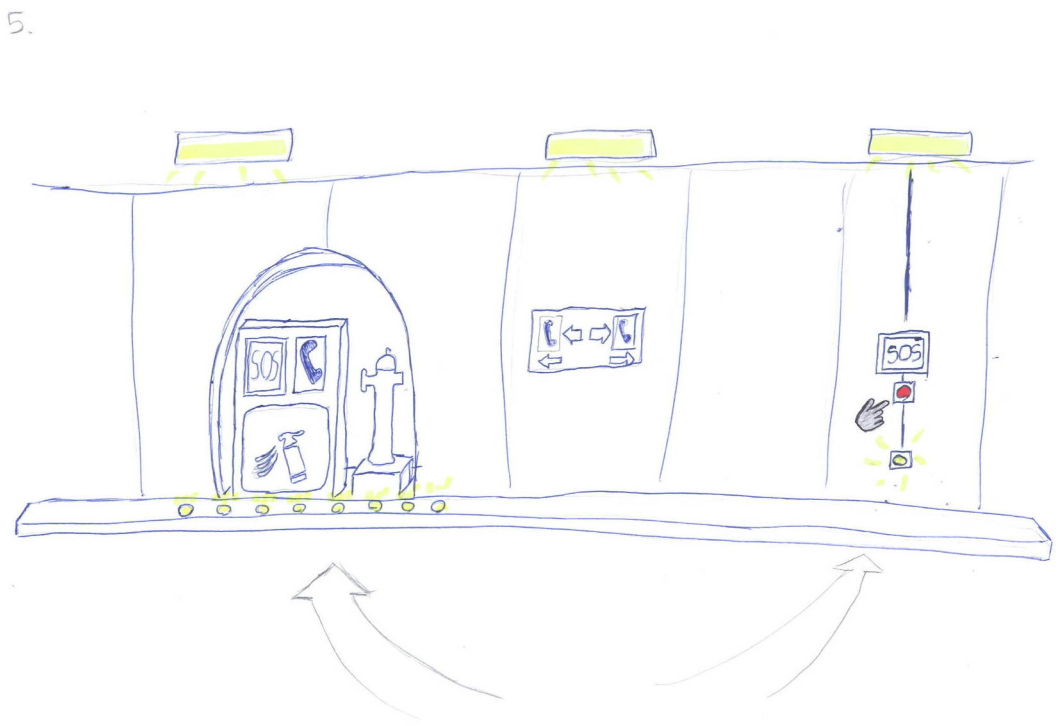

Figure 4.7: Page 5 of the storyboard.

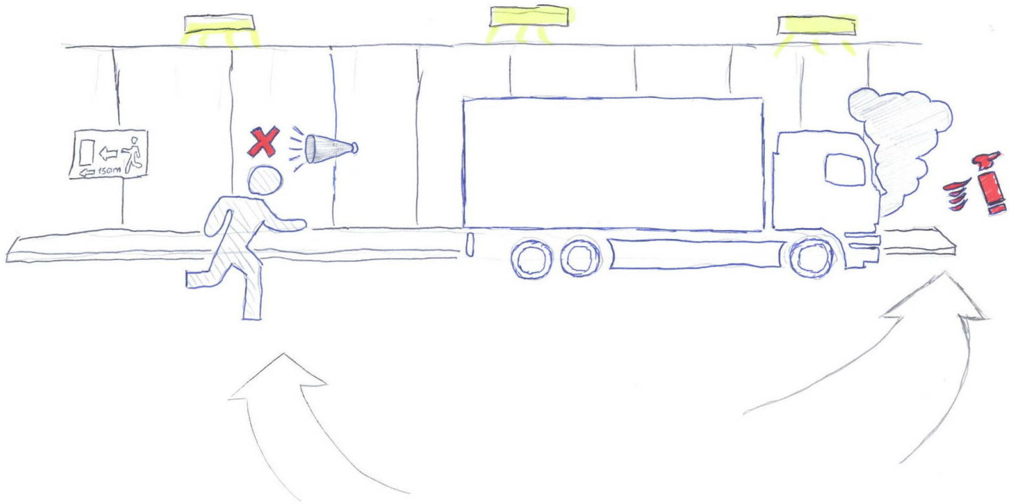

Figure 4.8: Page 6 of the storyboard.

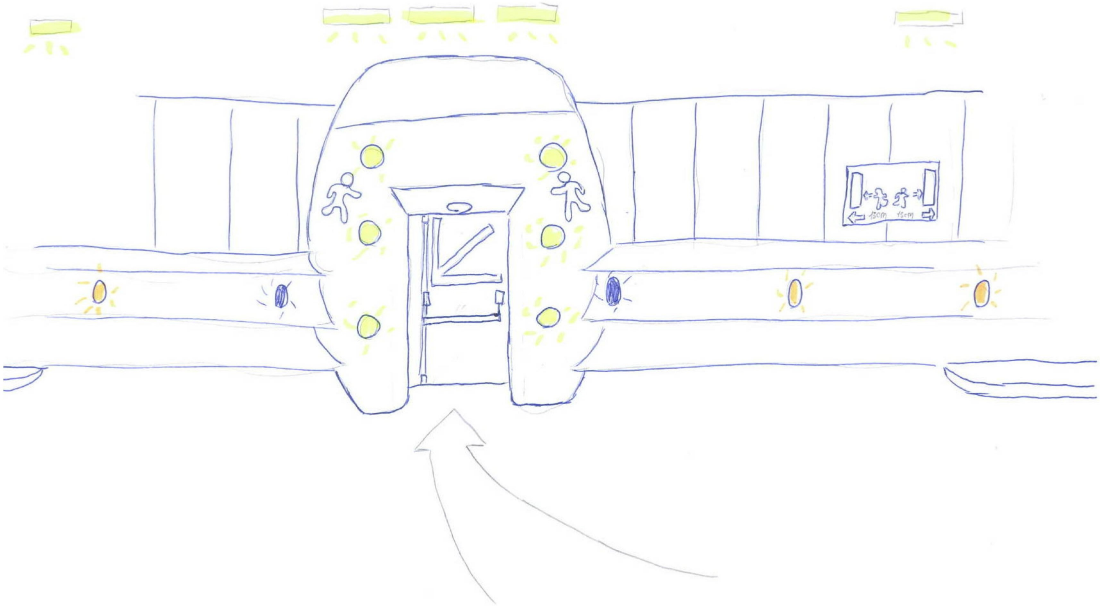

Figure 4.9: Page 7 of the storyboard.

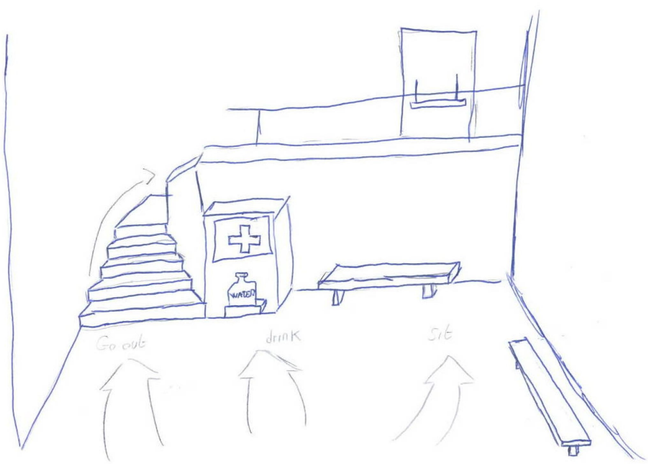

Figure 4.10: Page 8 of the storyboard.

4.2.3 3D modeling of the setting

Being the simulation fidelity a critical part of the work, it has been necessary to invest time and effort in the 3D modeling the setting.
Blender™[1] 2.77 was selected as 3D modeling and animation tool, because of its powerful tools, its gratuitousness and because it was widely explained in many academic courses.
The modeling was initially started from scratch, with the sole support of Internet, in order to develop a first version of a generic tunnel, mandatory for the start of work. Later, after the meeting with SITAF, a huge amount of real data was gathered, allowing the whole tunnel recreation on a realistic basis.

Figure 4.11: Tunnel versions comparison (before and after the on-site expedition).

4.2.4 Modeling of the physics effects

The modeling of the physics effects, performed by exploiting game-engine standard tools, should have included the rendering o the wind, the modeling of car movements and the physics of the dynamic objects. The fire modeling, while being a physics effect, is considered separately due its importance.

[1] A free multi-platform software for modeling, rigging, animating, simulating physics, compositing, rendering and game-developing

4.2.5 Audio effects implementation

Audio too was considered crucial for a completely immersive experience, therefore a large number of sound sources were inserted in the implementation plan. Most of them were obtained and through freesound.org[6], a free collaborative sound database full of Creative Commons Licensed sounds, while the most specific ones such radio messages and alarms were directly recorded inside the tunnel or received from the tunnel Authority. In order to edit and tweak the numerous audio clips, the tool Audacity®[2] 2.1.3 was employed.

4.2.6 Modeling of the fire sources

Differently from the other physics effects, the visual representation of fire sources was taken care to the last detail. In accordance with SiTI and as result of a small investment, the Ultra Real Fire Effects Volume 1[16] Unity asset was used as starting point to create every fire or smoke source useful for the scenario (Figure 4.12).

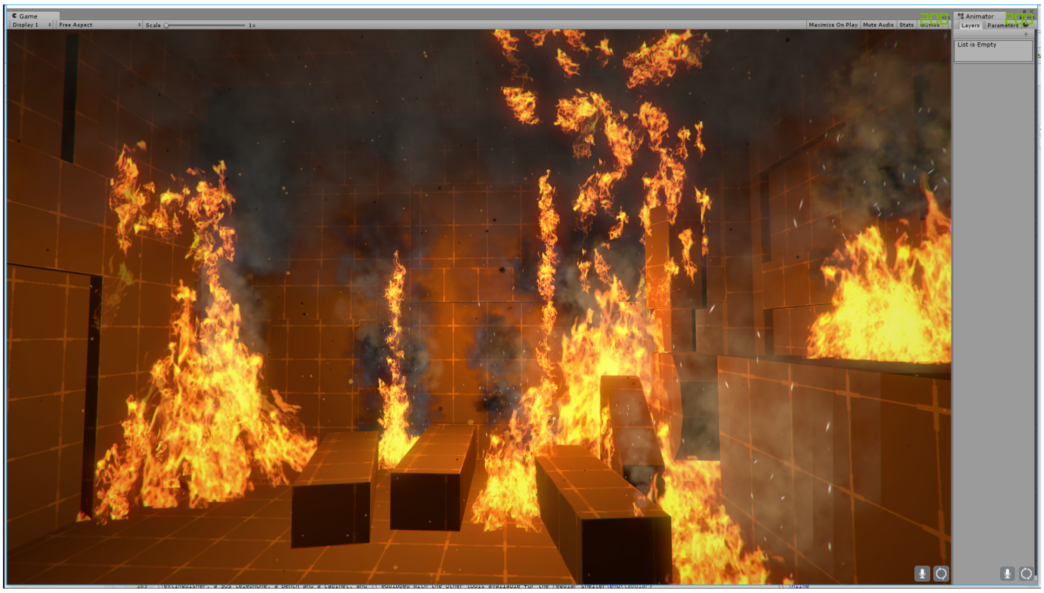

Figure 4.12: Demo scene from the Ultra Real Fire Effects Volume 1 by Ultra Real.

4.2.7 Modeling of the fire spread phenomenon

Along with the visual representation of fire, a form of fire spread logic was mandatory for the aim of the work. The non-scalable nature of this kind of algorithms makes it difficult to design a scenario-independent implementation, hence game engine usually don't provide a standard way to render the above-mentioned phenomenon. As for the Ultra Real, previous implementation were investigated in order to have a starting base; the The Simple Fire Engine[2] by Mike Boere was chosen and then heavily modified in order to implement the wide list of missing features required for the work (Figure 4.13).

[2]A free audio editing tool

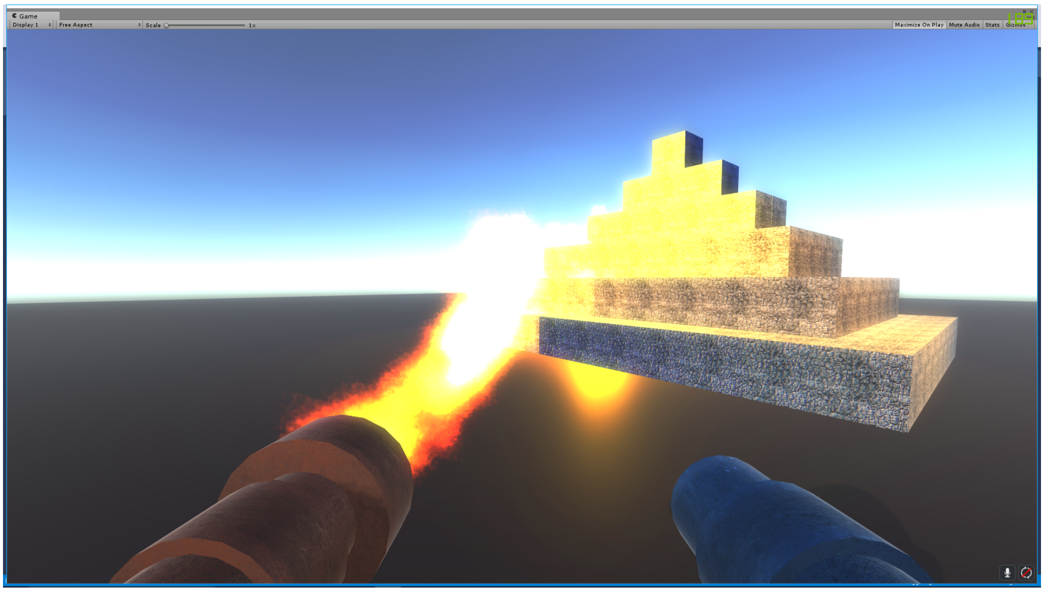

Figure 4.13: Test of the Simple Fire Engine my Mike Boere.

4.2.8 Unity Build for HTC® Vive™

As mentioned above, Unity 3D has been selected as game-engine. Among many different alternatives, Unity is probably one the most flexible one thanks to the wide asset store and the large number of supported devices. It also provides a dedicated mode for building VR applications, and among its target platforms one can find OpenVR, Oculus SDK and Windows Holographic platforms.

The HTC® Vive™ was preferred over the Oculus Rift due to the superior room-scale capabilities, along with the superior availability in SiTI.

For the locomotion, the initial design provided for the use of the Cyberith Virtualizer locomotion treadmill, already available in the SiTI VR laboratory. In addition, other available techniques were supposed to be investigated in order to find valid alternatives to the treadmill.

As result of this research, two methods have been identified as alternative to the treadmill.

- The Arm-Swinging, which was implemented starting from the ArmSwinger[**armswing**] by ElectricNightOwl free script, and by modifying it in order to maintain a similarity with the Cyberith SDK movement logic (necessary for the subsequent testing).

- The Foot-Swinging, a modified version of the Arm-Swinging which uses legs instead of arms for the movement, tracking them through additional sensors. For the purpose, two additional Vive™ Controllers were initially used, and later replaced with Vive™ Trackers[3].

[3]Additional tracking devices, officially available for developers only

4.3 Discretionary activities

These activities, even if optional, were all partly investigated, developed and tested during the development of the work.

4.3.1 Interaction with objects

The interaction with objects, initially considered discretionary, was moved to the list of mandatory activities after the storyboard definition, being an essential part of the simulation. Taking advantage of all the main HTC® Vive™ Controller features, an interaction system focused on robustness and intuitiveness was designed and implemented, along with every kind of interaction required by the storyboard.

4.3.2 Interaction with Non-Playable Characters (NPC)

The interaction with a single NPC was initially planned, implemented and later scrapped for various reason including a negative impact on the overall immersion (interaction with other people through controllers is highly unnatural), and the unsustainable effort to make it work flawlessly along with the rest of the simulation.
As result of this, the driver of the truck which was initially planned to be inside the tunnel requiring help from the user, and which was programmed to autonomously walk, run and eventually die by fire, intoxication or car hit, has been moved inside the shelter as simple aesthetic element.

4.3.3 Implementation of additional tools

Many additional tools were considered during the development, in particular the possibility to create a control-room mode[4] or to develop a multi-player build.
At the time these ideas were investigated, most of the scripting work was already designed and implemented for a string off-line usage, therefore the work required to completely re-write the already functioning logic wasn't compatible with the deadlines of the project.

[4]Training mode where along with the user there is a supervisor through a second machine who could alter the logic of the scenario at run-time

Chapter 5

Implementation

5.1 Research and data gathering

During the realization of the scenario the aim was to reach a high level of realism. To this aim, it was necessary to realize a virtual representation of the tunnel as much detailed as possible. After developing an initial version of the tunnel based on data gathered from public sources like newspapers and Internet, a meeting was arranged with the SITAF to get a first feedback about the work done and to obtain more reliable data to work with.

Successively, an on-site expedition was performed in the direction of gathering photos, sound recordings, plans and every other vital information to guarantee the replicability of the entire tunnel.

5.1.1 Preliminary research

As mentioned above, the first part of the research was focused on retrieving public data about the tunnel, principally from the Internet.
This choice was necessary due to the urge to start with the development of the application logic, without having to wait for an official interaction with the tunnel Authority.
The result was a first version of the tunnel modeled using the wrong geometric data (the new security tunnel measures were mistakenly used instead of the original ones), and depicted considering the old illumination system (yellow neon glow lamps) recently replaced with a totally different one (white led array lights).
Despite the issue, this rough version has been useful for designing, writing and testing the logic structure later adopted with the newer version.

