POLITECNICO DI TORINO

MASTER's Degree in AEROSPACE ENGINEERING

MASTER's Degree Thesis

Virtual Reality-based training scenario of a pressurized lunar rover's cockpit

Supervisors

Prof. Alfonso PAGANI

Prof. Magesh CHANDRAMOULI

Dr. Dario ZAMANI

Candidate

Marco CRESTINI

DECEMBER 2025

Virtual Reality-based training scenario of a pressurized lunar rover's cockpit

Marco Crestini

Abstract

During the last years a renewed interest in space exploration has ignited new attention and investments in designing pioneering solutions to be used in the upcoming manned missions to our natural satellite. In a wider ground exploration perspective, astronauts should perform long extra vehicular activities to conduct scientific experiments and technological validation campaigns. To ease this objective, some of the most important national space agencies are funding new research programs regarding the development of a vehicle able to transport and house its passengers into a pressurized environment, the so called pressurized lunar rover. Meanwhile, the industrial sector is exploiting a range of technologies known as Extended Reality (XR), a spectrum that includes Augmented Reality (AR) and Virtual Reality (VR); they allow to perceive the surrounding world adding layers of information and can even bring users into a completely different environment. The main objective of this project is to simulate, for training purposes, the cockpit of a pressurized lunar rover using the Virtual Reality technology. Users will be able to interact with devices that control the inner environmental conditions of the vehicle and with which the system can be set up in preparation for a driving mission.

ACKNOWLEDGMENTS

I would like to give special thanks to Alfonso Pagani and Magesh Chandramouli for creating the opportunity to work on this project and for following me throughout the entire process.

I would also like to thank Dario Zamani for being a very supportive figure during these months and Aleeha Zafar for the patience she put into helping me solve all my technical problems.

I would like to acknowledge the LPR Research Group part of the Space It Up! program, the Department of Mechanical and Aerospace Engineering of Politecnico di Torino, the Department of Computer Information Technology and Graphics of Purdue University Northwest, Thales Alenia Space Italia and the U.S. Consulate General of Naples

Table of Contents

List of Figures

List of Tables

Acronyms

AR	Augmented Reality.
AV	Augmented Virtuality.
CAD	Computer-Aided Design.
CGI	Computer-Generated Imagery.
CSA	Canadian Space Agency.
ESA	European Space Agency.
EVA	Extravehicular Activity.
HMD	Head-Mounted Display.
HUD	Head-Up Display.
ISS	International Space Station.
JAXA	Japan Aerospace eXploration Agency.
LOLA	Lunar Orbiter Laser Altimeter.
LSS	Life Support System.
MFD	Multi-Function Display.
MR	Mixed Reality.
NASA	National Aeronautics and Space Administration.
RE	Reality.
SANS	Spaceflight Associated Neuro-ocular Syndrome.
SBD	Simulation-Based Design.
UI	User Interface.
VR	Virtual Reality.

XR	Extended Reality.

Chapter 1

Introduction

1.1 Back to the Moon

Nowadays' interest in human space exploration has grown to the point that human beings are expected to return to the Moon with the Artemis program in few years. Artemis is a very ambitious campaign developed by the National Aeronautics and Space Administration (NASA) in collaboration with other national space agencies like the European Space Agency (ESA), the Japan Aerospace eXploration Agency (JAXA), the Canadian Space Agency (CSA) and other private companies. The program plans to bring astronauts to the Moon's South Pole and to build a lunar space station, the Gateway . These objectives are very challenging for all the players involved and they require pioneering technological developments. Accordingly to the Concept of Operations of Artemis III, astronauts will spend around six days on the lunar surface with the expectation of extending this stay in the near future; many systems are being further developed to allow it [1] [2].

Some members of industry and academia have proposed a new way of facing the design challenges: shifting from a product to a user centered design approach. This change of paradigm is justified by the new peculiarities of human lunar exploration, with astronauts performing several critical tasks for the success of the mission. A promising technology that could help users immerse themselves into operational scenarios is the Extended Reality (XR), a term that contains the different grades of hybridization between real and virtual environment. Since the last decade of 20^{th} century, industrial sector has experimented applications of XR and nowadays some technologies have reached a full operative capability to be insert in the design flow. Obviously human space exploration could greatly benefit from the addition of this range of applications to the range of technologies already employed.

1.2 XR overview

In 1994 Paul Milgram et al. [3] came up with the concept of a reality-virtuality continuum: XR can be seen as a spectrum, having at one end Reality (RE), then moving towards the other extreme there are Augmented Reality (AR), Mixed Reality (MR), Augmented Virtuality (AV) and finally Virtual Reality (VR) [4]. Shifting from one side to another, several information, images and videos progressively overlap the real field of view, creating a new environment where users can accomplish tasks in a more efficient way thanks to the multiple inputs they receive.

Probably the most investigated domains of the continuum have always been AR and VR. As many creative innovations, the first impulse came from an artistic field, cinematography in this case. In 1955, Morton Heilig, an American cinematographer and inventor, wrote a paper entitled "The cinema of the future" where he stated that if an artist controls the multisensory inputs an audience receives, he could trick them to be dived in first person into the movie scenario. Heilig also estimated a distribution of sensors contribution to define a person's perception: 70% sight, 20% hearing, 5% smell, 4% touch, 1% taste [5]. He was inspired by first 3D experiments and by *Cinerama*, a technology of movie projection that cast images on huge curved surfaces. Between 1957 and 1962 he built and patented the *Sensorama*, a device that allowed spectators to immerse themselves in five films he had created. This is considered one of the first examples of virtual reality device [6].

In the space sector first experiments with VR were done by NASA, resulting in the creation of a VR laboratory at Johnson Space Center, Houston in the early 1990s. The purpose of this research plant was to test procedures for Extravehicular Activities (EVAs) in microgravity. The main success of this program was the Hubble Space Telescope repair in 1993; astronauts involved in the mission were trained using a virtual environment reproducing the operative conditions they would have met in orbit. More recently the focus has been shifted to human-machine interface, developing a tool simulating the interaction with a computer in the Fluid Science Laboratory [7].

The expression Augmented Reality was introduced in 1992 by Thomas Preston Caudell, a Boeing researcher; he created an AR application to visualize assembly diagrams that manufacturers had to consult over and over during their work. The simultaneously acquisition of real world elements and virtual data in the same field of view, allowed workers to speed up their tasks and reduce mistakes. These kind of applications have been widely investigated during last decades in the medical sector too, where the level of jobs to carry out is often of critical importance, with life-related concerns. The video game market is always looking for new technologies to introduce that could revolutionize game experience for players. AR was surely a great innovation and many developers have created successful applications based on it [8].

Also VR has stepped forward during years since the first visionary projects of Morton Heilig, allowing access to a wide audience. Nowadays many products have been commercialized by tech companies to take advantage of this promising idea; the head-mounted device known as VR headset contains two displays, placed in front of each eye, and sensors like gyroscopes and accelerometers to create an artificial equivalent of field of view motions. It also has to take account of the reflexes generated by the vestibular system: the vestibulo-ocular reflex maintains images focused on the retinas of the eye when movement occurs and the ocular counter-rolling reflex keeps images horizontal when the head tilts [9] [10] [11] [12].

It's pretty evident that the immersion provided by commercial devices reaches only two senses: sight and hearing. Some critics have pointed out how the lack of a full perception of the virtual environment leads to a minor embodiment in it. This issue, according to them, marks a serious gap compared to real-world analogue studies and results in a low scientific validity of VR simulations. Dufresne et al. in [13] tried finding a way to add more perceptive inputs to user, focusing on kinesthetic and tactile sensations, summed up in the term haptics. To do so they integrated some items to the remote controllers: a device simulating astronauts gloves and a very simple physical mockup of the Apollo Lunar Surface Experiment Package. The gloves had the main purpose of reducing the movement capability of users hands, thus increasing the embodiment. The mockup was made of a horizontal rod to be grabbed and its mass was one sixth of the real object one, resembling its lunar weight. Authors, discussing the results obtained, showed that wearing gloves has positive influence on the sense of body ownership while the use of mockup doesn't seem to increase the feeling of being present in the virtual scenario.

Another limitation of some type of VR simulations is the connection between movement within it and in RE. Some headset are able to track user's walking and to reproduce it in the virtual environment. Laboratories where tests are performed have spacial limitations while simulated terrain can be created without any boundaries. In [14] Nilsson et al. presented redirection techniques to trick the user and to make him believe he's walking in a straight direction while in physical space he is describing circular trajectories. Authors summarized two main way of redirecting one's walking: by manipulating the mapping and by manipulating the architectural properties of the virtual environment. These studies are very interesting and necessary cause there are some side effects like motion sickness, spatial performances and cognitive load that need to be assessed.

The introduction of VR headset during spaceflight has allowed to monitor effects of the Spaceflight Associated Neuro-ocular Syndrome (SANS) too. This problem affects a great number of astronauts during their stay in microgravity and it seems to be caused by the constantly high value of the intracranial pressure due to fluid shift towards the upper body. The symptoms are a reduced visual acuity, visual scotomas and headaches. Other human body systems reach a new balance after a certain time in space, called adaptation, while for this syndrome, the longer astronauts remain, the worse it gets [9] [10] [11] [12]. Thanks to researches like [15], with the VR headset used as a diagnostic tool for eye-related pathologies, the comprehension of this phenomenon should increase, allowing for longer stays.

1.3 Related work

This section contains papers I have read to establish a starting point for the development of my work. They delve into applications of the VR technology to the space sector, mainly to lunar environment.

In [7] was proposed a new methodology to assist space mission design. The first problem authors faced, was the ability to explore and improve a future lunar outpost. To do so, in the paper, a VR lunar outpost was built allowing users to explore it via a first person point of view. This tool helped investigate ergonomics and details of the operations, hard to assess in other ways.

Another very critical point was the location of the outpost. The most important element for an inhabited settlement is water: basic processes like oxygen generation, electromagnetic shielding and fuel cell regeneration highly rely on this resource and an easy access to it could be crucial. The region chosen for landing is the Moon's South Pole where recent researches have outlined the presence of great quantity of icy water. In the article they tried finding some precise coordinates. The criteria that led to the evaluation of the most suitable point was the illumination percentage over a lunar day. Systems selected for Moon's surface exploration highly rely on sunlight to recharge solar panels, so optimizing the exposition during the whole day could improve operations duration. To conduct this analysis were used data of the terrain topography coming from the Kaguya/SELENE mission of JAXA. Another similar dataset, with a smaller latitude extension, was provided by NASA Lunar Orbiter Laser Altimeter (LOLA). The values were transformed into a Cartesian coordinate system to generate a mesh with a chosen resolution. In this VR environment, via a graphic engine, the Sun and the cast shadows were simulated. The analysis conducted during an entire lunar day resulted in a point of the region with the highest illumination percentage of 91.67 % using the Kaguya dataset, 86.11 % with LOLA's one. The two points slightly differ in position but both are located on the edge of Shackelton crater. The proximity of a crater is useful for a lunar outpost because its bottom area is permanently shadowed and an ice thin coat could be find.

With these two examples authors demonstrate the utility of VR tools in the design process and they suggest a possible location for human landing on Moon. They specify the intention to develop more accurate models in the future.

In [16] authors created a VR-based framework suitable for simulating astronauts operations on the lunar surface. They focused on three different issues: adaptation of astronauts to lunar environment, optimization of tools for EVA and human-robotic interaction.

The first application was developed according to Artemis mission concept: a simplified collection of assets reproducing the lunar base camp was located in a very faithful terrain. The simulation of ground details like craters and rocks and the implementation of special material textures, enhancing realistic reflections, helped pursuing the highest physical similarity. This whole process was conducted to get astronauts used to unique lighting conditions, very different from the ones they can find on Earth, mainly due to the absence of atmosphere and the consequent lack of light scattering. The reduced gravitational acceleration has large impact on object handling and any action implying the use of tool and equipment requires an additional attention by operators. Authors selected Unity as the perfect platform to create this (and the following) operative scenario, together with Meta Quest 3 headset.

The second implementation of VR investigated another critical aspect of lunar exploration: the ability to conduct scientific operation on the lunar surface assisted by tools. The development of EVA procedures and tools design, aiming at maximizing their usability and ergonomics, were analyzed with test campaigns using the VR environment. User can interact with several kind of tools, differing from each other in size, weight and properties and select the ones he considers more effective in carrying out the assigned task. The performances were studied to understand which are the best tools features, starting an iterative process of redesign. Furthermore the user himself delivered feedback to procedures analysts, helping them create an optimized sequence of actions to be executed during EVAs.

The third application focused on the presence of a rover in the environment and its interaction with astronauts. The robot is able to navigate autonomously on the surface thanks to sensors that map the surroundings and find the best path to reach targets. Human operator can monitor the rover activity accessing onboard cameras and, in case of out of nominal conditions, he has the authority to command the robot through a joystick. In the simulated scenario, the main data used by the rover algorithm are displayed, letting the user understand logic behind its movements and checking for upcoming failure risks. In the conclusions, authors highlighted the great realism obtained in these simulations, without issues of lags and the modularity of the created environment.

In [17] a Simulation-Based Design (SBD) approach was implemented in three different missions. This methodology employed the so called virtual prototypes: 3D assets around which the product development takes place. The three case studies authors decided to dive into are: an autonomous lander shuttling between the Gateway and Lunar surface, a cable system for surface material transportation and a navigation tool.

The lander was modeled keeping a very simple mesh and adding a crane system for payload unloading on the Moon. Approaching to the surface, four rockets ignite to slow down the descent and allow a safe landing for payload and system. To simulate a real operative scenario, the payload resembles the mass of the NASA Multi-Mission Space Exploration Vehicle.

The next challenge authors faced, was the design of a cable-based transportation system. They applied the SBD method to draft ideas and to easily assess their feasibility. The proposed concept is an infrastructure made of 4 cranes, each connected to the others, capable of carrying a payload on the lunar surface. The cranes are able to move and to rotate around their vertical axis, their telescopic boom allows to adjust payload height with respect to the ground and avoid obstacles. The VR environment stimulates many design implementations, assessing different variables in the crane features. A systems design approach was easily integrated and external reviewers could provide their contribution with feedback.

The third case study investigated another critical theme regarding the lunar exploration programs: path planning for surface transportation missions. Thanks to satellite images is possible to have an overview of the trip to be completed. The process presented in the paper starts with the generation of a grid that divides a pictures in many cells. The starting position marks the departure cell and the point to be reached the destination cell. Some element of the matrix can't be crossed as they represents obstacles that the specific vehicle is not able to overcame. Authors used the so called A* algorithm to evaluate the cheapest path, in terms of three cost variables: a first one which is the cost to move from the starting location to a given cell, the second one is the cost to go from that given cell to destination and the third variable sums the previous ones. The algorithm, through an iterative process, evaluates different free-collision paths and finally finds out the ideal trip. This last scenario is a less-visual, though very helpful application of SBD, similar to what is shown in [7] for the lunar light conditions.

In [18] authors used VR to generate a realistic lunar analog scenario where they could evaluate different interfaces to display information to operators. Apollo astronauts stated that EVAs on the Moon surface were very demanding due to excessive physical and mental workload. The main reasons are tightly related to the specific environmental conditions resulting in low color contrasts, elevated surface reflections, no light scattering as there's no atmosphere, absence of shadows in the illuminated areas and dark black regions when the Sun goes out. All these elements contribute to a difficult spatial navigation with several risks for human safety. In addition the spacesuits used in the Apollo missions got the tasks even harder with visual interferences and distortions issues. Different navigation interfaces have been studied during the years, such as head-down displays, suit-mounted displays, wearable head-up displays (HUDs), spatial HUDs and global orientation HUDs.

The research focused on the optimized layout of inputs to be created in the field of vision, combining HUD and AR technologies. Authors presented four possible configurations to a group of space experts, their proposals were: suit-mounted tablet, wearable HUD, spatial HUD and global orientation HUD. The first solution received negative reviews as it constantly requires one hand to visualize information. Experts pointed out the importance of having critical parameters inside the user's field of view; very essential values to be included are oxygen level, heart rate, inner temperature, EVA duration and battery status. They also suggested to add maps and navigation aids and to organize the inputs displaying them based on a hierarchical division. The control of this whole system could be managed via vocal commands.

1.4 Thesis outline

In the previous pages there are listed a bunch of reference papers where many scholars have tried to demonstrate the scientific validity and utility of the application of VR to the space sector. Some experiences were created to present test beds where designers could evaluate different solutions throughout the development process and astronauts could train themselves in the operative scenarios; others articles produced solid output through analyses of the lunar environment. The effort of many researchers in trying to validate this technology and to permanently integrate it in the engineering context results in a solid starting point for my Master Thesis project.

