Politecnico di Torino

POLITECNICO DI TORINO

College of Computer Engineering, Cinema and Mechatronics

Master's Degree Programme in Computer Engineering

Master's Degree Thesis

HoloFactory: Design and Layout of Industrial Production Lines in Virtual Reality

Supervisors
Prof. Bartolomeo MONTRUCCHIO
Dr. Antonio Costantino MARCEDDU

Candidate
Federico RASO

ACADEMIC YEAR 2023-2024

Summary

This thesis presents the development and use of "HoloFactory", an application that aims to facilitate the design and layout of industrial production lines through the use of Virtual Reality (VR). This project leverages the immersive capabilities of Head Mounted Displays (HMDs) and handheld Motion Controllers to provide a comprehensive tool for building industrial environments in a virtual space. The project was carried out in collaboration with SEICA S.p.A., a company specialized in the development and production of automatic test equipment for electronic components and electric batteries.

The main focus of this project is to provide an effective and intuitive alternative to traditional CAD tools for planning and layout activities. Through the use of VR, the application aims to reduce the time and technical expertise required to create detailed layouts of production lines, making these activities accessible to a wider range of users within the company and potentially allowing customers to actively participate in customizing the layout of their production line. Particular attention has been paid to connectivity between different systems, with visual indicators highlighting the input and output connections of each module and a wide range of real-time validity checks and tools implemented to assist the user in the build process. Thanks to these features, the application promotes a trial-and-error approach that allows users to experiment freely in the virtual environment, obtain immediate feedback on the validity of the configuration they have designed, and familiarize themselves with the functionalities of the different machines in the SEICA catalog. Another important goal of HoloFactory is to automate the creation of a layout document. This document consists of a top-down representation of the production line or area, highlighting the dimensions and relative distances between the different modules. The application automates this process through an integrated export function that can be used directly in the virtual environment. Thanks to the use of Python scripts for processing the layout data and creating the final PDF document, this process can also be fully customized to meet the requirements of the output file format.

The final part of the thesis analyzes the applicability of Virtual Reality and the tools offered by HoloFactory in each of the three phases of a production line's life cycle. In the pre-sales and commissioning phase, HoloFactory proves to be a useful tool to present SEICA's machine offering in an immersive and detailed way, as a customizable virtual showroom is directly integrated into the application's "Home" environment. In the design and development phase, the application offers the possibility to quickly and intuitively create a prototype of the layout of the line, reducing the time and need for technical staff to analyze requirements and develop an effective solution. The availability of a virtual model of the production line also enables important ergonomic and spatial assessments by simulating the movements and workflows of line personnel and ensuring the visibility of relevant components, such as machine status indicators, from any point on the line. Finally, the virtual twin of the production line installed at the customer's plant can be used in the after-sales and service phase to carry out training simulations for line personnel and as a tool to identify faulty components in the machines and guide technical staff in replacing these components.

Acknowledgements

First of all, I would like to thank SEICA S.p.A. and in particular the Software Research and Development Department for proposing the brilliant idea for this thesis and for providing me unlimited access to the resources required for its completion.
Next, I would like to thank my supervisor Antonio for his assistance and guidance, yet also for the trust shown to me by granting a high degree of independence in the planning and development of the project.

I am also grateful to my family for their unconditional support during my studies and, above all, for the excellent education provided to me since childhood, leading up to the achievement of this milestone.
Finally, a special thanks goes to my girlfriend Sara, for always lovingly standing by my side, encouraging me in challenging times and sharing my happiness during blissful moments of my career.

Contents

List of Figures V

List of Tables VI

Acronyms VII

1 Introduction 1

1.1 Motivations . . . 1

1.2 Background . . . 2

1.3 Advantages of Virtual Reality . . . 3

1.4 Thesis Objective . . . 5

1.5 Thesis Structure . . . 5

2 Technologies 6

2.1 Unreal Engine 5 . . . 6

2.2 Blender . . . 7

2.3 Python . . . 8

2.4 NSIS . . . 8

2.5 HTC VIVE . . . 9

3 Overview of the Application 11

3.1 Preparation and First Startup . . . 11

3.2 Application Manual . . . 12

3.2.1 Home . . . 12

3.2.2 Navigation . . . 12

3.2.3 Main Menu . . . 13

3.2.4 Project Area . . . 14

3.2.5 User Menu . . . 15

3.2.6 Buildings Catalog . . . 15

3.2.7 Build Mode . . . 16

3.2.8 Additional Settings . . . 20

3.2.9 Export . . . 23

3.2.10 Save Project . . . 25

3.3 Developer Manual 27
3.3.1 Adding Buildable Modules 27
3.3.2 Converting CAD Models 30
3.3.3 Export Customization 32
3.3.4 Sharing Project Save Files 34
3.3.5 Desktop Version 35

4 Conclusions 38
4.1 Outcomes and Use-Case 38
4.1.1 Phase 1: Pre-Sales 38
4.1.2 Phase 2: Design and Development 40
4.1.3 Phase 3: Maintenance and Support 42
4.2 Future Works 44

Bibliography 46

List of Figures

2.1 HTC VIVE Motion Controllers' command scheme. 10

3.1 Several test systems on display within the showroom area. 12
3.2 An example of valid and invalid teleport locations. 13
3.3 The main menu screen and interaction system. 14
3.4 The user menu window and interaction system. 15
3.5 The catalog page with a floating 3D preview of the selected system. 16
3.6 The "Pilot V8 Next" test system being previewed in build mode. 17
3.7 A previewed module overlapping an already built system. 19
3.8 Two modules snapping together through a pair of compatible sockets. 20
3.9 The "Settings" page inside the user menu. 21
3.10 User in build mode, with several optional settings enabled. 23
3.11 User selecting the area to be exported. 24
3.12 An example of generated layout document for a small production line. 26
3.13 An example content of a JSON file for the addition of a custom module. 29
3.14 An example content of the "Modules" folder. 30
3.15 The top-level global variables found inside the `layout.py` script. 33
3.16 A code snippet of the `main()` function of the `layout.py` script. 34
3.17 An example of JSON data received by the `layout.py` script. 35

List of Tables

3.1 Table of actions and key bindings. 36

Acronyms

Symbols

2D two-dimensional 4, 22, 39
3D three-dimensional 1, 3–9, 11–16, 18, 21–24, 27–33, 41, 43–45
6DOF six degrees-of-freedom 12

A

AGV Automated Guided Vehicle 4
AMOLED Active-Matrix Organic Light-Emitting Diode 9
API Application Programming Interface 7
AR Augmented Reality 7

B

BT Battery Tester 3

C

CAD Computer Aided Design 4, 5, 7, 30, 31, 40, 41, 43
CMS Complex Manufacturing System 4, 5, 38

E

EV Electric Vehicle 2, 3, 18, 28

F

FBX Filmbox 7, 8, 28, 30–32, 41
FPS Frames Per Second 1, 36, 43

G

GLB Graphics Library Binary transmission format 8, 30, 31
GLTF Graphics Library Transmission Format 7, 8, 30, 31
GPU Graphics Processing Unit 1, 9
GUI Graphical User Interface 13

H

HDMI High-Definition Multimedia Interface 9
HMD Head-Mounted Display 1, 4, 5, 7, 12, 13, 36, 37, 43
HUD Heads-Up Display 22

J

JSON JavaScript Object Notation 25, 27–29, 32–34, 41

L

LOD Level Of Detail 45

N

NSIS Nullsoft Scriptable Install System 8

P

PC Personal Computer 9
PCB Printed Circuit Board 2, 3, 16, 18, 19, 28, 44
PDF Portable Document Format 8, 23, 25, 34, 40
PIL Python Imaging Library 8

R

R&D Research and Development 8
RT Rotary Table 2

S

SL Stream Line 2, 44
STEP Standard for the Exchange of Product data 7, 30, 31, 41
STL Standard Tessellation Language or Stereolithography 30, 31

U

UI User Interface 22
UX User Experience 22

V

VR Virtual Reality 1, 3–7, 9, 11, 13, 18, 27, 30, 35–45

X

XR Extended Reality 7

Chapter 1

Introduction

1.1 Motivations

Virtual Reality (VR) is a simulated experience that uses three-dimensional (3D) near-eye displays and pose tracking to give the user an immersive feeling of a virtual world, replicating an environment, real or imaginary, in a three-dimensional space. Users interact with this environment using special electronic equipment, such as a Head-Mounted Display (HMD) with a screen in front of the eyes and often additional devices, such as motion controllers equipped with sensors, to enable tracking of hands and gestures.

Significant technological advances in recent years have enabled the development of high pixel-density screens and increased Graphics Processing Unit (GPU) performance to create VR headsets capable of reproducing images of unprecedented quality while maintaining a refresh rate high enough not to cause motion sickness in the user[1]. By overcoming the technological limitations that have most hindered its spread in the past, Virtual Reality has experienced a remarkable and constant increase in its diffusion in recent years, also thanks to increased public awareness and benefits of its usage during the COVID-19 pandemic. As a result, market pressure is also growing for the development of new ways to make the most of these technologies, which have finally reached a stage of maturity.

Although Virtual Reality has long been established as an entertainment and gaming method, its use as a productivity tool in the workplace is still under development. Research in this area has been very active over the past decade and has highlighted a variety of potential benefits of introducing this technology in areas such as industry where it is still struggling to gain traction. Although the potential of introducing Virtual Reality into development sectors is now well recognized, there is still a significant gap in the number of practical implementations of these solutions.

It is against this background that the motivation for this thesis arose, which aims to realize a practical implementation of an application that uses Virtual Reality technology and demonstrates the benefits of its adoption in an industrial environment. The aim of this work, carried out in collaboration with the company SEICA S.p.A., which provided the detailed specifications for the realization of the application, is to present an experimental but functional product, paying particular attention to the future possibilities of expanding and implementing additional functionalities. The application will leverage the qualities of Virtual Reality to provide an intuitive, fast, and easy-to-use tool for the planning and layout of industrial production lines and factory environments.

[1] The minimum refresh rate to avoid the impression of sluggishness on a classic monitor is usually 30 Frames Per Second (FPS), while a minimum refresh rate of 90 FPS is recommended to avoid motion sickness phenomena while using an HMD device. Considering that the standard resolution of a Full HD monitor is 1920 x 1080 pixels, while in a modern VR headset there are two displays (one for each eye) with resolutions ranging from 1920 x 1080 to 3660 x 3200 pixels, the load on the GPU increases by 6 to 12 times to render the same content in VR compared to a classic monitor.

1.2 Background

The thesis is carried out in collaboration with SEICA S.p.A. [1], a company based in Strambino (TO). SEICA's core business is the development and production of industrial machines for testing electronic Printed Circuit Boards (PCBs) and Electric Vehicle (EV) batteries. These machines are usually able to work automatically without the need for operator intervention, except for switching on and off and maintenance. For this reason, they are widely used in factories that manufacture electronic components for sectors such as automotive, medical, military, and consumer electronics. Testing the quality of electronic components at the end of their assembly enables customers who purchase SEICA systems to guarantee the quality and functionality of their products and to detect any design or assembly faults before they leave the factory where they were manufactured.

The catalog of test systems offered by SEICA, which is constantly being expanded to better meet the requirements of the market, now comprises more than 40 different models. These are divided into different product series, which differ in terms of performance and functionality. The most important of these are:

- **Pilot** line: These systems specialize in testing and performing electrical measurements on components built into PCBs. The main feature of these systems is the use of special probes (needles) to perform measurements. These probes are mounted on moving mechanical axes so that they can touch any point within the test area of the machine in a programmable and dynamic way. Thanks to this feature, the systems in this line are characterized by a high degree of flexibility in terms of the variety of electronic assemblies that they can perform test on. The various models in this line differ from each other in features such as the size of the test area inside the machine (and therefore the maximum size of the Printed Circuit Boards that can be tested), the number of mobile probes available (usually 4, 6, or 8), and the orientation in which the panels are transported and tested in the machine. In particular, the models with the prefix **H** in their name transport the PCBs horizontally, while the models with the prefix **V** transport them vertically. All systems in this line also have two input/output connections, one on the right and one on the left side of the system, through which the PCBs are loaded and unloaded after testing. Both connectors can be configured as inputs and/or outputs. This means that Pilot systems can work either in right-hand loading and left-hand unloading mode (R2L) or vice versa (L2R), but also with loading and unloading from the same side (R2R or L2L).

- **Compact** line: These are also systems that specialize in testing electronic Printed Circuit Boards. In contrast to the Pilot line systems, they use a press system on which fixtures are mounted that can accommodate dozens, hundreds, or even thousands of needles. Once the PCB is in the test area, the press is lowered onto it until the needles come into contact with the electronic components to be tested. In this way, it is possible to achieve a high degree of parallelism in the measurements, which significantly reduces the time required to test the entire PCB, but at the expense of the flexibility with which different products can be tested (as a specialized fixture is required for each product). With these systems, the PCBs are only tested in a horizontal orientation. The main difference between the various models in this line is the way in which they can be integrated within a line. In particular, the Stream Line (SL) models feature two configurable input/output connections (exactly as described for the Pilot line systems), while the Rotary Table (RT) models are not compatible with a conveyor belt transport system, but are designed for loading and unloading with a mechanical robotic arm that can place the PCB to be tested in a dedicated bay and remove it from the same position once the test is complete.

- **Rapid** line: These systems specialize in testing bare electronic boards, i.e., boards on which no components have yet been mounted (commonly referred to as *bare boards*). These systems validate the quality of the traces inside the Printed Circuit Boards, even for multi-layer PCBs. Functionally, these systems are very similar to the Pilot line. They are characterized by two configurable input/output connections and the presence of horizontal (**H**) or vertical (**V**) transport variants. They are not usually found in production lines in combination with models from the Pilot and Compact families, as they are aimed at testing PCBs at a different stage of the assembly production process.

- **BT** line: The systems in the Battery Tester (BT) line specialize in testing battery modules for the production of electric vehicles. They are also functionally similar to the Pilot systems, as they consist of two configurable input/output connections. These systems are not compatible with those seen so far, as they are designed to test a completely different product, which is much larger and heavier and is normally transported on metal pallets within a line, requiring a completely different transport infrastructure to that used for the thin and light PCBs.

As most test systems are designed for autonomous operation within a production line, SEICA offers, through its subsidiary SEICA Automation [2], a wide range of industrial modules for the transport and handling of the products to be tested. These modules are designed to be connected both to the systems manufactured by SEICA and to third-party machines. They can be used sequentially to control product transport through the various processing stages within a line. The automation modules offered by SEICA Automation consist of a large number of variants, but can be categorized into the following macro-categories according to their main function:

- PCB **Loaders**: These are the modules that form the beginning of a production line. The Printed Circuit Boards are usually stacked vertically and transported from one line to another in special containers known as *racks*. The purpose of these modules is, therefore, to receive these racks, remove the PCBs individually, and load them into the production line. The full racks are placed by an operator on a conveyor belt at the beginning of the module, in a queue that varies in length depending on the model. Once they are completely empty, they are returned via an output conveyor that runs parallel to the input conveyor belt.
- PCB **Unloaders**: Modules that have the opposite function to the loaders. They are the last module of a line and are responsible for collecting the processed PCBs and placing them inside the racks (which are loaded empty by the operators). Depending on the model, they can store a different number of racks and allocate the incoming cards to the different racks according to the results of the tests or processing carried out in the previous stations.
- **Conveyors**: These are the modules that make up a large part of the line and whose main function is transporting products from one station to the next. They can be made up of different types of conveyor belts depending on the type of product they need to transport (PCBs, bare boards or pallets with EV batteries).
- **Tilters**: special modules for turning Printed Circuit Boards from a horizontal to a vertical position and/or vice versa. They are indispensable for including systems that process PCBs in different positions within the same production line.
- **Robotic arms** and **cobots**: These specialized devices consist of a robotic arm with multiple pivot points that can perform programmable 360-degree movements. They are able to work together with other transport modules such as conveyor belts and with any type of workstation, picking up components from a line and placing them in a specific system.

In view of the large number of different models that SEICA offers and the fact that customers are increasingly not limiting themselves to purchasing individual systems, but require complete solutions consisting of entire lines or segments made up of several modules, the idea was born to develop an application that allows both the access to the complete catalog of the SEICA range and the creation of a layout for a production line from the modules and systems it contains, which can be adapted to suit different needs.

1.3 Advantages of Virtual Reality

Virtual Reality represents an innovative and promising tool for tackling the task of layout planning and management in industrial environments. During the development phase of an industrial assembly line, some of the complex three-dimensional issues - such as optimizing space, maintaining an appropriate distance between different subsystems, enabling proper interaction between different modules, and ensuring the human-machine ergonomics and safety (e.g. body posture, reach, visibility) - may be difficult to assess using conventional two-dimensional (2D) interfaces and traditional Computer Aided Design (CAD) tools. As a result, it is very common for these design issues to arise only during the final implementation of the Complex Manufacturing System (CMS), resulting in high costs and time spent on re-design [3]. To overcome these problems, the commissioning of CMSs is recently leveraging on the benefits of the novel concept of virtual commissioning [4]. This paradigm enables the validation and verification of a CMS through the virtual visualization and testing of the system and associated components, thereby reducing time to market, lowering costs and increasing productivity. Recently, several research papers have explored the use of VR in virtual commissioning in order to facilitate integration and assembly planning as well as testing and troubleshooting steps [5].

Focusing the analysis on the scope of this project and considering the detailed specifications provided by SEICA, several key benefits can be identified that arise from the adoption of VR as a means of implementing the proposed application.