5.1.2 Interactions with security corps

Along with the preliminary research, a significant number of interactions with different security corps were performed in order to familiarize with the subject of the work.

- Attending to the seminar "L'Approccio multidisciplinare per la gestione dell'emergenza degli incendi boschivi"[21], which concerned among other things about the fire spread modeling algorithm working on 2D grids where cells are characterized by a fuel value, made it clear that the complexity of that kind of methods was unsuitable for the work in object.
  At the end of the seminar, the interaction with fire department and forest service operators was helped with the identification of main hazards and frequent events characterizing a tunnel fire.

- At the presentation of the PRODIGE project at the river park of Cuneo, a second encounter with the fire department permitted to investigate the emergency procedures for both civilian and rescue teams, along with discussing about the common mistakes made by untrained people.

- Later with the development but before the expedition to the tunnel, other meetings with fire department and civil defense scheduled by SiTI were used to show the first version of the application and to obtain an initial feedback. This feedback was extremely valuable since it allowed to finely tune the behavior of the smoke, the timing of the fire and the effect of the intoxication on the player.

5.1.3 Interaction with the SITAF

The first meeting at Frèjus tunnel Authority has been a focal point for the development of the work. During this encounter, which involved SITAF, Politecnico di Torino and SiTI, the project has been presented, the design has been exposed, and a request with a list of required data has been submitted, asking for:

- photos of road surface, road marking, walls and ceiling;
- data about wall colors and lighting;
- data and photos of the various kinds of tunnel apparatuses (geometric data and distances);
- structural differences between French and Italian sections;
- data and photos of the lay-by;
- geometrical measures of the SOS niches;
- the content of the SOS niches;
- geometrical information of a shelter (possibly the 7);
- internal photos and content of a shelter;
- eventual security information about the tunnel;
- potos of the wall signaling for the closer shelter;
- information about what a user should do once reached a shelter;

- audio recording of the tunnel with normal/no traffic
- audio recording of the alarms;
- standard response of an operator when someone uses the SOS telephone;
- radio messages sent to the users;
- audio jingle used to signal a shelter.

As result of this encounter, along with a positive feedback about the design, the scheduling of a consecutive meeting for a guided expedition inside the tunnel, for the purpose of acquiring every information previously listed.
The expedition started from the Italian side of the tunnel and continued with the traversal till the border; after parking on the lay-by, we took photos and recording of the zone, including.

- Niche 50 (Figure 5.6).
- Niche 51.
- Shelter A6 (closed).
- Shelter A7 (Figure 5.4).
- Shelter S18 (by-pass) (Figure 5.2 and 5.3).
- Shelter A8 (closed).

The gathered data has been later intensively and extensively used in order to obtain the highest possible level of realism.

Figure 5.1: Rendering/photo comparison of the tunnel view from the lay-by

Figure 5.2: Rendering/photo comparison of the S18 shelter entrance

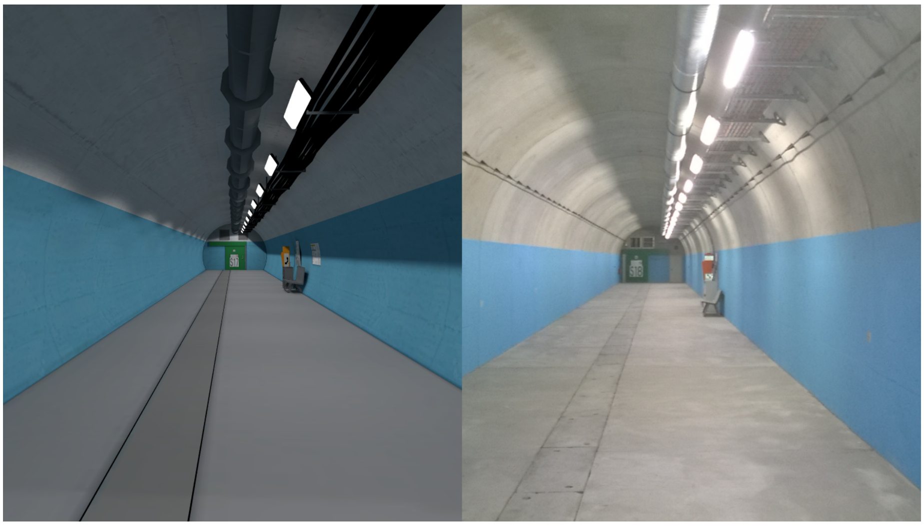

Figure 5.3: Rendering/photo comparison of the S18 shelter

Figure 5.4: Rendering/photo comparison of the S7 shelter entrance

Figure 5.5: Rendering/photo comparison of the SOS button

Figure 5.6: Rendering/photo comparison of the SOS niche 50

Figure 5.7: Rendering/photo comparison of the SOS telephone inside a niche

5.2 Development

The implementation activities were carried out since the starting of the project till the beginning of the testing, due to the continuous refinement operated on the application logic by following the feedback of the parties involved.

5.2.1 Tunnel segmentation

In order to simplify the manual assembly of the tunnel and to guarantee a good level of flexibility, it has been decided to segment the tunnel into basic units, characterized by a particular function and 10 meters long.
The tunnel section library created with this method is composed by 29 units, which can be used to recreate any section of the tunnel. The current scenario, set over the border between Italy and France, reproduces the section which goes from the km 7 to the km 6, and only required a subset of the assembled units.
On tables 5.1, 5.2 and 5.2 are listed and described all the tunnel section units available.

Table 5.1: Tunnel default sections

Section Type	**Content**
Default	The basic unit, composed by: • 1x Road • 1x Ceiling • 2x Wall • 2x Sidewalk • 2x White light LED element placed on a wire-way • 1x Set of reflective elements (at the center of the road)
Default Green Front	The default section, equipped with: • 1x Half-green wall (right side) • 1x Shelter proximity green strip (on the green wall) • 1x Shelter direction sign (front-facing, on the green wall)
Default Green Back	Similar to Default Green unit, but equipped with a rear-facing direction sign

Table 5.2: Tunnel sections equipped with niches

Section Type	Content
Hydrant	Unit equipped with: • 1x Firefight niche (left wall) • 1x Hydrant inside the niche (Italian or French) • 2x Safe distance monitor (1 per lane)
Niche	Unit equipped with: • 1x SOS niche (right side) • 1x SOS and firefighting niche (left side) • 1x Hydrant inside the left niche (Italian or French) • 4x Extinguishers (2 per niche) • 2x SOS telephones (1 per niche) • 2x Niche signal (1 per niche) • 2x Yellow light neon element (1 per niche) • 2x Semaphore • 2x Maximum speed monitor

Table 5.3: Tunnel sections related with shelters

Section Type	Content
Lay-By Start	A Niche unit equipped with: • 1x Start of the lay-by sidewalk (left side) • 1x Start of the lay-by wall (left side) • No hydrant
Lay-By	Default unit equipped with: • 1x Center of the lay-by sidewalk (left side) • 1x Center of the lay-by wall (left side)
Lay-By End	Default unit and equipped with: • 1x End of the lay-by sidewalk (left side) • 1x End of the lay-by wall (left side) • 1x Hydrant placed on the lay-by (Italian or French)
Shelter	Unit equipped with: • 1x Half-green wall (right side) • 1x Shelter proximity green strips (on the green wall) • 1x Shelter entrance (A6 or A7) • 4x Shelter signals (2 per lane) • 2x Stop monitor with flashing semaphore • 2x Half-barrier
By-pass Shelter	Shelter unit including a By-pass shelter (S17 or S18) containing: • 1x Extinguisher • 1x SOS telephone • 1x Bench • 1x Locker • 1x Medical Kit (inside the locker) • 1x 6-pack of water bottles (inside the locker) • The truck driver in case of the S18

Table 5.4: Tunnel SOS sections

Section Type	**Content**
SOS 20	Any of the previous sections, but equipped with: • 2x Yellow tracking yellow light • 2x SOS buttons
SOS 150	Any of the previous sections, but equipped with: • 2x Blue safe distance light • 2x SOS buttons

5.2.2 Lighting management

Managing the high number of light source has been a challenging task for the whole development. Even since the first short version of the tunnel the default Real-time[1] lighting showed its unsuitability for the given scenario in terms of performance. Advanced techniques were therefore investigated to get around the problem.

The first undertaken action was turning all the lights from Real-time to Mixed[2] mode in order to start lowering a the overall complexity.

After that, every stationary object inside the scenario was set as Static, in order to take advantage of the the Light Baking[3] Unity feature and drastically increasing the performance at run-time.
The drawback of this technique is the long pre-processing time required by the Bake run-time, which is proportional to the number of static objects and can last for multiple hours. The result of this onerous calculation are stored on special textures[4] called Light Maps[5].

Once solved the problem of illuminating static objects, a second technique was then necessary for managing the illumination of the dynamic ones.
For the purpose, Unity 3D offers the Light Probe; once placed inside the scenario, Light Probes encapsulates information about the light travelling through space during the Bake computation, and then at run-time use the pre-calculated values to cast lights on dynamic objects. Light Probe groups were then advisedly placed inside every tunnel section unit, creating a dene network covering the whole scene.

The combination of these three techniques drastically reduced the lighting impact on performance, allowing the usage of hundred of lights to faithfully reproduce the Frèjus tunnel illumination system.

5.2.3 Input management and player controller

One of the first task that was carried out during the development was the implementation of the various locomotion methods. As previously mentioned, the selected mode were as follows.

- Arm swinging with HTC® Vive™ Controllers (the user can walk by holding the grip key on both controllers and by swinging ahead and behind with his or her arms).
- Foot swinging with HTC® Vive™ Trackers (the user has to wear two additional tracking devices for his legs, which will let him move by walking in-place).

[1]Unity Real-time Lights are calculated and updated every frame at run-time.

[2]Unity Mixed Lights can change their visual properties during run time, but only within strong limitations.

[3]Unity pre-calculation of the illumination.

[4]Computer graphic technique which consists in mapping an image over the surfaces of a 3D model.

[5]Data structure containing the static lighting in formations about the object.

- Free movement with Cyberith Virtualizer.
- Mouse and keyboard: for debug use only, it does not require a VR headset to run.

Since the Foot swinging mechanic was not not directly supported by the HTC® Vive™, the logic had to be developed.
The first version was implemented by using two additional controllers connected with two USB extenders, which were placed on two supports modeled and 3D printed for that sake (figure 5.8 and 5.9). This solution offered the opportunity to develop a working solution, which was later reused when the two wired controllers were replaced with two wireless Vive™ Trackers.

Figure 5.8: 3D-printed support for the leg tracking device

Figure 5.9: Rendering of the support for the leg Controller on the left, Support adapted to work with Tracker on the right

The CybSDK[6] has been used as it is, being already compatible with the logic of the rest of the system; the provided player controller GameObject[7] is moved though the Camera Rig[8] by using a Character Controller[9] component, automatically managing gravity and collisions with the default layer.
In order to manage the height of the Character Controller, ad additional script has been developed, to track the height of the player through the headset position and to update the Collider[10] size every frame.

The swinging script, on the other hand, required a huge amount of work in order to behave in a similar manner; the ElectricNightOwl's Arm-Swinger, after calculating the space traveled by controllers since the previous frame, performs a set of operations in order to avoid unwanted behaviors, and then moves the Camera Rig through transform. This method was not as accurate as the Cyberith one for managing collisions (the player Collider is placer on the head only, while with the Virtualizer Prefab[11] it covers the whole body), and behaved in an unpredictable manner when dealing with height gaps.

Hence, a custom version of the above-mentioned script by re-implementing the movement functions, in the direction of reproducing the CybSDK behaviour (movement of the player through a Character Controller attached to the Camera Rig).
From this point on, the modified version of the arm-swinging script will be referred as Arm-Swing, in order to tell it apart from the original ElectricNightOwl's implementation (Arm-Swinger). Following this logic, Foot-Swing will refer to the leg-swinging version of the Arm-Swing. Arm-Swinging and Foot-Swinging will be

[6]Cyberith Virtualizer Unity Software Development Kit
[7]Name for objects inside the Unity game engine
[8]Room representation in the virtual world used by the Vive™ SDK
[9]Unity standard component for moving the player, integrating a capsule Collider
[10]Unity component used to manage collisions
[11]A Unity GameObject stored along with its properties and components

used to indicate both the action of swinging the relative limb and the locomotion technique depending on the context.

Being the controller Collider placed on the Camera Rig, the new system created a new problem; moving the Collider over the camera position required a continuous update of the Character Controller position, and this was negatively impacting on the player collision detection. The Cyberith Virtualizer did not require this sort of behaviour, since the player is physically locked all the time at the center of the room. For the swinging mode, the following solution was developed:
As soon as the player controller is spawned, the player location is selected as center of the playable area; the player will not be able to move from that position, if not through swinging (the headset will only track the vertical position along with the rotation of the head).

The huge drawback of this method is the bad feeling consequent to the movement of the head. To prevent this, the limiting script has been written with the possibility to define an offset which creates a safe area around the player where he could move the head in complete freedom; trying to exit from this area will trigger the script and start blocking the virtual movement as before.
This script was successfully used inside the player car in order to prevent glitching outside of the car borders. The final build of the application had the offset disabled for the swinging player controller due to problems with collision detection paired with Collider position update.

Along with this modifications, the customized Arm-Swing was also modified in order to support leg tracking, becoming the Foot-Swing, and offering the following swinging modes:

- both buttons pressed;
- left button pressed;
- right button pressed;
- one button pressed;
- no buttons pressed (only for leg devices).

The following device configurations are supported:

- arm-swinging;
- foot-swinging with side controllers;
- foot-swinging with rear controllers;
- foot-swinging with side trackers;
- foot-swinging with rear trackers.

An addition to these, an player controller, named First Person Controller, was implemented for debugging reasons; being it completely functional, it could be offered as main controller for an eventual non-VR version.

5.2.4 Modeling of the fire spread mechanic on flammable objects

As mentioned in the previous chapter, the fire spread logic was developed starting from the Simple Fire Engine implementation, and combined with the fire effects created using the Ultra Real Fire Effects pack as base.
Inside the Fire Engine, the propagation was modeled through the Unity message system between GameObjects with a CanBeOnFire[12] component attached, using the fire Unity Particle System particles as message carriers. An object on fire sends through its particle system's collisions a "burn" message to the colliding objects, spreading the fire around. A water particle, on the other hand, could extinguish a fire, leaving the object wet for some seconds. After burning for a given amount of time, the object is removed from the scene by a dedicated script named SmartDestroyed. The test scene also provides a FlameThrower script to test both fire spread and extinguishing.

Being this logic too simplistic for the purpose, the script has been integrated with all the required parameters and behaviors. In the new version of the spreading system, an object with a CanBeOnFire attached has a temperature, an ignition temperature, a transmission rate and a burn factor.
When the temperature reach the ignition value, it will catch fire; in order to obtain this, an initially disabled fire particle system (which has to be finely tuned to obtain a realistic look) must be added as child of the object.
The temperature can grow for direct contact with a flame, or through radiation (contact with hot smoke particles).

The heat exchange is influenced by the transmission rate, a parameter used to regulate the speed with which the object's temperature increases or decreases.
Hence, if a fire or smoke particle contacts with a burnable object and the object temperature is lower than the particle temperature, then a heat transfer will occur, and this will increase the temperature of the object by one degree multiplied for the transmission rate. An object with a higher temperature than the ambient one will tend to lose heat until it will reach the equilibrium.
In case of object on fire, it will also constantly gain a small random amount of heat from the combustion, keeping it far from the balance value. Once lit, the object will constantly lose "health", which represents the quantity of fuel left to burn; when the health comes to zero the fire will die out; if the object was tagged as "destroyable" it will be removed from the scenario, following the original Fire Engine logic, otherwise it will stay as burned object (no fuel available to catch fire again).

A custom Shader[13] capable of mixing different texture has been implemented to render the effect of burning on the object; if the material attached to the object has the given Shader, the health of the object will act as texture cross-fade coefficient to mix the original texture with another one representing the burned object.

Through the extinguisher, the user can decrease the temperature of an object on fire and eventually extinguish it after bringing it below the ignition temperature value.

[12]The Fire Engine script used to implement the fire spread logic

[13]Scripts containing calculations and algorithms for calculating the colour of each pixel rendered, based on the lighting input and the Material configuration

The massive fire involving the truck and spreading to the tunnel follows a different logic, with the aim to guarantee performance and a more realistic visual representation.
For this purpose, four fire stages have been designed:

- very Small Fire, composed by a single fire particle system (top cone) and a fire glow;
- small Fire, larger than the previous and composed by two fire particle systems (top cone and bottom cone), a fire glow and fire sparks;
- medium Fire, double size compared with the previous;
- large Fire, wide enough to cover the whole width of the tunnel.

This fire will originate from a CanBeOnFire script placed on the section of the tunnel containing the vehicle on fire, once the ignition temperature is reached. After being instantiated, it will start as Very Small Fire, and it will evolve through interpolation following the stages previously defined with a defined time-line; along with this, the Tunnel Fire Controller script will also generate various particle system for smoke and ashes in order to improve the realism of the result.

Figure 5.10: Final version of the tunnel fire, integrated with the spread, damage and intoxication logic

The smoke effect, in particular, required a huge amount of effort in order to reproduce the actual smoke behavior inside tunnel. The result of this effort was a smoke particle system emitting a huge amount of particles and influenced by the wind, which also reproduces the stratification behaviour observable during real tunnel fire events. Combining this with the intoxication mechanic explained below, the smoke implementation resulted as one of the most immersive and stress-inducing feature of the scenario.