The input for many investigations in the related work section came from issues reported by astronauts who have conducted space explorations in the last decades or from open questions regarding the Artemis program. In this moment a couple of important companies together with the relative national space agencies are working on a completely new kind of vehicle for the lunar surface: a pressurized rover. In the early '90 some concepts were presented by NASA but they weren't developed further [19]. In 2022 NASA and JAXA collaborated to test pressurized rover operations in the desert of Arizona [20]. At the moment it seems that the renewed interest in this type of transportation system could lead to raising enough funds to bring the projects to greater maturity.

The effort of important space agencies in creating analog missions around pressurized rover mockups, where users simulate procedures and EVAs, remarks the primary need of placing human beings at the center of the design process from the early phases. It's the perfect framework for applying the VR technology with the purpose of creating a scenario where astronauts could experience an high-fidelity simulation of a pressurized lunar rover. This project aims to develop such an environment. During the documentation phase of this Thesis, I wasn't able to find previous examples of the creation of VR scene applied to lunar transportation vehicles; deepening this thread, I've got in touch with a company which is developing a simulation of a rover they are designing, focusing on the creation of a Digital Twin to study in detail the subsystems of the product. This kind of approach is very profitable for industries but can't be the core of my job. The rover design phase is one of the first stage of my workflow, but the output produced during that initial analysis are taken to be the starting point of the main part of the Thesis: the creation of a likely environment where the simulation platform could be validated.

An aspect that I consider of particular value to the work that follows, is the combination between my major in Aerospace Engineering and the Computer Graphics Technology area. The possibility to work together with experts coming from both backgrounds, allows me to create an interdisciplinary set of skills and to have a wider view on the future trends industry will exploit. Obviously, to be prepared to face this challenge, I had to discover a whole new set of software regarding game development, coding, VR implementation and optimization. The final purpose of this project is to create a training environment running on a head-mounted display (HMD), an headset that combined with controllers allows to enjoy the experience.

This first introductory chapter summarizes the researches I've conducted to get acquainted with the subject, listing the most relevant papers I've found. The initial part of my documentation process concerned learning more about the XR technology, the people who set the first bricks for the construction of the whole world and how this sector has developed along the years, until now. After this primary dive into the environment, I started collecting materials regarding how aerospace industry has managed to blend with computer graphics technology to exploit the opportunities deriving from it. All the papers I have read allowed me to understand what is the state of the art, necessary to mark where my work would fit most effectively and its outlooks. Then I have presented the initial idea leading my project, the technical areas involved and the challenges that this activity poses to me. Finally I have declared what is the expected output that the Thesis should produce at the end of the process.

In the second chapter is presented the methodology. Following some general indications for the preproduction phase as presented by Unity, I am able to portray and investigate the needs of the users who are going to use my application. After some steps of preliminary design, prototyping and detailed design, the core production stage takes up, initializing an iterative process. That chapter is foundational for the development of the project as many choices deriving from analyses conducted in the preliminary design have a critical role during the following chapters. The documents produced in that phase represent a useful baseline to follow throughout the whole work, helping developers stay on track.

The 3D modeling find place in the third chapter. Obviously before starting modeling something, it's better to spend some time understanding which objects are going to be placed inside the rover; to do so it has been used a functional diagram. The elements at the bottom level of the tree have received a level of detail and a level of interaction, useful indicators for considerations about the modeling accuracy and the implementation. The first prototype built for volume allocation reasons has been uploaded on the headset to have the very first experience inside a VR scenario. In that chapter are also presented the human factors studies, with a focus on ergonomics. Finally the look and feel needs to be assessed, since it plays a relevant role for the immersion sensation of the user.

The fourth chapter is dedicated to Unity, the powerful graphic engine chosen to run the experience. There can be found all the details related to the software: locomotion systems, C# scripts, intractable objects, events, user interface (UI), accessibility, performance optimization and so on. The large part of the chapter is dedicated to the preproduction work, carried out to define precisely what will be the activities that will have to be developed in the second part, the actual implementation.

The fifth chapter presents the results of the work, divided into sections according to the different game modes and tasks, with pictures helping readers understand what is the visual output, hard to appreciate with textual descriptions alone.

The last chapter is the sixth, a place for budgets and final considerations on the work done and what could have done better. It also hosts hypotheses on what interesting future developments might be, laying the foundations for a continuation of the project by the author or by anyone who wants to pick up the torch.

Appendix A sums up the initial design process regarding the rover, some references and equations resulting in sizes to be used for the modeling part. The objective of this Thesis is not to model from draft to finish a vehicle, so those preliminary studies only aim at creating a real-scale habitat to set the simulation.

Chapter 2

Methodology

2.1 Development phases

The development of such a project requires a mix of engineering and game design workflow elements. However, whether you are working on a hardware or software product, the initial steps of the process are common and can be listed as follows, as suggested in the "Introduction to real-time 3D experience design" lesson of the Junior Programmer Pathway on Unity Learn [21].

2.1.1 Vision

This point has already been partially explored at the end of the previous chapter, in the Thesis Outline section. To address the problem outlined in the previous chapter, it is planned to create a training environment to accustom the future inhabitants of the pressurized lunar rover to the tasks they will have to perform, how to manage the subsystems around them and above all, the fundamental and unique element justifying the use of such technology, getting familiar with the effect of reduced gravity on surrounding objects and their bodies. Giving back to the user the sensation of lunar gravity is very difficult, but it's way easier to show the different effects of actions in these new conditions: an object gripped and dropped will take longer to reach the ground and will eventually bounce in a particular way; even the player's own body, if he decides to perform a jump, will more easily reach higher heights and return to the ground slowly. Similarly, with VR technology is possible to model the resistance conditions due to the presence of a particular atmosphere and evaluate its effects.

2.1.2 User identification

As a result of the previous point, the target audience for this application is first and foremost the crew of the rover. Secondly, such a platform is useful as Digital Twin to assist the classical system engineering concurrent design approaches during the development phase and thus usable by engineers. Finally, having such a digital product facilitates the creation of media content aimed at enriching the outreach aspect and even provides a simulator that can be used with general audiences.

As regards the first type of users, namely astronauts, many projects have been carried out in both the academic and industrial sectors which place the user and the actions they perform at the center of the experience, culminating in a test campaign carried out by people who have already visited space with stays on the International Space Station (ISS). From what I read in the papers and through an interesting visit to a major European space company, astronauts first make a fundamental contribution to the development of these scenarios by taking part in the evaluation phase by providing feedback on how much they find what they are testing to be similar to the in-orbit flight experience. In this way they are helping with the project growth even before such a scenario becomes useful to them. There is no question that once they have completed the virtual reality platform, after the various iterative cycles typical of the project, such technology will become relevant for their training and preparation.

There is also extensive literature on the use of simulations that integrate many aspects of the product, both structural and systemistic, into the design process. This theme is explored at an industrial level across the board, transcending the boundaries of space and aeronautics because it entails savings in time and money for every type of company. In this case, the virtual reality platform should be focused on those who must use it continuously, performing analyses and updating it whenever necessary to keep it up to date with the latest developments introduced. In addition to the literature, however, as far as industry is concerned, examples of Digital Twins are jealously guarded as part of strategic know-how.

Finally, in the field of products made available to non-experts, as well as those simulations that industries decide to release to entertain their visitors or publish on their social channels to convey some marketing message, you can count all those applications born with the sole purpose of being pure and simple games. The latter, developed by companies specialized in game development, have as their primary purpose entertainment and allow players to immerse themselves in extremely futuristic contexts, bordering on science fiction, trying to maintain a physical consistency with the real models they are inspired by. Clearly in these applications the gaming experience is maximized, creating stories and missions to follow and enriching the scenography with many aesthetic details, all aimed at attracting the user's attention and keeping him as much attached to the product.

2.1.3 User personas

From the three application threads discussed in the previous point, it is possible to derive three types of user personas that could benefit from the application. This categorization comes from marketing, in which it is necessary to divide the customer base into different macro categories in order to be able to associate certain buying habits with each segment and to make sure to identify the needs of these archetypal customers to target the right product. This practice also applies when there is no direct need for sale, when you do not have to attract a customer to buy but simply must tailor the experience on the user.

This requirement means that the three simulation platforms will differ in some aspects for the three user personas. The three of them and their identified needs are shown in Fig. 2.1.

Figure 2.1: The three user personas and their needs

2.1.4 Audience analysis results

Once the needs of the user personas are delineated more precisely, it is possible to focus on finding which simulation should be most relevant within the rover for each segment of audience and which combination of settings could better help the user in his experience. Trying to satisfy all the requirements raised in the previous section could lead to a very complex environment or eventually to three separate scenes targeted on each persona. To better address the problem, I am going to prioritize the astronaut's needs for the final product.

With this in mind, thanks to an opportunity that has arisen, towards the middle of the development of the application it will be possible for me to present what I have achieved up to then to a very diverse audience. This group of people will hardly include astronauts or space experts, so the early testers who are going to experience my simulation will likely have the visitor's needs listed in the third column of Fig. 2.1. To meet their demands, I'm going to build a first version, with limited functionalities but increased accessibility, that will be run and tested in that occasion.

2.1.5 Preliminary Design

At this stage, I proceed to analyze more specifically the concept of pressurized lunar rover, what it should contain in relation to the type of mission it must carry out and what size it could have. To do all this, I take inspiration from a historical rover design manual [22] and a series of prototypes and concepts under development. Then, during a preliminary design phase, the rover system is broken down into its sub-layers to identify the functional blocks to be included. Through reviews and feedback, I arrive at the bottom of this hierarchical diagram, at the component level. This way of proceeding allows to have always under the eye a baseline on the whole project and its everything that will be modeled and made interoperable.

During this phase, the menus that the user will be able to use in the monitors inside the environment are thought up and sketched. These User Interface (UI) elements are not about navigating between the app home pages and scenes but are in-game components.

A first 3D modeling activity on Blender allows to have a preliminary idea about the occupation of the internal volume, considering the available space that remains free after the insertion of all those hidden but fundamental support systems. At this level the functional blocks are modeled as colored parallelepipeds and it is possible to have a first view through the headset of the room under construction.

2.1.6 Prototyping

Further steps have always foreseen the work scheme that starts from the idea on paper, passes to a draft created in Photoshop for a 2D view or orthogonal projections of the part and continues with a 3D model equipped only with mesh. The individual component is presented to the work team's review for feedback. Prototyping initially takes place only up to the part of 3D modeling, without Unity and therefore virtual reality. The presentation of these ideas include renders and demonstration videos.

2.1.7 Iterative design

Since the creation of the models is fundamental to have the assets imported on Unity to enrich the scene and then create all the interactive elements, the first iterative part involves only the steps of the hierarchy diagram and Blender, excluding the game development software. Tests at this stage are also carried out using a mannequin inserted into the model to assess the concepts of human factors and ergonomics. Once all the components are created it is possible to add them to the project in the Unity Editor to start working on their physical properties and on the scripts that regulate all the game mechanics. Then UI needs to be added to facilitate the navigation through the app and to give instructions to the players regarding what they are able to do within the vehicle. A test campaign ends up the process to verify the presence of bugs and to solve misinterpretation issues regarding in-game texts.

This way of proceeding is theoretically wise but, deepening the knowledge of the working environment, some down-sides pops out. One of the reasons that has led me to follow this workflow, with 3D modeling executed before VR development, is my previous experience with assets generation and renders or animations production. The advantage of already owning this set of skills, brought me to create the components for the scene while watching tutorials regarding the Unity part, which I knew little about. The efficiency of proceeding on the two parallel tracks gave me the opportunity to showcase images and videos of the inner habitat during the first meetings of the team, demonstrating the creation of solid output for the project, while a long and time consuming back-end learning activity was taking place. The drawback of this approach is the fact that the performance of the VR simulation highly depend on the imported assets. The smoothness of the experience relies on the characteristics of the machine on which it runs. Especially if the app gets built on the headset, without any tethering to a computer, to reach an enjoyable number of frames per second, there are very precise limits to the geometry that need to be accounted. All these issues will be discussed in the following chapters but, to make a long story short, the workflow could have been organized differently. To have a first idea of the space distribution inside the rover, the creation of a prototype with basic 3D models is very helpful, but I think that before further enriching the details of the models, it would be better to switch to the Unity Editor to start working on the game mechanics with primitive objects. Only at this point, with a functioning app, the process of details enrichment should take place on the modeling software. However, this is my first experience discovering the giant world of XR and I certainly can't pretend to get all the right steps, specifically the most time-saving path, at the very first try. The back and forth between the game development software and the modeling software could be listed as part of the iterative process of design that, through thick and thin, characterizes my Thesis project.

Chapter 3

3D Modeling

3.1 Introduction to the modeling phase

The final purpose of this Thesis is to create a virtual scenario where users can experience the inner habitat of a pressurized lunar rover. All the elements in the scene need to be created in a proper software that allows to generate a basic mesh, modify it through a set of tools ad associate a material to it. A mesh is a net of polygons placed in the tridimensional space, it can be made of triangles or quadrilaterals that share edges or vertices.

The number of 3D software that offer the possibility to create solid objects is huge but, in my opinion, they can be divided in two big groups according to the purpose of those who use them. The first ones are all the tools for the so-called Computer-Aided Design (CAD); they help the industrial sector or architecture and they emphasize dimensions and tolerances. The goal is to have a drawing, a blueprint or a plan that another worker, in charge of production, can follow. The other group gathers the software for the Computer-Generated Imagery (CGI) and they are used for renders and animations. Since the components I plan to build will exist only in a virtual reality scenario, a software of the second category seems to fit the bill. Furthermore, I'm not completely new to the world of the Computer-Generated Imagery and I have explored some software for a couple of years now. Specifically, I learned about Blender [23], a fast- growing open source software with tons of functionality and a large community made of creators, artists and curious people eager to share their knowledge and their creations. Blender also offers several format to export the objects created in it; such a possibility is very important because all the assets created here have to be transferred to the Unity Editor as the next step in the workflow. To verify the compatibility, the list of exportation formats from Blender needs to be compared with the one of the importation format to Unity and check for common entries.

The positive side that made me choose this software over others I have used in the past like Solidworks and 3ds Max or over ones which I've never used but I known are very popular like SketchUp or Catia, is the very accessible UI, the open source philosophy behind it and the passion of the content creators that constantly publish tutorials on their channels. One flaw is that, in my opinion, Blender isn't so integrated with other software and when you export files to another program, then you have to fine-tune some aspects, usually the materials and textures. The CAD software listed above are more optimized for the migration between platforms as they are parts of the common engineering design phases.

3.2 Functional approach

To understand what kind of habitat should be included inside the vehicle, it is better to execute a preliminary mission analysis. The system should be able to move on the lunar surface like all the previous rovers that have been used during the Apollo missions to facilitate the exploration of the Moon around the landing site. The new feature of such a system is the possibility to generate a pressurized atmosphere inside it, while all the previous rovers where mainly made of a chassis and seats above it to house the astronauts, exposing them to the lunar environment. This peculiarity will probably set some limits to the operations of the vehicle, probably regarding the maximum reached speed, the slope of the obstacle to overcome and other mobility issues. Taking this into account, the first, important functional block to include inside the habitat is a cockpit where the crew is able to pilot the rover and to manage different parameters regarding the system status.

The livable habitat of the vehicle allows a longer stay for its inhabitants, not only limited to the short time to move between two points, but to use the rover as a base for proper exploration missions on the lunar surface. This new opportunity, however, requires the onboard presence of all those elements that must necessarily accompany human life in space. All those materials will be included in a proper functional block and will be later analyzed in detail.

Another relevant function that I want to include in the rover is a station to perform scientific experiments, mainly on lunar sample collected during EVAs or by a mechanism operated from inside. Since the first pioneers of space, every mission has represented an opportunity to carry experiments into orbit. The microgravity conditions offer incredible chances to study the behavior of natural processes in a peculiar environment and to evaluate the differences. The lunar gravity represents a new frontier as well and an in-situ research plant could help either with the development of Earth-centered test activities or with the knowledge of the Moon's structure. The habitat can be seen as a moving laboratory equipped with all the subsystems to support the life of a crew in an extreme environment for a certain amount of time.