- **Immersiveness**: Virtual Reality through the use of an HMD device represents one of the most immersive and realistic options available today for visualization and navigation in 3D scenarios. By tracking the position and rotation of the visor in space, the user is able to move freely within the virtual environment, explore the proposed solutions from all angles and obtain the most faithful representation of what the production environment could look like once realized.
- **Intuitive controls**: The HMD device enables the use of intuitive and natural controls to interact with the virtual environment. Thanks to position and orientation tracking, it is possible for the user to move around the virtual environment by simply walking inside the designated area and changing their view angle by naturally turning and tilting their head. Many headsets (including the model chosen for this project, the HTC VIVE) also support hand tracking through the use of specialized controllers, allowing interactions with virtual objects to be simulated.
- **Prototyping speed**: CAD tools commonly used for the design and layout of CMS usually require dedicated hardware and long processing times due to the complexity of the 3D models involved and the precision with which they are represented. By using a specialized application and a different format for displaying 3D models, it is possible to reduce not only the minimum system specifications but also the time required to create a prototype layout from a catalog of modules.
- **Space evaluation**: When designing the layout of a production area, consideration of space is crucial. An accurate assessment of the spatial requirements of the machines and their mutual distances and operating tolerances must be included in the layout document, which serves as the initial blueprint for the development of a production line (and later for its arrangement and installation). Other important spatial criteria, which are often overlooked until the first stages of installation at the customer's site, include the available routes for the transit of personnel, carriers, and Automated Guided Vehicles (AGVs) for material transport between the various lines, as well as the accessibility of emergency releases and the visibility of machine status indicators from all sides of the line. Thanks to the ability to navigate through the scene in a first-person perspective, the early inspection of a virtual representation of a production line through the use of Virtual Reality enables a more thorough assessment of these parameters than when using conventional 2D design techniques.
- **Ergonomic evaluations**: Many of the industrial systems manufactured by SEICA operate autonomously, but operators and technicians can monitor and control them for programming and maintenance work via the PCs on board the systems, which they can usually interact with using a touchscreen monitor or peripherals such as mouse and keyboard. It is, therefore, important to evaluate the ergonomics of the positioning of these components in order to take into account any structural changes to the design before the physical realization phase of the systems. Virtual Reality makes it possible to immersively test the interactions with the different systems and simulate first-hand the actions that can be performed by the operators on the line, allowing earlier detection and less costly correction of design flaws.

- **Availability of VR devices**: Another crucial, albeit less technical, aspect that strongly encouraged the use of Virtual Reality in this project is the fact that Virtual Reality-enabled devices (especially HMD headsets) are already readily available in many companies in the industrial manufacturing sector, but are still underutilized. This is mainly due to the widespread of Virtual Reality usage for marketing purposes during the COVID-19 pandemic. As it was not possible to attend annual exhibitions and congresses, many companies acquired HMD devices as a means of participating in virtual showcases and meetings. The use of these devices quickly declined in the following years due to the restoration of usual live events.

1.4 Thesis Objective

The objective of this work is, therefore, the development of a functional prototype of a Virtual Reality application that can be used with the HTC VIVE headset in combination with the two motion controllers supplied for hand tracking. Within the application, the user can access a complete catalog of the machines offered by SEICA and position the systems of interest in a virtual 3D environment to plan the layout of a production area. The application also automates and simplifies the process of creating a layout document (a document containing a top-down schematic representation of a line on which the dimensions of the different modules are indicated, together with the various distances between them), which is currently manually crafted by designers specializing in CAD modeling, in order to make the activity of prototyping the layout of a production line accessible to a wider range of users inside and outside the company. For this reason, particular attention is paid to the simplicity and intuitiveness of the controls through which the application is used: as the aim is to make the application usable by the widest possible audience, it is important that the commands are easy to understand and do not require in-depth technical knowledge of the device.

1.5 Thesis Structure

This thesis consists of four chapters. In addition to this introductory chapter, the following chapters are organized as follows: **Chapter 2** provides an overview of the technologies used to develop and utilize the HoloFactory application. This includes a description of the various software components (Unreal Engine 5, Blender, Python) and how they were used in the development phase of the application. There is also a detailed description of how the hardware system of the HMD headset used for the project, HTC VIVE, works. **Chapter 3** examines how the application works by providing an overview of all the functionalities implemented in it, divided into sections and organized according to the typical workflow of the application. This chapter also describes the more advanced functionalities, suitable for experienced users or developers who have the technical knowledge required to make changes to the functioning of the application itself (see section 3.3). **Chapter 4** not only draws the conclusions from the work carried out, but also analyzes in depth the benefits and opportunities offered by the use of Virtual Reality, in particular by using the tools provided by the HoloFactory application, within the different phases of the commissioning and development cycle of a production line or, more generally, of any type of Complex Manufacturing System. It also gives a brief overview of possible improvements and future developments.

Chapter 2

Technologies

This chapter provides an overview of the complete technological stack used for the development of HoloFactory. Here the main adopted software tools and programming languages are listed, giving insights about their key features and reasons for their choice.

2.1 Unreal Engine 5

The main component for the creation of a Virtual Reality application is the 3D graphical engine (also often referred to as game engine). The game engine represents the foundation on top of which the virtual experience is built, consisting of a framework that handles the common needs of an interactive 3D application - such as rendering models and scenes, handling different input and output devices, and running the core logic of the application loop. The chosen 3D engine for the development of HoloFactory is Unreal Engine 5 [6].

Unreal Engine is one of the two most popular and freely available game engines (the other being the Unity game engine). Unreal Engine heavily focuses on delivering high-quality graphics out of the box, implementing several cutting-edge technologies for optimizing visual performance, such as Lumen and Nanite. This is of critical importance when it comes to the realization of realistic and immersive Virtual Reality projects such as the one covered in this thesis, enabling the creation of a more faithful and convincing simulation. Unreal's main programming language - through which developers can implement their application-specific logic - is C++, but the engine also offers a practical and powerful alternative for faster and more intuitive development of features through the use of the Blueprint visual script language. The Blueprint language is completely integrated inside the Unreal Engine Editor (the main application used by developers to design projects using Unreal Engine) and allows for the integration of application logic by visually chaining together, in drag and drop fashion, functional blocks consisting in a high-level representation of C++ methods, inside the dedicated editor. This feature proves extremely useful for the fast prototyping of features, as well as for allowing developers to modify the application without the need for writing C++ code.

Unreal Engine 5 offers several integrated plugins and tools which have been extensively used to improve various aspects of the application, while also allowing the addition of third-parties plugins. Details about the integration of the most impactful of them are provided in the following sections.

Enhanced Input

The Enhanced Input system is a plugin included in Unreal Engine [7]. It is used to define input commands to be used by the user to interact with the application. This plugin enables decoupling input key bindings from the performed in-game action by defining two different entities inside the project. The Input Action objects are used to define actions that can be performed by the user, for example walking, turning, interacting, or building an object; they define the behavior of the action, allowing the developer to implement the logic to be executed once they are triggered. On the other hand, the Input Mapping Context is an asset that groups several Input Action together and assigns them to the input binding of choice. Several Input Mapping Contexts can be defined inside the project and dynamically switched at runtime in order to support different input methods.

Thanks to the integration of this system in HoloFactory, it has been possible to define two different Input Mapping Context: the first is used to bind the underlying Input Actions to the buttons and gestures supported by the VR headset and motion controllers, while the second maps the same actions to mouse and keyboard buttons. This way it has been possible to provide two different versions of the application - one that runs on HMD devices and the other as a standard desktop application - without the need for duplicating the code responsible for handling the different input devices.

RuntimeStaticMeshImporter

The RuntimeStaticMeshImporter is a third-party plugin that can be installed in the Unreal Engine Editor [8]. It provides several blueprint components that enable the dynamic loading of 3D models at runtime from Filmbox (FBX) files. FBX is a widely diffused file format for storing 3D meshes, materials, and animations, and is one of the preferred formats for working with Unreal Engine. Thanks to the implementation of this plugin, users have the possibility of importing custom 3D models to be added to HoloFactory's virtual catalog of buildable modules. More information about the usage of this feature can be found in the section 3.3.1.

OpenXR

The OpenXR plugin is an add-on provided by Unreal Engine for integrating the homonymous OpenXR Application Programming Interface (API). OpenXR is a royalty-free, open standard that provides a common set of APIs for developing Extended Reality (XR) applications that run across a wide range of Augmented Reality (AR) and VR devices [9]. By adopting this standard, developers can ensure that the crafted Virtual Reality application is compatible with the wide range of devices conformant to this standard - which at present include all the major VR device manufacturers. The Unreal Engine integration of OpenXR offers a variety of blueprint functions for interacting with the specific functionalities offered by the HMD hardware and for gathering the available inputs from the related accessories and tracking devices.

2.2 Blender

Blender is a popular open-source 3D modeling software featuring a suite of tools for 3D model editing, texturing, animation, rendering, and more [10]. In the scope of this project, it has been adopted as the main tool for the revision, modification, and eventual optimization of 3D meshes exported from the CAD files provided by the company before importing them into Unreal Engine. The usage of an external 3D modeling tool has been deemed necessary as the Unreal Engine Editor is not suited for performing heavy operations on complex 3D meshes - as the ones generated by the conversion of CAD project files.

In particular, during the development of the application, several CAD files had to be gathered from the company's source control repository in order to access the 3D models of the various systems developed and built by SEICA. CAD files are usually saved using the Standard for the Exchange of Product data (STEP) format, a popular standard for describing mechanical components [11]. This file format is not compatible with graphical real-time rendering engines and, therefore, not directly portable to Unreal Engine. For this reason, the free and open-source CAD modeler FreeCAD [12] was used during the early stages of this work, for converting STEP files provided by the company into a more suitable format, the Graphics Library Transmission Format (GLTF). Since the meshes generated from CAD file conversions are often very detailed and complex, the exported files are then loaded in Blender for an analysis of the complexity. If the 3D models are too heavy for real-time rendering on the target workstation, they can hereby be simplified by firstly removing any hidden component that may not be relevant for the usage of the model inside the HoloFactory application. After this step, several modifiers can be applied to the remaining geometry in order to reduce the number of vertex and polygons.

Once the simplification operations have been completed, it is possible to use Blender to assign materials to the various components of the resulting mesh. It is worth mentioning, though, that not every characteristic of Blender's material can be transferred to Unreal Engine due to the internally different implementation of their rendering engines. For this reason, a custom Editor Utility has been developed in Unreal Engine, that allows automatically assigning materials from the Unreal Engine library to the different vertex groups of an imported mesh by matching the names of the exported materials. This way it is possible to create a library of materials in Unreal Engine, and use placeholder materials with the same name in Blender for assigning different materials to the appropriate vertex group of the mesh. After this step has been completed, it is possible to export the 3D model in a format compatible with Unreal Engine, the most popular ones being FBX, GLTF and the Graphics Library Binary transmission format (GLB).

2.3 Python

Python is an interpreted high-level programming language ranking as one of the most popular programming languages in the world [13] thanks to its ease of learning, simplicity and versatility. Its interoperability with other languages and the availability of a wide range of libraries and frameworks make it a great tool for tackling a variety of development tasks.

The Python language is mainly used among the SEICA Software Research and Development (R&D) team as a way to offer customizable plugins and hooks for the main application responsible for the management of the test systems. By having the main application - which is written in C++ and therefore inherently not easily customizable - launch several predefined Python scripts during its execution cycle, an high level of personalization and flexibility is achieved without the need to rebuild the full project. This gives users and customers the possibility to adapt the application workflow to meet their specific needs, while not having to modify the common source code. Thanks to its interpreted nature the Python language is also well suited for fast prototyping and testing of new features, and the majority of the script files called by the main application can be modified and reloaded even without the need of restarting the program.

This methodology has also been adopted in the development of HoloFactory, specifically leveraging the flexibility of Python for accomplishing a complex task such as automating the generation of a layout document in Portable Document Format (PDF), in a scriptable and fully flexible way. Using this approach, advanced users and programmers are able to completely tailor the way the export function works based on their needs, and also utilize the information gathered by the application and sent to the Python script for any other kind of analysis or elaboration. More details on how the export function works and how it may be customized can be found in sections 3.2.9 and 3.3.3 respectively.

One of the main strength of the Python programming language is the availability and simple installation of packages. Several of them have been used inside the customizable Python script for the generation of the layout document; the main one is ReportLab [14], a library that allows procedural generation of PDF files from code, allowing organization of the entire content of the document, as well as the addition of images and tables. Another notable package, largely used in the project, is Pillow [15] - an active fork of the no longer maintained Python Imaging Library (PIL) - which provides methods for opening, manipulating, and saving images in several file formats.

2.4 NSIS

Nullsoft Scriptable Install System (NSIS) is used for the creation of an automatic installer for the deployment of HoloFactory. NSIS is a professional open source system for creating Windows installers [16]. It features a script-based language that allows the user to create custom logic for handling the installation of their products. This feature has been leveraged, as an example, in the HoloFactory installer, for automating the installation of a suitable version of the Python runtime on the target system (if not already available).

2.5 HTC VIVE

The Virtual Reality system adopted for the fruition of this project is HTC VIVE [17]. HTC VIVE is a solution that supports room-scale VR, i.e., the use of Virtual Reality while sitting, standing, or moving within the virtual environment by means of physically walking within an area of up to 15 square meters. This is also a tethered VR device, meaning that it does not host an onboard GPU, but is connected via a wired HDMI cable to a Personal Computer (PC), which renders and provides the images to be shown with each frame. This feature partially limits the user's freedom of movement in favor of the ability to reproduce high-quality images, as the performance of an on-board processor does not limit the rendering capabilities. The tethered model also allows unlimited use, not constrained by the battery life of the device.

The HTC VIVE hardware system is made up of three main components:

- **Headset**: this is the main part of the Virtual Reality system. It consists of two AMOLED screens of 3.6 inches in size, positioned behind a couple of lenses inside a visor. Both the distance of the lenses from the user's eyes as well as the distance between one lens and the other, can be adjusted using two different wheels. It also features adjustable straps and soft foam inserts to best suit a wide range of users, while preventing the cable attachments from excessive movements. The two underlying screens have a resolution of 1080 x 1200 pixels per eye, covering a field of view of 110 degrees. The refresh rate of this device is 90 Hz, which should be the minimum frame rate delivered by the connected workstation in order for the user to perceive a smooth experience. A front camera is also mounted on this device, which is tipically used for enabling a pass-through view of the surrounding environment when the user reaches the borders of the defined safe area;
- **Motion controllers**: these two handled controllers are designed to enable tracking of the position and rotation of the user's hands. They also provide a set of buttons and triggers which can be activated performing common and intuitive gestures (the complete list can be seen in Figure 2.1). As an example, the Trigger button (number 7) can be easily pulled with the index finger in a fairly ergonomic way; the so-called Grip button (number 8) is replicated on both sides of the controller and, as the name suggests, is meant to be pressed by tightening the grip on the controller. The Trackpad (number 9) can track the thumb's position when touched, and therefore be used to simulate vertical and horizontal scrolling; if it is instead pressed down, it can act as four different directional buttons (as shown with numbers 2, 3, 4 and 5). These controllers are wireless and also feature realistic haptic feedback;
- **Base stations**: two base stations are provided within the HTC VIVE kit. These are responsible for the real-time tracking of the position and rotation of both the headset and motion controllers inside the three-dimensional space. They only need a power connection, as they are wirelessly synced with each other and with the headset, which collects the gathered spatial data and transfers it to the connected PC. Base stations enable tracking of the user inside an area of up to 15 square meters (approximately 3.5 m x 3.5 m).

Apart from the hardware components, a suite of software tools must be installed in order to use the Virtual Reality device on a workstation. By following the instructions that come with the headset, software as VIVEPORT and SteamVR can be installed, both of which allows access to a variety of VR exclusive content. SteamVR in particular, works as an hub for entering the VR experience, offering a comprehensive library of VR applications and supporting different device models inside a common environment.

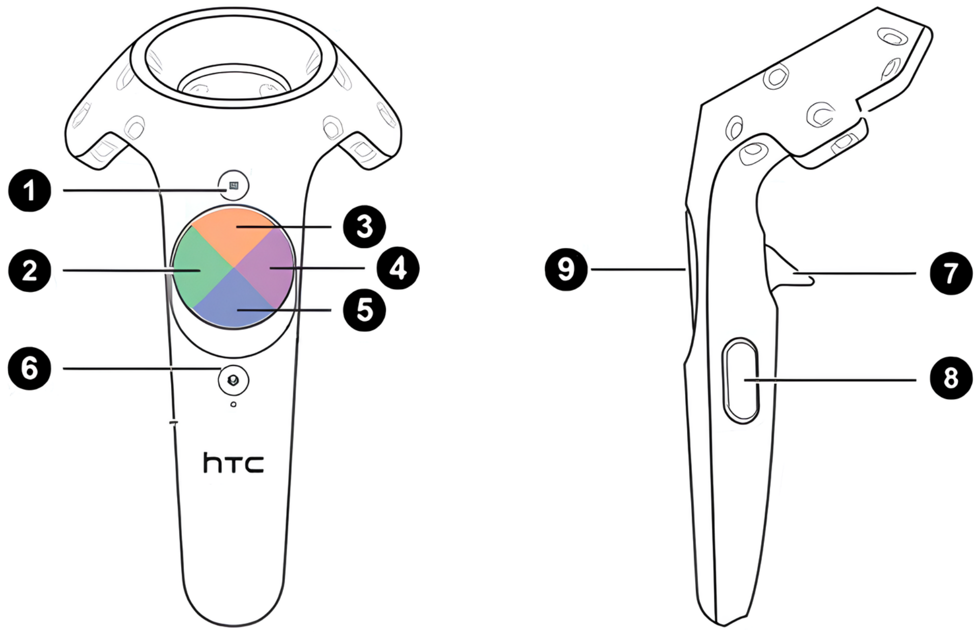


Figure 2.1: HTC VIVE Motion Controllers' command scheme.

Chapter 3

Overview of the Application

This chapter aims to provide an in-depth description of all the functionalities offered by the developed application, ranging from initial installation and configuration to a thorough description of the more technical tools available to advanced users.

3.1 Preparation and First Startup

Before starting the HoloFactory application for the first time, the HTC VIVE headset must be correctly configured and set up. This step includes the installation of the SteamVR software, an application that guides the user through the configuration of the Virtual Reality device and from which the library of installed VR applications can then be accessed.