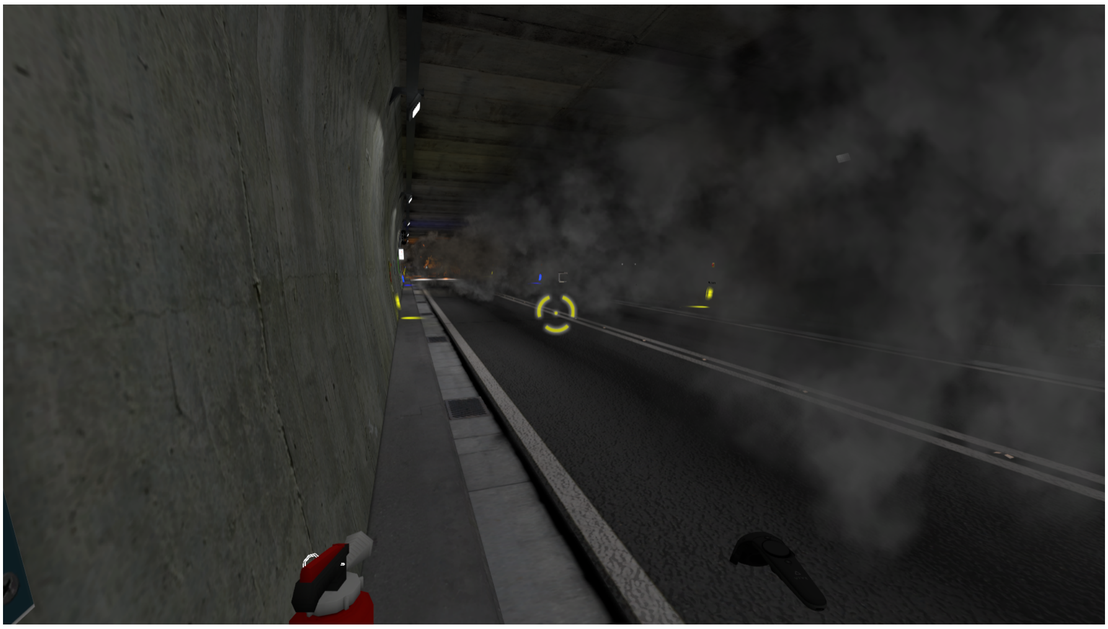

Figure 5.11: Smoke developing from the fire

5.2.5 Modeling of the fire damage and smoke intoxication on the player

In order to model the effect of the fire on the user, it was necessary to create a dedicated component; the script, named Player Status, is characterized by an health percentage, representing the health state of the player.
When the health reaches the zero, the player dies and the simulation ends. If the health is below 100% and the player is not taking further damage, a restore mechanic will gradually increase it till the full health value.

An object with a Player Status component could receive two types of damage:

- burning damage, an instantaneous damage caused by the contact with a flame or by coming close a strong heat source;
- intoxication damage, a small iterated damage which continues as long as the player is in contact with the smoke particles.

In order to inflict the damage to the player, previous message system was modified for the purpose:

- a fire particle will send a damage with pain message, characterized by a damage value (instantly subtracted from the health);
- a smoke particle will send an intoxicate message, characterized by an toxicity value (used as multiplier inside the intoxication logic).

In order to manage the effect of heat sources on the player, an additional script was implemented; the above-mentioned component is designed to send damage messages to every player found inside of a certain area at a given rate.

The feedback techniques relative to the above discussed health system are presented inside the following paragraph.

5.2.6 User Interface

When it comes to VR, the old and consolidate UI[14] paradigm used for years becomes unsuitable if not detrimental; any kind of overlay is not recommended, being the user unable to look around without dragging the display fastened to his or her head. Placing the UI over 2D planes inside the 3D environment could be an alternative, but this solution could both reduce the sense of immersion and impact on the field of view of the player.

In order to avoid these issues, the usage of 3D UI was limited to the Main Menu[15], and feedback system based on audio and visual effects was adopted for the simulation.
The most important information to be displayed to the player was the health value; in order to avoid the visual representation (health bar), an heartbeat sound was used, linking the volume and heart-rate to the health value.

- from 100 to 50 HP[16]: normal heart-rate, volume linearly interpolated between 0 and 1 using the health; (full health volume to 0, half health volume to 1);
- from 50 to 25 HP: doubled heart-rate;
- from 25 to 10 HP: tripled heart-rate;
- from 10 to 1 HP: quadrupled heart-rate.

The instantaneous damage derived from the contact with fire was rendered with a scream, while intoxication was implemented using a mix of audio (coughing sound) and visual feedback (decrease of visibility); in order to implement this visual feedback, the Unity built-in global fog feature was exploited inside the Player Status script, increasing and decreasing the fog density when necessary.

[14]User Interface, the software module designed to display information and eventually interact with the user.

[15]First scene of the application, where the user can select the locomotion device, do a small practice and then start the simulation

[16]Health Points, representing the percentage of health of the user.

Figure 5.12: Visual effect of the smoke intoxication

Another focus during the UI design was the development of a good feedback for the interactions; for the purpose, a large number of Vive^{TM} applications were analyzed in order to exploit every feature useful to make the interaction system as intuitive as possible. As result of the investigation, the following features were integrated within the interaction system.

- an Outline Shader, used to inform the player when his or her can perform an interaction by highlighting the interactive object;
- vibration, used to inform the player when his or her hand is touching an interactive object;
- a selection sound and a click sound, used for the Main Menu UI only;
- an additional Outline Shader for controller, used to highlight one or more buttons when the player can perform an interaction.

5.2.7 Interaction system

As previously mentioned, the interaction system was designed to work with the Vive^{TM} Controllers.
The button layout on the controllers includes.

- a trigger button, corresponding to the index finger;
- a directional pad on the top, which can be used as touch-pad or as button with the thumb;
- two grip buttons on both sides, which can be pressed with middle and ring fingers.

In the direction of reducing the confusion for inexperienced user, the known set of colors Cyan, Magenta, Yellow was used to respectively indicate Grips, Trigger and Pad. Along with the color, every button also have a specific function.

- the trigger button can be used to interact with objects (open/close, lift/drop, switch on/off);
- the pad can be used to show/hide the official brochure and to use the extinguisher (if already unlocked with the other hand);
- grips, useful with the Arm-Swing locomotion mode only, enable the movement inside the virtual environment.

The interaction is performed through a small spherical Trigger[17] placed on top of each hand. Every interactive object, to work with the system, requires:

- a Collider component;
- a Rigid Body component;
- a Generic Item component.

Generic Item contains various data related with the object, including various flags indicating if it ca be grabbed or not, if it is enabled or not and if the interaction with the item is enabled or disabled. In addition, the above-mentioned class can be expanded by exploiting the inheritance[18] technique in order to add useful information for specific objects while keeping the backward compatibility with the interaction system.

For objects which can be grabbed, an additional logic have been used; at the beginning of the simulation, the Item Controller object will instantiate an hidden copy of every object which could be eventually grabbed as children of every hand. The objects spawned with this logic are attached with a script named Grabbed Item, which can be extended to support the behavior of the different usable objects.

When the player will try to grab an object, the script will:

1. parent the original object to the hand, enabling inside the Rigid Body[19] component the IsKinematic[20] flag;
2. hide the original object;
3. show the corresponding grabbed object;
4. copy the state of the generic item to the grabbed object (for example, considering the extinguisher, the fuel amount would be assigned).

[17] A Collider component set as trigger will not be involved in the physic collision management, but will send collision messages to the attached script when a Rigid Body interacts with it

[18] Programming technique with which new classes can be created by reusing, extending, and modifying the behavior that is defined in other classes

[19] Unity component used to enable physics on a GameObject

[20] Flag used to make a Rigid Body unaffected by external forces

When the player will later drop the object, the script will:

1. copy the state of the grabbed item to the hidden original object;
2. hide the grabbed object;
3. show again the hidden original object;
4. remove the parenting with the hand;
5. set the IsKinematic flag to false.

Following this logic, the grabbed object can be placed in advance in the proper position in respect of the hand, and the object components can be separated between the Generic Item and the Grabbed Item (e.g. in case of the extinguisher, only the grabbed version will have the particle system and the relative script attached).

The complete list of the implemented interactions is reported in Tables 5.5 and 5.6.

Table 5.5: Interactions inside vehicle

Interaction Item	**Description**
Hazard lights button	The hazards lights button will enable the relative car function, signaling the danger to other drivers. Not enabling it will be counted as error
Car key	The car key let you to turn off the vehicle. Leaving it enabled will result in an error
Car handle	The car handle will let the user go outside of the vehicle, continuing with the simulation. Staying in the vehicle will temporarily protect him or her from the smoke, but the car could eventually catch fire due to radiation

Table 5.6: Interaction inside the tunnel

Interaction Item	**Description**
SOS button	The fire alarm, placed on the SOS column, will send an alarm to the tunnel headquarter
Niche door	Opening the niche door will grant the user access to the SOS telephone and to the extinguisher
SOS telephone	With the SOS telephone the user can ask for the help of an operator along with triggering an alarm
Extinguisher	The extinguisher can extinguish the fire, but has a limited amount of fuel and it must be unlocked before usage
Shelter door	When the user enters the shelter, the game will give him or her few seconds to use the SOS telephone, then will end the simulation
Car handle	Using the external car handle, the user can go back into the car

An additional interaction method was developed to manage the UI inside the Main Menu; for that scene, a ray-cast[21] logic with a visual ray coming out from the right controller is used to interact with the environment along with the classic method.

A similar system was also implemented inside the First Person Controller, in order to allow the previously mentioned interactions with mouse and keyboard too. To facilitate the usage of the system, a cross-hair was applied on the screen, whose color is used to signal the availability of any interaction.

Figure 5.13: Ray-cast interaction logic at work with the First Person Controller

[21]Given a point and a direction, a ray is projected in order to find intersections with other objects

5.2.8 Vehicles modeling

The work carried on to model the vehicles could be divided in four activities: 3D model segmentation, integration with the fire spread mechanic, implementation of the interactions (for the user's vehicle only) and physics development.

The initial models, obtained through free3d.com[22][7], were mostly unsuitable for the the a real-time simulation, being overly-detailed and monolithic. The first activity performed was a geometry simplification, manually operated by dissolving redundant vertices and removing unwanted parts, followed by a model segmentation, useful to implement at best the fire spread logic. Every part was then equipped with a CanBeOnFire script and characterized with rational values for its coefficients considering the real material composition, along with a fire effect tailored to the part. The truck, being the main prey for the main fire, was split in a much larger number of parts in comparison with regular vehicles.
The following table (5.7) illustrates all type of vehicle available.

Table 5.7: Vehicles

Vehicle Type	Description
Truck	The static heavy vehicle originating the fire, segmented in various sections which can burn separately: • Body • Left Door, Right Door, Back Door • Tarpaulin • Fuel Tank • Wheels Each section is equipped with a fire particle effect and characterized by different burning parameters.
User's Car	The dynamic vehicle driven by the user, segmented in: • Body • Wheels The wheel rotation is managed through a script which synchronizes the wheel colliders with the visual representation
NPC Cars	Other cars coming from the opposite lane, same logic of the user car but not integrated with the fire spread mechanic

The interactive elements inside the user's car were implemented using the already explained Generic Item logic. The list of interactions implemented for the car was previously presented in table 5.5.

Regarding the vehicle physics, its modeling was not contemplated by the initial design, and it became essential from the moment when it has been decided to provide for an initial interaction with the vehicle, in the form of controlling the brake system. The user vehicle, already integrated with the fire spread mechanic, required therefore a way to move inside the tunnel, and to eventually stop on user's command.

[22]A 3D model sharing site

The first two methods, movement through transform and through animation, were discarded for the poor flexibility and the difficulty to obtain a realistic behaviour. The third method, which was partially tested during the first stages of the development for other purposes, was picked back, made it working and integrated with the car.

This method consisted in using four Wheel Colliders[23] attached to a Rigid Body in order to simulate the engine of the vehicle. Being the player's vehicle composed by multiple interactive objects attached to Colliders and Rigid Body components themself, some expedients were necessary to make this complex system works:

1. at the beginning of the simulation, the interaction and the CanBeOnFire colliders inside the user's vehicle must be disabled to avoid misbehavior of the engine system;

2. once started, the car movement logic autonomously accelerates up to a maximum speed of 60 Km/h;

3. if the player presses any button of the controllers, the vehicle will brake, but he or she will not be able to completely stop the vehicle; releasing the button will let the vehicle to accelerate again;

4. when the vehicle will reach a first threshold, the ability to stop the vehicle will be enabled; these threshold are represented by trigger objects reacting at contact with the car;

5. if the player will stop the vehicle beyond this first threshold, the vehicle will completely stop; as soon as the speed comes closer to zero, the car Rigid Body is turned into kinematic, blocking any further movement, and all the colliders are enabled in order to permit interactions along with enabling the fire spread logic on the vehicle;

6. if the player will reach a second threshold placed at 100 m from the fire, the error will be recorded sent to the Stage Controller[24];

7. in case the vehicle will reach the third threshold, the vehicle will autonomously brake in order to force user to continue with the simulation.

Along with these precautions, the Rigid Body component over the player's car had it's movement limited to the Z axis in order to prevent deviations during the travel. If the vehicle will not leave the vehicle, the fire will evolve until reaching the car through its smoke, which could eventually catch fire due to radiation; staying inside a vehicle on flames will result in a continuous damage and eventually the end of the simulation.

[23]Unity component which physically model a vehicle wheel

[24]Script used to save the actual state of the current simulation

Figure 5.14: User braking too close to the fire during the simulation

A Vehicle Data script was then created to connect all the previous behaviours, along with offering a way to manage some of the usual car features, like:

- turning on and off the car;
- turning on and off lights;
- enabling and disabling the hazard lights;
- enabling and disabling the radio.

NPC cars were created starting from the player car and cleaning up the code from the unnecessary features.

5.2.9 Core Game Logic

Indispensable part of any 3D application, the core game logic is necessary to manage the flow of the simulation, to interconnect the various behaviours and to transfer data through scenes.

The Tunnel Controller script was developed as director of the tunnel scene, managing the player spawn, the traffic, the fire evolution and the fire procedures; along with the Tunnel Controller, the Stage Controller stores the actual results of the player and manages the end of the simulation.

Loading a new scene in Unity entails the destruction of every GameObject, making it difficult to maintain in memory the state of the player. In order to avoid this, a Global Controller script has been developed as static behaviour[25], preventing the

[25] Unity scripts usually derive from a behaviour class to allow them to be attached on GameObjects

destruction at the load of a different scene.
Inside the Global Controller, the following fields were created:

- the selected language;
- the selected platform;
- the user result in case of end of the simulation.

5.2.10 Main Menu

The Main Menu scene was conceived as a test room for introducing the user to the locomotion and interaction methods, once selected the proper locomotion device.
Once launched the executable, the user finds himself inside the by-pass shelter S18, in front of an UI which can be managed through the previously mentioned ray-cast logic. Once selected language, locomotion mode and eventually the tracking device, the user has the chance to move around and interact with SOS telephone, extinguisher and a locker containing a couple of objects which can be grabbed.
After this initial familiarization, the user has to place himself or herself in front of a Start button (in order to start aligned with the car) and then can start the simulation by interact with it. At the end of the simulation the user is brought back to the Main Menu scene, and the result of the simulation is made available over an UI panel.

Figure 5.15: User familiarizing with the extinguisher logic during the initial training in the Main Menu

Figure 5.16: Result panel at the end of the simulation

5.2.11 Tunnel fire procedures

An essential part of the simulation is the reproduction of the official procedures activated by the tunnel Authority in case of fire; using the data provided by SITAF, the procedures have been reproduced as accurate as possible.

At the beginning of the simulation, the tunnel is not on alert; the semaphores are green, the half-barriers are raised and the radio is broadcasting the usual message for normal situations. As soon as the fire becomes visible from the player, the Tunnel Controller automatically starts the fire procedures, mimicking the behavior of the fire detection system employed inside the real tunnel.

The fire procedures consists of:

- broadcasting the emergency message through the radio frequencies;
- enabling the visual (flashing lights) and audio (official jingle) indicators for the nearest shelter;
- turning the semaphores to red;
- lowering the half-barriers;
- enabling the large stop signals beside shelters along with their flashing yellow semaphores;
- enabling the ventilation audio effect inside every shelter.

Once launched, the fire procedures will remain enabled till the end of the simulation.

5.2.12 Optimization methods and other expedients

Given the complexity of geometries, behaviours and visual effects working altogether during the simulation, a huge work of optimization was necessary to maintain the performance required for a smooth VR experience.

A LOD management, for example, was used to reduce the impact of the farthermost geometries and fire effects, exploiting the unity LOD component by manually defining the LOD levels and manually selecting the component enabled in every level.

As for the vehicles, all the other models had undergone many simplification and optimization passes in order to reduce the complexity of the scene.

Being the initial fire and the tracking lights extremely hard to see in the distance due to the resolution of Vive™, is has been necessary to overcome the display limits through an expedient. Using a script, the scale of these objects was linked to the distance from the player, applying a larger scale at distance, and lowering during the approaching.

A similar script, was also used to keep two black planes at the clipping distance minus one unit, in order to avoid holes in the fog during the intoxication status (the fog is only applied on meshes).