The need of accessing the rover can be solved in different ways: for example the system could perform a docking to a permanent base allowing astronauts to move freely in a sort of extension of their primary habitat. Once the vehicle is ready to go, it performs an undocking. This solution is very efficient if the rover is used mainly as a transport system, to link two bases where it can be anchored. The astronauts might want to have the possibility to perform an EVA during their route, either for a planned reasons or for contingency, in this case a sort of airlock should be included to the structure. The airlock could be an entire room that gets pressurized and depressurized or a smaller device, like a manlock that allows to enter from the inside a suit hanged on the outer part of the rover and than to separate from it. The vehicle I plan to study shouldn't necessary include a docking system and so I'm going to add a functional block in the rear part to house the airlock and the space suits for the crew.

The last function I want to include groups all the emergency devices that should activate in the eventuality some subsystems fail or in case of outer threats, like micro meteoroids impact and so on. All the space operations procedures handbooks analyze the risks related to missions and implement several countermeasures to prevent potential fatal events to occur.

To get a baseline for the following steps, I've created a functional tree, also known as hierarchy diagram, where it is easy to visualize the decomposition of the vehicle from an high functional level to the components level at the bottom of the tree. In this first iteration phase of the design process, I've only descent on one level into the details, this will be enough to create a first prototype of the inner habitat with primitive shapes representing the functional blocks of Fig. 3.1.

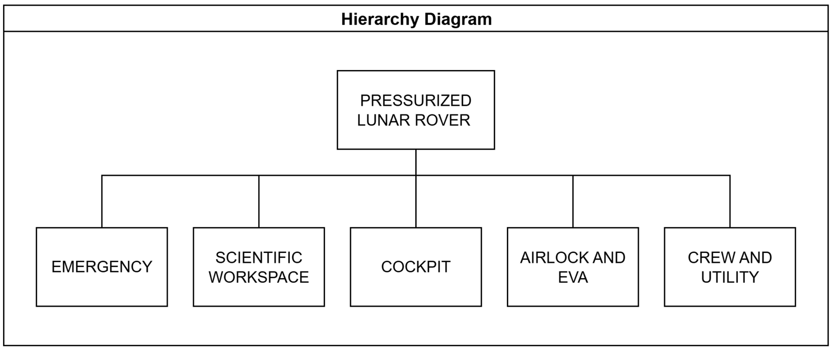

Figure 3.1: The five blocks that represent the first functional breakdown of the system

This kind of subdivision is very helpful for the future organization of the Unity Editor project: a tab of the layout is called Hierarchy and it lists all the assets included in the scene; to keep the window neat it is highly suggested to sort the different elements according to the functional tree scheme.

The choice of representing the development of the decomposition of the pressurized lunar rover with this method also allows to present the output of the analysis to a wider audience in an easy and time-saving manner. The diagram provides a self explanatory description of the direction the project is taking and lets members of the working team submit effective advises and feedback to me. This can still be considered as a pre-production phase, where all the efforts are aimed at focusing the goal in the best way. From the next section, some 3D modeling will take place.

3.3 First prototype

The analysis of the rover has been carried out by a functional point of view so far, no physical studies about the structure, sizes and volumes have yet been performed. To create a realistic inner habitat, though, I searched for the history of lunar vehicles, concepts and projects under development at the moment. Since the design of the system is not the goal of my Thesis, I include this part in appendix A, as support material to the main work.

The principal output from appendix A are the dimensions of the cylindrical shell that allows the pressurization of the habitat, creating the room where all the functional blocks have to be located. To understand how to distribute the volume, I start placing some low poly solids representing the functional blocks of Fig. 3.1. With this process I'm achieving a subsequent subdivision of the hierarchy diagram.

3.3.1 Cockpit

The main purpose of this zone is to accommodate the crew members during the rover movement and allow them to control the vehicle. A couple of seats are placed facing a wide windshield that allows to shift the gaze from the sky to the surface beneath the pilot. Multi-Function Displays (MFDs) in front of the seats help the exploration of the various menus to manage the system functions. Another group of screens, placed over the heads of the pilot and co-pilot, enrich the view of the ground around the rover thanks to a set of external cameras. The duration of the mission requires the presence of a resting facility: beds of some kind that allow astronauts to restore their energy. To save space, these elements are integrated with the seats and they deploy by rotating the backrest.

To easily identify assets being part of the same functional block, in the 3D scenario they all share the same color, blue for the cockpit.

3.3.2 Crew and utility

The services that the vehicle should offer to its guests include storage, hygiene and training equipment. The storage contains food, beverages, spare parts and maintenance tools; to a rough approximation these containers are pictured as boxes. The same strategy holds true for the hygiene sector, with a sink beside a larger block that gathers other necessary toilet elements. To address the need of physical exercise, a pop-up equipment is placed in the middle of the room but it will require further studies to understand what kind of machine it should represent and to turn it into a interactive asset for the VR experience.

This group of objects is portrayed with pink faces.

3.3.3 Scientific workspace

The scientific workspace is made of a desk with computer screens hung on the wall above it. The operator needs a working surface to interact with lunar samples, tools and other equipments to perform the experiments. Furthermore, all these working materials must be stored away once the analysis are over to ensure an hazard-less driving phase for the rover inhabitants.

The bed deployment mechanism could, in the maximum extension configuration, be hampered by the presence of the workstation; to avoid this, the depth of the desk needs to be carefully assessed: the creation of this first prototype represents a great opportunity to visualize the issue. I do not exclude that, as the project progresses, the positioning of the beds or their extension scheme may vary.

The color chosen to identify the elements of this functional block is red.

3.3.4 Airlock and EVA

The airlock is housed in the rear part and is initially represented as a free space between two hatches; an additional volume, made of two big parallelepipeds is considered for the space suit storage. The size of the airlock should allow the simultaneous presence of the two crew members inside it, with the additional bulk of the worn suits.

In this area may be useful the implementation of a dust removal tool, that let the astronauts remove all the particles that have settled on their suits during EVA. Throughout the pressurization time, they could execute such a task to avoid the accumulation of impurities in the filters that keep the inner atmosphere clean. This opportunity needs to be further studied but it is interesting to mention it in this brainstorming/preliminary design phase.

The tools to be used during the out-of-the-rover operations are imagined to be stored outside, to save space inside and to keep the entrance/exit routine as simple as possible.

All the blocks being part of the Airlock and EVA function have a greenish material applied.

3.3.5 Emergency

At this point of the project, the emergency functional block is still drafted as procedures rather than real objects. In the current version of the prototype aren't included 3D assets that fulfill these functions.

3.3.6 SimLab Soft and headset experience

The result of the previous subsections is the creation of a prototype in Blender. My desire now is to explore the model with the VR headset; to do this, various possibilities arise. The Meta Quest 2 device can be tethered to the computer via the proper link and allows the immersive navigation of several apps, Blender included. A Meta software has to be downloaded to start the communication between the headset and the computer. The main issue regarding the app, called Meta Quest Link, is the list of requirements that the user's machine has to fulfill. A fast overview of the list of hardware/software specs easily demonstrates the lack of graphical capability of my laptop and unquestionably marks the impossibility to use the software.

The solution to overcame this obstacle is SimLab VR Studio 14 [24], a very helpful platform that allows to import all the 3D assets previously created in Blender and easily set up a VR scenario. I consider this tool a simpler version of the Unity Editor, but my knowledge about the app is pretty limited so I could be wrong. A premium version of the software is available to dive deeper in its multiple functionalities but I've left it unexplored so far. The most important feature of SimLab is the uploading of a scene, created in a certain device, to a cloud space and the opportunity to access it from another support, like an headset.

Moving assets from Blender to the new software requires a compatible format to perform the export/import process. A possible answer is the .obj, also known as Wavefront .obj [25], which works well to transfer geometry since it includes information about vertices and faces of the meshes; to transfer also the materials properties, it needs to be twinned to a .mtl file.

Once a way to move 3D models has been found, it is a good idea to take another precaution into account before proceeding with the actual migration process. While creating objects in the 3D editor, it's pretty common to employ some very useful tools known as Objects Modifiers. These elements produce changes to the viewport output without affecting the real pattern of the meshes. If the purpose of the project is to produce renders or animations, everything is fine and the process runs smoothly, but, if the goal of the modeling is to export the solids to another working environment, is highly suggested to apply all the used modifiers to match the graphic output of the viewport with the real structure of the grid underneath. Not following this procedure would probably lead to problems when accessing the file on a different software.

Another setting to consider before finally exporting the file is the so-called packing of the external resources, a simple procedure to add all the textures and images files to the .blend project; this should avoid the unpleasant appearance of the fluorescent pink color instead of the original textures.

Inside SimLab, the main thing to do is the positioning of the starting point of the camera, corresponding to the initial location of the VR user in the scene. By opening the app on the Meta Quest 2, the experience is ready to be enjoyed and all the instructions about the way of moving inside the scene are displayed on the screen. At this point no type of interaction can be accomplished with the surrounding objects.

3.4 3D Development of the cockpit section

The general distribution of the space presented in the previous section and submitted to the review team was essentially approved and the process can go on. To model the prototype a first functional breakdown of the cockpit took place, with the differentiation of the structural part, the windshield from the inner elements like seats, monitors and control panel. The further subdivisions are explained in the next subsection.

3.4.1 Hierarchy diagram

Blocks like windshield and monitors don't require another decomposition since they are already elementary object that can be clearly created in 3D. The pilot seats need to include other two functions: the resting configuration and the command joystick to move the vehicle. The control panel houses three types of items: MFDs, analog indicator to backup the information of the MFDs and mechanical components such as buttons and a radio. In Figure 3.2 is also shown the breakdown of the analog indicators with a list of important components for the monitoring of the state of the rover.

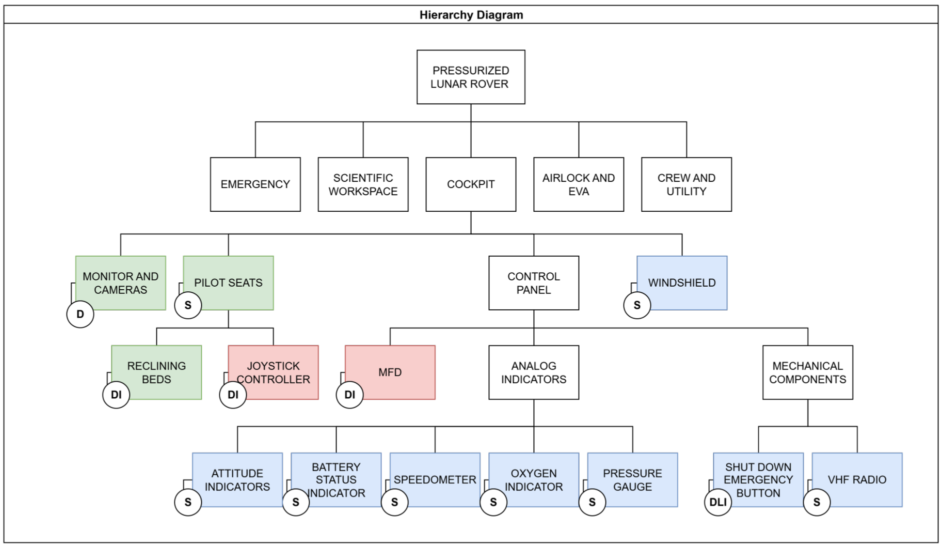

Figure 3.2: The breakdown of the cockpit functional block into its base components

Looking at the tree in Fig. 3.2, there are boxes that pique the eye with their different colors. White rectangles are components that still need to be subdivided into their basic elements since they represent functions or groups of physical objects. The varying highlighting marks a certain level of detail established to create 3D models. The need to differentiate the low branches of the diagram arises from time requirement and to take account of the limits of the processor unit that will run the simulation. Reaching an high level of detail for all the assets brings to very appealing renders but in the VR world might be dangerous; the number of poly, or of triangles, increases very quickly, leading to lags. A wise suggestion is to trade off between the mesh enrichment of an item and its importance in the whole habitat. An object that is grabbed by the user and observed closely or that is involved in particular interactions and movements should be created with greater detail than a less relevant one. The color scheme is: light blue corresponds to a low level of detail, green to a medium level and red to an high one.

Another categorization adopted in the diagram is about the degree of interaction between the player and an item. The letter in the little circle located in the bottom left corner of the box refers to the macro categories used to organize the workspace inside the Unity Editor, specifically in the Hierarchy Window. The first group includes the static (S) objects, all the assets that don't move during the simulation and that are not allowed to be grasped. The other group gathers all the dynamic (D) elements, which are able to move across the scene. To create a more precise partition, this last pool contains two subsets, the dynamic with interaction (DI) and the dynamic with low interaction (DLI) depending on the complexity of the actions performed. This partition will prove its importance for the lights set-up during the VR development for maximizing the optimization of the app; moreover this great containers will help keeping the Editor decluttered.

3.4.2 Referencing

Once all the components have been identified, before starting the actual modeling, it is a good idea to find references to take inspiration from and use as blueprints for the sizes. In the cockpit section are useful two slightly different referencing activities: the first type aims to find a version of a certain product that fits well into the selected context, from it, with the necessary modifications, the model derives. The second type of activity doesn't focus on the single item to be created, but on the layout of multiple components, which have to be carefully organized to fulfill a certain function; an example are the indicators that compose the control panel.

The first input for the cockpit configuration comes from the aeronautical sector and the reason for this almost spontaneous choice is to be identified in my education as an aerospace engineer. My familiarity with aircraft cockpits brings me to imagine a very high level of redundancy and the possibility to control the rover from both the front seats. This design philosophy leads to a very crowded space with several redundant system and low tasks differentiation between the two crew members during the movement of the vehicle. The review team suggested to shift from an aeronautical centered vision to an automotive point of view. Indeed, the system I'm studying is way more similar to a car than to an airplane and requires a shift in the way of thinking about it. The main change is the clear distinction between the role of the pilot and the co-pilot: the first focuses on driving the vehicle while the other monitors the indicators of the subsystems and eventually manages the navigation information. This change of paradigm allows a reduction of the elements in the cockpit, centering the driving functions around the pilot's seat.

3.4.3 Modeling

The seat modeling is probably the most complex of this entire section; the object gathers three different functions: housing the crew members during the movement of the rover, providing a resting place to sleep and operating the vehicle with a joystick. A first trial exploits an existing 3D asset from an online catalog. The mesh has undergone several changes to be adapted to the particular needs of the project. The main additions are the joysticks on the armrest and the deployable bed in the back. The risk of collision between the fully deployed structure and the sidewalls compartments suggests a revision of the seats concept; the possible solution is to take inspiration from the business class reclining seats of the middle/long range aircraft that allow the passenger to rest in an almost laying position. This precious advice comes from the team of experts of Thales Alenia Space that hosted some meeting on the Lunar Pressurized Rover topic.

The main element of the control panel is the MFD, its modeling is pretty simple because it is a scaled cube with an inset of a face. The core activity around this element concerns the windows that are shown on its screen, but are sort of UI elements and will be discussed in the following chapter. All the other indicators that enrich the panel have been modeled taking inspiration from hydraulic gauge, speedometers, attitude indicators and other elements from the automotive and aeronautical sector. They are mainly made of a very simple mesh that encloses a picture portraying a scale or a clock-like background with the values of the unit to be monitored.

The windshield at the moment is very large, with several transparent panels to enjoy a wide view of the surrounding area. However, this surface of windshield is unrealistic, the risk of micro-meteoroids impacts with the outer structure makes this design choice unsafe. A limited extension of panels, combined with a set of cameras to be aware of the hidden terrain below driver's field of view, shall be more effective.

A sneak peak of the cockpit is shown is Fig. 3.3.

3.4.4 Ergonomics

Most of the dimensions of the elements are essentially deduced from the reference images; this is especially true for radio, screens and the indicators. Nevertheless, some require a greater attention when it comes to their sizing and positioning, especially in relation to the distance from other components. These topics are investigated by a discipline known as ergonomics. The huge research field applied to this particular project can be divided in two parts: the ability of certain objects, like seats and beds, to accommodate astronauts and the possibility of the latter, placed in an established location, to carry out all the required actions. The first issue can be summarized as the needs to create comfortable objects, following rules of sizing, angles and ratio. The second problem focuses the layout of a group of items around the pilot, everything should be reachable by the user's hand or be inside his field of view, for example a display. The prioritization based on the use frequency of buttons, indicators, gauges and monitors affects their position inside the cockpit.

The first strategy adopted to assess all these ergonomics aspects is the inclusion of a medium-height mannequin model to the 3D project. Blender offers a whole set of tools for rigging and animating character that turn very helpful to establish whether or not the place of a certain button is reachable from the pilot seat. With a skeleton parented to the mesh, the arm and leg extensions are easy to reproduce and all the ergonomics studies can be carried out.