The HTC VIVE package includes a set of two detectors that enable tracking the user's position via the headset and the motion controllers (to track the position and rotation of the hands). To use them, it is necessary to define a working area in which the user is able to move safely. The area must be roughly square or rectangular in shape, with side lengths of up to 5 meters each. The two detectors must be positioned along the perimeter of this area at two opposite corners and aligned facing towards the center. Once the physical setup is complete, it is possible to launch SteamVR to continue with the software configuration. Here the user is first guided through the calibration of the orientation sensors in both the headset and the two handheld motion controllers. After that, a more in-depth definition of the working area must be performed by virtually drawing its perimeter using one of the motion controllers.

Once the configuration of the workspace is complete, the user is given access to the SteamVR Home, a small virtual 3D environment where the library of recently used VR applications can be browsed. For the first time, the HoloFactory must be launched by directly opening its executable file; after that it should automatically be registered in the SteamVR library, allowing users to also launch it from there. In order to facilitate the deployment of the application together with all the required dependencies, a convenient automatic installer has been developed. This should be launched after the configuration of the VR device has been performed on the target workstation, and is responsible for installing a suitable version of the Python runtime as well as deploying all the files needed for the application to run. The default installation folder for the HoloFactory application is

```
C:\Program Files\HoloFactory
```

Here the main executable for the application - `HoloFactory.exe` - can be found, and a shortcut to it can be optionally created on the user's desktop.

3.2 Application Manual

This section delves into the standard use of the application in order to demonstrate the various features offered by HoloFactory. These functionalities are presented in such an order as to follow the steps performed by a user during a typical application usage session, starting from the first access to the main menu, up to the final export of the layout document for a production line. This section guides the user in accomplishing the core tasks within the application, such as moving around and interacting with the 3D environment, creating and saving a project, building virtual modules, and exporting the resulting layout.

3.2.1 Home

When starting the application, the user lands in a 3D scene that serves as the application's main menu.

This area is designed to simulate a virtual showroom: different machines and automation modules have been arranged in stands within the available space and each of them has been provided with a description of their properties and features so that the user can move freely between them to get an overview of the buildable systems available in their catalog. This is particularly useful given the application's intent to target a wide range of users. By navigating through this showroom area, users can familiarize themselves with the catalog of modules offered by SEICA. This makes the application suitable both as a learning tool for new employees and as a presentation environment for showcasing solutions to customers or visitors in a commercial context and at industrial trade fairs.

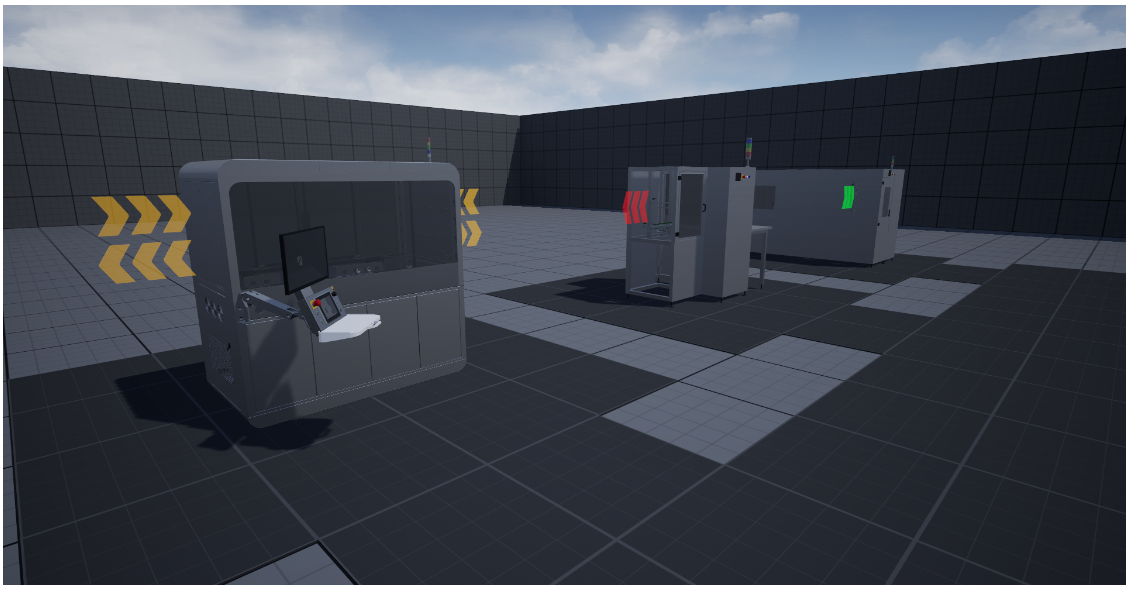


Figure 3.1: Several test systems on display within the showroom area.

3.2.2 Navigation

The user can move around the virtual environment by moving within the defined working area of the HMD device. Thanks to the six degrees-of-freedom (6DOF) tracking supported by the headset, the position and orientation of the user's head in space are faithfully reproduced in the virtual scene.

Teleport

As the actual working area available is limited in relation to the size of the virtual scene, a solution for changing the user position through teleportation has also been implemented, which enables immediate movement even at great distances.

The teleportation function is activated by pressing the Trackpad Up button on the left motion controller (which is generally used for the movement functions), labeled with the number 3 in the illustration 2.1. As long as the button is pressed, a circular indicator is visible to the user, which shows the target position and can be moved to the desired area by aiming with the left motion controller. If the selected teleportation zone is valid, the indicator lights up in a green color, while it turns red if it points to an invalid position. Once a suitable destination has been selected, the Trackpad Up button may be released to perform the teleportation.

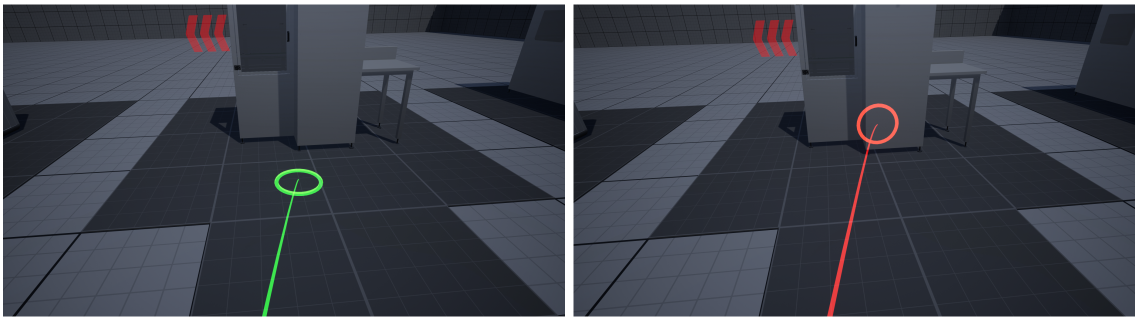


Figure 3.2: An example of valid and invalid teleport locations.

Rotation

Although the orientation of the user's head is constantly tracked by the HMD device, an alternative method of performing large rotations quickly can sometimes be useful. Therefore, the ability to perform a 45° rotation of the view to the left or right has been added by pressing the Trackpad Left (labeled with the number 2 in Figure 2.1) and Trackpad Right (labeled with the number 4 in Figure 2.1) buttons on the left motion controller.

The implementation of an impulsive rotation was preferred to a gradual rotation, in accordance with the application of best practises for the use of Virtual Reality, as the simulation of gradual rotations and accelerations that are not associated with a real movement on the part of the user can lead to so-called cybersickness, i.e., a feeling of disorientation and discomfort experienced while wearing an HMD device [18].

3.2.3 Main Menu

The main function of the Home environment is to provide access to the application's main menu. This menu is represented in a diegetic[1] way, namely as a large screen located in the center of the showroom within the 3D scene. The user can enable interaction with the main menu screen by moving closer to it. Once close enough, it is possible to aim the right motion controller towards the screen to see a light blue beam appear that follows the movements and orientation of the user's hand. This tool allows the selection of the various buttons and other controls available in the menu interface. By pressing the Trigger button located on the back of the right-hand motion controller (labeled with the number 7 in the illustration 2.1), it is possible to interact with the currently hovered button.

[1] In VR, it is referred to as diegetic the representation of user interfaces and information through the use of elements located within the virtual world, as opposed to the traditional method of visualization through static Graphical User Interfaces (GUIs) overlaying the user's view. The diegetic approach is preferred in VR as it contributes to increasing user involvement [19].

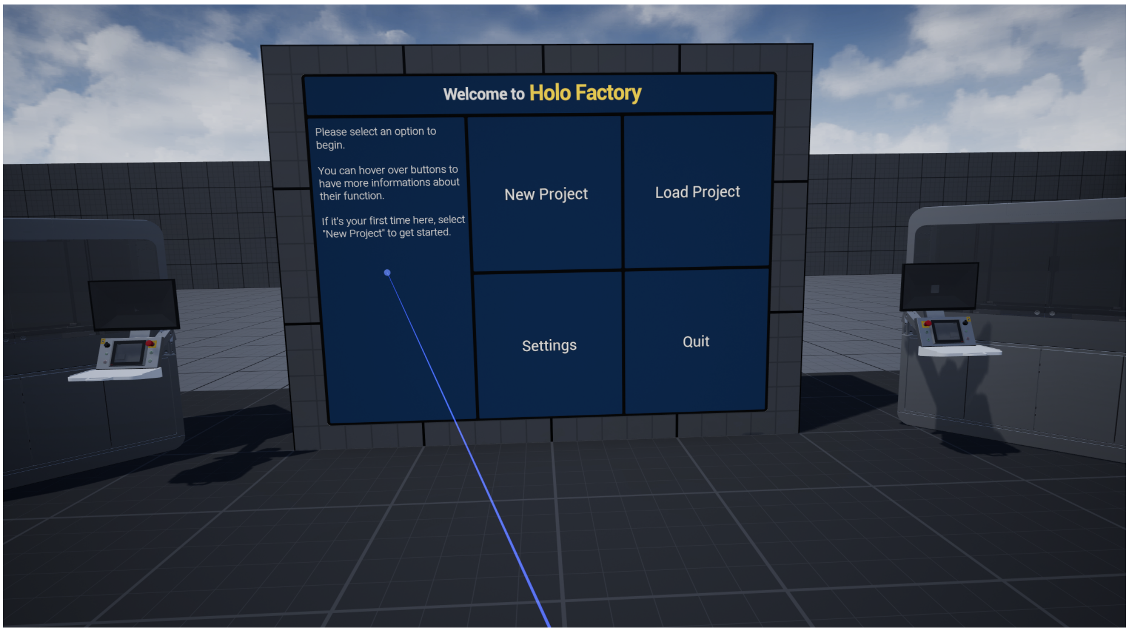


Figure 3.3: The main menu screen and interaction system.

The following options are available on the home page of the main menu screen: New Project, Load Project, Settings, and Quit.

Creating a New Project

A new project can be created by selecting the New Project option. After selecting this option, the user is prompted to enter a unique name for the project, which will be used as an identifier for saving, storing in the file system, and loading in future sessions. Once the name has been confirmed, a new empty environment is created. The user is immediately teleported to this new scene where they can start placing the various modules available in the virtual catalog and plan the desired production area.

Loading an Existing Project

By selecting the Load Project option, a project saved in a previous session can be loaded. After selecting this option, a list of all saved projects is displayed on the menu screen, from which the user can select the desired one. Once a level has been selected, the associated saved 3D environment is loaded, including all modules previously placed in it. The user is then immediately teleported to the loaded project scene.

Settings and Quit

By selecting the Settings option, the user is presented with a screen from which a number of different options can be changed, mostly related to how modules are displayed and built in the virtual environment. The complete list of settings with the documentation of their respective functions can be found in the section 3.2.8.

Selecting the Exit option, the application will be closed and the user will return to the SteamVR Home environment.

3.2.4 Project Area

After the creation of a new project, the user is placed in a new, completely empty scene. The default 3D environment of a new project consists of a factory floor that extends over an area of

50 by 50 meters with walls delimiting its bounds. The center of this square area is marked by a 3D gizmo with arrows indicating the orientation of the 3 main axes that define the coordinates system adopted by the scene (red for the positive X-axis, green for the positive Y-axis and blue for the positive Z-axis). This is important for later considerations, as it marks the origin point from which all measurements and module positions are calculated.

The project area is the environment in which the user spends most of the time within the application. This is where functions such as browsing the virtual catalog, building modules and exporting a layout are activated, which are not available in the home/showroom environment.

3.2.5 User Menu

While users are inside a project environment, they can access the user menu by pressing the Menu button on either the right or left motion controller (labeled with the number 1 in the illustration 2.1). As soon as the user menu is opened, it appears as a floating window within the 3D scene. The menu interface automatically positions itself according to the user's position and rotation so that it appears in a location that is convenient to look at and interact with. Similar to the main menu, the different elements of the user menu can be selected using a virtual laser beam controlled by the right motion controller. It is then possible to interact with a hovered button using the trigger located on the back of the right motion controller (labeled with the number 7 in Figure 2.1).

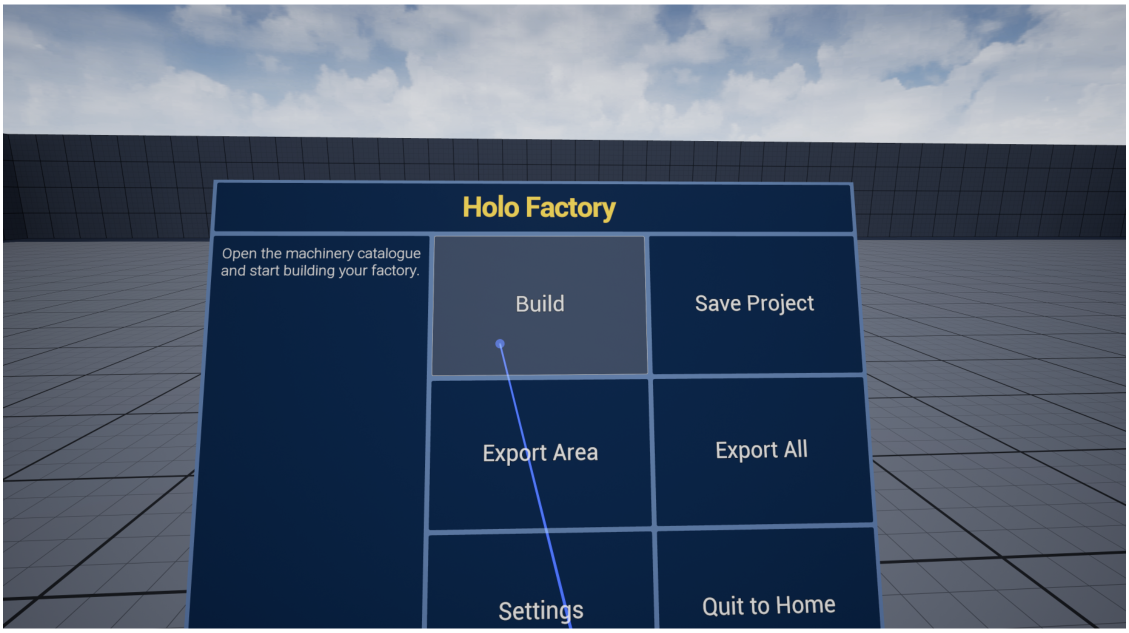


Figure 3.4: The user menu window and interaction system.

3.2.6 Buildings Catalog

By selecting the Build option in the user menu, a page showing the catalog of available modules (as shown in the illustration 3.5) is open. The main section of this page shows the complete list of modules, grouped by their category (by default, the buildings are categorized as "Test systems" and "Automation modules", but more categories can be added if custom modules are imported). The user can highlight a module of interest by pointing to it with the selection laser beam: as long as a module is highlighted, its description is displayed in the detail area situated on the left-hand side of the page. As long as a building remains highlighted, a rotatable 3D preview of the module is also visible to the left of the personal menu. Once the desired module is selected, the user can press the Trigger button on the right-hand motion controller to activate the build mode and place the module in the project environment.

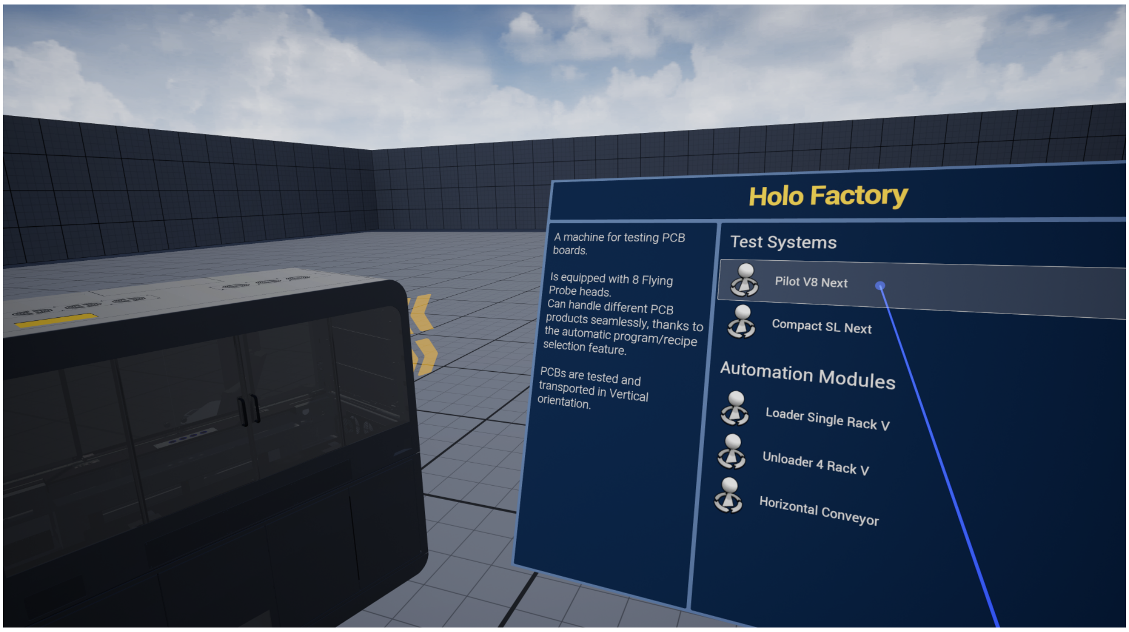


Figure 3.5: The catalog page with a floating 3D preview of the selected system.