Chapter 6

Evaluation

6.1 Introduction

The work in object performed a comparison of the user-friendliness and the motion sickness between two different Virtual Reality locomotion strategies wearing the HTC® Vive™ device. In the first case, the user has been equipped with an omni-directional treadmill (Cyberith Virtualizer) which allowed free movement, in the second case the user, while staying in place, generated the movement by swinging back and forth the arms (Arm-Swinging). Thirty users were selected for the test, fifteen per locomotion device; twenty-nine of them completed the simulation, while a single user withdrew from the simulation before the end because of extreme cyber-sickness symptoms.
Someone referred a slight worsening of some symptoms already present before the simulation, but overall the group of tester did not suffer from particular worsening of the previous health state. The biggest differences were observed among the evaluation of the comfort and of the usability of the locomotion method.

6.1.1 Usability

Usability is a multidimensional term which refers to different aspects of a product or of a system, and it includes factors such the ease of use, the manipulation of the errors, the help, etc. Shackel (1981)[17] stated that: "The usability of a computer is measured by how easily and how effectively the computer can be used by a specific set of users, given particular kinds of support, to carry out a fixed set of tasks, in a defined set of environments".

With the growing interest for Virtual Reality applications, there is a urgent need to quantify the benefits the VR technology can bring, because the cost for supplying a VR solution could be high when high performance are required.
When people experience VR systems, usually complain about many aspects related with the UI, the resolution of the device and the method of interaction.
When the complaint is about the jerkiness of the display, the natural assumption would be that the computer is probably not capable of sustaining the proper refresh rate. However, a more detailed analysis would show that this latency could derive

from a various number of system areas, for example from the head tracking system, the graphic engine, a badly-designed model, an inappropriate simulation or even from the interaction device.
Only after an accurate interrogation followed by a punctual analysis the real cause could be discovered and fixed. The user should not be aware of the internal design of the VR system, but the perception over the usefulness of the system is fundamental in order to develop effective VR systems.

Hence, "the three-dimensional nature of VR has introduced new issues concerning the most suitable way to design input devices in order to make movement around a VE as natural and intuitive as possible." (Sturman and Zeltzer, 1994 as cited by Sarah Nichols, 1999)[14]. The locomotion inside the VE represents at the moment the major challenge for the VR designers. With the increase of the computational power of the graphic processors, and with the increasing realism of the virtual environments, providing an effective locomotion method would represent the missing piece in order to maximize the sense of presence of the user inside the environment. Usability measures are necessary in order to evaluate the developed locomotion methods.

6.1.2 Motion Sickness

One of the most common errors within the design of VR applications is believing that being immersed inside the virtual environment would be enough to define the simulation as realistic. As matter of fact, the immersion is not free of problems.
As Kay M. Stanney and others (1999)[18] said "varying degrees of motion sickness-like symptoms have been reported in nearly every flight simulator fielded by the military services (Kennedy and others, 1989). Similar symptoms, sometimes known as cybersickness, are now being reported with increasing regularity by users of virtual environment (VE) devices (Kennedy and others, 1995) and the problem seems even greater for VE devices". These manifestations derive from the motion sickness.

The motion sickness, also known as kinetosis and travel sickness, is a condition in which there is a disagreement between perceived movement and the vestibular system's sense of movement.
This is exactly the case of VR, where the user could be subjected to virtual movement while staying still in the real world. When wearing and HMD, there are numerous anomalies which could bring to modification of the perception, including visual distortions and positioning errors.
The touch and the pertinence allow the user to perceive the dimension, shape and position of the objects; as ay M. Stanney and others (1999)[18] stated, "Gibson (1966) has suggested that coordinated locomotion throughout an environment is a result of a learned correlation between the optic flow that indicates self-motion and proprioceptive feedback. When in a VE, a person will use their proprioceptive senses to operate the haptic interface (e.g. glove), joystick, mouse, or other device that moves them about the virtual world. Based on observations in our laboratory, when a person initially interacts with a VE their movements are relatively jerky and uncoordinated. James and Caird (1995) attributed this difficulty to a lack of correlation between the visual scene provided by the HMD and the haptic experience"

As P.A. Howarth and M. Finch (1999)[10] stated "The cardinal signs of motion sickness are nausea and vomiting, changes in pallor, and cold sweating (Reason and Brand, 1975; Benson, 1984, 1988). However, these signs are normally preceded by some (if not all) of the following physiological symptoms: stomach awareness, dizziness, bodily warmth, headache, sweating, increased or decreased salivation, drowsiness, and excess wind."

6.2 Methodology

The premsise of the test was that the users would have tried the experience with one locomotion method only, in order to avoid learning effects. The test sessions were performed in Turin and covered two days; the treadmill testing was performed inside the SiTI VR laboratory, and the Arm-Swinging testing inside the laboratory of the GRAINS group at the Department of Control and Computer Engineering (DAUIN) of Politecnico di Torino.

For the scenario evaluation, we gathered data in order to.

- formulate general observations regarding quality of the simulation in terms of usability;
- operate a comparison between two locomotion methods, Arm-Swinging and treadmill, in order to establish if they were equivalent or if one would have directly impacted on the performance during the simulation.

For this analysis, we used a custom version of the VRUSE[11] questionnaire along with the SSQ[1][12] questionnaire.

For the performance evaluation, the following data have been gathered.

- log of the simulation outcome (errors, time-stamps, total time);
- recording of the in-game footage (few representative samples);
- recording of the real-life footage (in parallel with in-game recording);

This second analysis was then used to formulate assumptions on the user behaviour.

[1]Simulator Sickness Questionnaire, a method for quantifying simulator sickness

6.3 Subjects

The thirty voluntaries were composed by twenty-two males and eight females (Figure 6.1), all in good health and between 20 and 40 years old. Most of them were SiTI employee or students from the Politecnico di Torino; few of them had a good level of knowledge about the VR device in use, but no one was proficient about serious gaming, or ever travelled through the Frèjus tunnel. However, most of them had a general background about gaming, VR and road safety measures (Figure 6.2).

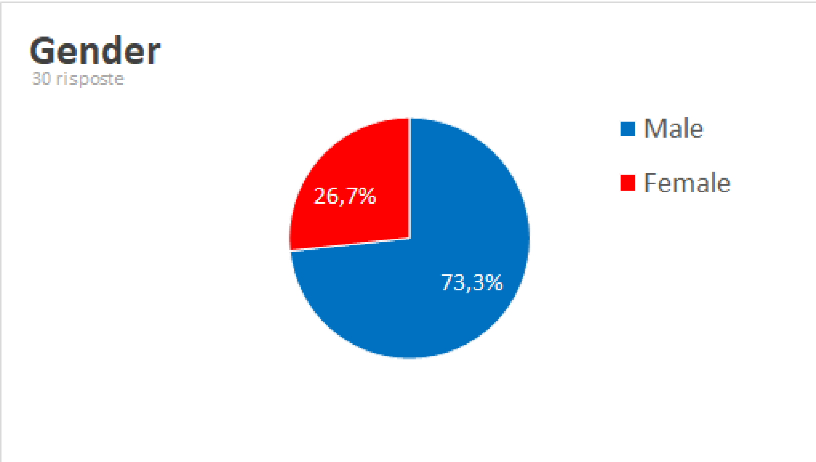

Figure 6.1: Gender of the users

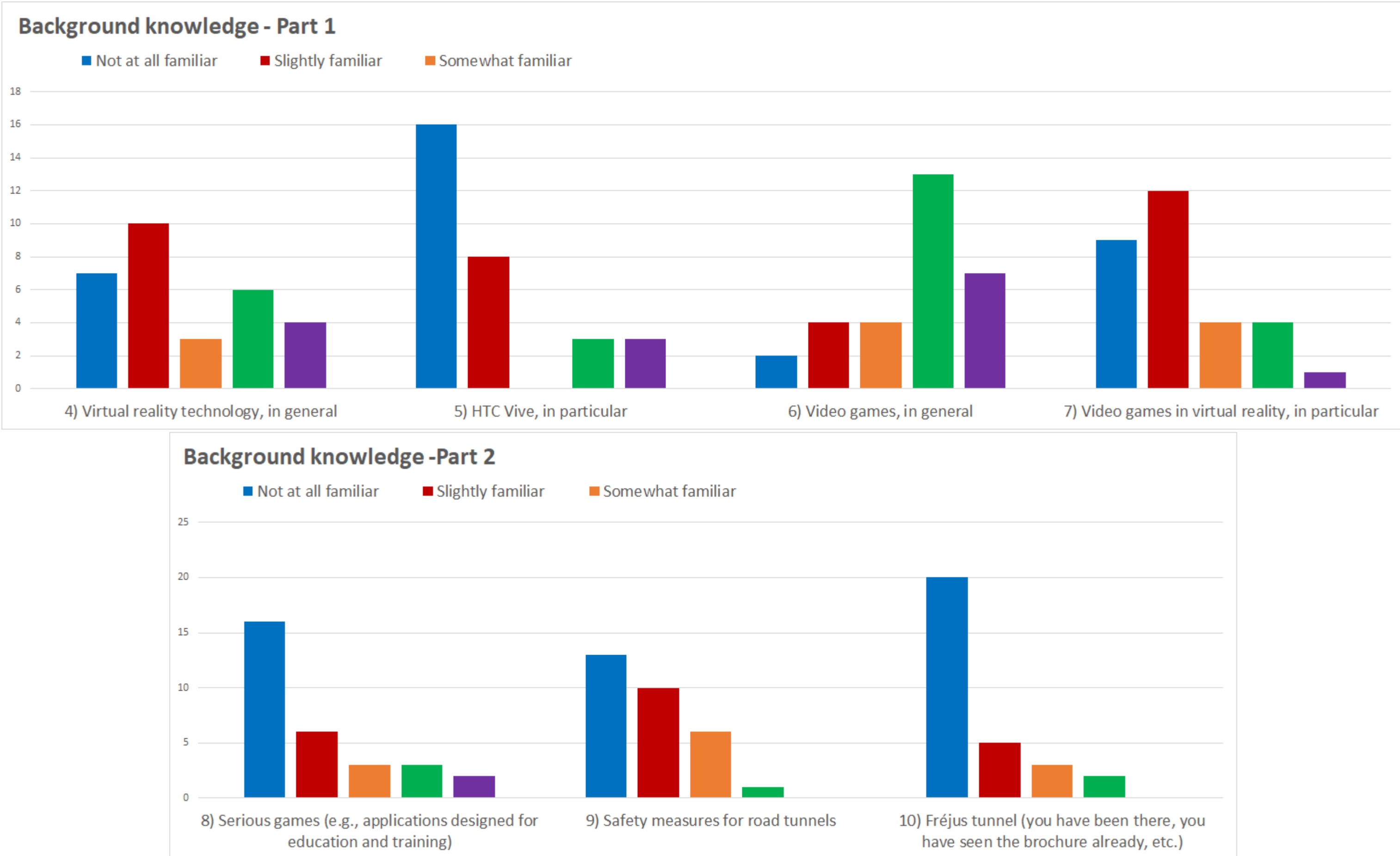

Figure 6.2: Background knowledge of the users

6.4 Supports

6.4.1 Presentation

In order to introduce the user to the simulation and to the technologies he was going to use, it was necessary to arrange different presentations, one per locomotion method.
Presentations had the purpose of giving a first understanding about the general usage of the controllers and about the locomotion system, as well as about the purpose of the simulation. The user should have made aware of being inside a real situation with real consequences, along with familiarizing with the interaction tools.
After the introduction, the user was given a chance to perform a training session inside the VE; for the purpose, a test room was equipped with some of the available interactions (a SOS telephone, an extinguisher and a locker) and made navigable through the selected locomotion method.

6.4.2 VRUSE Questionnaire

The VRUSE represents a standard evaluation method for the usability of VR interfaces. The purposes of the diagnostic tool are to provide a sensitive computer-based diagnostic aid to assist in usability evaluations in virtual interfaces, along with providing a structured method of capturing important user feedback about the usability issues of a VR system.
The VRUSE it therefore a considerable indicator for the problematic areas of a VR user interface.
It uses a five points likert scale[2], and the questionnaire is subdivided in specific categories. For this testing, the questionnaire has not been completely used, and it was adapted in order to evaluate the locomotion method along with the interaction one already covered. Table 6.1 summarizes the key aspects of the questionnaire. The complete evaluation questionnaire can be found in the Appendix A.

[2]Strongly Disagree, Disagree, Undecided, Agree, Strongly Agree

Table 6.1: Usability key factors and objectives

Usability Factor	**Objective**
Functionality	The interface should be able to provide the level of functionality (control) the user expects in order to complete a task
Locomotion	The user should be able to move inside virtual environment in a natural manner
Interaction	The user should be able to interact with and control the virtual environment in a natural manner
Flexibility	The VR system should not constrain the user who should be able to interact with the system in a flexible manner
Simulation Fidelity	In order to be useful a VR system needs an underlying model or simulation to control the virtual environment
Sense of Immersion/Presence	A VR system should allow a user to feel part of (or immersed in) a virtual environment
Overall System Usability	Overall a VR system should be intuitive and easy to use

6.4.3 SSQ

In order to measure the cyber-sickness experienced after the exposition to the VE, the users compiled the SSQ. The SSQ questionnaire mentions sixteen symptoms, each of which is correlated in terms of severity levels (none, slight, moderate, severe), with a maximum score of 235, and it is delivered before and after the simulation, in order to measure the impact of the exposure to the VR system on voluntaries.
A diagnostic score procedure is used to obtain a global score reflecting the general discomfort known as the total severity score (TS).
The SSQ also provides scores over three sub-scales representing the different dimensions of the cyber-sickness (nausea, oculomotor disorder and disorientation).

6.5 Procedure

The users were received one by one inside the laboratories, and then were introduced to the experience. With the help of the presentation, context and goals were explained. The official brochure was displayed and commented.
Then, they were asked to compile the first part of the questionnaire regarding.

- personal data (1-3)
- background knowledge (4-10)
- sickness (SSQ) (11-26)

In brackets the questions belonging to the each section of the evaluation questionnaire in Appendix A.

Successively, the devices had been showed and the controller buttons had been commented. After that, the locomotion method was been explained.
Finally, the subject were equipped with the VR devices and transported in the Main Menu of the application; here they were able to familiarize with the environment by testing the locomotion mode and by interacting with some sample items available (Figure 6.3).

Figure 6.3: User familiarizing with the interaction system inside the test scene

After the simulation, users had been helped free themselves from the VR devices, and briefly asked to complete the second part of the questionnaire, composed by.

- functionality (27-33)
- locomotion (34-43)
- interaction with objects (44-54)
- sickness (SSQ) (55-70)

- flexibility (71-74)
- simulation fidelity (75-84)
- sense of immersion and presence (85-94)
- overall system usability (95-106)
- simulation report (107-115)

6.6 Results

6.6.1 Application Evaluation

6.6.2 General Observations

In terms of Functionality (Figure 6.4), the system responded in a positive way to the demand of control from users. In fact questions 27, 29 and 31 which respectively concern to the level of functionality, the ease of access and the comprehension showed values mostly leading towards the right (agree) while the questions regarding the the ambiguity (28), the difficulty of memorization (30) and the confusion (33) leaded to the left (disagree).

Figure 6.4: Functionality

For the Sickness (Figure 6.8 and 6.7), we performed the previously-mentioned calculus in order to obtain the subdivision into the three sub-scales (nausea, oculomotor, disorientation), along with the total score value, which resulted in four surprisingly low values (Figure 6.5). The given result, probably caused by the lack of significant differences between the before and after symptoms, could be related with the short duration of the simulation (Figure 6.6).