Thanks to an additional discussion about the human factors world, it turns out that a more precise way of conducting such a research is to consider the height percentiles. This measure represents the portion of population that reaches a certain tallness: if someone is under the first percentile, it means that he's shorter than a certain height and that the share of humans that are under that value are less than 1% of the world population. The same rules for being over the ninety-ninth percentile, with a group of people taller than a certain measure that are less than 1% of the whole world. This way of thinking lets designers take into account a greater variety of heights than minding only about the average body size. For example, assuming a 3% - 97% percentiles range, the lowest size (corresponding to women third percentile) is 151 cm and the highest value accepted (which is men ninety-seventh percentile) is 186 cm; this means that the 94% of people are physically able to command the rover. All this reasoning about heights doesn't take into account other kind of obstacles that could limit someone's ability to use the system, but it belongs to accessibility considerations that will be partially addressed in the next chapter.

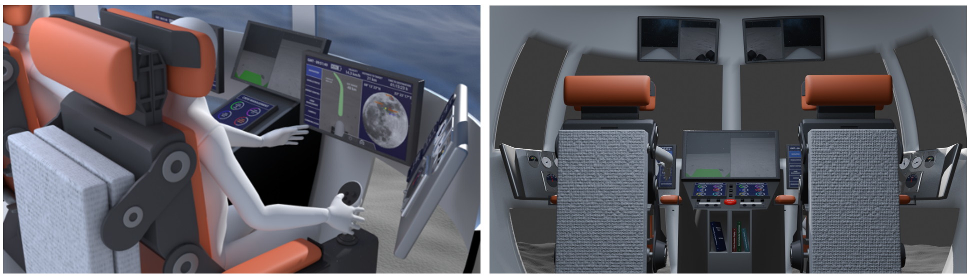

Figure 3.3: Views of the cockpit section

3.5 3D Development of the crew and utility section

This section brings together all those objects that serve the function of supporting the lives of astronauts inside the rover along the mission. The main element that characterizes the habitable modules is the Life Support System (LSS), with the management of the pressurized atmosphere, the monitoring of temperature, humidity, contaminants concentrations and so on. The further fundamental function of the LSS is to supply all consumables to crew members to maintain correct metabolism and to keep them hydrated. In this logical flow follows the management of products of the cycle, divided into solids and liquids. This whole system is mainly located behind the scenes and it's not the primary focus of this work; in the simulation, it is interesting to include the parts that the user interacts with and uses to feed himself or train.

3.5.1 Hierarchy diagram

The first division of the high-level functional block is into exercise area, storage, personal hygiene and galley. A training equipment is mandatory in all the manned missions in micro-gravity cause the zero-g condition drastically affects the whole body. Many body systems are subject to negative effects during the phases of space flight, some are able to adapt in a short period while others fail to reach a balance and their decline continues progressively. The musculoskeletal system is one of them: the low gravity environment corresponds to a more relaxed condition than common terrestrial activities and this leads to a loss of muscle tone. Another consequence of the lack of gravitational field is the acceleration of the process of decreasing bone density, known as osteoporosis. A good point to consider is the difference between the microgravity environment and the lunar one: the latter is somehow less impacting since the level of stress applied on the muscles is higher than in zero-g, but it still represents a reduced load with respect to Earth, plus astronaut surely comes from a period of interplanetary transfer in which his body has already weakened. To contain these problems the exercise is compulsory and the rover needs to offer an exercise area to its inhabitants. The dimensions of such equipment are critical, probably a cycle that disappears in the floor can be a solution.

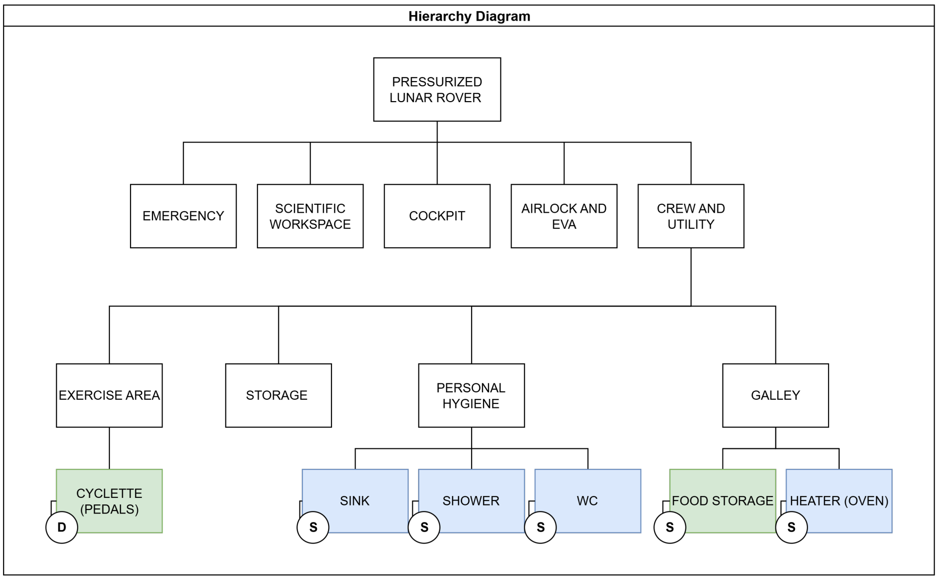

Figure 3.4: The breakdown of the crew and utility functional block into its base components

A large amount of space must be dedicated to store all the spare parts, repairing tools, emergency kits, astronauts belongings and other useful stuff. Another kind of depot, the galley, has to house all the goods, organic consumables and a very simple set of kitchen equipments. Lastly there's the block of the personal hygiene, branched into a sink, a shower (optional) and the toilet. Everything is neatly pictured in Fig. 3.4. The level of detail of the components of this section is low or medium. The mesh enrichment is also related to the level of interaction: the training equipment, which should be utilized by the user and is the most complex item of the crew and utility section, requires a medium level of detail. The other assets are not particularly relevant to the simulation.

3.5.2 Referencing

To design an effective training tool, the inspiration surely comes from the numerous devices sent to space on the ISS during the last decades. The limited space leads to a simplification of the usual structure of the cycles, with the removal of the saddle, resulting in a pair of pedals.

For the hygiene part, the model, considering the limited volume, is the one of the bathroom that can be found on the airplanes or on the sailboats.

The storage on the ISS is mainly made of soft fabric bags, fastened to the structure with special straps. This weight efficient solution could have disadvantages if used inside a rover that has to move on rough terrain.

3.5.3 Modeling

The first idea is to group all the hygiene components into a space that provides some privacy; a folding door creates the boundary between the two areas. The storage, including food, is divided into labeled boxes with handles to grab them. A simple oven and a water dispenser complete the collection of 3D models shown in Fig. 3.5. The pedals are located inside the floor and they can be pulled out when necessary.

Figure 3.5: View of the crew and utility section

3.5.4 Ergonomics

They main ergonomics study in this section is about the restroom: the mannequin introduced in the previous section has been used to evaluate the distance between the toilet bowl and the sink. The training equipment requires further investigation that will be carried out during the test part with the complete VR setup.

3.6 3D Development of the scientific workspace section

The core function that the rover includes is the laboratory. All the other blocks are somehow designed to support this major area that will serve as a base point for important studies and EVAs. The results brought by the scientific missions could lead to relevant discoveries regarding the highly debated birth of the Moon, lunar soil composition, presence of resources and other important issues that concern human stay on our natural satellite and far beyond.

3.6.1 Hierarchy diagram

The first subdivision of the high level function aims to draft the main elements of the working setup: a workbench where operators can handle all the tools and work on the samples, some monitors to access information and a storage to stow the scientific specimens. The storage could also be used for the EVA tools but, as mentioned above, it is probably a wiser choice to place them outside. The items that can be used on the workbench are listed in its next breakdown in Fig. 3.6.

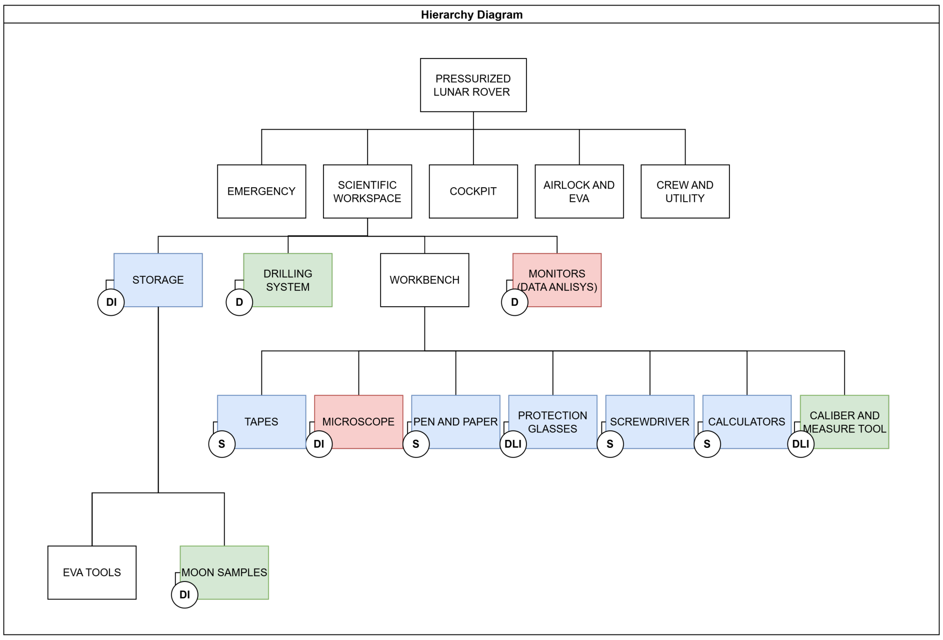

Figure 3.6: The breakdown of the scientific workspace functional block into its base components

Another block has been added to the tree, a drilling system and it represents a system to gather samples without leaving the habitat. Many of the objects that can be found on the desk are very common and simple and don't necessary need to be included in the simulation as dynamic elements since they can be effortlessly experienced in everyday life.

3.6.2 Referencing

The main inspiration for the area comes from the laboratory placed on board the ISS. Looking at pictures taken during the missions, can be distinguished two types of scientific workspaces: a glovebox to perform atmosphere-controlled experiments or to contain dangerous vapors and a sort of crafting table to perform repairs and maintenance tasks. The second workstation is equipped with a large variety of objects that is good to have on hand and which, unlike the in-orbit version, must be properly stored in drawers once the task has been accomplished.

3.6.3 Modeling

The workstation is modeled has a desk with some drawers, a stool for the operator and a monitor on the wall in front of him. In Figure 3.7 can be seen on the left side the glovebox made with transparent walls and a pair of gloves to handle the object inside it. All the items that enrich the section are pretty easy to model as they are little objects of everyday life with shapes that resemble the primitive solids and therefore require little mesh manipulation.

Figure 3.7: View of the scientific workspace section

3.6.4 Ergonomics

The only sizes to consider are about the stool design and the height of the gloves from the floor to let users easily access them.

3.7 3D Development of the airlock and EVA section

This is the last functional block to model to complete the final set of assets for the VR implementation. The airlock is necessary to access the rover whenever a scientific exploration mission on the lunar surface is scheduled and, after that, to leave the vehicle to perform an EVA to return to the main outpost or to conduct tasks outside. This system allows the presence of a pressurized atmosphere while the external environment is characterized by harsh conditions. The core principle of this device is to create an intermediate room that can transition from on extreme to another, leading the astronauts from a completely depressurized situation to a livable one. A common example to describe an airlock is the similarity to a dam for boats in canals: the ship enters the portion of water between the two bulkheads, the one from which it entered is closed, while the other is already sealed. Then the level of water increases (or decreases, depending on the way of transit) and the vessel reaches the height of the destination channel; at this point the second bulkhead is opened and the procedure is completed. Another popular example is the two doors system to enter the pressostatic balloons for indoor sports during winter.

3.7.1 Hierarchy diagram

As previously introduced in the last subsection, the airlock is made of two ports that identify an internal space, where crew members await the completion of the procedure. Handles and a locking device must be added to act on the hatches and to ensure the space is sealed.

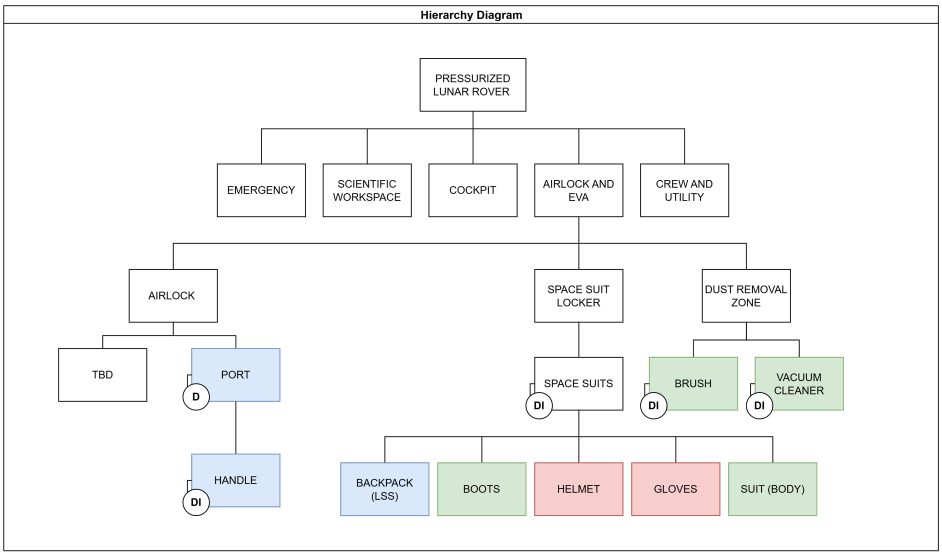

Figure 3.8: The breakdown of the airlock and EVA functional block into its base components

In the inner habitat, a portion of the room has to be dedicated to storing the space suits. In Figure 3.8, the garment is divided into the different pieces that must be worn separately. Another relevant area included in this functional block regards the procedures to remove dust and other particles from the worn suits; to do so a brush can be added to the collection together with a vacuum cleaner for more detailed cleansing.

The detail scales with the complexity of the objects: port, handle and backpack are very similar to primitive solids, while other elements need for a more careful modeling. All the items of the section are very interactive, in fact they show at least the dynamic label.

3.7.2 Referencing

The model for the space suits comes from the latest versions which were presented to the media as a result of agreements between NASA and partner companies and which should be used for the Artemis program. The airlocks located on the ISS are also a good starting point for the development of the rover one but need a greater grade of manipulation to consider the different gravity conditions that affect astronauts posture when crossing it. Components used to remove dust and contaminants are already common in space modules and are usually very similar to the tools designed for the daily housekeeping tasks.

3.7.3 Modeling

The high number of details and of little elements that enrich a space suits make it complex to reproduce in 3D. Another typical problem with worn components is their deformability and ability to be constantly attached to the body that moves them. Especially in the bending parts, the deformed fabric should form wrinkles without intersecting the mesh underneath. This graphical output needs a very high number of polygons and is in contrast with the smoothness of the simulation.

All the other items of this section are way easier to model since they are solid components that do not behave like cloths.

3.7.4 Ergonomics

The ports must have dimensions that can accommodate the passage of astronauts wearing a suit. The evolution of technology has led to a decreasing in the overall size of space suits and a gain in flexibility in the points that correspond to the joints of the body. The room between the two doors must be able to house two astronauts simultaneously and, to facilitate entry and exit procedures, the doors should be opened outwards.

Chapter 4

VR development

4.1 Introduction

4.1.1 Software selection

In the previous chapters, the building of the fundamental elements of the project was completed through a preproduction phase at first and then the modeling; now the single bricks must be put together to raise the final complex construction that represents the simulation experience.

There are two main tools that could allow me to reach this objective, they are very popular in many sectors, from the game design to the industrial field with very effective applications also in health care. This great, common ability to spread across a wide range of disciplines, due to their versatility, lets anybody who is eager to move further the boundaries of his field of studies express creativity and produce interesting pioneering outputs. The two platforms I'm talking about are Unity and Unreal, two powerful graphic engines that mark the standard in this share of market. The decision on which of the two to use in this project was mainly dictated by the preliminary materials received by Prof. Magesh Chandramouli, member of the Computer Information Technology & Graphics Department at Purdue University Northwest, one of the two creators of this entire Thesis opportunity. The chosen software is Unity.