3.2.7 Build Mode

Build mode allows the user to visualize a real-size three-dimensional preview of the selected module, move it around the project area, and finally build it at the desired position. As the user moves the preview model, a series of real-time checks are automatically performed in the background, providing immediate visual feedback on the validity of the chosen position.

As can be seen in Figure 3.6, a full-size preview of the currently selected module is always displayed in build mode. The user can move the preview using a pointer system similar to the one used to interact with the various menu interfaces. The build mode laser beam is displayed in orange color and can be controlled by moving the right-hand motion controller.

The preview of the module follows the movement of the laser pointer and is positioned where the beam intersects with a suitable surface (mainly the factory floor). The origin point of the 3D model of the module is highlighted by a gizmo with the arrows describing the local coordinate system of the object; it is this point that sticks to the intersection between the pointer beam and the floor surface during the positioning of the building.

The preview of the module is displayed as long as the position specified by the user remains valid for the construction of the module. In particular, the floor represents the main surface on which the modules are to be positioned, while the positioning is not valid if the laser is aimed towards a wall or another module. In both cases, the preview is no longer visible and a message informs the user that the selected location is not suitable for the construction of the module.

Furthermore, the area in which the module can be built is limited to a radius of 10 meters from the user's position. This limit is graphically highlighted by an orange-colored circle that delimits the perimeter within which the module can be placed. The build location is considered invalid if the user aims the laser beam outside the permitted area, resulting in the preview of the module being no longer visible.

Another important piece of information visible while in build mode are visual indicators representing the available transport sockets to connect the module to other systems by forming a line. Sockets represent the input and output connection of the module, through which the required materials and products are loaded before processing takes place,and unloaded afterwards. They are represented by semi-transparent arrows that indicate the direction in which the materials can flow within the machine. For products that can be transported in different orientations - such as PCBs, which can be handled by different systems in a vertical or horizontal position - the arrow indicators are also placed in the same orientation as the transport.

Depending on the direction in which the products can be transported through them, three different types of sockets can be visualized:

- **Inlet**: represents a connection through which products enter the module. They are displayed as green arrows pointing towards the module itself;
- **Outlet**: represents a connection through which products leave the module. They are shown as red arrows pointing out of the module;
- **Bidirectional**: represents a connection through which products can both enter and leave the module. Some modules (such as the Pilot line models described in section 1.2) can allow products to enter and exit via the same connection, while others can be configured to work in one direction - for example, with an input on the right side and an output on the left - or in the opposite one. In both scenarios, these connections are represented by bidirectional sockets. They are visualized as yellow double arrows, one pointing in towards the module itself and the other in the opposite direction.

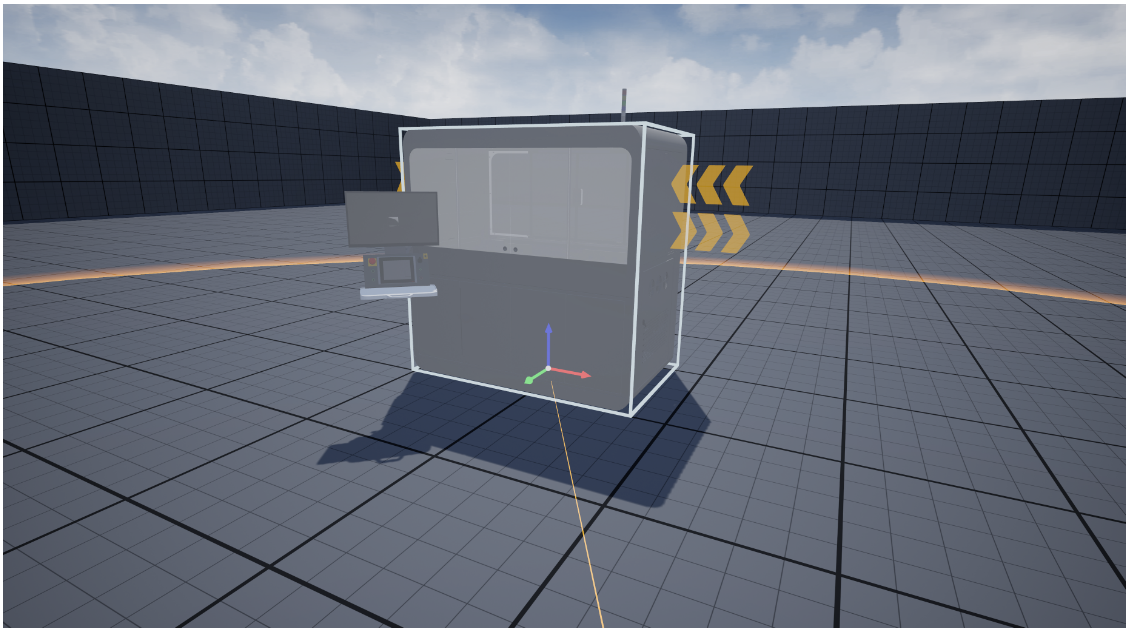


Figure 3.6: The "Pilot V8 Next" test system being previewed in build mode.

Build Mode Commands

In build mode, several new commands become available to interact with the preview of the selected module, mostly by using the buttons on the right-hand motion controller, which is generally used for tasks such as building and editing modules. Here is a complete list of the commands available in build mode:

- **Rotate preview to the left**: rotates the preview 90° to the left. It is activated by pressing the Trackpad Left button on the right motion controller (labeled with the number 2 in Figure 2.1);
- **Rotate preview to the right**: rotates the preview 90° to the right. It is activated by pressing the Trackpad Right button on the right motion controller (labeled with the number 4 in Figure 2.1);
- **Next module**: allows the user to quickly select the next module within the catalog without having to leave build mode and open the user menu. It is activated by pressing the Trackpad Up button on the right-hand motion controller (labeled with the number 3 in Figure 2.1);

- **Previous module**: Allows the user to quickly select the previous module within the catalog without having to exit build mode and open the user menu. It is activated by pressing the Trackpad Down button on the right-hand motion controller (labeled with the number 5 in Figure 2.1);
- **Confirm**: attempts to build the module at the position selected by the preview. The operation is only successful if the user selects a valid location for building the module (the preview must, therefore, be visible). If the selected position is not valid, the user is informed that the operation failed by a message on the screen. It is activated by pressing the Trigger button on the back of the right-hand motion controller (labeled with the number 7 in Figure 2.1);
- **Cycle anchor point**: allows the user to change the anchor point used to position the module in the environment. By changing the anchor point, it is possible to cyclically move the origin point of the module between five positions, namely the center of the module and the four corners of its bounding box. The active anchor point is always highlighted by the 3D gizmo indicating the origin point of the module and always follows the movement of the laser pointer performed by the user. In this way, it is possible to position a module according to the location of one of its corners instead of its center point. This is particularly useful when it is necessary to align a specific side of the module in relation to other systems or objects already present in the project. This function is activated by pressing the Grip button on the right-hand motion controller (labeled with the number 8 in Figure 2.1);
- **Open user menu**: cancels the positioning of the selected module and opens the user menu. This function is activated by pressing the Menu button on the right-hand motion controller (labeled with the number 1 in Figure 2.1);
- **Exit build mode**: cancels the positioning of the selected module and exits build mode. It is activated by pressing the Grip button on the left-hand motion controller (labeled with the number 8 in Figure 2.1);
- **Enter build mode** (shortcut key): to avoid having to open the user menu and select the module from the catalog, a shortcut key is also available to enter build mode by selecting the last module used. The button provided for this purpose is the same as the one used to quickly exit build mode, namely the Grip button on the left-hand motion controller (labeled with the number 8 in Figure 2.1). In this way, it is possible to quickly switch into and out of build mode using a single button;

Collision Handling

In addition to the above-mentioned checks on the validity of the positioning of a module, the build mode also checks for collisions or overlaps between the system in the preview and modules that have already been built. If the module's build preview is positioned in such a way that it partially overlaps an existing module, this becomes highlighted in red, as demonstrated in Figure 3.7. The overlap invalidates the location of the system in the preview, therefore if the user attempts to confirm the construction, a pop-up message informs them that the operation cannot be finalized.

Snapping Between Modules

When two industrial automation modules are placed within a production line and components (in this case mainly PCBs or pallets transporting EV batteries) need to move from one to the other, it is essential that they are positioned and aligned correctly, with tolerances generally being a few millimeters at most.

It is, therefore, important that a high level of precision is used in the design phase to align compatible modules within a line. This precision can be difficult to achieve when positioning modules by hand using a Virtual Reality laser pointing system, especially considering that the inlet and outlet connections of the various modules are often not aligned with the origin and their exact

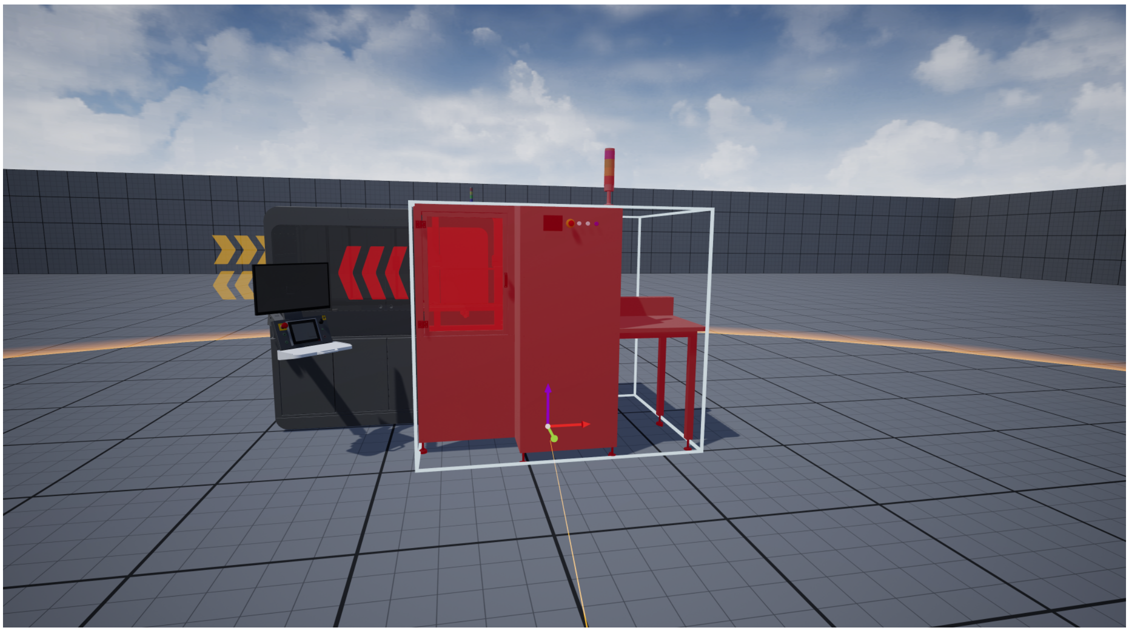


Figure 3.7: A previewed module overlapping an already built system.

position may not be even known to the user. Another source of inaccuracy is caused by micro-movements of the user's hand, constantly tracked by the very sensible motion controllers, which are more amplified the greater the distance between the user and the location they are aiming at. Even with steady hands, this effect cannot be fully canceled out and causes positioning errors ranging from a few millimeters at a very short range up to 20-30 centimeters near the maximum build range.

For these reasons, as well as for the sake of facilitating and speeding up this fundamental step of a production line design process, a snapping function has been implemented. This function consists of an assisted positioning of the module under construction when one of its sockets is close to a compatible free socket belonging to a different already constructed module. In particular, each socket has its own influence volume. If the volume belonging to a socket of the module under construction overlaps - even partially - with that of a compatible existing socket, the preview of the module under construction automatically moves to the position that allows perfect alignment with the existing module. When the module under construction successfully reaches a snapping position and does not overlap with other modules, the preview is highlighted in green and it becomes possible to confirm the positioning and build the module.

In order to activate the snapping between the module under construction and another target module, it is necessary that the two sockets concerned are compatible with each other. The following conditions must be true for two sockets to be compatible:

- They must have a compatible socket type (inlet/outlet/bidirectional). Inlet and outlet sockets are compatible with each other and with bidirectional sockets, but not with sockets of the same type. Bidirectional sockets are compatible with any type of socket;
- They must be aligned in the same transport direction: the arrows representing their direction of flow must have the same orientation. They can not form a 90° angle to each other, but must be aligned in a straight line;
- They must transport the same type of product or material (by default this can be either PCBs or pallets, but other products can be added by importing custom modules);
- The product must be transported in the same orientation (horizontal or vertical). This applies in particular to modules that transport and process PCBs. These items can be moved horizontally or vertically positioned through a production line and even change their orientation within a module. This is highlighted by the orientation of the arrows representing

the socket, which can be positioned horizontally or vertically depending on the orientation of the product it is intended to transport;

- The target socket must not already be occupied by an existing connection.

An example of compatible sockets successfully snapping together is shown in Figure 3.8. In particular it is clearly visible how the two sockets have a compatible type (outlet and bidirectional), do not form any angle and are both vertically oriented.

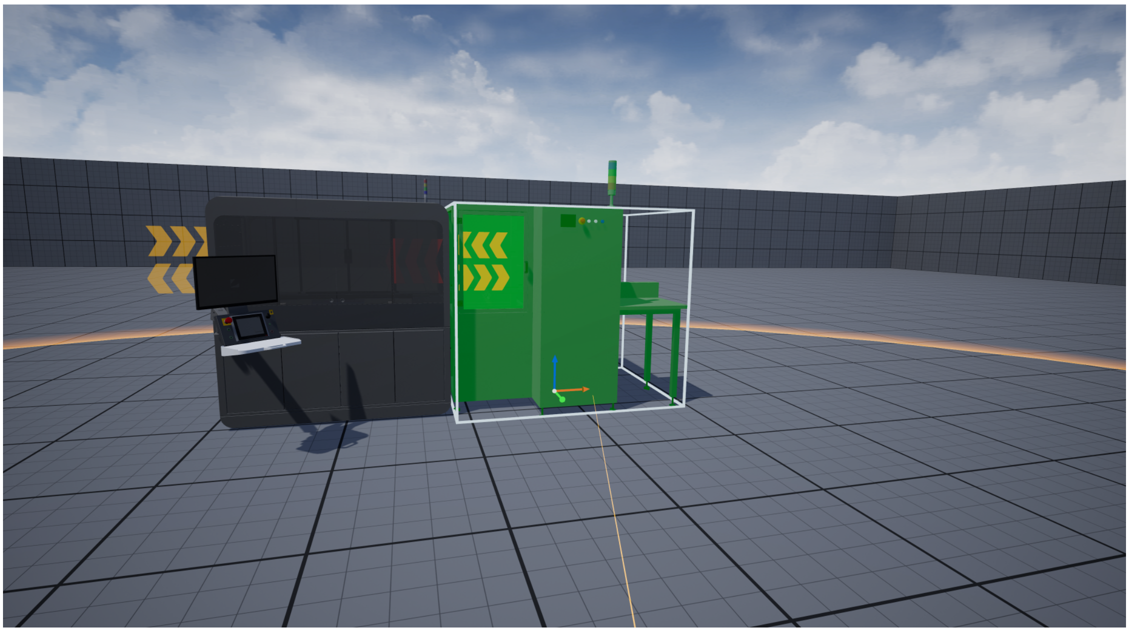


Figure 3.8: Two modules snapping together through a pair of compatible sockets.

Editing Existing Modules

By pressing the Trackpad button on the right-hand motion controller (labeled with the number 9 in Figure 2.1), the function for selecting and editing existing modules within the project can be switched on and off. When this function is active, a blue laser pointer becomes visible, which can be controlled with the movement of the right hand, similarly to what happens while in build mode. Pointing this laser beam towards an existing module and pressing the selection button - the rear Trigger on the right-hand motion controller (labeled with the number 7 in Figure 2.1) - a small menu opens in front of the selected module, through which it is possible to move or delete it.

By choosing the move option, the selected module is removed from its position and the user automatically enters build mode in order to reposition it. All sockets that may have been occupied by the targeted module before being moved are freed and can be occupied again by other modules. By selecting the delete option, the selected module is deleted and, once again, all sockets that may have been occupied by the targeted module are freed.

3.2.8 Additional Settings

Several optional settings are available, which can be activated to allow more accurate positioning of the modules while in build mode or to display more detailed information about the constructed modules. These settings can be accessed by opening the Settings page in both the user menu and/or the main menu located in the Home scene. The settings are divided into two categories:

- **Selection**: affect the appearance of a module when it is hovered using the selection tool;
- **Build Mode**: affect the appearance of a module when it is previewed in build mode.

Figure 3.9: The "Settings" page inside the user menu.

Bounding Box Visualization

By enabling the "Show bounding box on hover" option in the Selection section, it is possible to display the bounding box of modules when they are aimed at with the laser pointer of the selection tool.

A bounding box is defined as the rectangular prism with the smallest volume that contains the module in question. Displaying this information can be important as the bounding box dimensions of each module are specified in the layout document once the project is finalized and the export has been performed (as described in the section 3.2.9). The bounding box of each module can be calculated automatically by the application or defined by the designer through the use of a custom 3D mesh: this option is particularly useful in order to prevent certain moving appendages present on different machines from affecting the bounding box dimensions.

The display of the bounding box can also be activated while in build mode for the selected module preview by activating the "Show bounding box in preview" option in the Build Mode section.