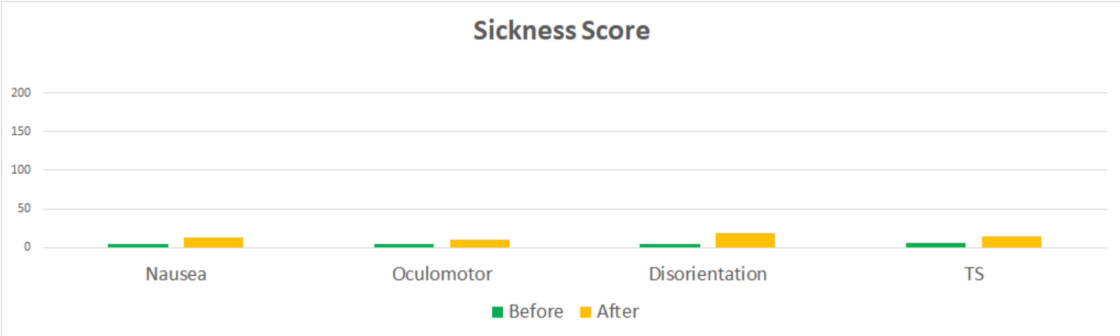

Figure 6.5: Sickness SSQ score

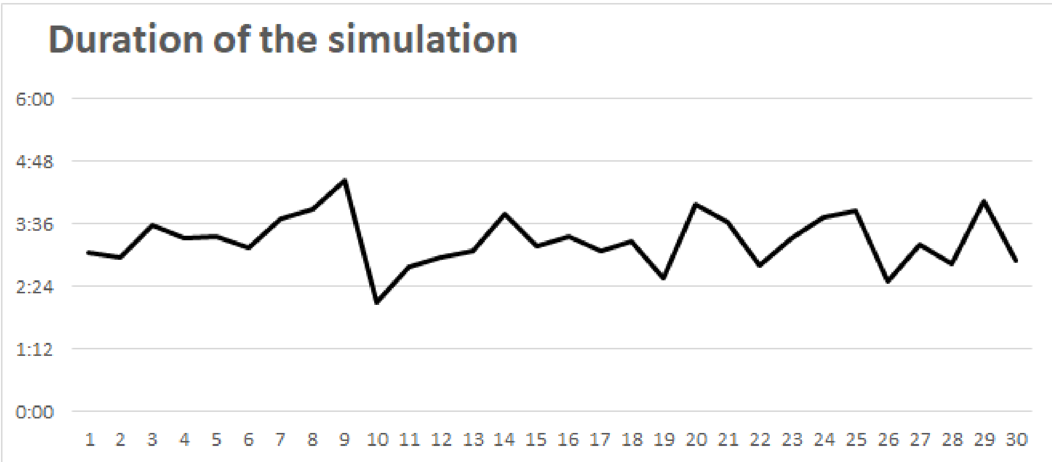

Figure 6.6: Duration of the simulation per user

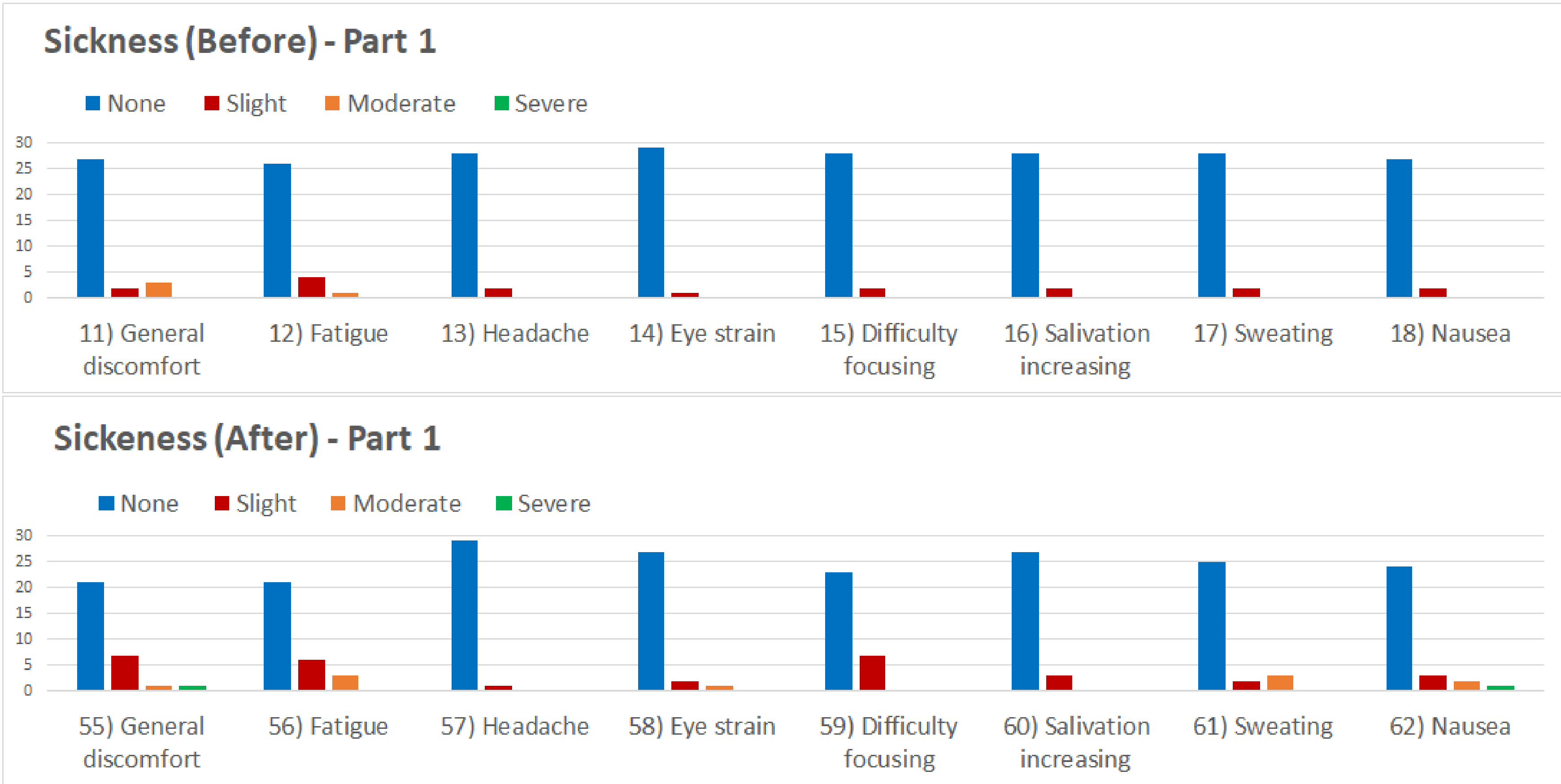

Figure 6.7: Sickness part 2 (Before and After)

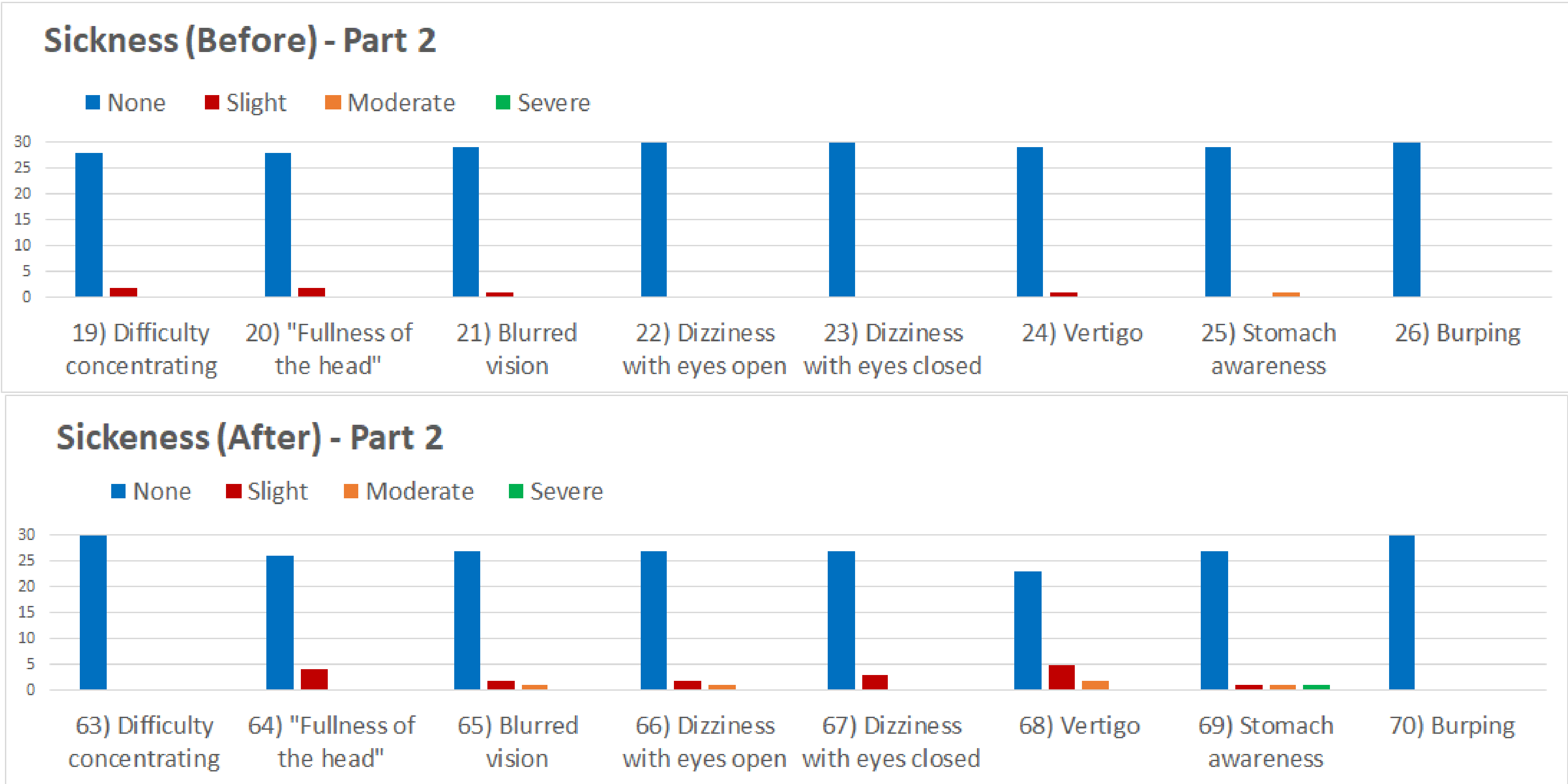

Figure 6.8: Sickness part 1 (Before and After)

Being the tests performed on two separated groups with two different locomotion methods, values obtained from the Locomotion section (Figure 6.9) did not provide in general significant data. Despite this, the question 39 showed that both systems were considered adequate for the purpose and the question 42 showed that both were easy to use effectively.

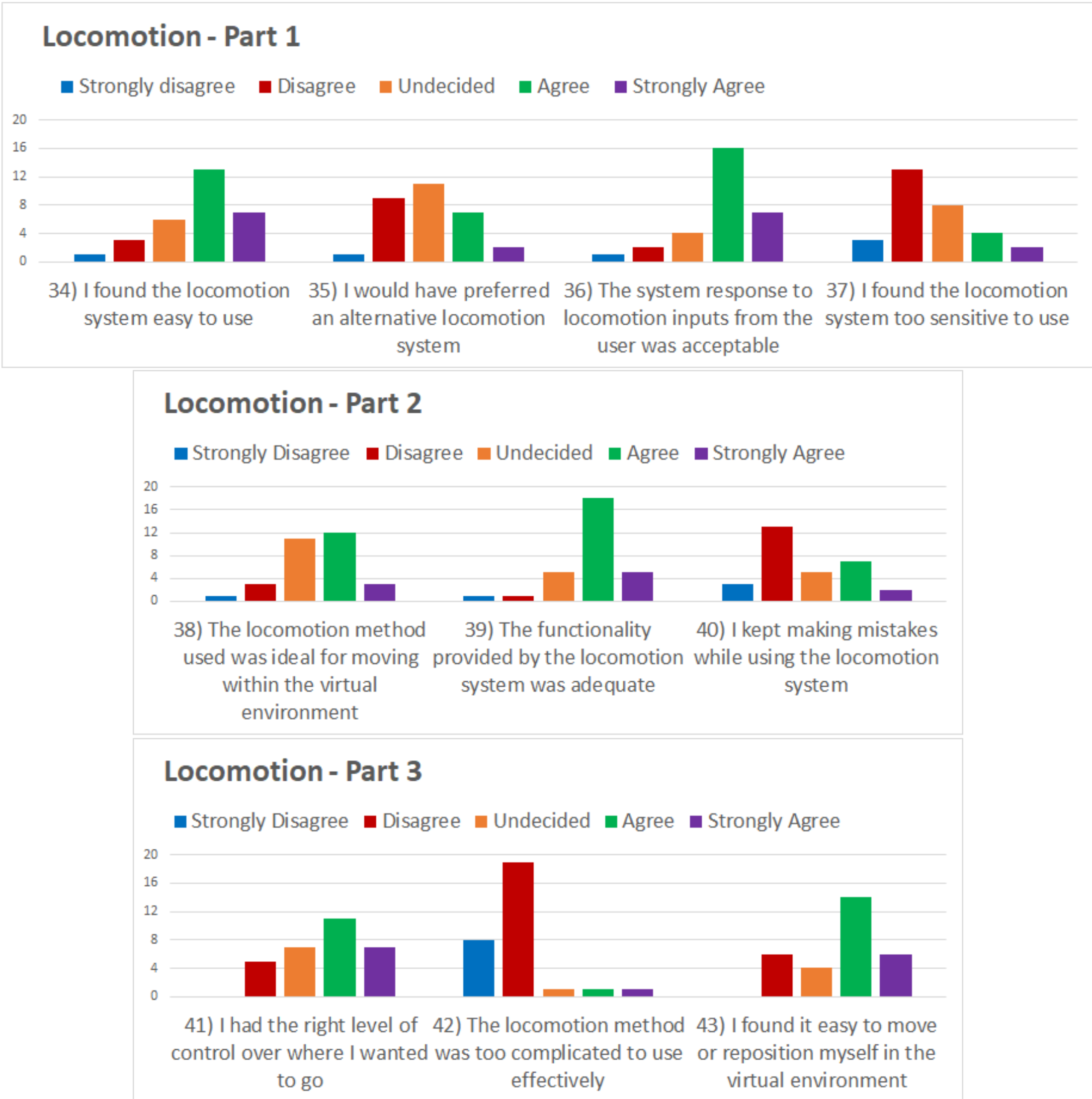

Figure 6.9: Locomotion

Regarding the Interaction with Object section (Figure 6.10), the overall positive can be seen as an high-quality indicator for the interaction system. This result could demonstrate the performance of the HTC® Vive™ in terms of interaction, but also the quality of the interaction system developed for the scenario.

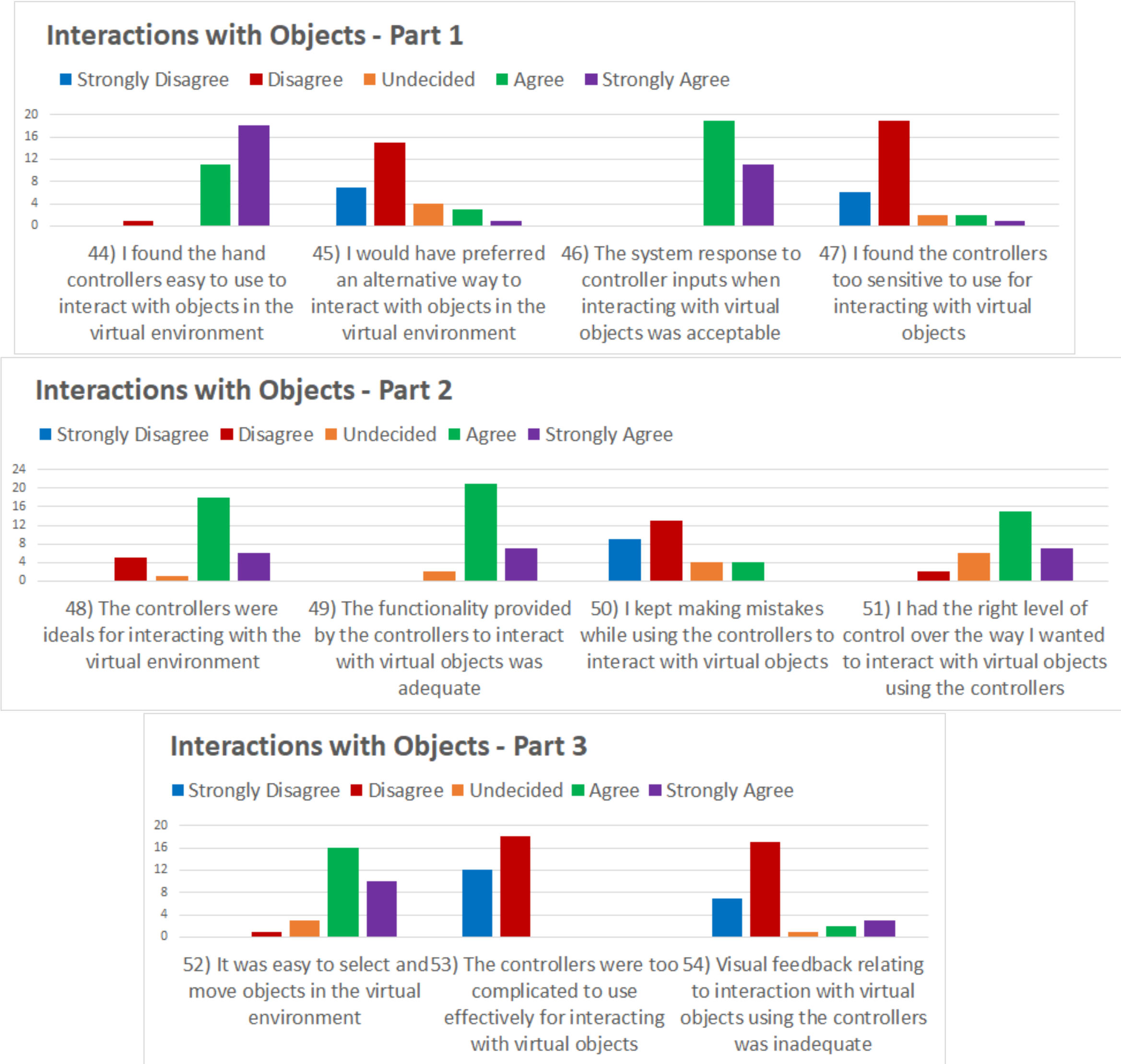

Figure 6.10: Interactions with Objects

For the Flexibility section (Figure 6.11), similarly to the interaction one, shows a positive trend in all the questions. Users found easy to perform tasks the way they chose (71) and in the order they chose (72).