4.1.2 Unity, an overview

Unity is a platform created in 2005 by Unity Technologies, initially to be used in the Mac Os X environment as a game engine and then enlarged to include Windows. During the last twenty years it has been constantly updated till the actual version, Unity 6. In my opinion, to easily explain this marvelous software to someone completely unaware of it, it could be useful trying summarizing its core functionality as follows: it allows you to import 3D objects coming from an external source and work on them to add a desired behavior, usually through the built-in coding language C# (pronounced C-Sharp); this process could led to the creation of a game, more in general to an interactive experience where inputs coming from an user affect what is happening in the scene. Unity includes the graphic engine to process all the information needed and it can build executable applications for multiple platforms. In addition, many packages can be added to extend the functionalities and to allow for wider interoperability between different environments [26].

As previously anticipated, the imported assets by themselves don't provide any interesting content to the scene more than visual enrichment. To create something a little more complex than a static rendered picture of the game space, assets need to be paired to a script to unlock a wider range of implementing possibilities. Scripts are files attached to a certain game object that can, for example, manage its position, its rotation and make it appear or disappear. Usually, the files aren't anything more than simple lines of text that get interpreted by a tool known as compiler that translates the strings of text into useful instructions for the game engine. Since a translation process takes place, compiler needs to receive the input file written in a standard language to be able to understand it; in the case of Unity, scripts must be written using C#. C# was created by Microsoft Corporation back in 2000 and on their documentation is described as a cross-platform, general purpose language [27]. It's the most popular one of a set introduced as part of the .NET platform, a useful tool that allows developers to work in multiple environments for the creation of their applications [28]. The suggested source code editor to write C# instructions is Visual Studio Code, developed by the already mentioned Microsoft Corporation [29]. This powerful editor also offers extensions to optimize the writing process with useful tips and shortcuts. The most famous plug-in is the C# Dev Kit [30].

4.1.3 The learning pathways

This kind of software is rarely used during the common aerospace engineering education programs. In my student's career, before this very project, I've gotten in touch with Unity only once, in an elective course during my bachelor's degree; its purpose was to list all the most advanced and innovative technologies for the digital representation and Unity was introduced as a tool to create AR experiences for smartphones. Since then, Unity has been an unused icon floating in my desktop, till now. The level of knowledge required to complete this kind of project is no way comparable to the information I gathered back in 2022 to produce my simple AR app. So, I started following the tutorials provided by Unity Learn, a section of the website where many contents are available for everyone [31]. The amount of learning material is huge, so the whole set of tutorials is organized into different courses called pathways: a student can choose between the different options depending on his level of proficiency in using the software and on the final output he aims to reach. My final objective was to get acquainted with all the most important VR features, so I immediately selected the path called VR Development. The site though suggested me to attend some preliminary courses before jumping into the core subject, to familiarize myself with the interface and better understand the coding basis implied in the development. My final learning plan included: Unity Essentials, Junior Programmer and lastly VR Development. This particular sequence of classes was suggested by the platform itself for whoever wants to master the main aspects of the software and it's organized as follows: the first pathway is considered foundational and has an estimated completion time of two weeks; the second one, probably the most complex of the three, has received a beginner level categorization and requires approximately twelve weeks to be finished; lastly the VR one is again beginner and takes six weeks.

4.1.3.1 Unity Essentials Pathway

In Unity Essentials students receive their first instructions about the creation of a new project exploiting the Unity Hub; then they're introduced to the Editor interface, where all the main functions are gathered and organized in different windows. After a brief list of ways to explore the scene view, they have all the instruments to dive into the software. This first pathway provides an overview of different core development elements and mainly aims at triggering the curiosity of users by providing interesting examples on possible application of the presented material. The tutorials are divided into several essentials categories: after the Editor Essentials it's presented the 3D Essentials, where students place their first blocks into the scene to decorate a kid's room and exploit the first physics simulation by creating a little homemade bowling alley. In this lesson the first concepts related to light setting, background images and camera positioning are also introduced. The following section is Audio Essentials, which are sometimes overlooked by developers who prioritize the visual result of their products but are crucial to immersive virtual reality scenarios. Audio sources can represent sounds generated by the outer environment, in case of indoor scenes or, for outdoor settings, by sources not visible to the player; in general, they are considered as background sounds, merging to form the so-called soundscape. Another category of possible audio sources that can enrich this sound-based landscape is way more related to the objects included in the scene: to each of them, developers can add audio clips resembling typical noises of that particular item, for example a bell can produce a ring, a cat a meow and a turntable a catchy tune. The main difference is in the duration of the sound and in the play mode: continuous audio or triggered mode, like the doorbell. The Programming Essentials provides a first sneak peek of the huge world of C# scripting. This subject will be explained in a more detailed way in the Junior Programmer pathway later. The 2D Essentials covers a different concept of game design, probably the oldest thus still very popular one, with all the objects moving only on a plane and the camera placed on the normal of that plane, resulting in the loss of a dimension; arcade is a perfect example. The last section of the pathway is named Publishing Essentials, pretty self-explanatory in the title and explains how to create a portfolio and make your works available for the community.

4.1.3.2 Junior Programmer Pathway

The Junior Programmer Pathway is probably the most effective way to get acquainted with the C# basics. The first mission focuses on the player controller, meaning how to transfer input that a person provides to a controller into the change of position and orientation of a game object or a camera that represent respectively the alter ego or the point of view of the player inside the game scenario. The first script created performs exactly this function. The next mission provides information for basic gameplay implementation, as the player alone, being able to explore an empty world, doesn't seem to be such an entertaining activity. Therefore, other objects are added to the scene, with events associated with them whenever a collision happens with other game objects or the player itself. Then a brief focus on additional details that can be added to enrich the experience, like particles and sound effects to provide greater feedback on the actions of the player through visual and audio aids. The Gameplay Mechanics section helps creating an additional scripted structure to the game with methods, portions of C# code with very specific functionalities, that regulate the behavior of the gameplay like the drop of powerups, increasing difficulty or the regulation of the enemies' creation. The following mission focuses on the UI, the set of text and menus that get displayed on the screen to help player getting information about the game he's experiencing. Examples of this interface are the in-game text showing the score reached during the current game session, the game-over message that pops up whenever the player loses his match or the initial and settings menus. User interface needs accurate study because it is a sort of frame that contains all the instructions and hints that will guide players during the whole game experience, it must be very clear and concise. In the following mission, it is explained how to properly publish your project. Like many others code-related activities, before submitting your work to a wider audience, deep optimization and troubleshooting must be conducted, preferably throughout a test campaign. The two last missions are somehow more theoretical; they try to explain some important underlying concepts about object-oriented programming and some advanced technics to manage the scripts during their compilation to achieve a higher optimization.

4.1.3.3 VR Development Pathway

This is the last pathway, finally focused on the creation of an interactive virtual reality experience. The first mission, VR Basics, introduce students to the main elements of a well-designed interactive scenario, with several additional components that must be included to manage the main peculiarity of this type of application, the tight relation between movements of user and the transformation of his in-game point of view. These settings are somehow gathered in the XR Interaction Toolkit. Students receive a scene of an empty room, they get asked to decorate it and convert some specific assets into interactable objects, stuff that can be grabbed pressing buttons on the controller. The other main feature explained in this foundational first mission is about sockets, special elements that allow user to release certain objects only in its appropriate position of destination (e.g. hats on a hanger). The focus of this first mission also covers a relevant aspect of the preliminary building of a scenario: locomotion. Developers must decide how players will be able to explore the in-game scenario; given three main different locomotion systems, each has pros and cons for user experience. The second core topic deals with Events and Interactions, starting with an introduction of audio and haptics feedback to increase immersion, then explaining how a user input can lead to an output, which is basically the definition of events. These actions can take place thanks to special instruments known as interactors. Finally, the lesson introduces the UI elements applied to the VR world, with several additional functionalities compared to the one showcased in the previous pathways. The last tutorial-based section of the course provides simple concepts of Ergonomics and Optimization, crucial in this application. The first important lesson learned about ergonomics includes considerations on comfort and accessibility, allowing everybody to perform the in-game activities no matter what are is physical abilities or constraints. The optimization instead is more code-centered and deals with organizing the whole scripted structure in a way that speeds up the execution of the back-end computing; lighting plays a very important role too. The very last section represents a do it yourself scenario where learners should demonstrate the skills they've acquired throughout the whole pathway.

4.2 Unity preproduction job

4.2.1 Game modes

After following three different Pathways on Unity Learn, it seems time to start creating the experience through the Editor. However, as an essay needs an outline before being written effectively, this project must be carefully studied to identify the different actions that the user will be able to perform within it. Until now it has not yet been clarified what the simulation will include in detail and, as a result, there is still no precise plan to follow step by step during development; undertaking such an advanced stage of the project without having it would be a big risk, so first it is better to work on a storyboard.

Fortunately, the user personas created in the second chapter come to the rescue and allow me to establish an initial separation between possible experiences within the app. The main purpose for which this whole simulation is going to be built is the need for a training facility, in this case a VR one. To address this first objective, strictly related to the identification of the astronaut user persona, the simulation will certainly include a Single-task mode, that allows players to focus on a certain task and eventually repeat it until they feel confident enough to perform it. Furthermore, to create deeper immersion and get users more involved in the scenario, an additional mode will provide the same tasks organized in a logical sequence and it will be called Storymode. The side objective of this application is for outreach, so a third mode will be implemented displaying the whole inner habitat and letting users explore it and interact with a group of items. This last gaming way will be the Free-roaming mode.

4.2.2 Tasks identification

In the previous chapter I described the modelling procedure leading to the 3D rover generation. The inner habitat is made up of five main high-level blocks, two of which are the most important ones according to the functions' allocation. To carry out a more in-depth study, I will focus on a single block: the cockpit. To produce a storyboard of any type, it is necessary to have a narrative in your head that you want to tell; from an engineering point of view, it is more than good to have a mission to accomplish and break it down into the tasks that composed it. The main purpose of the cockpit section is to provide all the instruments to drive and to control the status of the vehicle. The simulation won't include the effective movement of the rover on the lunar terrain, because it'd require the implementation of complex laws that regulate wheel soil interactions, known as Terramechanics models, investigated by a very fast-developing field of study. After having defined the boundaries of the project, it's possible to identify the relevant mission the players inside the simulation will try to accomplish as training. Users will be required to perform a series of tasks finalized to set up the vehicle subsystems for an upcoming trip to be executed by the rover. An additional set of tasks, placed after the trip completion, allows players to acquire more confidence in the inner habitat by converting the section into a resting-ready zone and experience an off-nominal condition.

The preliminary activities to be carried out before the driving takes place are turn on the Multi-Functional Display to conduct a thorough check of the subsystems, activate the outer cameras view in the monitors placed all around the pilot seat to have a greater view of the surroundings and finally to set a destination in the navigation menu. These three preparatory driving tasks and the following two make up the mission on which the Storymode is based. Subsequently, each individual operation must be further broken down to create the final detailed storyboard that can be used as a precise blueprint to follow along during the building process. There every input action must be listed and related to a result, in an input-output relationship so that everything can be implemented in the C# code scripts.

4.2.3 User interface planning

After having identified a first draft structure for the app, another very important step regards the UI. User interface is the set of elements that allow users to interact effectively with a device located near them; part of this family of graphic items are all the messages that inform or instruct the player on an event that is happening and how to deal with it, the buttons for making decisions and proceeding with the activity, sliders to toggle the intensity of a certain parameter and many other icons.

Within the application, UI elements can be classified in at least two ways, based on the criteria that someone prefers to follow. The first subdivision can be somehow summarized as menu versus gaming interfaces: the menu category is made of icons initially shown to players to guide them through the menus to their desired destination, the game scene, and they represent kind of a transition between different branches of the app architecture; setting windows are part of this category too. Gaming interfaces are instead included in the game-play and are directly related to the flow of the player's experience while he/she is enjoying the game: texts appearing during a conversation with another character, instructions and all the fundamental elements necessary for the continuation of the adventure. If the previous group is like the frame, this family represents the paint inside it, the focus of the attention. Another way of distinguishing UI components is less functional and more technical, related to the Unity Editor built-in logic and specifically to the Canvas Render Mode: Screen Space versus World Space UI. The Screen Space overlay places elements on top of everything else and is scaled to fit the screen; it's probably the most common method used to display important information such as health status, timers or inventories. World Space overlay instead places UI objects within the scene, like any other asset included in the collection and players are able to move relative to it and they can watch it from different perspectives.

The experience presented in this Thesis is full of interfaces, mainly due to the peculiarity of the section it focuses on. The core device placed in the middle of the cockpit, right in front of the pilot seat, is the Multi-Functional Display, that can be seen as a computer and all the design activities related to its implementation are no different from common Web Design procedures, with the difference that MFD's developing doesn't exploit HTML, CSS or Json but is limited to a bunch of images and TextMesh Pro tools. All the menus, tabs, sections of the MFD are created as World Space overlay UI elements inserted in the computer frame coming from Blender. Another great number of items increasing the count of UI instances are the messages displayed to users to guide them during the whole experience, updating whenever they complete a task and helping them with useful piece of advice.

4.3 Creation of the project

4.3.1 Version control

The time has finally arrived to dive into Unity. Before starting to work in depth on a project, in the world of computer information technology, it is good practice to set up a version control system, which is a way to keep track of all the changes that a file undergoes. This is useful both for going back to a previous version of the product in case some new changes have affected the functionality of the project, and in case of data loss since a version control is basically a backup that is updated whenever the developer deems it appropriate. It is also very useful because it allows sharing a repository where all team members can access and transfer the content they're working on. To avoid problems of overwriting, there is also a whole further system that manages the main strand of the project and all the branches that derive from it, ensuring that everyone acts on the section under their responsibility without disturbing the labor of others.

Unity is equipped with its own integrated version control system which therefore seems to be the wisest choice given the type of activities I'm about to carry out. However, it has limitations, mainly due to the need to configure it via a developer account, which requires being part of an organization that essentially bears the cost of recognition by Unity.

A great alternative, strange to say, is proposed by Unity in one of its Learning Pathway: it's GitHub, the platform used by millions of developers worldwide that exploits Git version control features. All file upload and download activities, known as push and pull respectively, take place via Command Prompt, which does not make this system very user friendly, but with a bit of habit anyone can get hang of it. That said, it's better, even before creating the project via Unity Hub, to prepare a folder that is connected to the GitHub repository; at this point it's possible to generate the new working environment by placing it in the location just prepared [32] [33].

4.3.2 Unity Hub

Unity Hub is the tool used to manage projects in progress and create new ones. Furthermore, it is also necessary to install any desired version of Unity on the device as it has a list of all available releases from which developers can choose. Novice users are advised to equip themselves with the latest version marked with *lts* icon, i.e. long-term support, which means that it is stable and will be maintained in the near future. In the Installs tab there's the list of all the editors downloaded in the computer and for each is necessary to add a module, the Android Build Support, to be able to build the application on the VR headset.

Unity Hub allows developers to set up a project from scratch using templates based on the final purpose of their application. To have the previously created version control system up and running, user needs to save the newly generated working environment in the folder linked to the GitHub repository. At this point the chosen Editor will be launched and after a reasonable amount of time it will be possible to start working.

4.3.3 XR Interaction Toolkit

Having created the project via the default template called Universal 3D, in the opening sample scene there is mainly a directional light and a camera. To ensure that the first features related to VR can be correctly tested, it is necessary to import a series of tools; they are gathered in a single place, known as XR Interaction Toolkit, critical throughout the development of the entire experience. This important element can be easily imported in its most up-to-date release within the software, via the built-in package manager. In addition to the toolkit itself, it is possible to download a rich collection of examples that delve into peculiar and innovative aspects of the tool; certainly, however, the most precious additional resource that a developer can equip himself with is the XR Device Simulator, which will be explored in depth in the next section.

For the moment it is a priority to focus on the XR Interaction Toolkit since it is the basis of the mechanics that manage the experience; to do this it relates the inputs coming from the user via the controllers and the headset to the scene he sees live through the camera. As regards the interpretation and processing of the movement data coming from the VR headset, they occur automatically in the depths of the software and there is no margin for intervention during the implementation phase. A little more can be said about the inputs coming from the controllers. Each one has five pressable buttons and a joystick: usually the joysticks are used for locomotion input, with one controller that takes care of the translational movement and the other of rotations. In the rear part of the object, usually operated with the index finger, is the trigger button, responsible for the main selection event, similar to the left click of a traditional mouse. On the internal side of the device there is another important snap, known as grip, which is mainly used to grab items in the scene and to hold them. Near the joystick can be found two additional buttons; these two, in the initial XR interaction setup, don't correspond to a precise function, but developers are able to assign one depending on the game commands requirements. The last key in importance allows users to pause playing the game and return to the main Meta Quest menu at any time.