Size Visualization

In a similar way to that described for the bounding box display option, it is possible to enable the display of a module's dimensions by ticking the "Show size on hover" option in the Selection section. When this option is enabled, the footprint dimensions - width and depth - of the module are shown as an overlay located at its base, when the module is indicated with the selection pointer tool. The dimensions are shown in millimeters and represent the exact size of the module's bounding box. They can therefore be modified by the designer by using a specifically crafted 3D mesh as a customized bounding box. For the same reason, it is also most useful to have this option enabled together with the "Show bounding box on hover" option.

The display of a module footprint dimensions can also be enabled during build mode for the system being previewed, by enabling the "Show size in preview" option in the Build Mode section of this page.

Snap to World Grid

Another setting intended to facilitate the positioning of new modules while in build mode is the "Snap to grid" option. When "Snap to grid" is enabled, the preview of the module under construction will no longer strictly follow the laser pointer intersection with the ground plane, but will move in fixed increments along the two main axes of the surface. These fixed-size increments define precisely a grid composed of squares with sides equal to the size of the minimum displacement. The module under construction always positions itself on the point of the grid closest to the laser pointer. This option is very useful for positioning modules precisely or at a given distance from other elements already present inside the project area, as it facilitates the selection of a precise point on the factory floor surface. This can be difficult to achieve freehand - without enabling this feature - due to the high sensitivity of the laser pointing system, especially as the distance to the selected point increases.

When "Snap to grid" is enabled, the grid on which positioning takes place also becomes visible in the are near the preview of the module under construction. An additional grid (highlighted by thicker lines) is also always shown, displaying fixed increments of 1 meter. This helps the user in placing the module in the desired position, by giving a visual indication of where the different grid points are located. The grid overlay updates in real-time following the current position of the preview and fades out at a distance of about 5 meters from the anchor point of the module.

To enable this option, it is necessary to tick the "Snap to grid" entry in the Build Mode section. It is also possible to customize the size of the grid on which the snap takes place (i.e., the size of the minimum movement of the module), by modifying the value of the "Snap to grid size" option. The available values are 10, 50, 100, 200, 250 and 500, all expressed in millimeters.

Location Visualization

With the same purpose of facilitating the precise positioning of modules in space (at times when the automatic placement by snapping between sockets is not exploited), it is possible to enable the visualization of the module position in real-time, while the user is in build mode. To enable this option, the "Show position" setting must be ticked in the Build Mode section.

With this option, as long as the build mode is active, the position of the selected module is displayed as an overlay following the active anchor point. This means that by changing the selected anchor point, or by activating the "Snap to grid" option, the position will vary, always indicating the active anchor point of the module, highlighted by the 3D gizmo.

The position of the module's preview is shown as a pair of coordinates (X, Y). These two values follow the coordinates system of the project environment, which has its origin marked by the 3D gizmo located at the center of the factory floor. As for previous measurements, also the position of the module is expressed in millimeters.

Distances on Main Axes

Another need that may arise with some frequency during the design of a line is that of having to position a module at a certain known distance from another element (already present within the project area). In order to facilitate this operation, an option has been added that makes it possible to calculate the distances between the module being constructed and other items, along the main horizontal axes (X and Y) of its bounding box. These values are then updated and visualized in real-time as spatial UX components[2] inside the scene.

The distances are visually represented by lines that originate from each side of the bounding box of the module under construction and branch off in the four directions described by its two

[2] Spatial components are informational elements, such as text overlays and visual effects, presented in the virtual world. These elements are used to provide additional information relative to virtual objects in the scene, although not being represented as virtual objects themselves (as for diegetic elements). Spatial UX components are used to convey information to the user in a more immersive way than with traditional 2D UI or HUD elements [20].

main axes. Each of these lines is only shown if there is an object from which to calculate the distance within a 10-meter radius of the module preview. The value of each calculated distance is shown (in millimeters) slightly above the corresponding line. The distances are calculated and updated for each frame by projecting the bounding box 3D mesh along its two horizontal axes, both in the positive and negative direction, for a maximum of 10 meters. If the projection detects an overlap, the closest hit point is used for calculating the distance from the nearest object in that specific direction. If no hits occur during the projection, then no distance is calculated and the visual indicator and text are hidden.

To enable this option, the "Show distance on axes" item must be toggled in the Build Mode section.

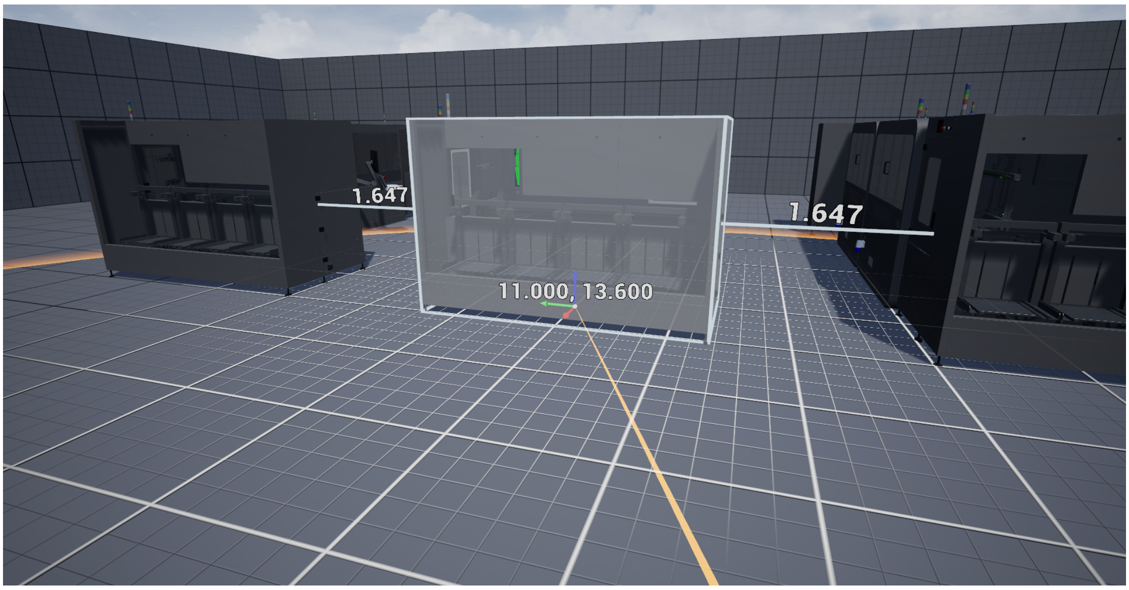


Figure 3.10: User in build mode, with several optional settings enabled.

3.2.9 Export

Once the design of the production line (or part of it) is complete, the export function can be used to automatically generate a layout document in PDF format, containing data and dimensions of the modules built inside the virtual environment.

The layout document consists of a top view of the design area, where the modules are represented schematically, and all their positions, dimensions, and respective distances are specified. This document serves as the starting point for the construction of a production line, precisely defining the overall dimensions and relative positioning of the various machines that make up the line. In addition to the layout diagram, there is also a table summarising the data of the modules in the line.

The user has two export modes available, both of which can be accessed via the personal menu. The "Export all" option allows the entire project area to be quickly exported, thus considering all elements located in the scene. With the "Export area" option, on the other hand, it is possible to select a subset of the project area and proceed with the export of just the modules lying within it.

By selecting the "Export area" option, the user automatically enters the build mode, where they must position two markers in the scene, in order to identify the two opposite corners of the rectangular area they wish to export. The first marker is characterized by a green color, while the second is red. After the positioning of the first marker, a semi-transparent yellow cubic volume appears, highlighting the size of the area that will be exported; this volume resizes in real-time while the user chooses a suitable location for placing the second marker (as can be seen in Figure 3.11). During this phase, it is possible to interrupt the export procedure at any time by exiting the construction mode before positioning the second marker. Otherwise, once the second marker has been positioned, the export procedure immediately starts.

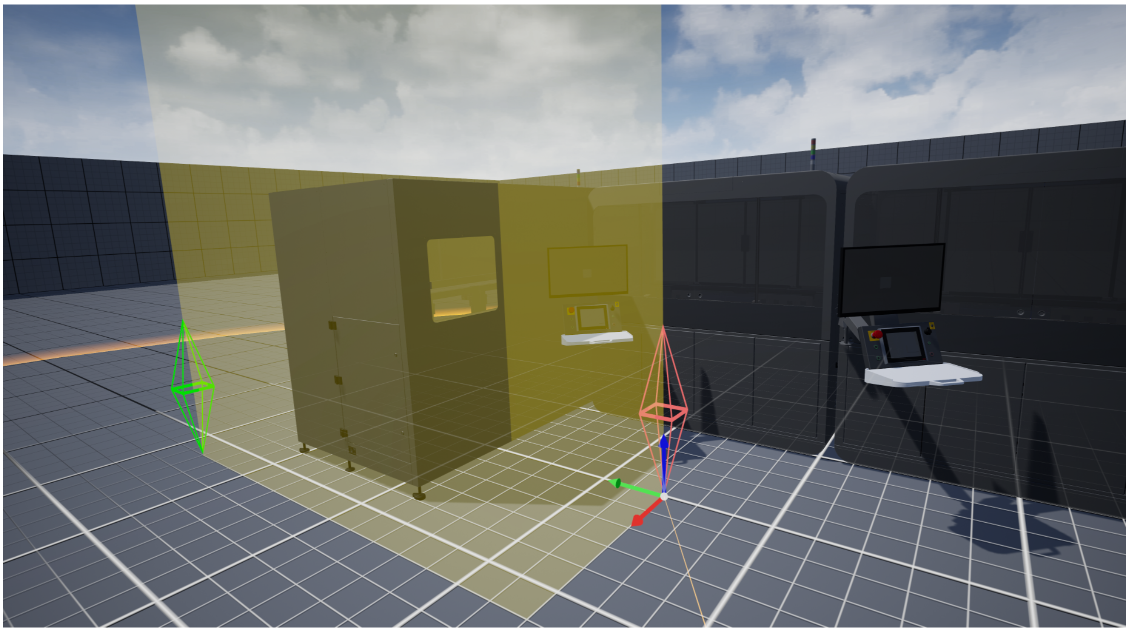


Figure 3.11: User selecting the area to be exported.

The Export Algorithm

Once the export procedure has been started, either by selecting the "Export all" option or after completing the selection of the desired area in the case of "Export area", a 3D text indicator will be displayed - showing the message "Layout export in progress..." - inviting the user to wait until the export has been completed. It will not be possible to start a new export procedure until the previous one has been finalized.

The export procedure consists of two phases executed in succession: the first phase is carried out within the application and consists of cycling through all the modules within the selected area (the entire project in the case the export has been triggered through the "Export all" option), collecting, for each one of them, the information necessary for generating the layout document. This information consists of:

- **Name**: the name identifying the model of the module;
- **Description**: a brief description of the module;
- **Identifier**: usually one or more letters used to identify the schematic representation of the module in the layout document;
- **Location**: the coordinates (X, Y) defining the position of the module in the project area, expressed in millimeters;
- **Rotation**: the rotation of the module about the Z (vertical) axis, expressed in degrees;
- **Size**: the footprint dimensions (width and depth) of the module, expressed in millimeters;

Along with this information, a high-definition image of the module seen from above is also saved, after applying a post-production filter that extracts a schematic representation of it, highlighting its contours and geometry.

Since image capture is the most time-consuming operation within the export procedure, it has been optimized to maintain a cache in which the images are saved. In this way, if the image of a certain module already exists in the cache, either because that module appears several times within the project or because it has been previously exported (even from other projects), it is not necessary to capture and process it again, which drastically reduces the time needed for the export procedure to complete. After the capture, the image is saved to the file system (if not already present) and its path location is added to the information collected for each module.

The second stage of the procedure is the actual generation of the PDF document. To do this, the application executes a Python script containing the instructions for generating the file; the data collected by the application is placed inside a string in JavaScript Object Notation (JSON) format and used as a parameter for the execution of the script. Here, thanks to the information received for each module, it is possible not only to reconstruct the positioning of each element within the project area but also all combinations of relative distances.

The result is the creation of a file in PDF format, within which a schematic representation of the production area seen from above can be found, on which all dimensions of the various modules and the distances between them (expressed in millimeters, with a precision of 0.5 mm) have been automatically quoted. Modules are identified within the layout by a label consisting of their specific identifier followed by a progressive number. There is also a table at the bottom of the page summarising the list of the various system models that take place in the layout, with their respective quantities and unique identifiers. An example of the layout of a small automatic production line, fully generated using the export procedure provided by HoloFactory, is shown in Figure 3.12.

If the application has been deployed using the dedicated installer, the generated PDF layout document is saved in the application's output folder located at:

`C:\Program Files\HoloFactory\Output`

The name of the created file is composed by the project name and the date and time of export so that subsequent executions of the export procedure do not overwrite previous versions. An example of file name is `ProjectName_20240704T180735.pdf`.

The choice of using a Python script to perform this part of the export procedure stems from the need for it to be easily configurable. By modifying the code and/or the parameters defined within the Python script, it is indeed possible for an experienced user or programmer to radically change the appearance and the way the data is presented within the PDF document. More information about the possibility of customizing the export script can be found in the section 3.3.3.

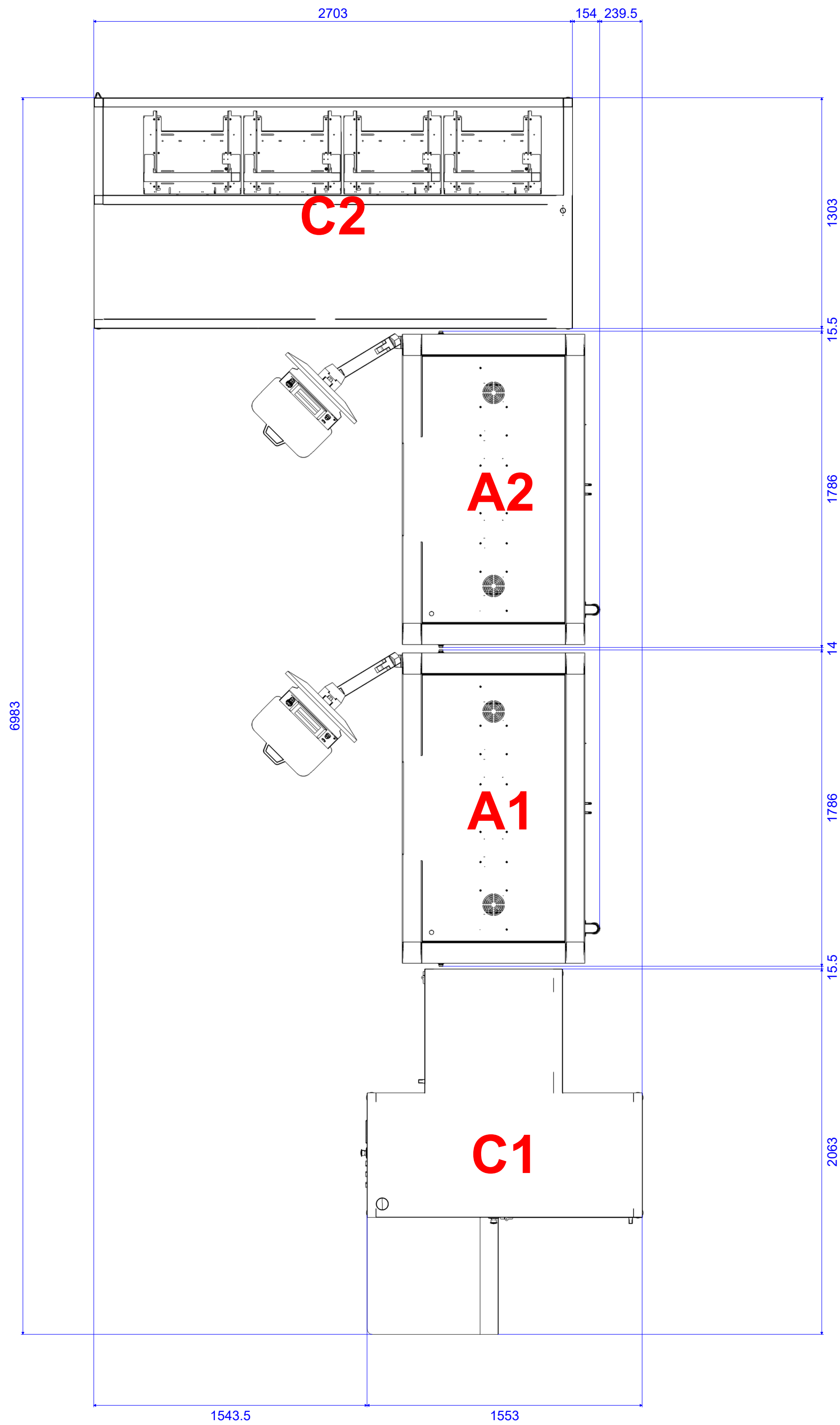
3.2.10 Save Project

The "Save project" function, found on the main page of the user menu, allows for storing all information associated with the current project environment in system memory, in order to be able to reload and resume editing the project in future sessions. Once a project has been saved, it then becomes available in the list of projects that can be loaded when using the "Load project" option from the main menu of the Home scene.

When saving the layout of a project, a list of every module present inside the virtual environment, as well as a map containing all their connection with each other, are saved. This ensures that, when loading the project, every machine will be correctly placed inside the production area and every socket will be restored to the correct state, either free or connected to another socket.

Quit

After saving the modifications made to the project, it is possible to go back to the Home area by using the "Quit to home" button found in the user menu. This operation concludes the standard work cycle of the application.



ITEM	PART NUMBER	DESCRIPTION	QUANTITY
C1	Loader Single Rack V	Automation module for loading PCB panels from racks to Vertical test systems.	1
A1, A2	Pilot V8 Next	Automatic Vertical PCB panels tester.	2
C2	Unloader 4 Rack V	Automation module for unloading PCB panels from Vertical test systems into racks.	1

Figure 3.12: An example of generated layout document for a small production line.