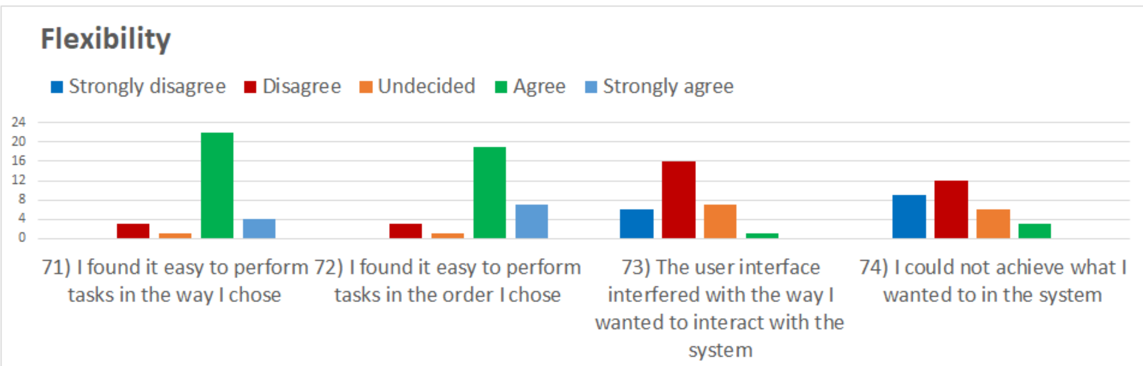

Figure 6.11: Flexibility

The simulation Fidelity (Figure 6.12), being one of the focus points of the work, showed a large number of highly positive results along with a general positive trend. The simulation was considered accurate (75), the VE was not seen as too complicated (82) and the system did not suffer from freezing or pausing at intervals (84).

Figure 6.12: Simulation Fidelity

For the Sense of Immersion and Presence section (Figure 6.13), the previous positive trend persists, showing an high level of immersion (85), a good level of realism of the objects in the environment (93) and an high sense of presence (87).

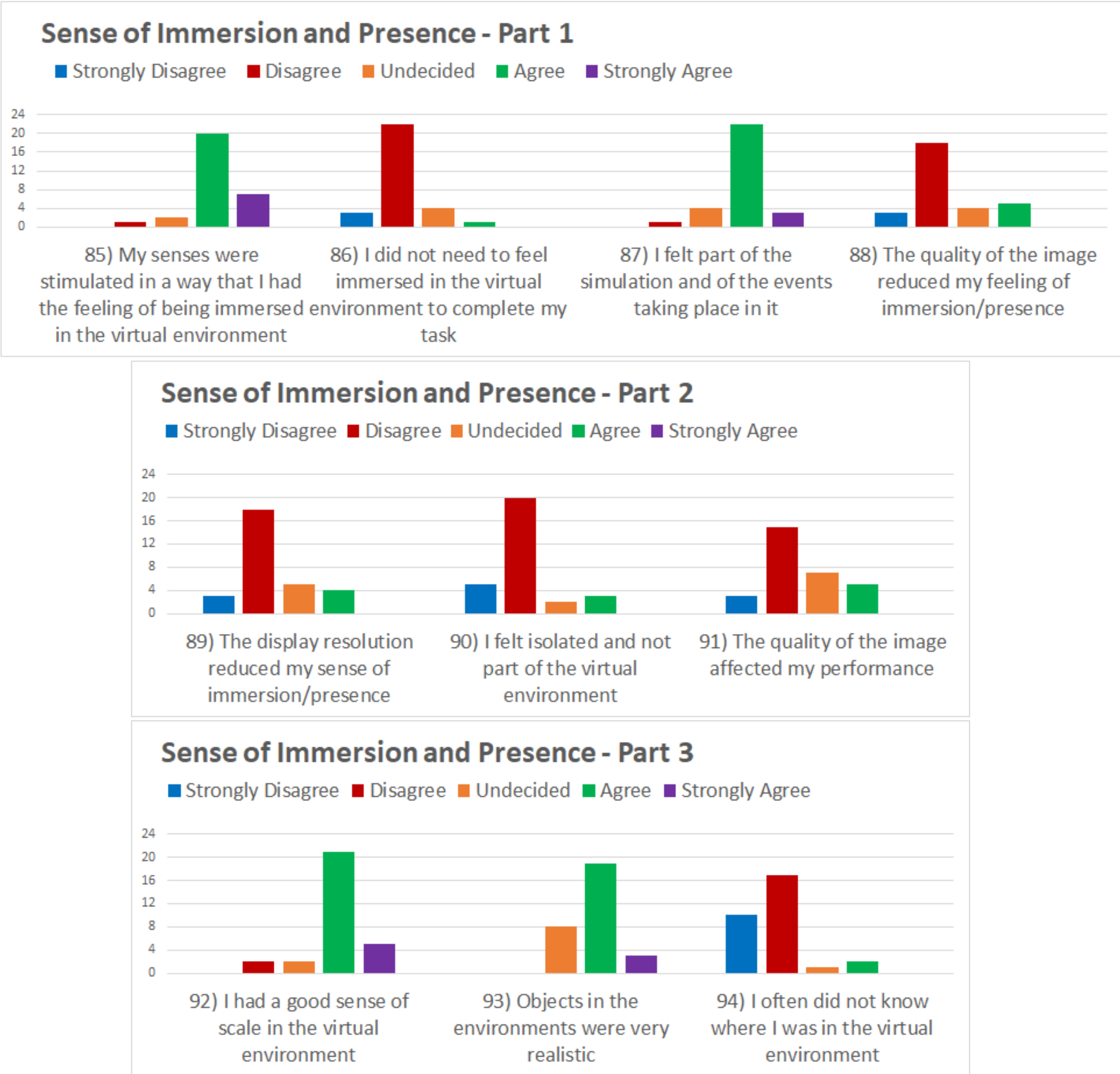

Figure 6.13: Sense of Immersion and Presence

Finally, the Overall System Usability (Figure 6.14) confirmed what already observed before; the response time (98) was considered acceptable, learning to use the system was considered easy (99), the system worked as expected (102) and the users enjoyed using the system (106).

Figure 6.14: Overall System Usability

6.6.3 Analysis based on the locomotion mode

A further investigation was performed by subdividing the users in two groups based on the locomotion system used during the test.
Locomotion data have been classified and subdivided on the basis of the method used by the participants. In order to evaluate the answers, a numerical scale was overlapped on the likert one. For the VRUSE:

- strongly Disagree = 0;
- disagree = 1;
- undecided = 2;
- agree = 3;
- strongly Agree = 4.

For the SSQ:

- none = 0;
- slight = 1;
- moderate = 2;
- severe = 3.

In order to evaluate the statistical significance of the collected answers, a two-tailed unpaired Student's t-test[3] was executed on the data. Hence, it has been possible to establish if the differences between the average values between the two groups (e.g. Arm-Swing and Treadmill) were or not result of the randomness, and, more precisely, if the two locomotion methods were equivalent or not.
The Student's t-test, starting from the values of the answers of two separated groups, calculates a p value which is usually compared with the standard threshold of 0.05; if the value is lower, the difference between the averages of the two groups is significant, whereas if it exceeds the value then the collected data cannot be used to validate hypotheses. There are then a couple of sub-levels of significance:

- $p >= 0.05$ not significant;
- $p < 0.05$ significant;
- $p <= 0.01$ highly significant;
- $p <= 0.001$ extremely significant.

In order to indicate the significance of the data represented inside the following histograms, the bars corresponding to answers with a non significant p value ($p >= 0.05$) will be characterized by a faded color, hence highlighting the significant ones. In the following analysis, Arm-Swinger will be referred as AS, and Virtualizer as TD.

The null hypothesis for this investigation is that the two methods are equivalent.

Concerning the Functionality section (Figure 6.15) in the VRUSE questionnaire (in Appendix A), significance differences between the average values were not observed. This could suggest that the two methods are considered equivalent from this point of view. Only the question 29 ($p = 0.023$) shows how, with the AS method, accessing all the functionalists of the system is considered slightly easier (3 vs 3.47).

[3] The t-test is a statistical hypothesis test in which the test statistic follows a Student's t-distribution under the null hypothesis

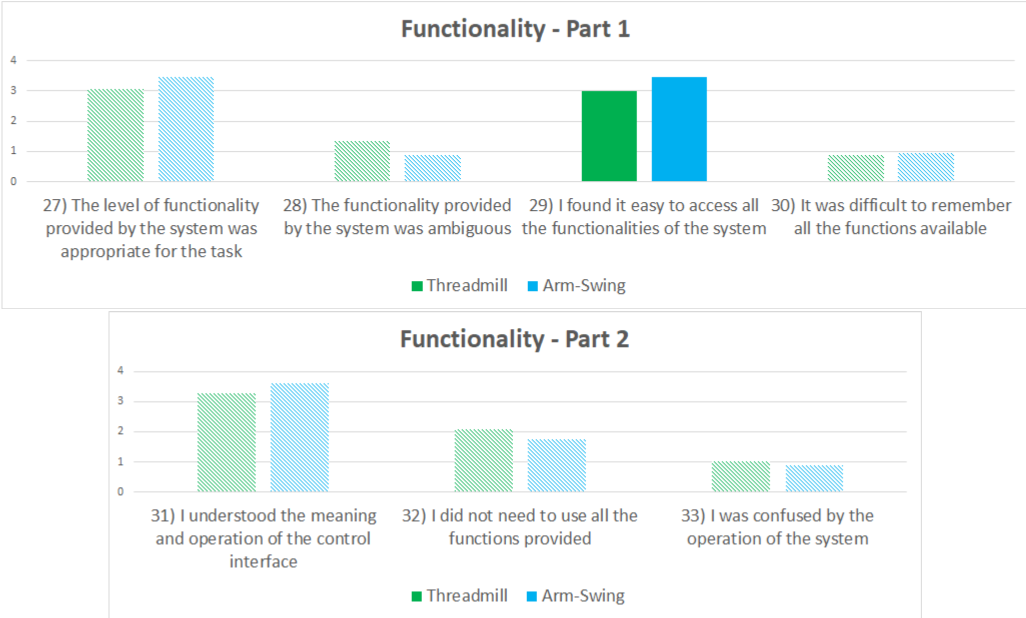

Figure 6.15: Functionality t-test

The analysis of the Locomotion section (Figure 6.16) showed an average value in favour of the AS. Question 34 ($p = 0.0001$) showed how AS was considered easy to use an higher more than the VT (3.5 vs 2.1) and question 36 ($p = 0.0003$) showed how it was also more well-performing in terms of system response to locomotion inputs (3.5 vs 2.3).

Question 38 ($p = 0.012$) showed how AS was seen as ideal for moving within the VE (2.98 vs 1.9), question 39 ($p = 0.019$) that the AS was seen as slightly more adequate as locomotion system (2.46 vs 3.2), and question 40 ($p = 0.0001$) showed that users kept making mistake while using the locomotion system much frequently with the VT (2.47 vs 1).

Question 41 ($p = 0.0005$), 42 ($p = 0.009$) and 43 ($p = 0.01$) respectively showed that AS provided more times the right level of control over the user wanted to go (3.02 vs 2.06), the VT was more frequently seen as too complicated to use effectively (1.34 vs 0.53) and that with AS was also easier to move or re-position inside the VE (3.13 vs 2.2).

Figure 6.16: Locomotion t-test

Regarding the Interaction section (Figure 6.17), the two locomotion methods basically resulted as equivalent. This outcome was mostly expected, since the interaction devices and logic was identical among the two versions. Interesting to note that being forced to hold two buttons and to swing the arms in order to move around with the AS did not have a significant impact on the interaction system (question 52).

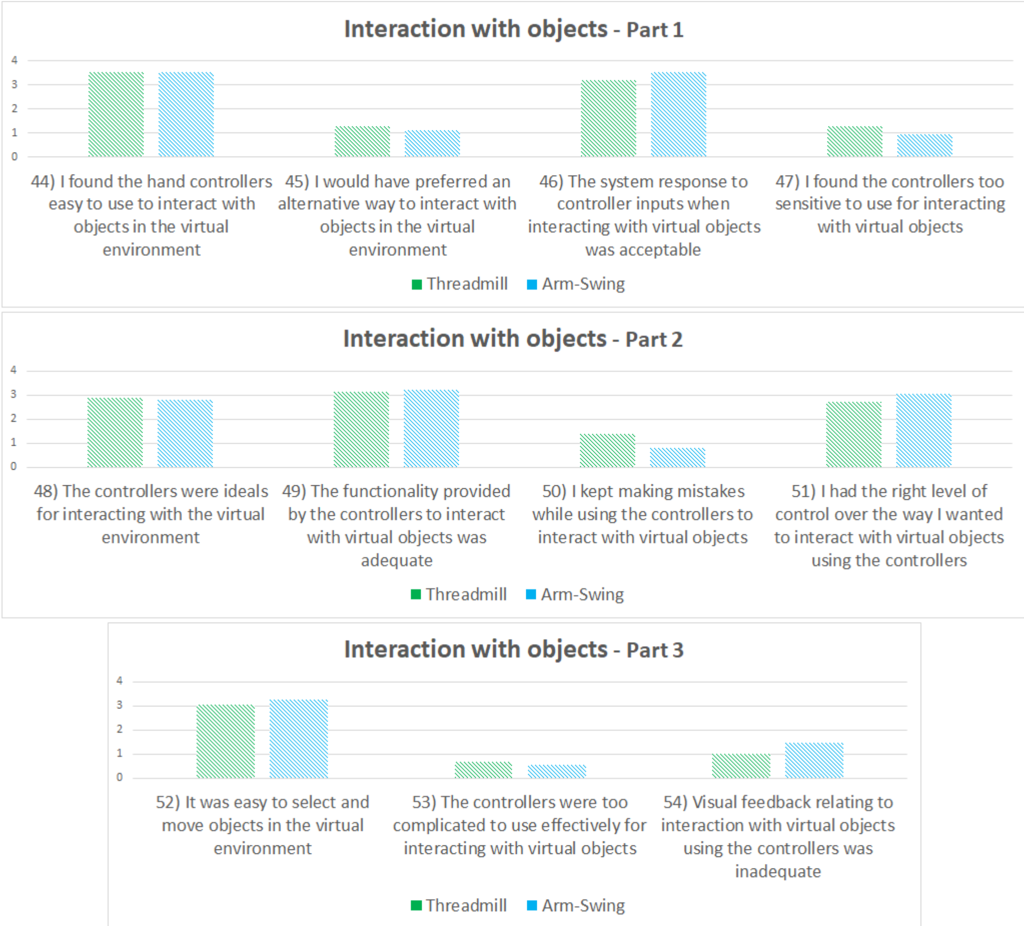

Figure 6.17: Interaction with Objects t-test

Another interesting note is that the two systems seem comparable in terms of Sickness (Figure 6.18). Despite the treadmill users suffered on average from worsening of the already present symptoms, the difference did not resulted significant for the t-test.

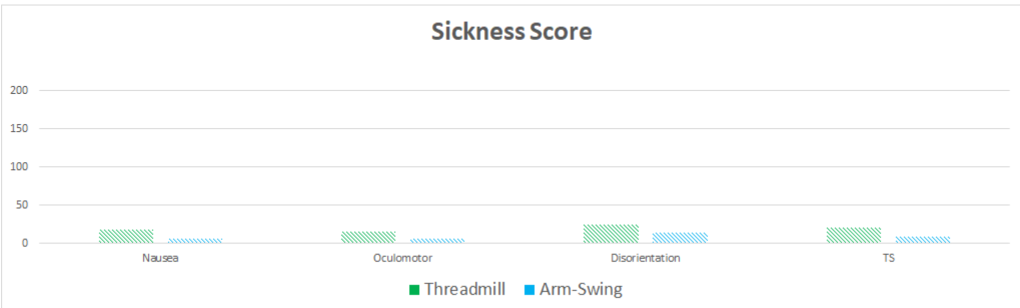

Figure 6.18: Sickness t-test

Regarding the Flexibility (Figure 6.19), question 72 ($p = 0.025$) shows that it has been easier to perform the tasks in the order the user chose with AS (3.3 vs 2.67). With the VT, question 74 ($p = 0.012$) shows how users were not able to achieve what they wanted to in the system more frequently (1.53 vs 0.67) for the user to perform what he or she wanted to do.

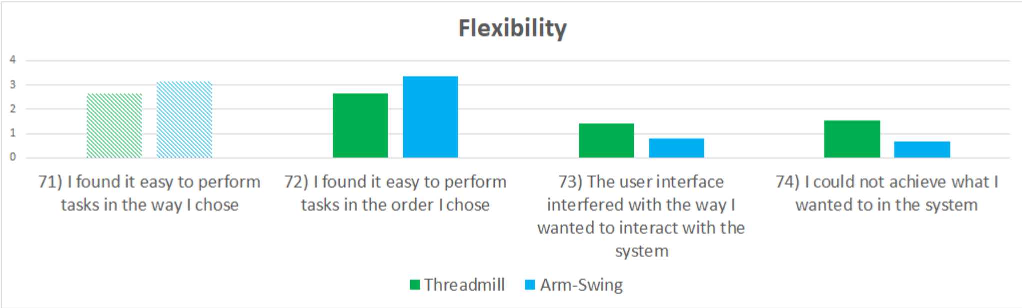

Figure 6.19: Flexibility t-test

For the Fidelity (Figure 6.20), question 83 (p = 0.007) shows how user thought more frequently with AS that the quality of the simulation enhanced their performance (2.6 vs 1.73), and question 84 (p = 0.015) showed how the simulation appeared to freeze more times for the VT users (0.7 vs 0.007), confirming how the locomotion device could impact on the feeling of performances.

Figure 6.20: Fidelity t-test

It has been also observed a correlation between the locomotion method and the Sense of Immersion and Presence (Figure 6.21). Question 94 (p = 0.38) shows how VT users often did not know where they were inside the VE more frequently (1.13 vs 0.53).