One of the basic features to know and to master within the Toolkit is the choice of the types of interactors that can be assigned to the left- and right-hand controllers. In the previous versions there were mainly two choices: ray or direct interactor. The fundamental concept behind this distinction concerns the range of action of the player's in-game hands. One interactor associates a ray that originates from the hand and points endlessly frontally; any interactable object that is crossed by this line can be selected, grabbed or any other scripted action, regardless of the distance from the player. The second possibility is a direct interactor, in this case the position of the user's controller with respect to the element with which to operate is fundamental to determining whether the interaction can occur or not. This type of setup is not very complex to get because it is exactly what people experience in everyday life using hands. In the latest versions, hybrids have been created between these two main interactors to make the gaming experience more pleasant and smooth; a further type has also been introduced exploiting data coming from user's gaze recorded by the headset.

Within the functionalities managed by the XR Toolkit there are obviously also setting for locomotion. There are three ways to move around the scene: the first records the real movement of the person wearing the headset reproducing it in the simulated scenario; the second uses inputs coming from the joysticks to create a continuous movement, typical of most games in which players control a character. Finally, the last method uses teleportation, therefore always with joysticks inputs thus creating a non-continuous movement; this may seem absurd but there is a clear motivation behind the success of this latest mode in the VR world. The first mode is certainly the most immersive, but there are major limitations related to the space available for the user to move around while playing, especially if the experience is set in large outdoor spaces. The second locomotion choice certainly represents a simple solution to the previous problem, but it brings with it a further particularly debilitating side effect which makes the experience very uncomfortable: the illusion of movement perceived by the player's eyes conflicts with the perception of immobility sensed by the vestibular system in ears leading to simulator sickness. For this reason, the third option of locomotion has gained great popularity, creating instantaneous movement from one place to another through teleportation, not producing this effect of discordance between the two senses

4.3.4 XR Device Simulator

As mentioned previously, together with the XR Interaction Toolkit, among the additional resources, it can also be imported the XR Device Simulator. This component allows developers to easily launch the simulation within the editor's play mode window, without being forced to link the headset to the computer and experience it from there. Convenience becomes a necessity if the computer is unable to meet the minimum requirements required by the Meta Quest Link software and is therefore unable to connect with the headset. In this case, if the Device Simulator were not used, every time developers wanted to test the latest improvements added to the project, they would have to build and run it as an application, a very long process.

Obviously, there are limitations on the functionalities that can be exploited replacing the controllers with mouse and keyboard during simulation; it is not a complete experience, but it constitutes a quick useful check of the achieved progress. Finally, before launching the app's build process, it is necessary to disable the Device Simulator from the scene, otherwise it will conflict with the controllers leading to major errors making the game unusable.

4.3.5 3D Assets import

After testing the basic features of the VR experience, all the 3D assets can be imported to enrich the scene. It is preferable that the models are exported from Blender in .fbx format, with all modifiers, rotations and scales applied, so that there are as few problems as possible when opening them on Unity. It would be better if objects made up of many child elements with a mainly scenographic role were joined under the parent object to shorten the list of instances in the Hierarchy window. For materials, anything in the Blender's Node Editor more complex than the Principled BSDF connected to the Material Output will be discarded from the transfer file format. Finally, to avoid strange effects such as transparent planes, it is a good idea to check the orientation of faces and eventually flip normals.

Let's move onto Unity: first, the rover shell is dragged and dropped in the Hierarchy window to make a Game Object out of it. It is essential that the walls of the rover are impenetrable to the object inside it and to the player himself; for this to happen, the object must be provided with a mesh collider, i.e. a component that makes the grid resistant to any other object that tries to pass through it. The problem that has arisen is that by applying the mesh collider to a closed object (like a cube, cylinder but also the rover's shell) the software will reason in such a way as to keep anything out of it, to guarantee its physical integrity and avoid interpenetration; obviously, this reasoning makes no sense with the structure of a rover which is by definition closed but which must necessarily contain elements inside it and should not push them back out. After some research and tests, the best solution devised is to return to Blender, separate the shell structure into smaller pieces like left and right shell, ceiling and floor and then import them as separate parts onto Unity. At this point the component is applied to each of these objects which are no longer interpreted as a single box but as different walls that must not be crossed, regardless of the sense of movement.

4.3.6 Gravitational field

Within the simulation environment there is a preset gravitational field, corresponding to the commonly accepted mean value of the acceleration of Earth's gravity. In Unity's Project Settings, under the Physics tab, it is possible to insert any desired gravitational field by entering the numerical values along the three axes. As regards this case, just enter the lunar gravity, approximately one sixth of the pre-inserted entry on the y-axis, Unity's vertical direction or rather the normal one emerging from a spherical celestial body.

4.4 Free roaming mode

The first possible game mode within the application is free roaming, where users can move undisturbed inside the pressurized lunar rover. The right scene can be selected from the main menu through an element located in mid-air; a special interactable button will launch the loading of the chosen experience. In this kind of experience, users do not receive specific instructions on actions to be performed aimed at carrying out system control routine or any type of complex mission. For this reason, the use of UI messages in this scenario is much more limited than in the other two modes available, in which players must be accompanied step by step to achieve tangible objectives; the only useful piece of information for those testing the free roaming simulation is an image that explains how the controller keys work, both to manage movement and to interact with the surrounding objects.

Having already all the imported assets and the XR Interaction Toolkit set up and running, the main step to implement this mode involves creating the interactable objects. The interactions between the player and the items present in the scene can essentially be of two types: human-machine if they take place via an interface of some on-board system, providing that a user's click corresponds to a certain reaction from the menu in front of him; grabbable object if user can select an object and hold it in his hand as long as a key remains triggered. In free roaming mode, the possibility of grabbing object is preferred, so as to make players feel the sensation of handling components in reduced gravity. Due to the VR device used, it is not possible to make the weight of the picked elements perceived, but through their release can be appreciated the lunar gravitational conditions.

On a technical level, adding an XR Grab Interactable component to each graspable object also allow to manage its physical properties of mass, drag and establish how the movement of the controller influences the kinematics of the held item. In this way a collider is also assigned, a sort of cage that prevents the solids from interpenetrating and makes the simulation extremely more realistic by adding collisions and rebound against other surfaces.

The ability to throw a large number of tools around the room seems to be very appealing to many users, whether they are interested in the rover from a scientific point of view or not. Their enjoyment would probably decline if they were asked to clean up the confusion they have created; for this reason, a convenient reset button allows them to return all the instances to their original position, reliving everyone of their tidying up responsibilities.

4.5 Story mode

4.5.1 Introduction

The second game mode in the application is the so-called Story mode. The setting is the same as the one presented in the previous section, the well-known inner habitat of the pressurized lunar rover, but this time instead of exploring the whole environment, users are invited to focus on the cockpit section. To do so, instruction messages appear in front of them, delivering a sequence of tasks they should achieve and useful advice to succeed in them. The detailed sequence of the tasks has been presented earlier in this chapter, in the Tasks planning subsection of the Unity preproduction job section. Later on, they will be delved into, breaking them down into each single action (or input) that users should perform and the related result (or output) provided by the interface they are operating. The core idea behind the whole experience is to connect the events happening in the MFD to the scripts that manage the instruction messages and the pop-ups appearing on the display itself whenever an off-nominal condition is detected. For the good result of the narrative, the initial situation has pre-existing issues which are revealed during the execution of the simulation as the actions proceed. Furthermore, users intervention on the system leads to further problems, triggering a chain reaction that encourages those who work to delve deeper into the close link between various elements of the rover and how they interact with each other to guarantee an effective division of the available resources.

Now the time has come to implement everything in the Unity Editor, following the tasks tables as blue print of the project.

4.5.2 Task 1: Life-support System

The initial task is part of the first group of activities, those that revolve around the MFD and that are preparatory to a driving mission. This in particular investigates the LSS and the power section. Table 4.1 shows the list of events that defines this first mission. There is a continuous repetition of a precise pattern: certain instructions provided by some element of UI, one or more user's actions aimed at satisfying the request and a message to mark the completion of it.

#	INPUT	OUTPUT
a.		A World Space UI invites users to switch on the MFD and conduct a check of the rover subsystems
b.	Users press the MFD power button to turn it on	A welcome screen pops up in the MFD, then the home window
c.	Users explore the MFD's menu and selects the Vehicle Status Tab	The Vehicle Status Tab opens showing all the different indicators
d.		A warning message informs users about a problem in the oxygen generation, indicator turns red to highlight the issue
e.	Users click on the oxygen indicator to access the specific menu	The menu shows a low level of oxygen for the internal habitat
f.	Users can modify the oxygen production grid to enable a new line of supply	The screen shows an increasing level of oxygen production; the indicator turns white
g.		A warning message informs users about a problem in power generation (consequence of the previous action)
h.	Users open the energy control section	They find out that the power budget/balance is negative
i.	Users explore the solar panels menu by clicking on the proper icon	A map of the solar panels mounted on the rovers shows up, indicating which ones are deployed or not (+ information on how much a certain panel produces/could produce if deployed)
j.	Users must open enough panels to turn the balance to positive by clicking on the closed panels in the menu	When the energy production is higher than the expenditure, the warning message disappears, the UI marks the task completed

Table 4.1: Detail of the Life-support System task

One peculiar element used in this first task is a physical button, an object which exploits the properties of rigid bodies to collide with each other to produce signals to be associated with an action: the activation of the MFD. As regards navigation within the screen menus, most events are managed by UI buttons that enable or disable the visibility of images or text strings. Another way to control the on/off status of a variable, usually a boolean in the script, is the toggle switch.

The core idea, common to the whole story mode experience implementation, is to create a script, known as UI manager, that rules over the entire sequence of instruction messages; it is specialized in receiving input information about players progress and provide them, as an output, with updated advice throughout the path. The second key script is the MFD manager, that controls the windows and the various UI elements included. Finally there is a series of shorter scripts, very specialized on a certain function to be performed or a subsystem to be controlled, such as the codes that manage oxygen and power emergencies.

4.5.3 Task 2: Cameras

The second task derives from the need to increase the pilot's visibility over the surrounding environment. For structural reasons linked to the pressurization of the internal habitat and the resistance of the shell to the impact of micrometeoroids, the surface occupied by windscreen must be limited; to compensate, monitors are placed in specific positions to allow pilot to grasp with a quick glance what is happening around the rover. The VR version of this driving aid is no different from the real one, with cameras placed outside framing a portion of lunar ground and broadcasting the captured view to internal monitors.

From a technical point of view, Unity provides an interesting tool for transposing images coming from a camera into a texture, the Render Texture components, as if it were to expose an old analogue camera's film. The textures thus created can be used as Base Map input for a material and it can ultimately be assigned to a plane to simulate the presence of a video on a screen.

At this point, reading Table 4.2, arises a dilemma over the design of the window of the MFD which manages the allocation of the cameras. The first option is to have icons representing the cameras on the menu and for each a list of the possible monitors where to display each of them. The other possibility is to have the layout of the screens and by selecting each of them the list of possible cameras appears. From a coding point of view, the second path turned out to be simpler; furthermore in this way the user has the opportunity to locate all the different monitors in the scheme in front of him.

An issue related to the 3D assets becomes apparent while working on this task. The structure of the shell and all the internal components were imported into the scene; however, this mission also requires the exterior to be enriched to make it decidedly more lunar. Right now Moon's soil is a simple flat surface and a default Skybox fakes the common Earth's cloudy horizon. The presence of clouds, if we exclude those of dust, and a blue sky is a bit unreal given the lack of atmosphere. The default Skybox should be changed into a dark one, with eventually a little Earth showing in the distance. The terrain also requires a more in-depth analysis to make its representation realistic: the shape should reproduce all the roughness formed by meteoroids impacts over time. Finally, the material should try to imitate the appearance of regolith, its poor contrast and its crumbly and yielding consistency. As regards the external equipment of the rover, adding wheels, solar panels, antennas and special coverings would help to make the external view more faithful and immersive.

#	INPUT	OUTPUT
a.		A World Space UI invites users to activate the external cameras of the rover, to place each in a monitor and to turn on the lights
b.	Users must press the cameras tab button in the MFD menu list	Cameras window appears with all the info to manage them
c.	In a drop-down list, users select in which monitor he wants to display every view	Every camera view appears in the chosen monitor
d.	In the MFD, users can control the external light settings by clicking the proper icon on each camera panel	Each view gets lighted up; a message marks the completion of the task

Table 4.2: Detail of the cameras task

4.5.4 Task 3: Navigation

The third activity involves inserting a destination for the trip by choosing from a list of options. Ideally it should be possible to enter lunar coordinates to be able to reach any place on the Moon's surface; in this simulation though, to simplify the work and to avoid making bulky keyboards appear on the screen, the reachable destinations are preset and users must stick to them. The list of development steps in Table 4.3 mainly reproduce elements of the previous tasks like UI messages and drop down menu-ruled events. In addition the navigation menu includes several images of the lunar terrain created in Blender. Probably one of the most interesting development points of this section is the addition of a series of icons, sort of navigation aids to guide pilot toward his final goal, known as pictograms. This group of arrows is displayed on the MFD but should not be visible to the user watching through the windscreen. The simplest way to solve this issue is to place all the icons under a special layer and then mask out that layer from the main camera visibility options, the one which is moving with the XR rig. The images displayed in the central screen come from the front camera and if the pictograms layer is correctly listed as one of the visible, the whole mechanism perfectly works. The conclusion of the task requires a triggerable object that can track the collision of the hand controller entering the borders of a defined volume. This event should set a direct connection between users' right hand and the joystick placed on the arm of the pilot seat; one of the buttons previously used to select or to grab items, now controls the brake system of the rover and its release. At the end of this mission, a scripted series of events should fake the ongoing driving phase, not part of the simulation and fast forwardly bring players to the desired destination of their voyage.

#	INPUT	OUTPUT
a.		A World Space UI prompts users to set a destination in the navigation section of the MFD
b.	Users select the navigation tab in the MFD	The navigation menu appears with a map of the lunar surface around the rover's current location
c.	Users click on the new route UI button	A list of possible destinations appears
d.	Users can highlight each of them	When highlighted, a possible route expands showing the distance to target, the time to target and the planned path
e.	After deciding, user can click on the confirmation UI button to start the trip	The window shifts to a different view, with labels overlayed on the map to mark the planned route the pilot should follow and info about speed and range
f.		A World Space UI marks the end of the task. The following instruction asks user to remove the parking brake and start the route
g.	Users must select the joystick on the right side of the seat	Now, with their right-hand controller, users can directly control the command joystick
h.	Users hold a certain controller button to remove the parking brake	The additional task is completed, an image overlays the screen, needed as transition to the arrival destination site since the simulation of the real movement is not included in the experience

Table 4.3: Detail of the navigation task

4.5.5 Task 4: Seat to bed conversion

This is the first activity within the story mode that doesn't incorporate the MFD as main interface to succeed in the proposed mission.

#	INPUT	OUTPUT
a.		A World Space UI invites users to prepare for the night
b.	The first thing to do is to shadow the windscreen by clicking a proper physical button on the central cockpit	The windscreen darkens and the view on the surrounding land is limited
c.	Then users turn the MFD into sleeping mode by clicking a proper UI button inside the home window (called night mode)	All the camera monitors and the main screen of the MFD go off
d.	By opening a little panel, users can access a console placed on the arm of the seat	A new set of buttons shows up, allowing usesr to manage the reclining of the seat
e.	By rotating a control knob, users can adjust the inclination of the backrest and footrest	When users toggle the correct angle for backrest and footrest, a green light appears on the console
f.	Once the seat is in resting configuration, users select a physical button to turn off the lights	All the internal illumination goes off and the emergency path (plane-like) shows up in the semi-darkness
g.	Now users can fall asleep by leaning their head on the headrest	A scripted black overlay appears simulating the eyes shutting
h.		After a certain amount of time (10-20 s) the eyes open and a good morning message pops up; then it suggests following the whole procedure backwards to turn the cockpit to the driving configuration
i.	Users press the physical button to turn back on the lights	The internal lights turn on and the emergency light path disappears
j.	Users rotate the control knob to bring the seat to the vertical layout	The seat follows the commanded rotation and when the vertical is reached a green light appears
k.	Users reach the central cockpit to clear the windscreen by clicking the proper physical button	The windscreen turns transparent; a task completion message appears

Table 4.4: Detail of the night routine task

The first detailed point listed in Table 4.4 asks users to prepare the system for the night, setting up a series of interactable elements to a resting condition. The first step involves darkening the wind screen by clicking a proper button located in the central console of the cockpit; the event should trigger a scripted method that changes the transparency property, called alpha in the software, of the material linked to the two windscreen assets in the scene. The only interaction with the MFD aims at turning it into the sleeping mode. The leading element of this task is the pilot seat and the controls on it. The recline is controlled by a knob which has a completely vertical configuration at one hand and a barely horizontal layout at the other. The player is asked to bring the seat into sleep mode by acting on the knob; at his command the rotation of the backrest and footrest begins. The last thing that divides the exhausted astronaut from a good sleep is the rover internal light on, which is why a handy button near the tilt control allows him to turn it off without leaving the laying position. A sequence, similar to the driving part, fakes the whole duration of the bed-time and in less than a minute user is awakened and prompted to reset all settings to make the vehicle operational again.