3.3 Developer Manual

This section highlights the more technical features offered by the application, providing advanced users and developers with the opportunity to further improve, extend and customize the behavior of HoloFactory. The following sections also provide additional insights into the development process and the technical choices required at this stage, in order to deliver as complete an analysis as possible of the functioning of these functionalities.

3.3.1 Adding Buildable Modules

HoloFactory features a complete catalog consisting of all the test systems and automation modules that are currently manufactured by SEICA. However, in the future the content of this catalog may often have to change for several reasons. In the first place, new automation modules and test systems are constantly being developed by SEICA's mechanical engineers and designers to better satisfy the needs of the evolving industry - therefore new models may likely need to be added to the HoloFactory virtual catalog. Another common reason is that customers sometimes require specific hardware modifications to be implemented on their test equipment, leading to the creation of slight variations of an existing model. Moreover, designers using the HoloFactory application to plan the layout of a production area may also be interested in adding to their virtual environment other structural elements - such as protective barriers, pillars, ground indicators, signs, dashboard screens, carts for the transportation of material, shelves, and depots for storing products, and so on - that may be relevant to the layout or for carrying out some of the different spatial assessment possible in Virtual Reality.

For these reasons, the HoloFactory application ha been equipped with the possibility to dynamically load custom modules and add them to the catalog. This feature makes it possible for advanced users to customize the content of the catalog, by adding the desired 3D models and tailoring their behavior to their specific needs. This feature also contributes to broadening the scope of the HoloFactory project by potentially allowing users to leverage the planning tools offered by the application not only for the design of industrial production lines but virtually any kind of environment.

In order to add a custom module to the buildable catalog, the files containing its 3D model, as well as all the needed information, must be provided to the application. In particular (assuming that the application has been deployed using the provided installer) a JSON formatted file containing the information of the module to be added must be created inside the folder:

```
C:\Program Files\HoloFactory\Modules
```

During startup, the application scans this directory for any JSON file, and parses its content to gather information about custom modules that may need to be shown in the virtual catalog. This means that it is up to the user whether to create a different file for each custom module or to combine the definition of all modules inside a single file. Aside from this file, it is recommended to create a subdirectory for each custom module, containing the main 3D model file (defining the shape of the module) as well as any other file that may be needed for defining different features, as described in the next step. An example of the suggested folder structure is represented in Figure 3.14.

For each added module, an object must defined inside of one of the above-mentioned JSON files, specifying all the attributes needed for the application to create a functional virtual representation of the given element. The list of attributes consists of several fields, some of which are optional.

- **Name**: a string representing the name of the model of the module. It must be uniquely defined within the project, as it identifies the specific type of module. If a single JSON file is created for describing the characteristics of the module, it is recommended to name it as this field;

- **Image**: [optional] the full path of an image to be used as a thumbnail for representing the module inside the virtual catalog. It should be squared in shape and with a resolution of up to 512 x 512 pixels. It is recommended to put this file inside a specific subfolder named after the module;
- **Category**: [optional] a string representing the category to which the modules belong to. It can be used to group related modules together in the catalog. By default, only two categories are used inside the catalog, "Test System" and "Automation Module", but custom categories can be added by the user by simply specifying them inside the definition of new modules;
- **Description**: [optional] a string containing a detailed description of the characteristics of the module. It is used to display information about the module while it is selected by the user inside the virtual catalog;
- **Short description**: a string containing a short description (max 150 characters) of the module. It is used as the description of the specific element during the generation of the layout document;
- **Module Id**: a short string (max 3 characters) used as an identifier for labeling the module's schematic representation inside the generated layout document. Does not need to be unique; in the standard catalog for example the same letter "A" is used for labeling all test systems from the "Pilot" line, while the letter "C" is used for all the "Automation modules";
- Mesh: the full path of the file containing the 3D model of the module. The file must be in FBX format;
- **Bounding box**: [optional] the full path of the file containing a simple 3D model to be used as the bounding box of the module. The file must be in FBX format. If this field is not specified, the bounding box of the module is automatically calculated inside the application;
- **Sockets**: a list of JSON objects defining the various socket connections available on this module (if any). The JSON structure of a socket object is described below.

Each socket object listed inside the module definition represents an interface point for the module to be connected with other modules featuring compatible sockets. The JSON representation of a socket object contains the following attributes:

- **Direction**: a string representing the direction of the socket. It can be either "inlet", "outlet", or "bidirectional". Inlet and outlet sockets are compatible with each other and with bidirectional sockets, but not with sockets of the same kind. Bidirectional sockets are instead compatible with any kind of socket.
- **Type**: a string representing the type of materials/products that are supposed to be transferred through the socket. By default, two types are defined by the modules already present inside the catalog, "PCB" for Printed Circuit Boards and panels and "battery" for EV batteries carried by metal pallets. Custom type identifiers may be defined by the user. Sockets are only compatible with other sockets with the same type;
- **Location**: an array with three numbers defining the 3D coordinates (X, Y and Z) of the connection point of the socket. This position must be specified relative to the coordinate system and origin point defined by the 3D model used as the Mesh for the module. This parameter defines the exact point to which other compatible sockets snap during build mode;
- **Rotation**: an array with three numbers defining the rotation around the three axes (X, Y, and Z) of the connection point of the socket. For a socket to work properly when connecting to other sockets, its forward vector (i.e., its local positive X axis) must always face towards the center of the parent module. Usually, a rotation around the Z (vertical) axis is enough to satisfy this condition;
- **Snap area**: the full path of the file containing a simple 3D model (usually a box) defining the volume of influence for this socket - i.e., the volume in which another compatible socket must be located in order for the automatic positioning (snapping) to be triggered. The file must be in FBX format;

- **Visible**: a boolean value (true or false) indicating whether the socket must be highlighted by the semi-transparent arrow indicators inside the application. By default, all modules' sockets are represented this way, but it may be useful to disable this visualization to enable a more flexible usage of sockets in order to exploit the snapping behavior also for elements that do not actively transport any item.

An example of the content of a JSON file created for the addition of a module inside the HoloFactory library is shown in Figure 3.13. Particular attention should be put to how the rotations of the two sockets have been defined: the first one, located on the left side of the machine does not need to be rotated as its forward vector is pointing towards the positive X-axis direction, hence towards the center of the module itself; on the contrary, the second one, which is located on the right side, has a rotation of 180 degrees around the Z (vertical) axis in order for its forward vector to also face towards the center of the module.

```
{
      "name": "Pilot V8 Next",
      "image": "C:/Program Files/HoloFactory/Modules/Pilot V8 Next/image.png",
      "category": "Test System",
      "descrition": "A machine for testing PCB panels. Is equipped with 8
          Flying Probe heads. Can handle different PCB products seamlessly,
          thanks to the automatic program/recipe selection feature. PCBs are
          tested and transported in Vertical orientation.",
      "shortDescription": "Automatic Vertical PCB panels tester.",
      "moduleId": "A",
      "mesh": "C:/Program Files/HoloFactory/Modules/Pilot V8 Next/mesh.fbx",
      "boundingBox": "C:/Program Files/HoloFactory/Modules/Pilot V8
          Next/bb.fbx",
      "sockets":
      [
            {
                  "direction": "bidirectional",
                  "type": "PCB",
                  "location": [ -90, 4.25, 130 ],
                  "rotation": [ 0, 0, 0 ],
                  "snapArea": "C:/Program Files/HoloFactory/Modules/Pilot V8
                      Next/socketL.fbx",
                  "visible": true
            },
            {
                  "direction": "bidirectional",
                  "type": "PCB",
                  "location": [ 90, 4.25, 130 ],
                  "rotation": [ 0, 0, 180 ],
                  "snapArea": "C:/Program Files/HoloFactory/Modules/Pilot V8
                      Next/socketR.fbx",
                  "visible": true
            }
      ]
}
```

Figure 3.13: An example content of a JSON file for the addition of a custom module.

Once the correct files have been placed inside the `Modules` folder, the related elements should become available inside the build catalog. Custom-added modules may show a loading delay - proportional to the size and geometric complexity of their 3D model - when being selected for the

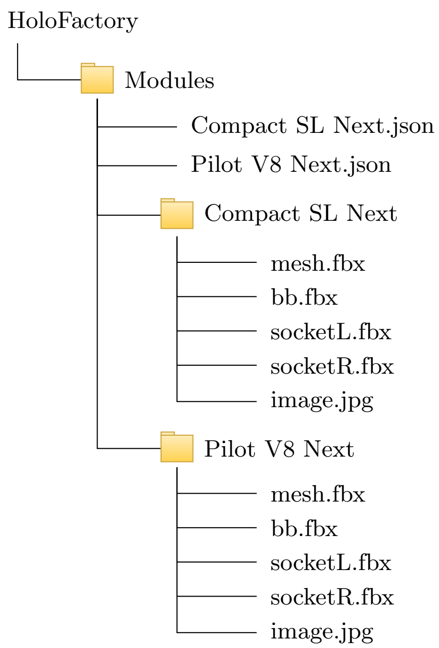


Figure 3.14: An example content of the "Modules" folder.

first time from the catalog inside the application. This is due to the fact that the corresponding 3D models must be dynamically loaded from the file system during runtime, while modules from the standard catalog (which have been imported directly inside the Unreal Engine environment before building the application project) have been compiled and optimized by Unreal Engine and are directly stored inside the application executable and related asset data files.

The only file format currently supported by HoloFactory for adding modules to the catalog is FBX, mainly due to the limitations of the Unreal Engine plugin "RuntimeStaticMeshImporter", which is responsible for dynamically importing and rendering meshes at runtime. Even though FBX is one of the more versatile and diffused file formats for 3D modeling, support for other popular formats such as GLTF, GLB, OBJ, and Standard Tessellation Language or Stereolithography (STL) might be added as a future work by integrating the glTFRuntime plugin and its variants [21], which allows for loading meshes it the mentioned file formats at runtime.

3.3.2 Converting CAD Models

One of the most challenging steps when it comes to using 3D CAD models in a Virtual Reality environment, is the export and conversion of such models from CAD data to a graphical mesh representation, suited for rendering inside a real-time application [3]. Usually, files containing technical projects and mechanical drawings are stored in CAD-specific file formats, specific to the CAD design tools adopted by the company. In the case study proposed by this project, the main CAD editing platform used by the company is SolidWorks, therefore the entire library of every module's technical drawing is available in the CAD native STEP format. This file format is not suitable for the representation of 3D models in Virtual Reality engines like Unreal Engine, therefore a conversion step must be performed in order to simplify the model and convert it to a common graphical mesh format like FBX, GLTF, or GLB. Moreover, mechanical drawings often contain an enormous quantity of components, which can lead to extremely high polygon counts when converted to a mesh format. For this reason, while performing the format conversion step, it is also useful to verify the complexity of the generated mesh and, if needed, proceed to a simplification by removing some of the hidden internal parts that are not crucial for the positioning and evaluation of the characteristics of the module within a production line. This conversion step may be achieved in different ways, depending on whether it takes place during the development steps of the application (meaning it is being performed by a developer who has access to the Unreal Engine programming environment) or for the preparation of the 3D models for being dynamically imported in HoloFactory through the technique showcased in section 3.3.1.

A general workflow for the complete conversion from a STEP project file to a collection of 3D meshes represented in a format compatible with Unreal Engine, can be articulated in four different steps. Operations performed in the last two steps may differ depending on which of the two available routes for adding a module to the HoloFactory library is being followed.

1. 3D CAD export: the first step consists of opening the STEP file containing the module of interest end exporting it in a graphical format. To do this the CAD file must be opened within a suitable editor. Possible options include:

   - SolidWorks can natively open and edit STEP files, as it is the main CAD tool used by the company. It can export into very few graphical mesh formats, with STL being the most common option, although not a widely diffused file format overall. Furthermore, STL files can not be directly imported into Unreal Engine, meaning another conversion step is needed with this option. This does not represent a real problem most of the time because a revision and simplification step, performed with a more general purpose 3D modeling software, is usually required anyway before importing the final model in Unreal Engine;
   - FreeCAD is a free and more versatile alternative. It is an open-source CAD design tool capable of importing and editing STEP files, while offering more flexibility for exporting object as graphical meshes. Aside from the STL format, this tool also provides the possibility to export in OBJ as well as the more popular GLTF and GLB formats. Both these file formats are widely supported by 3D modeling software and can also be directly imported into Unreal Engine; for these reasons, FreeCAD has been extensively adopted as the main tool for converting STEP files to GLTF format during the development of HoloFactory;
   - The popular open source 3D modeling software Blender can also be directly used for importing and converting project files in STEP format, although it requires the installation of the STEPper [22] paid add-on to do so. Being a complete 3D modeling tool, Blender has the advantage that can be used also for the following step of revisioning and simplifying the converted 3D model. It can then export the converted model to a variety of graphical file formats, including OBJ, GLTF, GLB, and FBX.

   During this step, it is also possible to perform an initial optimization of the modules' complexity, by excluding some of their internal components from the selection of exported geometry.

2. 3D model review and optimization: this step is aimed at the verification and eventual simplification of the mesh generated after the conversion. During the conversion step, several features of the original CAD model, such as materials, lights, and animations, are likely to be lost, and artifacts in the 3D mesh may appear due to the loss of accuracy suffered when porting the model from a mathematically exact representation to a graphical (approximated) one. It is, therefore, necessary to open the resulting model file with a 3D modeling tool in order to recreate the lost features - mainly materials, as lights and animations do not need to be imported into Unreal Engine. The main editor used for performing these operations during the development of the application is Blender, which offers a comprehensive suite of tools for modeling, texturing, rendering, and exporting 3D models. Based on the performance of the targeted workstation for the utilization of the application, it is also possible that some optimization may have to be applied to the mesh during this phase. In this regard, Blender offers several modifiers that can be applied to 3D objects in order to reduce the number of vertex and polygons constituting them.

3. Additional components generation: after the mesh representing the main 3D model for the module of interest has been polished, several more mesh may need to be generated and exported, in order to describe features such as the customized bounding box and/or the volumes of influence relative to the module's sockets (if any is present). These additional components should then be saved as independent mesh files and - together with the main 3D model - either be imported directly into Unreal Engine for creating modules that are part

of HoloFactory's standard catalog, or exported as FBX and copied inside a subdirectory of the "Modules" folder in order for the module to be dynamically loaded at runtime. Several components and information can be gathered using the 3D modeling tool of choice during this phase:

- A custom bounding box may be modeled and exported as a separate mesh file. This can be used to define the bounding box of the given module, useful, for example, in cases where some parts of the module's main mesh should not contribute to the calculation of its bounding box. Should be exported in FBX format if used for a dynamically loaded module, or in any other Unreal Engine supported formats if directly added to the application project files;
- Several boxes, defining the volumes of influence for each of the sockets available on the given modules, may be modeled and singularly exported. These meshes should be positioned accordingly to the module main mesh, and exported from that position; this ensures that they will maintain the correct position relative to the parent module when imported in Unreal Engine or dynamically loaded into the application. The file format must follow the general rule mentioned for the bounding box file;
- Placeholders representing each socket's connection point may be placed in the scene, according to the main mesh. By doing so, the 3D location and rotation of each of these sockets can be accurately measured in order to correctly compile the corresponding attributes when defining the socket in the application (either in the JSON file for dynamically imported modules or directly in the Unreal Engine project for standard catalog's modules).

4. Final import: in this last phase, the various 3D model files exported and converted in the previous steps are finalized to be used inside the HoloFactory application. During the development stages of the application, these files may be directly imported into Unreal Engine and packed inside the main executable of the application. If the module is instead being prepared for getting dynamically added to the application's catalog, the resulting FBX files must be copied inside a suitable subdirectory of the "Modules" folder and a JSON file containing the definition of the needed module's characteristics must be compiled (as described in section 3.3.1).

3.3.3 Export Customization

The export procedure, i.e., the algorithm responsible for the creation of a layout document based on the information that can be gathered from the virtual representation of a line in HoloFactory, has been implemented using the Python scripting language. All the logic is self-contained inside a single Python file, which is run by the application as part of the export procedure. The input data needed for the layout generation is gathered within the application and passed to the Python script as a command line argument, after being serialized in JSON format.

This paradigm is very common inside the suite of software tools developed by SEICA. By demanding the implementation of different functions to specifically defined Python scripts, these can be used as plugins for the main application. This enables a strong customization of the behavior of the application by enabling users to modify well-defined (and possibly safe) sections of the code that are executed without the need to rebuild the entire project.

The Python script file containing the code responsible for the generation of the layout document can be found, after deploying the application using the dedicated installer, at the following location:

`C:\Program Files\HoloFactory\Python\layout.py`

By editing the file content some of the document generation parameters can be quickly tuned by changing the values of several top-level global variables defined at the top of the script (listed in Figure 3.15). These settings include a scale factor for resizing the document, the base size of the font, and several options for padding between different elements of the layout.

```
DOC_PADDING = 500
SCALE_FACTOR = 0.5
FONT_SIZE = 50
BORDER_PADDING = 8
BORDER_WIDTH = 8
ANNOTATION_PADDING = 400
ANNOTATION_TEXT_PADDING = 20
ANNOTATION_LINE_OVERSHOT = 10
ANNOTATION_ARROW_HEIGHT = 6
ANNOTATION_ARROW_LENGTH = 12
```

Figure 3.15: The top-level global variables found inside the `layout.py` script.

A more in depth personalization is achievable for an advanced user by editing the content of the various methods within the file and adding custom logic tailored to their needs. Particular attention must be paid at this stage not to edit the basic interface used by the HoloFactory application to start the execution of this script. The application launches the Python script through the Windows command prompt shell, using the following command:

```
python "C:/Program Files/HoloFactory/Python/layout.py" %JSON_DATA%
```

Therefore it is important to always include in this script the `main()` function handling the conversion and parsing of the first received argument, which is used to transfer the information gathered from the 3D scene to the script. By default, the script also implements a convenient conversion of the received JSON data to a list of `Machine` objects, specifically defined to facilitate access and usage of the available information. A code snippet highlighting this section of the script is shown in Figure 3.16.