Unexpectedly, question 93 (p = 0.029) shows how users inside the VT considered the objects inside the VE less realistic than AS ones (2.6 vs 3.07).

Figure 6.21: Sense of Immersion and Presence t-test

For the Overall System Usability (Figure 6.22), question 95 ($p = 0.0034$) showed how the VT gave the impression of being working against the user more frequently (1.13 vs 0.33), and question 102 ($p = 0.0014$) showed that it worked more often in an unusual manner (1.17 vs 0.47).

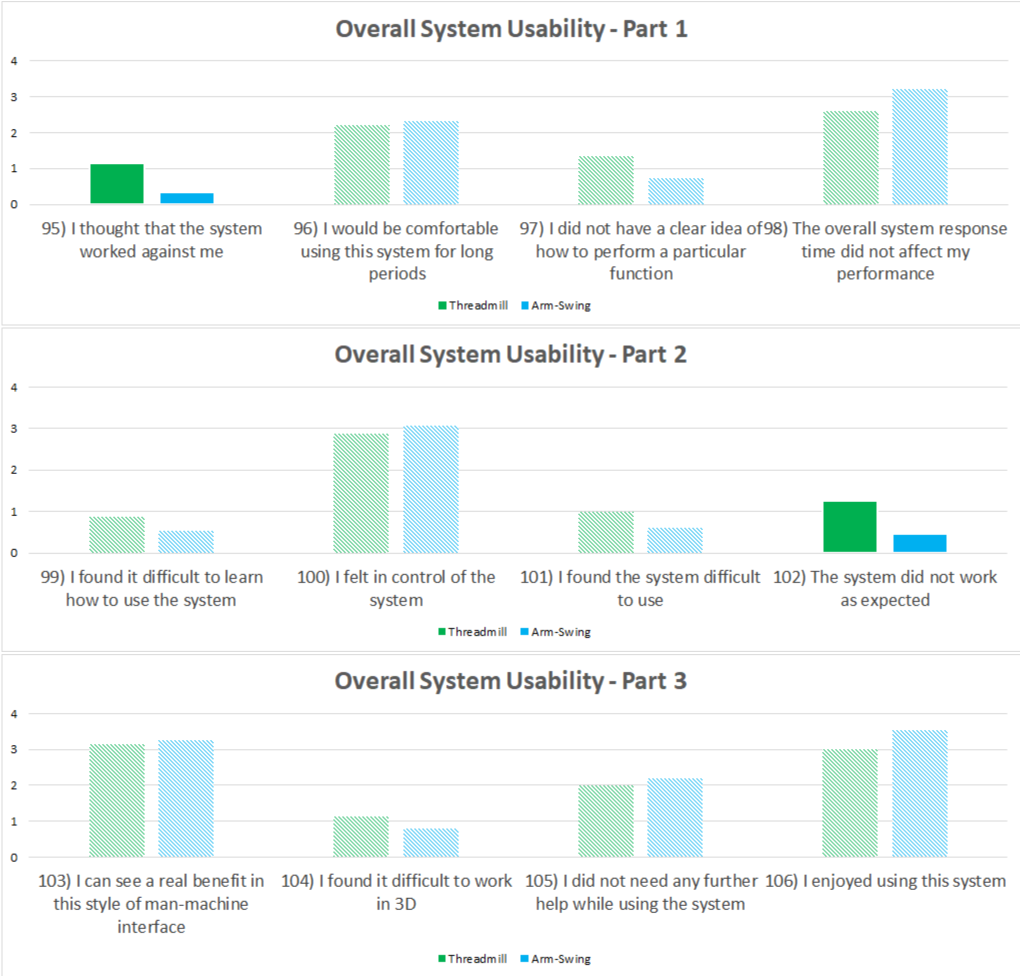

Figure 6.22: Overall System Usability t-test

In conclusion, among the two methods, the AS resulted the most well-performing and preferable.

6.6.4 User Behaviour

When it comes to User Behaviour (Figure 6.23 and Figure 6.24), the majority of the users (26) activated the alarm, only 19 asked for help through the SOS telephone, 18 reached the emergency shelter, 16 turned off the car engine and only 14 enabled the hazard lights.

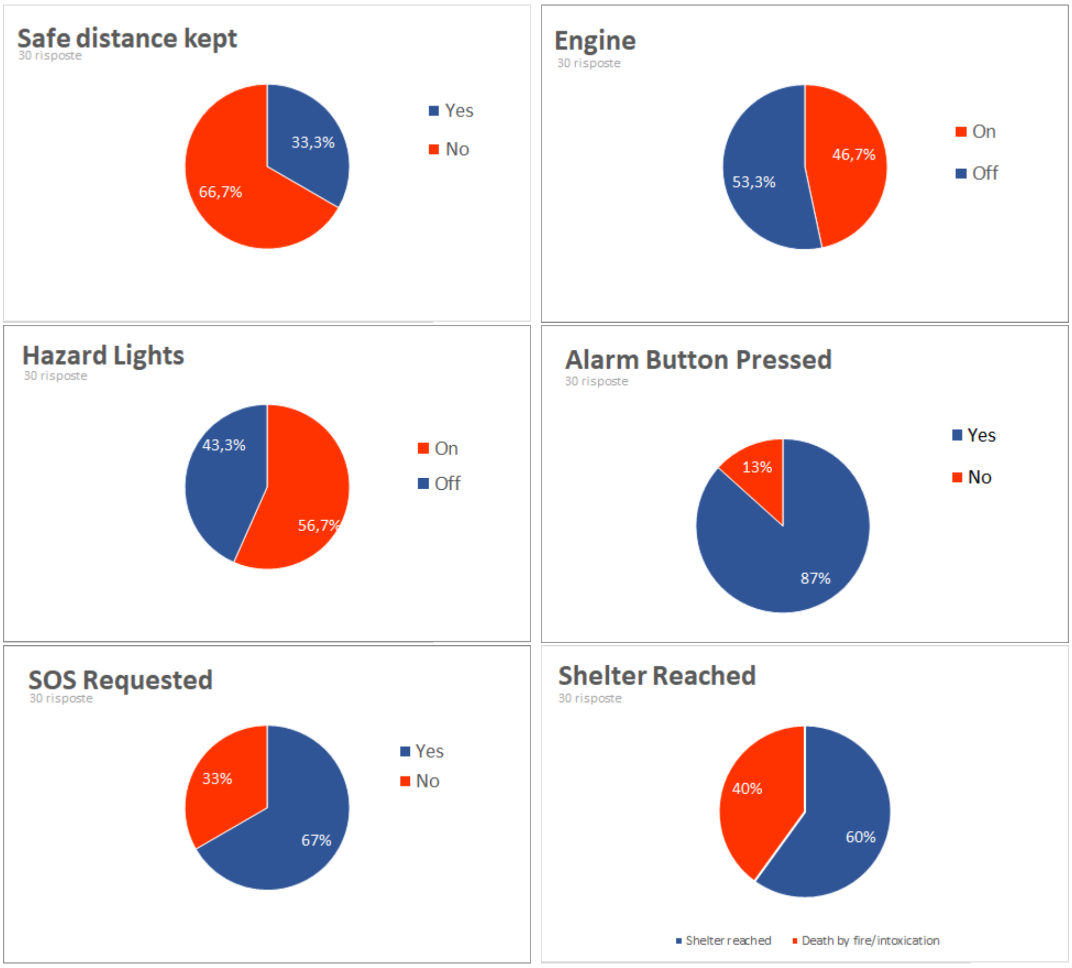

Figure 6.23: User Behaviour

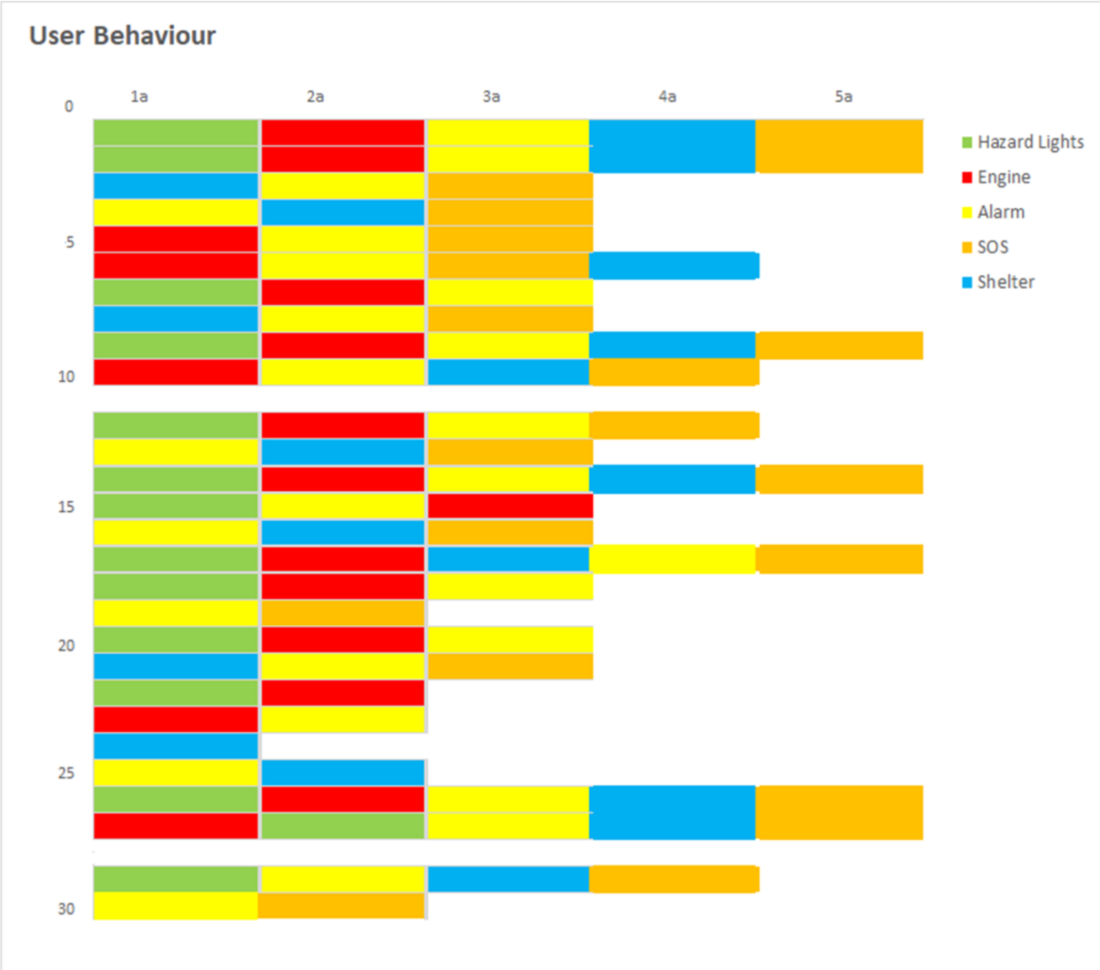

Figure 6.24: User Behaviour Time-lines

Some users, probably unaware of the possibility to move around the head inside the car, found difficult to localize the keys.

Although the initial explanation, no one used the brochure during the simulation. There have been cases of users died inside the car because they did not know what to do, or of users which, once gotten out from the car, died by intoxication while roaming in confusion.

There have been a single case of user died trying to open the door of an closed shelter (the A6 shelter, next to the S18, is going to be converted to technical room interdicted for the public soon) while being intoxicated by the smoke.

Some users re entered the car because, once got out, they remembered a required task the previously forgot; this usually caused the death of the tested, due to the quick spreading of the smoke.

When comparing again AS and VT (Figure 6.25), a correlation has been found between the reaching of the shelter ($p = 0.025$) and the the activation of the alarm ($p = 0.04$). These actions were performed an higher number of times by the AS users, hitting a peak of 100% for the alarm activation, against the 73% of the treadmill users (the shelter was reached by 80% of the Arm-Swing users and by 40% of the VT users).

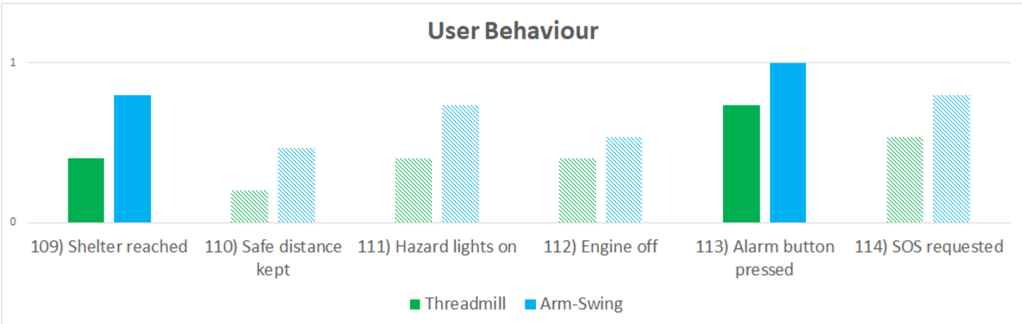

Figure 6.25: User Behaviour per locomotion mode

As before, no correlation is found between the subdivision (Figure 6.26) criteria and the time required to complete the simulation (again an average of 3 minutes, $p = 0.59$). Female users reached the shelter with more difficulty (25% vs 73%, $p = 0.027$) and also were dramatically less inclined to request help through the SOS telephone (12.5% vs 86%, $p = 0.00025$).

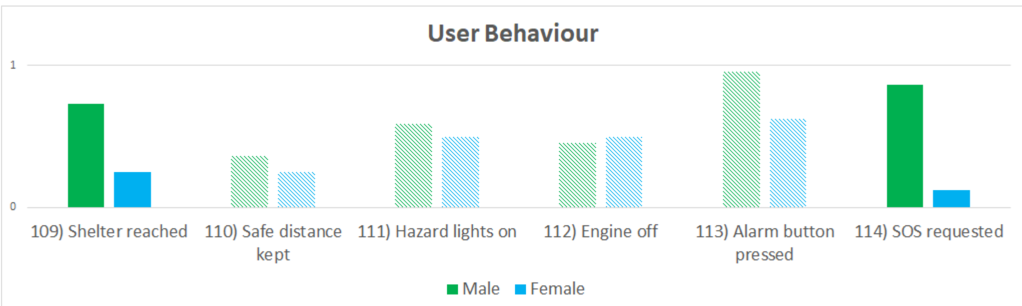

Figure 6.26: User Behaviour per gender

6.6.5 Analysis based on gender

Even in the case of a subdivision of users according to the gender specified before the test, the data remained insignificant for any further evaluation about the Sickness (Figure 6.27).

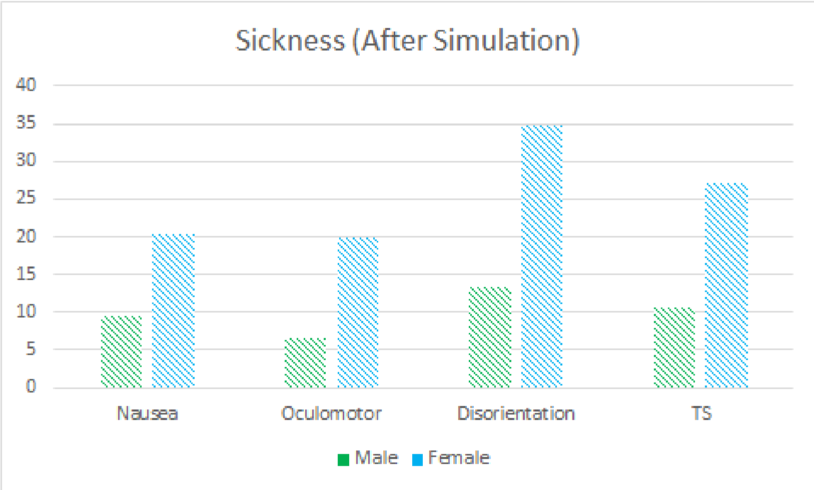

Figure 6.27: Sickness (After Simulation) per gender

The different subdivision shows that the female users needed the Immersion (Figure 6.28) in order to perform the required tasks in comparison with the male ones (1.23 vs 0.75, p = 0.036).

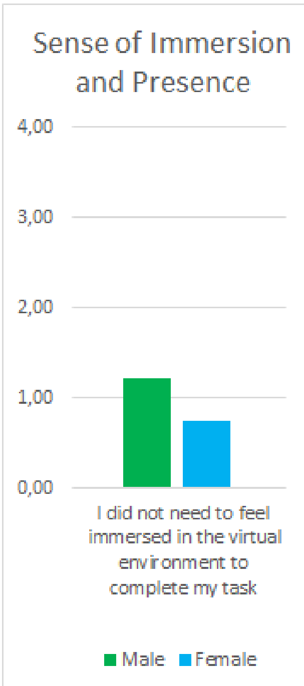

Figure 6.28: Sense of Immersion and Presence per gender

Furthermore, the female users (Figure 6.29) also required more help to complete the simulation than the male users (2.45 vs 1.125, p = 0.002).

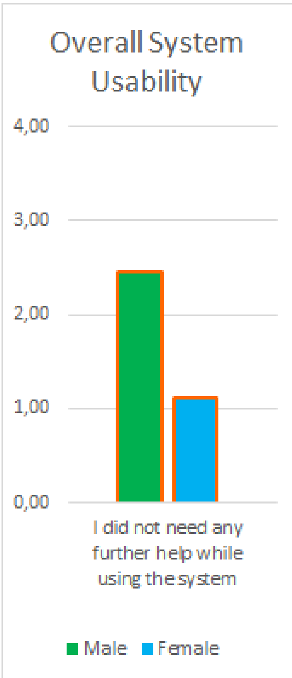

Figure 6.29: Overall System Usability per gender

Chapter 7

Conclusion and Future Development

7.1 Conclusions

The thesis work developed has led to a training tool, which could be employed for a broad spectrum of purposes, from the communication of the emergency procedures to the civilians to the evaluation of the security level of the selected scenario, e.g., in collaboration with emergency staff.