By the technical point of view, all the functions invoked in this task are recycled from the previous activities, except for the tilt control of the seat's elements. Until now, none of the physical or UI buttons directly operate on the kinematics of an object in the scene, in this case to regulate its tilt. For the very first type in the project, three dimensional vectors need to be accessed in the script to couple the output produced by the knob to the real rotation of the backrest and footrest. Furthermore this new type of command is very useful in the first task too: the switch toggles that deploy the solar arrays can be updated to rule over real 3d panels placed outside the rover to rule over their configuration, enriching the scenario with more interactable elements visible in some external cameras. The difference between the cases is the on/off nature of switch toggles opposed to the high angular precision of a knob that allows players to select intermediate angles between the two extremes.

4.5.6 Task 5: Emergency

The last task is intended as an off-nominal situation, with alarms and lights marking the emergency status crew members are into. Until now, players have been able to complete all proposed missions without leaving their seats; the sudden condition of danger presses them to get up and head towards the spacesuits. Table 4.5 specifies the need to create a button that activates the component related to the locomotion that allows users to stand up and move away from the cockpit. Suit equipment can be managed via socket interactor, a tool that allows an object, when grasped and brought close to a certain area, to occupy a predefined position and to be stuck to it during movements. Then each part of the suit will occupy the most natural position above user's imaginary body. The most important element is the helmet, because it will have to be placed so that the visor is perceptible in front of the main camera and vital data will have to appear superimposed on it. These UI elements are completely similar to those created previously, for example to those that provide instructions in front of the seat, but have the particularity that they must be hierarchically child objects of the helmet to remain fixed in a portion of astronaut's field of view once the gear is worn.

#	INPUT	OUTPUT
a.		Suddenly, an emergency message shows up on the MFD, saying that a severe pressurization loss is happening. Red emergency lights begin to flash; a UI countdown starts. A World Space UI urges users to wear the spacesuit
b.	By pressing a VR controller button, users can leave the seat and stand up	Now users can freely move and rotate in the inner habitat
c.	Users walk to the suit locker and start selecting each part of the suit	If clicked in the right order, the different parts of the equipment are worn by the users; when all the elements are placed correctly, a success message appears
d.		A World Space UI invites users to activate the Life Support System of their suit
e.	Users can click on their wrist tablet to manage all the vital parameters and activate the vital related subsystems	When activated, O2 level indicator and others (°F, pressure. . .) get displayed on the Head Mounted Display of the helmet
f.		A World Space UI appears marking the completion of the last task and of the whole training experience

Table 4.5: Detail of the emergency task

4.6 Task mode

The last game mode is called Task mode to accentuate the goal of focusing on executing a single task to train in carrying it out most efficiently. From the starting window of the main menu scene, selecting the proper UI button, a list of the five tasks appears and users can pick the one they are most interested in. The scene that loads is similar to the story mode one, but now the instruction messages prompt players to fulfill one single mission instead of the whole set of them. Although the first idea was to create a new scene for every single task to include only the necessary scripts to each of them, ultimately for this training experience users are directed to the exact same scene used in the story mode. It is then necessary to create some sort of index with which to precisely target players to the line of code where the task of their choice begins. A variable must be created in the main menu to store the number of the chosen option, then it needs to be transferred to the game scene to locate the correct starting point. When a boolean variable marks the completion of the task, a specific method should stop the scene from executing and return to the starting menu.

Chapter 5

Results

5.1 Main menu

Launching the RoVR app, the very first scenario appearing in front of users is an empty lunar terrain, without vehicles, astronauts or any remarkable evidence of human presence. In Figure 5.1 the dark sky in the background is full of blinking stars and planets. This breathtaking view is made of a simple plane with a displacement modifier applied to it to fake the lunar bumpiness and a Skybox, a cubic item used to reproduce sunlight and weather conditions, in this case to cast the Milky Way on the celestial dome. One single object suggests the possibility for players to perform some kind of activities in this marvelous but a bit desolating setting: a big, mid-air floating menu interface breaks the monotony of the surrounding area and provides the very first possibility of interaction.

Figure 5.1: Main menu scene

The UI element let users pick their desired experience between the first set of choices: free roaming, story mode and task mode. Each pickable option is created as a button, which provides a bunch of feedback to players' eyes: the box changes color whenever the controller hovers over it, helping to locate the end effector of the ray interactor at the distance, producing a sound when pressed and allowing to implement all a new series of additional properties changes from the Inspector window. There are mainly two levels of selection, one which allows to distinguish between the playable modes and another, accessible from the previous one, which focuses on a more detailed choice between a deeper level of scene subdivision. By clicking on the story mode text, the new page that appears offers two possible choices, the cockpit or the scientific workspace experience. The task mode button instead brings to a list of the five possible missions. The free roaming loads directly the gaming scene. As I mentioned before, the handiest type of interactor to be used in this sort of pre-game lobby is the ray interactor: with its long-range capability lets players from every distance location reach the elements inside the starting menu.

5.2 Free roaming mode

The free roaming mode has to be the most accessible of the three gaming experiences, since it's mainly designed for the outreach and has to provide a non-technical overview of the inner habitat of the rover. The interactions in this section are not very training-centered but have the main goal of providing an example of the peculiar lunar gravitational field. To show the effects of the reduced gravity, users can grab objects in Fig. 5.2 using a direct interactor and can throw them to experience their strange way of falling and bouncing against walls and other assets. Each graspable instance has a proper empty object that serves as handle to guide the correct position of the item in players' virtual hands.

Figure 5.2: Free roaming mode scene

The movement inside the room is managed by a locomotion script that detects input coming from the analog controllers. The section where the highest number of interactable objects are located is the scientific workspace, with a large group of measuring instruments placed on the desk or in the drawers below. The reason why the workstation is so rich of active elements it's mainly due to the lack of them in the cockpit zone during the story mode experience; to balance the poverty of grabbable object during the execution of the sequence of the tasks, the free mode has been willingly structured in a way that the defects of a game mode are somehow compensated by another.

5.3 Story mode

5.3.1 First task

The story mode path starts in a different spot than the free roaming, users are located seated in the driver's seat and initially are not able to leave this position unless a button is pressed on the controller. During the first four tasks there is no need for players to stand up and walk away from the cockpit, they can technically do it but a warning message would immediately invite them to reach their seat. The distance of the seat from the MFD is adjustable through a screen installed on the left arm of the chair, everyone can adapt the ergonomics of the section according to his own sizes. Directly in front of the pilot's seat the MFD and the instructions are gathered in the same field of view to highlight their priority level. The first message displayed asks to turn on the main display using a green button placed at the right of it. To be pressed this jointed cylinder needs a collider parented with a hand to interact with it. The movement of the pressable cylinder triggers another element underneath and this event invoke a method that activates the MFD. All the following actions are completely managed through the touch screen and UI buttons. During the exploration of the home window a warning message informs the users of a problem in the concentration of the oxygen in the breathable mixture, as shown in Fig. 5.3.

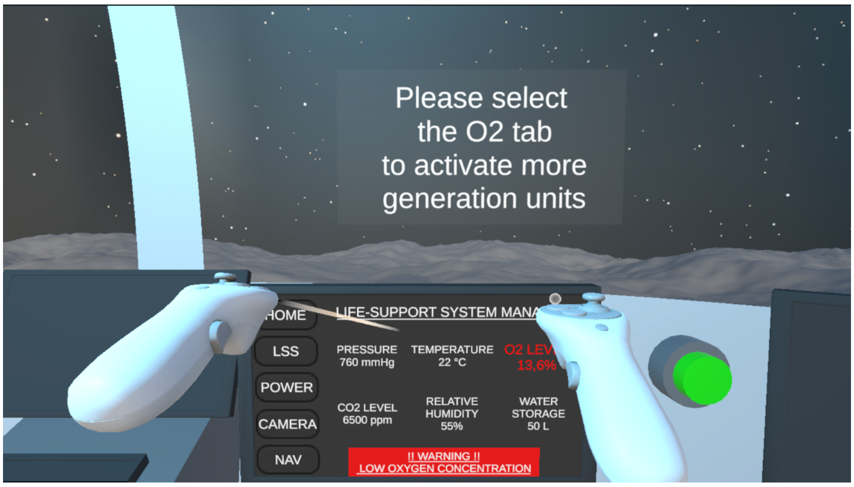

Figure 5.3: Oxygen crisis in the first task

Accessing the detailed page where the oxygen generators status is provided, it turns out that one of the three production units is deactivated and needs to be brought back to its operational mode. With a set of handy switch toggles players can manage the generators and solve this first crisis. The consequence of this additional active element is the power budget turning to negative: the instantaneous power expenditure exceeds the rate of production, this long-term condition is not sustainable, even if there are emergency accumulators. Shortly, as displayed in Fig. 5.4, a new warning message comes out prompting crew members to deploy additional solar panels to increase the rate of power generation. A new window with another set of toggle switches controls the rotation of the solar arrays, activating enough panels increases the budget and makes it positive and sustainable.

Figure 5.4: Power crisis in the first task

5.3.2 Second task

The second task doesn't delve into the subsystems as the previous one did but explores the configuration of the camera views inside a group of screens installed all around the main display. The result displayed in the MFD is a window with icons representing the four screens and the main computer in the middle. Selecting any box except the central one, a dropdown list appears presenting the possible camera views, as pictured in Fig. 5.5. The UI element that shows up is always the same, no matter which screen icon was selected, so the information on which screen was chosen must be preliminarily stored in an array during the On Click event. The choice of a given view involves assigning the material corresponding to the object that reproduces the screen and turning on the latter, result shown in Fig. 5.6.

An additional difficulty given by using a single menu was managing the pointer that saves the information about the choice just made in the list, here is an example of the problem: the pointer is an integer variable initialized to zero, the zero position houses the "No screen selected" string; when users first click on one of the buttons, the dropdown menu that appears rightly shows the "No screen selected" wording. If a camera choice is made, the pointer gets updated with the index of that particular view. The problem occurs when a second icon is selected and the list reappears, unfortunately still showing the information about the previous preference. The solution is to create an additional array initialized to zero that stores in each row, i.e. for each screen, the information about the camera chosen for it; in this way, every time the selection menu opens, it reports the indication of the current view and, if there's none, the "No screen selected" message.

Figure 5.5: Dropdown menu in the second task

Finally, on the menu window there is also a switch to turn on the external headlights relating to the portion of the terrain framed, to ensure a clear vision even in case of low external brightness or areas shadowed due to the volume of the rover itself.

Figure 5.6: Left camera view in the central screen in the second task

5.3.3 Third task

Another very important activity to be conducted in the cockpit before departure is setting up a destination; the Navigation menu window is specifically created to do so. The same dropdown UI element from the previous task appears, this time to select a possible destination. In Figure 5.7 can be seen that each target has its own lunar map with pins to highlight starting and finish points, duration of the trip and distance to be covered.

Figure 5.7: Trip preview window in the third task

A button at the bottom right corner of the screen sends to the proper trip page, displayed in Fig. 5.8, with a view coming from the central camera with arrows superimposed on the ground to guide drivers along the path. Now the movement of the vehicle should take place through the joystick located on the right arm pf the seat. The selection of the element allows users to gain control over the commands and remove the parking brake.

Figure 5.8: Route directions window in the third task

This last event triggers the shadowing of the windscreen to create a transition that skips the real driving part of the mission and that brings the rover and its inhabitants immediately to the desired destination. The arrival at the target is followed by the return of the windshield to transparent, allowing the pilot to appreciate the difference in the landscape compared to the initial one.

5.3.4 Fourth task

The output for the fourth task implementation is no different from the previous ones, with messages prompting users to perform certain actions in order to, in this case, convert the cockpit into a sleeping zone. To complete this duty, three different areas of the control panel need to be used, first the central console, then the physical button and finally the tablet attached to the seat left arm. The blue cylindrical button placed in the central console manages the darkening of the windshield; every pressure switches the state of material transparency. Thanks to the peculiar layout used for the UI instructions, they keep being visible when the screen turns dark, as shown in Fig. 5.9.

Figure 5.9: Darkened windshield view in the fourth task

The most interesting part of the task is the movement of the seat, through the tablet attached to it; three tabs house the main seat control: translation, rotation and light manager. The main camera, representing the view of the player, is parented to the backrest of the chair and moves and rotates together with it.

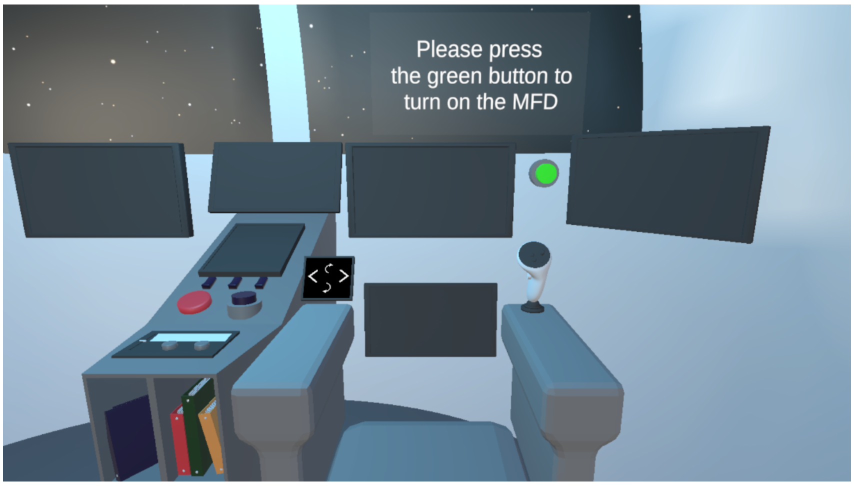

Figure 5.10: Bed configuration in the fourth task

The view users perceive at the end of the rotation is the one in Fig. 5.10. Lastly, by turning off the light, the bed routine is completed, an animation shuts player's eyes and forces him to rest. Don't worry, the night only lasts few seconds and shortly a new day starts full of activities for the crew members.

5.3.5 Fifth task

The beginning of the fifth task from a technical point of view is nothing different from the crises that occurred during the first activity. The difference in the final result is mainly in the level of severity assigned to this emergency, much more serious than the previous situations; consequently it becomes necessary to wear the spacesuit. The suit is made up of various elements: a helmet, a central body, gloves, boots and a backpack. They are all sockets that can be worn in different parts of the body. So far everything seems to be working, in reality the implementation of all this is much more complex than the rest of the project. First of all it must be considered that until now, the player's avatar in the simulation does not have a body, only his hands or the controllers he uses exist. Adding a body makes everything much more complex, because through inverse kinematics the arms should move depending on the position of the hands, something similar for the legs. The simplest element should be the helmet, it does not require joints and bones and should simply partially cover the view of the main camera; this also seems to cause problems due to the high proximity which causes it to disappear due to the limits on the minimum distance of the objects to be perceived by the sensor. As can be seen in Fig. 5.11, the helmet visor is not visible, but the relevant thing is that the values of the vital parameters projected on it appear.