The parsing strategy for the JSON data received by the application can also be customized. HoloFactory always calls the script by passing as the first argument a JSON format string containing an array of objects. Each object describes a module built inside the selected project area and contains the following attributes:

- **Name**: the name of the machine's model;
- **Description**: a short description of the machine;
- **Item Id**: a unique identifier for the module, built by concatenating the machine's identifier string (usually composed by 1 to 3 characters) with a numerical index incremented every time machines of the same model are found in the exported area;
- **Location**: an array with two positions, indicating the X and Y coordinates of the machine position in the 3D environment. They are expressed as integer numbers, in millimeters and following the project coordinate system;
- **Rotation**: an integer number describing the rotation of the module about the Z (vertical) axis. It is expressed in degrees;
- **Size**: an array with two positions, indicating the X and Y size of the machine bounding box. They are expressed as integer numbers, in millimeters and in the local coordinate system of the machine;
- **Image Path**: the full path of the image file containing the schematic representation of the module as seen from above.

These objects are natively used by the script to draw the production line schematic over a canvas using the `ReportLab` and `Pillow` libraries, calculate the relative distances between the modules

```
MACHINES = []

def main():
    if len(sys.argv) < 2:
        print("Error: a valid JSON must be provided as first argument!")
    machines_json = sys.argv[1]

    machines_json = machines_json.replace("'", "\"")
    if len(machines_json) == 0:
        print("Warning: empty machine list received!")

    for machine_json in json.loads(machines_json):
        machine = Machine.from_json(machine_json)

        # Conversion from Unreal Engine coordinate system
        machine.location.y = -machine.location.y
        machine.rotation = -machine.rotation

        [...]

        MACHINES.append(machine)

    # Add custom logic here...


if __name__ == "__main__":
    main()
```

Figure 3.16: A code snippet of the `main()` function of the `layout.py` script.

and save it as a PDF file. The code provided by the script can be used as a guideline for the elaboration of the received data and the addition of custom functionalities. An example of JSON data string received by the script, representing three different modules, is provided in Figure 3.17; notice how the different string fields are delimited by single quotes and not double quotes (as per the JSON standard): this is done to facilitate sending the string as an argument using the Windows command prompt, therefore it is crucial to convert it back to valid JSON format before parsing it inside the Python script.

3.3.4 Sharing Project Save Files

As already mentioned in section 3.2.10, the layout of a project can be saved for later use and further modification using the ”Save project” function. When a project is saved, all the data related to its content are stored into a save file named after the name of the project, using an encoding specific to Unreal Engine. An example of a save file name is `ProjectName.sav` and, assuming that the application has been deployed using the provided installer, the save files for every HoloFactory project can be found in the folder located at:

`C:\Program Files\HoloFactory\Saved\SaveGames`

It is possible to copy and paste these files from one installation of HoloFactory to a different one, in order to share the same project across multiple workstations. This can be helpful for transferring project files to customers and importing them back for reviewing possible modifications. It also offers the possibility to store and backup project files associated with specific production lines after their commissioning, thus allowing to maintain a virtual duplicate of the installed line which can be useful for future developments and maintenance purposes.

```
[
    {
        'name': 'Loader Single Rack',
        'description': 'Automation module for loading PCB panels from
            racks into Vertical test systems.',
        'itemId': 'C1',
        'location': [ -10118, 8302 ],
        'rotation': 180,
        'size': [2063, 1553],
        'imagePath': 'C:/Program Files/HoloFactory/Images/Loader Single
            Rack V_capture-top.png'
    },
    {
        'name': 'Pilot V8 Next',
        'description': 'Automatic Vertical PCB panels tester.',
        'itemId': 'A1',
        'location': [ -6378, 8282 ],
        'rotation': 180,
        'size': [1786, 1115],
        'imagePath': 'C:/Program Files/HoloFactory/Images/Pilot V8
            Next_capture-top.png'
    },
    {
        'name': 'Unloader 4 Rack V',
        'description': 'Automation module for unloading PCB panels from
            Vertical test systems into racks.',
        'itemId': 'C2',
        'location': [ -3018, 7334 ],
        'rotation': 0,
        'size': [ 1303, 2703 ],
        'imagePath': 'C:/Program Files/HoloFactory/Images/Unloader 4 Rack
            V_capture-top.png'
    }
]
```


Figure 3.17: An example of JSON data received by the `layout.py` script.

3.3.5 Desktop Version

In order to favour a wider adoption of the application, a desktop version which does not rely on the use of Virtual Reality hardware, has also been developed parallel to the main version of HoloFactory. Thanks to how Unreal Engine projects are structured and to the integration of the well-established Enhanced Input plugin, it is possible for both versions to coexist inside the same project repository, thus requiring minimal code and asset duplication and allowing the core logic to be independent of the targeted hardware.

As a result, the usage and workflow of the desktop version are exactly the same as the VR one. Obvious adjustments had to be made to the input methods for moving around the scene and interacting with the various components. Since the desktop version requires a mouse and keyboard input method, the implemented movement technique follows the standard inherited by the gaming industry; specifically the arrow keys and/or the W-A-S-D keys can be used for moving around the scene, while the movement of the mouse determines the rotation of the view camera. A complete list of the actions that can be performed and their key bindings are provided in the table 3.1 (the "R" and "L" notation in the VR binding column identifies which motion controller the button belongs to).

Action	VR binding	Desktop binding
Movement and interaction		
Camera rotation	Head rotation	Mouse movement
Move forward	User's movement	W / Up Arrow Key
Move backward	User's movement	S / Down Arrow Key
Move left	User's movement	A / Left Arrow Key
Move right	User's movement	D / Right Arrow Key
Toggle user menu	Menu button (R/L)	Tab
Select/interact	Trigger (R)	Left Mouse Button
Toggle selection pointer	Trackpad button (R)	Z
Turn left (45 degrees)	Trackpad Left button (L)	Q
Turn right (45 degrees)	Trackpad Right button (L)	E
Teleport	Trackpad Up/Down button (L)	Right Mouse Button
Toggle SteamVR menu	System button (R/L)	-
Enter build mode (hotkey)	Grip button (L)	B
Build mode		
Place module	Trigger (R)	Left Mouse Button
Rotate module left	Trackpad Left button (R)	Mouse Wheel Up
Rotate module right	Trackpad Right button (R)	Mouse Wheel Down
Next module	Trackpad Up button (R)	R
Previous module	Trackpad Down button (R)	L
Cycle anchor position	Grip button (R)	Mouse Wheel Click
Exit build mode	Grip button (L)	B

Table 3.1: Table of actions and key bindings.

Although this project heavily focuses on the benefits of VR technology for planning operations, many of which are lost in the desktop version of the application, there are still several trade-offs that make the desktop version useful in different scenarios. The first and most apparent one is that it allows access to the HoloFactory application to a wider range of users inside the company. Most companies like SEICA may only have access to a few Virtual Reality capable systems[3], while it may be useful for several people inside the company to leverage the use of HoloFactory as a fast prototyping tool for virtually assembling a production line and automatically generating the related layout document - both of which are functions completely independent of the means through which the application is used. Not having to provide the minimum refresh rate and screen resolution needed for a smooth VR experience means that the desktop version features far lower system requirements, again enabling a wider range of workstations to run the application, and also giving the possibility of enhancing graphic quality and visual effects without the strict constraint of having to provide a minimum refresh rate of 90 FPS. Furthermore, project save files created by the desktop version are fully compatible with the main VR application, meaning that an approach that splits the workflow between the two is possible. As an example, it is possible to build a prototype layout using the desktop version and then export its saved file to the Virtual Reality version for a first-person analysis of those metrics such as ergonomics and visibility that can not be easily evaluated using traditional non-immersive means.

[3]Virtual Reality capable system refers to a setup consisting of an HMD device, motion tracking tools, a large enough space to ensure comfortable movement, and a fairly high-end workstation capable of running VR applications smoothly.

It is also worth noting that the availability of a desktop version has proved greatly beneficial during the development phase, giving the opportunity to implement new features and test most of them without the need for the HMD device, aside that for final validation.

From the first tests of both versions of HoloFactory, a trend emerged in the speed at which different categories of users became comfortable with using the two versions of the application. In particular, the time required for acquiring a basic understanding of the controls of the application (sufficient for its usage and build of basic line configurations) was similar among the entire user base for the Virtual Reality version using the HMD device. In contrast, when testing the desktop version, younger users showed far superior speed in learning the basic interactions, as compared to the older audience. This is largely due to the game-like approach and control scheme used by the desktop version, which proves to be very easy to learn for people familiar with computer games.

Chapter 4

Conclusions

This concluding chapter aims to analyze the contribution offered by the use of Virtual Reality, and particularly the tools provided by HoloFactory, within the design, manufacturing, and maintenance cycle of an industrial production line, listing the opportunities for its employment and the potential benefits that can derive from it. At the end, a brief mention is also made of the possible future uses and developments of this project.

4.1 Outcomes and Use-Case

Although HoloFactory is still in the prototype stage of its development, the analysis of how the tools it provides can be integrated within SEICA's production cycle has allowed an assessment of the potential benefits of its adoption at different stages of the productive process of a generic CMS.

In particular, the manufacturing cycle of an industrial production line, consisting of several independent machines and material transport modules, can be divided into three phases. The first phase, the pre-sales phase, consists of collecting detailed specifications from the customer in order to be able to propose a solution best suited to their needs. This phase generally ends with the creation of a commercial offer, which is then followed by the beginning of the second phase, the one of development and construction of the systems that form the production line. This phase consists of several steps and covers all the activities ranging from the initial specification analysis and design to the construction and installation of the line at the customer's plant. This phase concludes with the installation and commissioning of the line, and the after-sales and support phase begins, in which the company undertakes to complete staff training activities and provide support for maintenance or any requested modification of the systems that make up the line.

In the following sections it will be shown in detail how the use of Virtual Reality, in particular through the HoloFactory application, can bring a positive contribution in each of the three phases of the life cycle of a production line.

4.1.1 Phase 1: Pre-Sales

During the pre-sales phase of a production system, the customer's ability to understand and visualize the characteristics of the various machine models offered by the company is of fundamental importance. It is, therefore, crucial to be able to immediately convey and make potential customers aware of the entire range of systems offered in the company's catalog, making the qualities of each one clear, so that it is easy for customers to imagine a solution suited to their needs.

The use of Virtual Reality, and in particular of the HoloFactory application in this circumstance, can be very effective in order to present the company's catalog of systems to the customer in an immersive and intuitive way, allowing for a 360-degree analysis of the product. HoloFactory has been specifically designed to support the sales personnel when showcasing the different products offered by the company: the Home scene within which the user is projected when the application is launched is, in fact, designed as a virtual showroom. Here, the various systems in the SEICA catalog are displayed individually, and each of them is supplemented by an informational kiosk on which the user can find an in-depth description of the module's functionalities - just as if they were inside a stand at a trade fair.

Trade fairs represent a key environment for introducing customers to the potential of the systems offered by the company; however, the stands dedicated to companies have limited space and, given the size of the systems, it is often not possible to exhibit more than two or three of them. By organizing a dedicated area for the use of a Virtual Reality system, HoloFactory can be used as a virtual extension of the exhibition space. The mirroring functionality offered by the headset makes it possible to display what the user sees on an external monitor or video device; in this way, it becomes possible for the sales staff to use the VR system in order to show certain systems or features to the interested customers, who are able to follow the visit from the outside by looking at the external monitor. On the other hand, it is also possible for the customer to directly use the headset and explore the virtual environment while being instructed and supported by the expert staff who share their view-perspective from the outside. Compared to what is possible with traditional 2D video supports, the user can have a more in-depth knowledge of the systems and fully appreciate critical aspects such as size, ergonomics, and compatibility between the different modules.

The showroom area inside HoloFactory's Home is fully customizable according to the needs and possible evolutions of the products offered by the company. The number of systems on display and their positioning can be easily adjusted. Even the informative content displayed on the totems next to each system can be widely extended and customized, supporting not only the addition of text descriptions but also the display of images and videos.

Also in the field of pre-sales, once the end customer has acquired a good knowledge of the company's product catalog, HoloFactory can be used as a tool to configure a virtual line prototype to suit the customer's needs. Thanks to the simplicity and smoothness of the movement and construction methods, as well as the large amount of planning aids integrated into the application (such as snapping between compatible modules and real-time collision control), even users without extensive experience in design or in the use of Virtual Reality tools are able to create a prototype line in a short amount of time. At this stage, it is also of paramount importance to reduce as much as possible the time elapsed between the receipt of specifications from the customer and the production of a preliminary design proposal that can meet them. This operation often takes several weeks since information must be transmitted through different departments of the company. If there are inconsistencies within the specifications received, their recycling and rectification can take several days. In particular, the process leading to the definition of a preliminary layout proposal for a production line can be summarised as follows:

1. The customer acquires a sufficient understanding of the range of systems offered by the company. In this phase, they are supported by the marketing and pre-sales departments;

2. The customer formulates specifications for the realization of a production line. These may include product specifications, targeted cycle times, specific layout, and encumbrance constraints and requirements for the integration with existing production environments;

3. The specifications are handed over to the pre-sales department, which carries out an initial validation and, if they are not clear enough or reveal inaccuracies, cycles back on them from point (2);

4. Once the specifications successfully pass the pre-sales department's analysis, they are forwarded to the Research and Development and Systems Design departments. These conduct a preliminary feasibility assessment and are responsible for producing a line design proposal, with the creation of a layout document. If inconsistencies arise within this phase, it is necessary to cycle back again from step (2);

5. The preliminary layout document is sent to the customer, who may be satisfied with it and thus initiate the request for a commercial offer, or request some adjustments. In the

occurrence of modifications to the customer's specifications, it is necessary to start again from step (2).

The use of a virtual configurator tool such as HoloFactory (either using Virtual Reality or the desktop version) can contribute to a substantial reduction in the time required to complete the above cycle through a reduction in the number of steps required and affected departments within the company. HoloFactory has been designed to integrate rigorous validity checks on the positioning and compatibility of the modules that can be constructed within the virtual project; this ensures that any set of production systems connected to form a line in the virtual environment is also viable and possible to realize in real space. In this way, it is possible to automate almost the entire verification process, normally entrusted to the Research and Development and Systems Design departments (step number 4). Furthermore, thanks to the virtual project export function, through the generation of a layout document in PDF format, it is possible to eliminate the need to use specialized design software (usually CAD design tools employed by mechanical designers from the Systems Design department) for the production of the preliminary layout document. With this approach, the operations carried out within step (4) are thereby automated and made available also to less specialized personnel, making the direct involvement of certain company departments no longer mandatory, with benefits in terms of the time needed to submit a preliminary design. Should the end customer then have the opportunity to install and autonomously utilize a version of HoloFactory, this would further reduce the number of re-cycles within the specifications' acquisition phase. Once instructed on the basic operation of the application, the customer would be able to independently create several iterations of possible layouts for a production area that reflects his requirements by finding information about the characteristics of each module and the compatibility between them directly within the application. This would also minimize the number of re-cycles between steps number (2) and (3), as the customer would be made independent in the prototyping phase of his virtual line, enabling them only to consult the pre-sales department for a final validation once they are satisfied with the result obtained through the HoloFactory application.

In conclusion, it can be stated that the use of Virtual Reality, and in particular the functionalities offered by the HoloFactory application, makes it possible to share information regarding the systems offered by the company in a detailed and highly immersive manner compared to the use of traditional methods. Features such as the user-friendliness and simplicity of the controls also help to make the task of planning the layout of a production area available to a wider and less specialized public, thereby minimizing the time that elapses from the availability of customer's specifications to the submission of a realization proposal.

4.1.2 Phase 2: Design and Development

This is the phase for which most of the functions in HoloFactory are designed. This phase starts after the formalization of a commercial offer for the construction of a production line for a specific customer and is divided into several sub-phases, ending with the final completion and installation of the line at the customer's plant. HoloFactory represents a fundamentally useful tool in the prototyping, design, and finalization stages of the layout of a line and of the systems that are part of it. As already seen in the previous section, HoloFactory can be used, even by personnel not specialized in systems design and engineering, as a convenient way of providing a line prototype in a short time, ensuring its feasibility and consistency. Within the design stage, it is possible to further develop and detail the prototype layout realized during the pre-sales phase through the use of the different optional construction tools made available by the application, so as to precisely calculate distances and positioning of the various modules in space. The possibility of importing customized models into the application is particularly relevant to more thoroughly examine the placement of the line and the visibility and navigation aspects of its surroundings. By adding custom modules (even just simple mock-up boxes of various sizes), it is possible to populate the environment within the virtual project, replicating the characteristics - such as architectural elements, columns, other already existing machinery, passageways, and barriers for operators - of the customer's plant where the line will be located after the commissioning. In this way, it is possible to anticipate a series of spatial and visibility assessments that could highlight potential issues or possible improvements that can be applied to the layout.

Another area in which the use of Virtual Reality as a review and analysis tool is particularly beneficial is during the mechanical design phase of a new system (i.e., of a model not yet included in the company's catalog) or the modification/customization of an existing model to suit a customer's specific requirements. In this phase, It is, therefore, possible to add a new step in which a revision of the CAD project is conducted through the use of HoloFactory, following the completion of the CAD design step. Depending on the outcome of the revision carried out in this step, it is then possible to decide whether to continue with the finalization of the concept and the start of the physical realization, or whether it is necessary to re-cycle from the design phase by introducing changes to the design. The integration of this review step through the use of Virtual Reality aims to allow for assessments regarding the ergonomic aspects of the different components and activities that operators might perform on the system, as well as the visibility of the various elements and indicators from the different viewpoints.

A possible workflow for the introduction of this audit step - using the HoloFactory application - within the design phase of a new system is proposed below. Particular care has been taken during the development of the application to ensure that the operations required in this phase can be undertaken by the personnel who have an interest in carrying out the review - i.e., mainly designers, mechanical engineers, and project managers - in the most independent possible manner, without requiring the intervention of developers and technicians specialized in VR.