The analysis of the data obtained through the testing allowed to get interesting results about users' experience in the designed VE, especially considering the quality of the simulation and users' attitudes towards a specific interaction mode.

As secondary result of the testing, the simulation also highlighted some unforeseen recurring events:

- multiple users were unable to find the SOS niche, thus of requesting help through the communication module; when interrogated, they blamed the lack of signals as detailed as the shelter indications

- the brochure, despite being presented as an essential tool during the briefing stage, was only employed by an insignificant percentage of users, usually resulting in faulty execution of the tasks; when asked for explanations, the user admitted to not having felt the necessity to recall the tool, being too sure about on self to even suspect an eventual forgetfulness;

- some users, as result of disregarding the safe distance, blamed the fact that the radio emergency instructions were requesting to stop the vehicle at the red light, and being the vehicle on fire placed in correspondence of a semaphore led up to the above-mentioned failure;

- in certain cases, as result of confusing the SOS button with the SOS telephone, a group of users wasted precious seconds in waiting for a response which would never come, kept pressing the button waiting for a visual or sound feedback confirming the operation;

- it is interesting to note that the SOS button tends to become greatly more attractive for the user lost inside the smoke and dying due to intoxication; during the excited moments preceding the end of he simulation, the above-mentioned tool is seen as a desperate last resort and pressed multiple times in the forlorn hope of being somehow noticed and rescued.

Given this, the user feedback also drawn the attention over a number of issues still affecting the developed VE.
In the first place, some user's feedback revealed that the initial part inside the car apparently did not provide the same level of immersion in comparison with the rest of the simulation; this lies mainly in the lack of control over the vehicle, in the form of.

- acceleration;
- steering;
- possibility of making an U-turn;
- a chance of surpassing the fire.

Being thrown inside the car at the beginning of the simulation, seeing the car moving autonomously and having little to no interactions available probably led some users to believe they were just watching a passive 360° introduction; in support of this, many of the testers who forgot to perform one or more of the car tasks adduced as justification the fact they were not really driving the car, and that turning off the engine and enabling the hazard lights would have been two natural actions in the event that they were using a real vehicle.

Furthermore, the Arm-Swing testing showed huge drawback of the current implementation which could lead to a negative impact in terms of sickness. Blocking the tracking of the head movements along the X and Z axis in order prevent wall glitching caused a huge disorientation in the users who tried to move around the head; luckily, the issue becomes irrelevant as soon as the user starts walking through the Arm-Swing mechanic.
As workaround, replacing the head blocking (or maybe just hiding it) with a fade to black could be prevent the bothersome side effect.
A single user also experienced an increase of hearth-rate, but it was not recorded since it is not an usual symptom related with the cyber-sickness, not being included in the SSQ.

Regarding the quality of the simulation fidelity, feedback and evaluation result have been mostly positive or extremely positive, and the same can be said for the interaction logic apart from the locomotion. A single criticism was made regarding the fidelity and the immersion, and it concerned the lack of additional NPCs in the simulation; the earlier versions of the scenario had moving cars during the whole simulation and the truck driver was running around inside the tunnel, but these features were scrapped in response to early feedback.
Surely the addition of traffic, along with other stopped car and other NPCs escaping from the fire would have been a huge improvement in therms of immersion.

The activities also brought to light the limits of the actual technologies and devices available for VR; the limited resolution of the HMD, for example, had a negative impact on the perception of the farthermost objects inside the tunnel, making it necessary to find workarounds in order to obtain the desired result.
The evaluation of the locomotion methods showed their intrinsic weakness, being the principal reason of discomfort, sickness and lack of immersion.

The next step would be testing of the third locomotion mechanic implemented, the Foot-Swing, in the direction of comparing it with the Arm-Swing and the treadmill. Along with this, new locomotion modes could be investigated and tested in order to find the most suitable for the purpose.

Another possible development could be a testing session involving the regular users of the tunnel, in order to evaluate the differences in terms of results comparing the new data with the analysis already performed.

Finally, the training tool could be reshaped in order to support:

- new use-cases (a different kind of emergency situation, a different zone of the tunnel involved);
- new scenarios (a different tunnel);
- new target audiences (regular user, operators, etc.);
- new training modes (a multi-player build with two or more users, a control room mode useful to dynamically modify the scenario);
- new procedures (the fire department procedures, the civil defense procedures, etc).

It will be up to the involved parties to set the direction for the future development of the tool in question.

Appendix A

FrejusVR: evaluation questionnaire

A.1 Personal Data

1. Age
2. Gender
3. Company

A.2 Before simulation

A.2.1 Background Knowledge

How familiar are you with these subjects? (Not at all familiar, slightly familiar, somewhat familiar, moderately familiar, extremely familiar)

4. Virtual reality technology, in general
5. HTC Vive, in particular
6. Video games, in general
7. Video games in virtual reality, in particular
8. Serious games (e.g., applications designed for education and training)
9. Safety measures for road tunnels
10. Fréjus tunnel (you have been there, you have seen the brochure already, etc.)

A.2.2 Sickness

Right now, are you affected by any of these symptoms? (None, Slight, Moderate, Severe)

11. General discomfort
12. Fatigue
13. Headache
14. Eye strain
15. Difficulty focusing
16. Salivation increasing
17. Sweating
18. Nausea
19. Difficulty concentrating
20. "Fullness of the head"
21. Blurred vision
22. Dizziness with eyes open
23. Dizziness with eyes closed
24. Vertigo
25. Stomach awareness
26. Burping

A.3 After simulation

A.3.1 Functionality

How would you judge the functionality (control) provided by the system? (strongly disagree, disagree, undecided, agree, strongly agree)

27. The level of functionality provided by the system was appropriate for the task
28. The functionality provided by the system was ambiguous
29. I found it easy to access all the functionalities of the system
30. It was difficult to remember all the functions available
31. I understood the meaning and operation of the control interface
32. I did not need to use all the functions provided
33. I was confused by the operation of the system

A.3.2 Locomotion

How would you judge the locomotion method (the method used to move in the virtual environment)? (strongly disagree, disagree, undecided, agree, strongly agree)

34. I found the locomotion system easy to use
35. I would have preferred an alternative locomotion system
36. The system response to locomotion inputs from the user was acceptable
37. I found the locomotion system too sensitive to use
38. The locomotion method used was ideal for moving within the virtual environment
39. The functionality provided by the locomotion system was adequate
40. I kept making mistakes while using the locomotion system
41. I had the right level of control over where I wanted to go
42. The locomotion method was too complicated to use effectively
43. I found it easy to move or reposition myself in the virtual environment

A.3.3 Interaction with objects

How would you judge the use of controllers for interacting with virtual objects? (strongly disagree, disagree, undecided, agree, strongly agree)

44. I found the hand controllers easy to use to interact with objects in the virtual environment
45. I would have preferred an alternative way to interact with objects in the virtual environment
46. The system response to controller inputs when interacting with virtual objects was acceptable
47. I found the controllers too sensitive to use for interacting with virtual objects
48. The controllers were ideals for interacting with the virtual environment
49. The functionality provided by the controllers to interact with virtual objects was adequate
50. I kept making mistakes while using the controllers to interact with virtual objects
51. I had the right level of control over the way I wanted to interact with virtual objects using the controllers

52. It was easy to select and move objects in the virtual environment
53. The controllers were too complicated to use effectively for interacting with virtual objects
54. Visual feedback relating to interaction with virtual objects using the controllers (e.g., buttons highlighting) was inadequate

A.3.4 Sickness

How do you feel after the simulation (right now, are you affected by any of these symptoms)? (None, Slight, Moderate, Severe)

55. General discomfort
56. Fatigue
57. Headache
58. Eye strain
59. Difficulty focusing
60. Salivation increasing
61. Sweating
62. Nausea
63. Difficulty concentrating
64. "Fullness of the head"
65. Blurred vision
66. Dizziness with eyes open
67. Dizziness with eyes closed
68. Vertigo
69. Stomach awareness
70. Burping

A.3.5 Flexibility

How would you judge the flexibility provided by the system? (strongly disagree, disagree, undecided, agree, strongly agree)

71. I found it easy to perform tasks in the way I chose
72. I found it easy to perform tasks in the order I chose

73. The user interface interfered with the way I wanted to interact with the system
74. I could not achieve what I wanted to in the system

A.3.6 Simulation fidelity

How would you judge the fidelity of the simulation? (strongly disagree, disagree, undecided, agree, strongly agree)

75. The simulation was accurate
76. The simulation was too simplistic to be of use
77. I was impressed with the way I could interact with the simulation
78. The simulation behaved in a very unusual manner
79. Objects in the virtual environment moved in a natural manner
80. I felt disorientated in the virtual environment
81. I had the right level of control over the simulation
82. The virtual environment was too complicated
83. I thought that the quality of the simulation enhanced my performance
84. The simulation appeared to freeze or pause at intervals

A.3.7 Sense of immersion and presence

How would you judge your sense of immersion and presence in the virtual environment? (strongly disagree, disagree, undecided, agree, strongly agree)

85. My senses were stimulated in a way that I had the feeling of being immersed in the virtual environment (being there, not at SiTI)
86. I did not need to feel immersed in the virtual environment to complete my task
87. I got a sense of presence (i.e., part of the simulation and of the events taking place in it)
88. The quality of the image reduced my feeling of immersion/presence
89. The display resolution reduced my sense of immersion/presence
90. I felt isolated and not part of the virtual environment
91. The quality of the image affected my performance
92. I had a good sense of scale in the virtual environment
93. Objects in the environments were very realistic
94. I often did not know where I was in the virtual environment

A.3.8 Overall system usability

How would you judge the system usability? (strongly disagree, disagree, undecided, agree, strongly agree)

95. I thought that the system worked against me
96. I would be comfortable using this system for long periods
97. I did not have a clear idea of how to perform a particular function
98. The overall system response time did not affect my performance
99. I found it difficult to learn how to use the system
100. I felt in control of the system
101. I found the system difficult to use
102. The system did not work as expected
103. I can see a real benefit in this style of man-machine interface
104. I found it difficult to work in 3D
105. I did not need any further help while using the system
106. I enjoyed using this system

A.3.9 Simulation report

107. Which locomotion system has been tested? (Virtualizer treadmill, Arm-Swinging, Foot-Swingig)
108. Duration of the simulation (time)
109. What caused the end of the simulation? (death by fire/intoxication, shelter reached)
110. Safe distance kept (yes, no)
111. Hazard lights on (time)
112. Engine off (time)
113. Alarm button pressed (time)
114. SOS requested (time)
115. Shelter reached (time)

Bibliography

[1] Stefano Torresani Bernardo Magrì Massimo Berti. "La galleria di sicurezza del Frejus. La variante del progetto definitivo per l'apertura al traffico della galleria al termine dei lavori". In: *Strade & Autostrade* (2013). URL: http://www.stradeeautostrade.it/infrastrutture/gallerie-e-tunnelling/2013-12-03/la-galleria-di-sicurezza-del-frejus-3786/.

[2] M. Boere. *The Simple Fire Engine*. 2014. URL: https://www.assetstore.unity3d.com/en/#!/content/23064.

[3] Moohyun Cha et al. "A virtual reality based fire training simulator integrated with fire dynamics data". In: *Fire Safety Journal* 50.Supplement C (2012), pp. 12–24. ISSN: 0379-7112. DOI: https://doi.org/10.1016/j.firesaf.2012.01.004. URL: http://www.sciencedirect.com/science/article/pii/S0379711212000136.

[4] P. Cialdini. "La sicurezza delle infrastrutture: casi di progettazione flessibile". In: Turin, Italy, 15 June 2007. URL: https://ingegneria.tesionline.it/ingegneria/articolo.jsp?id=1136.

[5] European Parliament and Council of European Union. *Directive 2004/54/EC of the European Parliament and of the Council of 29 April 2004 on minimum safety requirements for tunnels in the Trans-European Road Network*. 2004. URL: http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=OJ:L:2004:167:TOC.

[6] Frederic Font, Gerard Roma, and Xavier Serra. "Freesound Technical Demo". In: *ACM International Conference on Multimedia (MM'13)*. ACM. Barcelona, Spain: ACM, 21/10/2013 2013, pp. 411–412. ISBN: 978-1-4503-2404-5. DOI: 10.1145/2502081.2502245.

[7] *free3d.com*. URL: https://free3d.com.

[8] "Frejus, bruciano 5 mezzi: un inferno Due morti, venti persone intossicate". In: *La Repubblica* (Apr. 2005). URL: http://www.repubblica.it/2005/f/sezioni/cronaca/tunnelfrej/tunnelfrej/tunnelfrej.html.

[9] "Gottardo: incendio nel tunnel, almeno 14 morti". In: *La Stampa* (25 October 2001). URL: http://www.corriere.it/Primo_Piano/Cronache/10_Ottobre/24/gottardo.shtml.

[10] Peter Howarth and M Finch. "The nauseogenicity of two methods of navigating within a virtual environment". In: 30 (Mar. 1999), pp. 39–45.

[11] Roy S. Kalawsky. “VRUSE—a computerised diagnostic tool: for usability evaluation of virtual/synthetic environment systems”. In: *Applied Ergonomics* 30.1 (1999), pp. 11–25. ISSN: 0003-6870. DOI: https://doi.org/10.1016/S0003-6870(98)00047-7. URL: http://www.sciencedirect.com/science/article/pii/S0003687098000477.

[12] Robert S. Kennedy et al. “Simulator Sickness Questionnaire: An Enhanced Method for Quantifying Simulator Sickness”. In: *The International Journal of Aviation Psychology* 3.3 (1993), pp. 203–220. DOI: 10.1207/s15327108ijap0303_3. eprint: http://dx.doi.org/10.1207/s15327108ijap0303_3. URL: http://dx.doi.org/10.1207/s15327108ijap0303_3.

[13] Corpo Nazionale dei Vigili del Fuoco - Servizio Tecnico Centrale Ispettorato per la Formazione Professionale Ministero dell’Interno. *Attrezzature di Protezione Individuale dei VV.F.* 2010. URL: http://www.vigilfuoco.it/sitiVVF/ascolipiceno/downloadFile.aspx?s=85&f=27266.

[14] Sarah Nichols. “Physical ergonomics of virtual environment use”. In: *Applied Ergonomics* 30.1 (1999), pp. 79–90. ISSN: 0003-6870. DOI: https://doi.org/10.1016/S0003-6870(98)00045-3. URL: http://www.sciencedirect.com/science/article/pii/S0003687098000453.

[15] Groupement d’Exploitation du Fréjus Protezione Civile Prefettura di Torino Protection Civile Prefecture de la Savoie. *Piano di Soccorso Binazionale Traforo Autostradale del Frejus.* 2008. URL: http://www.cittametropolitana.torino.it/cms/risorse/protciv/dwd/prevenzione_pianificazione/pppp_ppe/integrazioni_pppp/04_PPPP2014_ALL_09_4_PIANO_FREJUS.pdf.

[16] Ultra Real. *Ultra Real Fire Effects Volume 1 : Static Fire Props.* 2014. URL: https://www.assetstore.unity3d.com/en/#!/content/20923.

[17] B. Shackel. “The Concept of Usability”. In: *Proc IBM Software and Information Usability Symposium.* Poughkeepsie, NY: IBM, 15-18 September 1981, pp. 1–30.

[18] Kay Stanney et al. “Motion sickness and proprioceptive aftereffects following virtual environment exposure”. In: 30 (Mar. 1999), pp. 27–38.

[19] “Tir in fiamme nel tunnel del Frejus”. In: *La Stampa* (29 November 2010). URL: http://www.lastampa.it/2010/11/29/cronaca/tir-in-fiamme-nel-tunnel-del-frejus-nzqhaSAotLapc2ggYgcDgM/pagina.html.

[20] “Un errore italiano ha aggravato il rogo”. In: *La Repubblica* (July 1999). URL: http://www.repubblica.it/online/fatti/bianco/rapporto/rapporto.html.

[21] V. Verda et al. “L’Approccio multidisciplinare per la gestione dell’emergenza degli incendi boschivi. L’esperienza del Politecnico di Torino nel progetto europeo AF3 “Advanced Forest Fire Fighting””. In: Turin, Italy, 20 March 2017.

[22] Wikipedia. *HTC Vive — Wikipedia, The Free Encyclopedia.* [Online; accessed 16-October-2017]. 2017. URL: https://en.wikipedia.org/w/index.php?title=HTC_Vive&oldid=805583971.

[23] Wikipedia. *Oculus Rift — Wikipedia, The Free Encyclopedia.* [Online; accessed 16-October-2017]. 2017. URL: https://en.wikipedia.org/w/index.php?title=Oculus_Rift&oldid=805634982.

[24] Z. Xu et al. "A virtual reality based fire training simulator with smoke hazard assessment capacity". In: *Advances in Engineering Software* 68.Supplement C (2014), pp. 1–8. ISSN: 0965-9978. DOI: https://doi.org/10.1016/j.advengsoft.2013.10.004. URL: http://www.sciencedirect.com/science/article/pii/S096599781300166X.