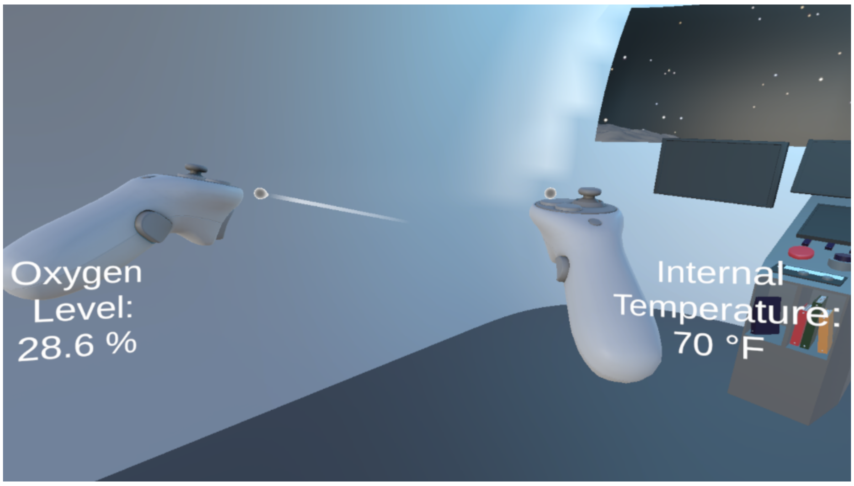

Figure 5.11: View with the helmet and life-status data in the fifth task

Ultimately this fifth task is only half complete: the functional part is present and working properly, but the visual output has gaps that require further productive effort.

5.4 Task mode

The implementation process for the task mode is no different from the story mode, the structure of every single task is the same and the only detail that changes is the extension of the training: in the task mode the experience is limited at one task per time, while in the story mode users can experience the whole sequence of activities. This time the main menu scene gains importance as core selection manager between the different tasks, as shown in Fig. 5.12.

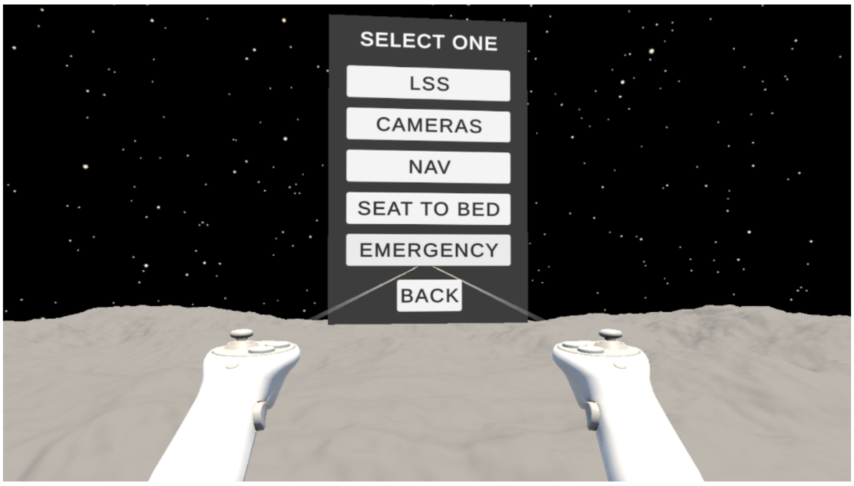

Figure 5.12: Selection panel for the task mode

Chapter 6

Conclusions

6.1 Positive sides

In the first chapters of this Thesis is stated that the main purpose of the entire project is to build a training environment for astronauts. This very ambitious goal has been achieved, as set out in the results section. The final output is an app that can be used on the Meta Quest 2 VR headset; in it, within the main menu scene is possible to select also another kind of story mode experience, different from the cockpit which has been extensively explored in this writing, result of a project focused on the scientific workspace. Users are guided throughout the different game modes, with advices on the actions they have to perform, tutorials on how to use the handheld controllers and the buttons on them and settings pages to manage game settings. In the cockpit, players start understanding the basic interface elements of the Multi-functional display and the windows containing the most relevant subsystems for rover housekeeping.

This project attempts to lay the foundation for an adaptable platform where many additional functional blocks can be added to upgrade the complexity and completeness of the model. Each task also represents a renewed effort to experiment with a particular Unity feature: the sequence of task, defined by a functional point of view, has been developed in order from the first to the fifth, giving the opportunity to use the skills learned in the previous blocks to face the next implementation challenge. Usually speeches that want to be a justification for obvious lack in the project contain a but at a certain point of the text, the following sentence will not deviate from this established tradition. From the game development point of view, what was added to the project could be considered quite simple, but the starting point at the beginning of the thesis must be taken into account.

6.2 Negative sides

Many problems arose throughout the implementation process: some of them were too complex to solve and required alternatives to patch the situation involved, for others the logical motivation was too rooted in the early stages of the projects to change them. The main methodological issue concerns the order in which the various stages were carried out and has already been mentioned in a previous chapter. Many complex elements already created in the 3D modeling phase, after a detailed study in the task planning and VR development, required an additional revision in the 3D software to update them and make them eligible to the latest requirement. In fact, the official Unity Pathways tutorials suggest a different worklogic, with the first phase of VR implementation carried out with primitives that can be created within the editor itself. Once the features managed by the C# scripts have been created and optimized, then the scenes can be enriched with the most detailed assets; possibly, a further step can be taken to add secondary elements that are not involved in the tasks interactions but that return greater fidelity and immersiveness.

6.3 Future developments

The list of tasks has undergone several iteration cycles and has been reduced both in the number and in the assortment of actions included within them. An initial expansion of the work could cover a whole series of activities which for reasons of time have been excluded from the final product; first of all, only two functional blocks of the system have been explored in detail, neglecting three others, one of which is particularly important, the airlock section. Then it must be considered that the simulation built is essentially static, there is no movement of the rover with respect to the lunar surface. An attempt to guide and control the vehicle over Moon's bumps was produced by a member of the Lunar Pressurized Rover research group and could be merged with this simulation to have a more complete and rich platform.

From a purely immersive point of view, the product provides visible and acoustic feedback, one interesting development is the inclusion of haptic response to create the feeling of weight of lifted objects. Currently the controllers used to operate the in-game avatar are able to vibrate at different intensities depending of the type of triggered event. This vibration is often used to highlight the passage of the selector over an object with some peculiar property or degree of interactability; it is also used to mark the non-correctness of an action in progress.

An interesting addition to the app, as it is intended as a training environment, would be a live monitoring system and recording of the user's performance. Saving the time required to complete a single or entire sequence of tasks can be the first step, then moving on to monitoring candidate's heartbeat, breathing and sweating to understand in more detail in which phases he or she encounters the most difficulty. A further level of complexity to add, absolutely outside my expertise, would be an algorithm capable of reorganizing the information provided, perhaps through artificial intelligence, to improve the experience from time to time until the user's mental load is minimized. All credit for this last idea for future development goes to Aleeha Zafar and her research group at Purdue University Northwest, with whom I had interesting conversations on the topic.

To conclude, this project aims to become a digital twin, to do so it will have to integrate many analysis and many other components that right now have not been modeled yet . I don't even know if the software where everything was created are the best for achieving this ultimate goal, but I think it's a great starting point to implement everything you need one piece at a time.

Thank you for having the patience to read this work to the end; if you are one of those people who, as soon as they open a book, go to the very last page to read the ending, you will not find any spoilers about the story here.

Appendix A

Rover design

Pressurized space structures are usually cylindrical for stress distribution reasons, an example of this are the modules of the ISS. Pressurized rover concepts range in the use of one or more cylindrical or spherical elements to compose an articulated and highly posable vehicle. From the very first moment, the idea was to create a system made of a single cylindrical module so that it could accommodate in a single room the scientific laboratory and all other functional blocks.

The first important design parameter is the diameter of such a solid; to choose it, data from some prototypes or vehicles in the conceptual design were collected and several 3D models were created to evaluate what could be the most suitable measurement for the purpose of the mission. A good solution was the choice of a diameter of 3.5 m and a length around 7 m, with a diameter length ratio very close to 2 which seems to be a general rule for this kind of construction.

Another important point is the way in which the two ends of the cylinder are closed. A circular flat face is absolutely unacceptable for the sharp edge it would generate. The two main possibilities are a sort of half sphere grafted onto the final circumference of the solid or a tapering of it. In Figure A.1 the basic face, the hemispherical and the tapered.

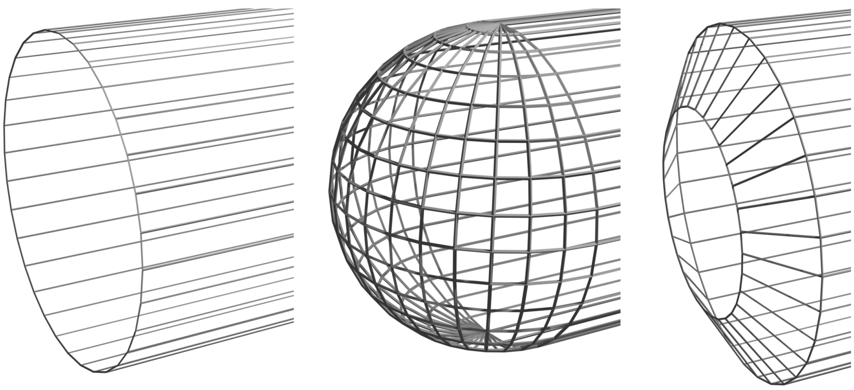

Figure A.1: Different border solutions

The tapered end is much simpler and from a construction point of view and is commonly used for in-orbits modules. For the rover under development, this solution will be used for the rear wall, in which the airlock is located, while for the front end, which houses the cockpit, the hemispherical solution will be chosen. Furthermore at first it was thought to make this wall equipped with many transparent panels, but after some technical reviews, their number was significantly reduced for reasons of safety and feasibility.

Inside the outer shell is a second structure, scaled in diameter to ensure that 20% of the total volume remains between the two layers. This value represents a good estimate of the thickness required by the subsystems in the gap between the two shells. The space between floor and ceiling is 2.10 m, the entire upper and lower space is dedicated to tanks.

Bibliography

[1] L. Mohon and C. E. Williams. *Artemis III: NASA's First Human Mission to the Lunar South Pole*. 2023. URL: `https://www.nasa.gov/missions/artemis/artemis-iii/` (cit. on p. 1).

[2] K. Hambleton and C. E. Williams. *NASA's Artemis IV: Building First Lunar Space Station*. 2024. URL: `https://www.nasa.gov/general/nasas-artemis-iv-building-first-lunar-space-station/` (cit. on p. 1).

[3] P. Milgram, H. Takemura, A. Utsumi, and F. Kishino. "Augmented reality: a class of displays on the reality-virtuality continuum". In: *Telemanipulator and Telepresence Technologies – SPIE Vol. 2351, International Society for Optics and Photonics* (1994) (cit. on p. 2).

[4] M. Chandramouli. "VR, AR, MR, Desktop VR". In: *Proceedings of VR Design and Implementation: Applications, Tools and Techniques*. 2024 (cit. on p. 2).

[5] M. Heilig. "The cinema of the future". In: *Espacios 23-24* (1955) (cit. on p. 2).

[6] J. M. Norman. *The Sensorama: One of the First Functioning Efforts in Virtual Reality*. 2025. URL: `https://www.historyofinformation.com/detail.php?id=2785` (cit. on p. 2).

[7] A. E. M. Casini, P. Maggiore, N. Viola, V. Basso, M. Ferrino, J. Hoffman, and A. Cowley. "Analysis of a Moon outpost for Mars enabling technologies through a Virtual Reality environment". In: *Acta Astronautica 143* (2018) (cit. on pp. 2, 4, 6).

[8] F. Arena, M. Collotta, G. Pau, and F. Termine. "An Overview of Augmented Reality". In: *Computers 11, 28.* (2022) (cit. on p. 2).

[9] H. Planel. *Space and Life: An Introduction to Space Biology and Medicine*. CRC Press, 2004 (cit. on p. 3).

[10] H. C. Gunga. *Human Physiology in Extreme Environments*. Elsevir, 2015 (cit. on p. 3).

[11] J. C. Jr Buckey. *Space Physiology*. Oxford University Press, 2006 (cit. on p. 3).

[12] G. Clément. *Fundamentals of Space Medicine*. Springer, 2011 (cit. on p. 3).

[13] F. Dufresne, T. Nilsson, G. Gorisse, E. Guerra, A. Zenner, O. Christmann, L. Bensch, N. A. Callus, and A. Cowley. “Touching the Moon: Leveraging Passive Haptics, Embodiment and Presence for Operational Assessments in Virtual Reality”. In: *Association for Computing Machinery, International Conference Proceeding Series* (2024) (cit. on p. 3).

[14] N. C. Nilsson, T. Peck, G. Bruder, E. Hodgson, S. Serafin, M. Whitton, E. S. Rosenberg, and F. Steinicke. “15 Years of Research on Redirected Walking in Immersive Virtual Environments”. In: *IEEE Computer Society* (2024) (cit. on p. 3).

[15] R. Sampige, J. Ong, E. Waisberg, N. Zaman, P. Sarker, A. Tavakkoli, and A. G. Lee. “XR-SANS: a multi-modal framework for analyzing visual changes with extended reality (XR) in Spaceflight Associated Neuro-Ocular Syndrome (SANS)”. In: *Eye 38:2680–2685* (2024) (cit. on p. 3).

[16] G. Franchini, B. Tuberga, and M. Chiaberge. “Advancing lunar exploration through virtual reality simulations: a framework for future human missions”. In: *Proceedings of the 2024 International Astronautical Congress, Milan, Italy*. 2024 (cit. on p. 4).

[17] L. Hochberg, J. Mulino, D. Phillips, C. Griffith, A. Gupta, J. Cecil, and A. Lopez-Aramburo. “Exploring the role of Simulation-Based Design principles in support of transportation and material handling activities for the Lunar Mission”. In: *Proceeding of the IEEE International Conference on Systems, Man and Cybernetics (SMC) Bari, Italy*. 2019 (cit. on p. 5).

[18] L. Bensch, T. Nilsson, J. Wulkop, P. de Medeiros, N. D. Herzberger, M. Preutenborbeck, A. Gerndt, F. Flemisch, F. Dufresne, G. Albuquerque, and A. Cowley. “Designing for Human Operations on the Moon: Challenges and Opportunities of Navigational HUD Interfaces”. In: *Association for Computing Machinery, International Conference Proceeding Series*. 2024 (cit. on p. 6).

[19] K. Creel, J. Frampton, D. Honaker, K. McClure, M. Zeinali, M. Bhardwaj, V. Bulsara, D. Kokan, S. Shariff, E. Svarverud, and R. Wirz. “Pressurized Lunar Rover (PLR)”. In: *Proceedings of the 8th Annual Summer NASA/USRA Advanced Design Program*. 1992 (cit. on p. 7).

[20] NASA Communications. *NASA Practices Rover Operations in Simulated Lunar Environment*. 2022. URL: https://www.nasa.gov/humans-in-space/nasa-practices-rover-operations-in-simulated-lunar-environment/ (cit. on p. 7).

[21] Unity Technologies. *Introduction to real-time 3D experience design*. 2025. URL: https://learn.unity.com/pathway/junior-programmer/unit/manage-scene-flow-and-data/tutorial/introduction-to-real-time-3d-experience-design (cit. on p. 10).

[22] M. Bhardwaj, V. Bulsara, D. Kokan, S. Shariff, E. Svarverud, and R. Wirz. *Design of a Pressurized Lunar Rover*. Tech. rep. Department of Aerospace, Ocean Engineering Virginia Polytechnic Institute, and State University, 1992 (cit. on p. 13).

[23] Blender Foundation. *Blender: The Freedom to Create*. 2025. URL: https://www.blender.org/about/ (cit. on p. 15).

[24] Simulation Lab Software L.L.C. *Professional VR Creation & 3D Visualization Software*. 2025. URL: https://www.simlab-soft.com/ (cit. on p. 20).

[25] Blender 4.3 Manual. *Wavefront OBJ*. 2025. URL: https://docs.blender.org/manual/en/4.3/files/import_export/obj.html (cit. on p. 20).

[26] Unity Technologies. *Unity Documentation*. 2025. URL: https://docs.unity.com/en-us (cit. on p. 32).

[27] Microsoft. *A tour of the C# language*. 2025. URL: https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/overview (cit. on p. 32).

[28] Microsoft. *Introduction to .NET*. 2025. URL: https://learn.microsoft.com/en-us/dotnet/core/introduction?WT.mc_id=dotnet-35129-website (cit. on p. 32).

[29] Visual Studio Code. *The open source AI code editor*. 2025. URL: https://code.visualstudio.com/ (cit. on p. 32).

[30] Visual Studio Code. *Working with C#*. 2025. URL: https://code.visualstudio.com/docs/languages/csharp (cit. on p. 32).

[31] Unity Technologies. *Welcome to Unity Learn*. 2025. URL: https://learn.unity.com/ (cit. on p. 32).

[32] GitHub. *The future of building happens together*. 2025. URL: https://github.com/ (cit. on p. 38).

[33] git. *–everything-is-local*. 2025. URL: https://git-scm.com/ (cit. on p. 38).

Dedications

For Esmé - with Love and Squalor