1. Creation/modification of the CAD design. In this step, the CAD drawing of the system in question is created or modified (in case of a specific customer's request or a re-cycle following a failed VR validation). The result of this stage is the creation of a STEP file containing the system blueprint;

2. Conversion of the 3D model from CAD to FBX format. In this phase, the model is prepared for being imported into HoloFactory, starting with the conversion to FBX format. This may be accomplished directly within the CAD design tool if this format is supported among the output ones or alternatively with the use of one or more external tools such as FreeCAD and Blender. In this phase, the user can perform a preliminary simplification of the model through the exclusion of design components before the conversion; since different types of machine projects can have significantly different levels of complexity, it is advisable to employ a trials-and-errors approach in order to obtain the proper balance between the performance achieved within the virtual environment and the quality of detail required for the evaluations to be conducted. This approach suggests the use of an ad-hoc HoloFactory project for the evaluation of each design revision, hence minimizing the need for optimization of the 3D model's mesh, as only one module will be present within the scene at a time;

3. Import into HoloFactory. Once the main mesh of the module has been exported in FBX format, it is possible to copy the produced file into a specifically designated subfolder within the imported modules directory

   `C:\Program Files\HoloFactory\Modules`

   It is also necessary to create a JSON file in which the minimum set of characteristics, necessary for the application to import the custom model and to add it to the library, is specified (as explained in detail in the section 3.3.1). This operation is only mandatory for the first revision cycle, as in future reviews it will be sufficient to simply replace the previous FBX model file with the updated version, leaving the related information unchanged;

4. New project creation and module placement. It is now possible to open HoloFactory and create a new project in which the module just added to the catalog can be placed. In the case of subsequent revisions, it may be sufficient to load the project in which the module has been previously placed (provided it has been saved) but, in general, it is preferred to delete the existing project and create a new, clean project, as changing the 3D model associated with the module may result in the new model being misplaced inside a previously saved project;

5. Module inspection. Once the module has been placed in the virtual environment, it is possible to proceed with ergonomic evaluations, by simulating the interactions with the

different system components and the activities performed by the operators. If the analysis reveals critical aspects concerning the design of the system, it is possible to proceed with a revision starting from point (1) of this workflow. Otherwise, if no issues arise from this evaluation stage, it is possible to proceed with the finalization of the design and start the production phase of the concerned system.

It is important to emphasize how the possibility of carrying out ergonomic and spatial evaluations within a virtual environment, therefore before proceeding to the physical realization of the concerned system, may lead to the preventive identification of design faults that are often only detected at advanced stages of the module's construction, leading to a significant reduction in the time and costs required to address them.

4.1.3 Phase 3: Maintenance and Support

Once the installation and initial start-up of the production line in the customer's plant is complete, the last phase of the product life cycle begins, that of providing maintenance and technical support. This phase extends throughout the entire warranty period and often beyond, with the stipulation of support contracts. In this phase, the company's main objectives are:

- To promptly respond to reports of unforeseen failures and malfunctions of the systems that make up the production line. Once manufacturing has started, production lines may have to operate 24 hours a day to fulfill orders received from customers, which is why any malfunction that interrupts the productive cycle must have a timely resolution. Therefore, an emergency management mechanism is needed that can provide assistance quickly even in locations with several time zone differences.
- To provide training courses: once a production line has been delivered and installed, it is necessary to train the customer's personnel who will be responsible for operating and supervising the line.
- To handle modification requests: it is not uncommon for new customer requests to emerge following the commissioning and initial start-up of production on a line, whether they are aimed at improving the integration and usage of systems within the plant environment, or dictated by a change in demand for certain products processed on the line. In both cases, it is in the company's best interest to be able to support and fulfil the customer's requests as best as possible, even at this stage.

The use of Virtual Reality, in particular through the functionalities integrated into HoloFactory, can help to improve the quality of the support provided at this stage. In particular, it is advisable, during the design and realization phase of a production line, to maintain an up-to-date virtual project that replicates its structure within the application; this way it is possible to have a virtual twin of the line available at any time, which can be used as a support for the delivery of training courses, both during the phases prior to the delivery of the systems (therefore before the real line becomes available), and later once the line has been installed at the customer's site and is, therefore, no longer at the disposal of the SEICA's personnel. By having a virtual representation of the line, it is in fact possible to simulate the activities that will have to be carried out by the operators on the real line. Although the modules are represented statically within the HoloFactory environments, it is possible to physically emulate activities such as the loading and unloading sequences of the line, preparation of the machinery for automated production, and routine maintenance of the systems. The training of line personnel can thus be supplemented with the use of Virtual Reality in different ways:

- **First-person**: the final operators of the line receive training in turn, using the Virtual Reality headset first-hand, supervised by experienced personnel who - through the mirroring functionality - guide them from the outside. In this way, it is possible to carry out some stages of personnel training independently of the physical availability of the line, allowing this activity to overlap with construction and installation activities, thus reducing the overall delivery and commissioning times for the line. Thanks to the first-person involvement of the user, this mode guarantees the greatest immersiveness and the most faithful level of simulation possible.

- **Third-person**: the line operators receive training in a group, without wearing the Virtual Reality headset directly, but following a demonstration session via mirroring on an external monitor. A specialized technician from the company controls the visor and simulates the training routines within the virtual project. In this way, more participants can be accommodated in less time, at the expense of their level of immersion within the simulation. This approach is likewise independent of the availability of the physical version of the line.
- **In remote mixed mode**: once the line has been installed at the customer's facility, it is possible to carry out this form of training in which a specialized technician from SEICA guides the operators via a remote audio and video connection, in performing the training routines directly on the real line. Specifically, the technician delivering the training, who is located in the company's workplace, uses the HMD device to operate within the virtual version of the line, performing the training procedures and explaining the steps. By mirroring this on an external screen and sharing it via teleconferencing software (Google Meet, Microsoft Teams, Zoom, etc.), video images of what is simulated in the virtual environment can be broadcast in real-time. At the customer's plant, the operators receiving the training can be connected to the video-conference via computer or mobile devices, and replicate what is being shown to them directly on the real version of the line.

In summary, it is shown how maintaining a Virtual Reality twin of a production area can bring several benefits at this stage, opening up a range of possibilities for delivering training in a more immersive, immersive manner that is faithful to the real experience. Also important are the time benefits of being able to conduct such training activities in contemporary with the manufacturing of the equipment itself, thanks to the exploitation of their virtual counterparts.

Even in the phase of technical support and response to fault reports, the use of Virtual Reality can contribute to greater effectiveness and timeliness in reaching a solution, especially in cases where support must be provided remotely. The automatic test systems produced by SEICA internally contain a large number of electronic components, cabling, drives, and moving elements; therefore, there are various circumstances that may lead to the need for replacement of some of these components, whether dictated by a malfunction or by preventive maintenance of the equipment. It is, therefore, important, especially in cases where an on-site service intervention would be expensive (both in time and economic costs), that the company is able to transmit clear instructions to the plant's technical staff so that they can directly undertake the necessary corrective actions. Thanks to the use of Virtual Reality and HoloFactory, it is possible to have a virtual version of the system on which intervention is required; this makes it easier for a SEICA specialist technician to identify and locate the components requiring replacement. Once possible corrective actions have been identified, the technician can record a short video using the mirroring mode, showing first-hand how to locate the component inside the test equipment and, if possible, simulating the proper repair operations to be performed. This recording can then be shared with the plant's technical crew and used as a reference guide for the intervention on the real system. As previously described for remote mixed-mode training, this walk-through can also be performed via real-time video communication. Once again, the possibility offered by a virtual twin of a system to visualize components accurately and simulate operations on it without having to access it physically is crucial in this scenario.

When designing a line within the HoloFactory application, the 3D models used to represent the various systems usually consist of a simplified version of the complete CAD designs. This simplification is essential to reduce the overall polygon count in the virtual scene and allow the application to maintain a high (above 90 FPS) and stable refresh rate. The first aspect is crucial for automatic test systems, for which the CAD files contain a vast number of elements. In contrast, the second is necessary to guarantee a fluid user experience free of motion sickness effects, even in projects where dozens of systems are visible at the same time. To have access to a model with the maximum level of detail possible, which can be optimally employed to identify and showcase faulty components in the service phase, the proposed approach consists of importing an additional custom module within the HoloFactory library, representing a high-definition version of the system concerned. As seen in section 3.3.1, this can easily be achieved by a developer or advanced user through the add module function. Given its high complexity, the module added in this way is, therefore, not intended to be placed during the creation of a line layout, but can be positioned on its own within a specifically created project and used whenever a detailed representation of the system in question is required.

To make clearer the number and content of projects that this approach suggests to maintain during the development and after installation of a production line, an example of practical application to a real line segment, used as a case study, is presented below. The line under consideration is composed as follows:

- 2 automatic test systems model Pilot VX NEXT, positioned so that they can work in parallel with each other;
- 1 automatic test and programming system model Compact SL NEXT, positioned at the end of the segment, sequentially to the previous systems;
- 7 automation modules, serving the function of loading and transporting products (in this case PCBs) between one system and another. The distinction between the various models and functionalities of the transport modules is not relevant in this analysis.

The following projects must be maintained within the HoloFactory application to ensure that a sufficient set of virtual resources is available to exhaustively address technical support and training requests for the line described:

1. Complete layout project of the line: this is the main project, which should already be available since the prototyping stages prior to the start of development. It should be maintained and updated with any changes that may be necessary during the construction phases of the line. It can be used during the final stages of the manufacturing and after installation to provide general training on the use of the line to the operators working on it;
2. Pilot VX NEXT detail project: a project in which only a highly detailed model of the system in question is present. The model may either already be present in the catalog or specifically imported if necessary, for example in the event that customizations to the standard model have been implemented in order to better meet the customer's requirements. This project is used to provide specialized in-depth assistance on this particular test equipment model;
3. Compact SL NEXT detail project: a project in which only a high level of detail version of the system in question is present, in the same way as specified in point (2).

Details concerning individual automation modules have not been considered in this analysis, as these machines are generally simpler than test systems. For these modules, it is often possible to use a sufficiently detailed 3D model even for the design and layout phase of the entire line, therefore removing the need for employment of the particular approach described.

In conclusion, it can be stated that, although the main objective of the HoloFactory application is to facilitate the design phase of a production line, some of its functionalities make it a valid tool for the use of Virtual Reality as a support in the maintenance and training phase. Particularly useful in this phase is the possibility offered by the application to save, maintain, and share a library of virtual projects, which can be used as a support to carry out after-sales activities with a greater level of detail and immersion than using traditional methods and even in the absence of the availability of real systems.

4.2 Future Works

Several possibilities open up when it comes to the future application and possible developments of the HoloFactory project.

The main future goal is its integration within SEICA systems' production cycle, which can take place starting from two of the stages mentioned above. In the pre-sales phase, it can be adopted as a tool to better publicize the range of the offer and let customers experience the flexibility of the solutions proposed by the company. This objective can be achieved by introducing the use of HoloFactory as multimedia support during the company's participation in trade fairs. Such usage, which seeks to target as broad an audience as possible, will be able to provide insights and suggestions to further improve the ease of use of the application. The second phase where the adoption of HoloFactory can be consolidated quickly is in prototyping and refining the layout of new production lines. Through the roll-out of this employment, in addition to measuring the benefits already outlined in previous sections, it will be possible to evaluate the utility of the different tools and building aids provided by the application for planning the layout, possibly expanding their number if necessary.

Another crucial aspect for the manufacturing industry, where production timelines are critical, is a careful estimation of the cycle times of the individual modules contained within a line. During the design stage, this allows an understanding of the proper number of modules of different kinds to be included in the line to meet the throughput required by the customer, avoiding the occurrence of bottlenecks. This opens up the possibility of extending the capabilities of HoloFactory to associate each module with a cycle time (which can be dynamically adjusted according to the type of production running on the line), intending to simulate in real-time the line's production speed and throughput and give the user a dynamic indication of how they vary according to the structural changes made by adding or removing modules.

One more aspect that could be explored further is the amount of features from the real systems that can be represented within the simulated environment. In particular, the addition of spatialized audio sources to the various modules in the catalog could open up the possibility of conducting acoustic evaluations, such as assessing the perceived amount of noise from the different workstations on the line or estimating the distance at which alarms and buzzers, used by autonomous machines in case of emergency or need for operator intervention, can be heard and distinguished. Another detail that could be further developed is the implementation of movable and interactive parts - such as doors, drawers, buttons, and racks - within the various modules to render their use more faithful when simulating training or maintenance operations.

Finally, from a more technical point of view, it is worth mentioning that with the release of the 5.3 and 5.4 updates of Unreal Engine (this project uses version 5.2), support for Nanite was also introduced for VR applications [23]. Nanite is a feature offered by Unreal Engine, which aims to enable very complex 3D meshes to be utilized within projects through the combination of particular compression techniques and the automatic generation of LODs. This feature could fit particularly well with HoloFactory, as it would potentially allow the direct import of 3D models with very high polygon count, therefore removing the need to perform mesh simplification and optimization steps before importing them into Unreal Engine and providing a higher level of detail when observing the models from a very close distance within the virtual environment.

Bibliography

[1] Seica, *Automatic test equipment*, Accessed: 2024-06-22, Jul. 2021. [Online]. Available: https://www.seica.com/.

[2] Seica Automation, *Seica automation website*, Accessed: 2024-06-22. [Online]. Available: http://www.seica-automation.it/.

[3] L. Dammacco, R. Carli, V. Lazazzera, M. Fiorentino, and M. Dotoli, "Designing complex manufacturing systems by virtual reality: A novel approach and its application to the virtual commissioning of a production line", *Computers in Industry*, vol. 143, p. 103 761, 2022, ISSN: 0166-3615. DOI: https://doi.org/10.1016/j.compind.2022.103761. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166361522001580.

[4] C. Cimino, E. Negri, and L. Fumagalli, "Review of digital twin applications in manufacturing", *Computers in Industry*, vol. 113, p. 103 130, 2019, ISSN: 0166-3615. DOI: https://doi.org/10.1016/j.compind.2019.103130. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166361519304385.

[5] V. Kuts, T. Otto, T. Tähemaa, and Y. Bondarenko, "Digital twin based synchronised control and simulation of the industrial robotic cell using virtual reality", *Journal of Machine Engineering*, vol. 19, no. 1, pp. 128–144, Feb. 2019. [Online]. Available: https://www.infona.pl/resource/bwmeta1.element.baztech-864b3bb2-f435-4a9e-af97-ab8b3470a8c8.

[6] Epic Games, *Unreal engine 5*, Accessed: 2024-07-01. [Online]. Available: https://www.unrealengine.com/en-US/unreal-engine-5.

[7] Epic Games, *Enhanced input in unreal engine*, Accessed: 2024-07-04. [Online]. Available: https://dev.epicgames.com/documentation/en-us/unreal-engine/enhanced-input-in-unreal-engine.

[8] RianeDev, *RuntimeStaticMeshImporter: Plugin for loading static mesh at runtime from FBX files*, Accessed: 2024-07-10. [Online]. Available: https://github.com/RianeDev/RuntimeStaticMeshImporter.

[9] The Khronos Group Inc, *Openxr overview*, Accessed: 2024-07-04, Dec. 2016. [Online]. Available: https://www.khronos.org/openxr/.

[10] Blender Foundation, *Home of the Blender project*, Accessed: 2024-07-04. [Online]. Available: https://www.blender.org/.

[11] P. AG, *3d formats in the field of engineering - a comparison*, 2011. [Online]. Available: https://isicad.ru/uploads/http___www.prostep.pdf.

[12] FreeCAD Project Association, *Freecad: Your own 3d parametric modeler*, Accessed: 2024-07-04. [Online]. Available: https://www.freecad.org/.

[13] P. Carbonnelle, *PYPL popularity of programming language index*, Accessed: 2024-07-01, 2023. [Online]. Available: https://pypl.github.io/PYPL.html.

[14] A. Robinson and R. Becker, *The ReportLab toolkit*, Accessed: 2024-07-12, Jun. 2024. [Online]. Available: https://pypi.org/project/reportlab/.

[15] J. A. Clark, *Pillow - python imaging library (fork)*, Accessed: 2024-07-12, Jul. 2024. [Online]. Available: https://pypi.org/project/pillow/.

[16] Nullsoft, *Nsis - main page*, Accessed: 2024-07-12. [Online]. Available: https://nsis.sourceforge.io/Main_Page.

[17] HTC Corporation, *HTC VIVE*, Accessed: 2024-07-12. [Online]. Available: https://www.vive.com/eu/product/vive/.

[18] J. J. LaViola, "A discussion of cybersickness in virtual environments", *SIGCHI Bull.*, vol. 32, no. 1, pp. 47–56, Jan. 2000, ISSN: 0736-6906. DOI: 10.1145/333329.333344. [Online]. Available: https://doi.org/10.1145/333329.333344.

[19] F. Saling, D. Bernhardt, A. Lysek, and M. Smekal, "Diegetic vs. Non-Diegetic GUIs: What do Virtual Reality Players Prefer?", in *ICAT-EGVE 2021 - International Conference on Artificial Reality and Telexistence and Eurographics Symposium on Virtual Environments - Posters and Demos*, J. Maiero, M. Weier, and D. Zielasko, Eds., The Eurographics Association, 2021, ISBN: 978-3-03868-159-5. DOI: 10.2312/egve.20211333. [Online]. Available: https://doi.org/10.2312/egve.20211333.

[20] E. Galati, "Understanding UX in video games: The diegesis theory", *Medium*, Mar. 2022. [Online]. Available: https://uxdesign.cc/understanding-ux-in-video-games-diegesis-theory-f59d5a94cbcf.

[21] R. D. Ioris, *glTFRuntime: Unreal engine plugin for loading glTF files at runtime*, Accessed: 2024-07-10. [Online]. Available: https://github.com/rdeioris/glTFRuntime.

[22] Ambi, *STEPper: STEP file importer for Blender3D*, Accessed: 2024-07-04. [Online]. Available: https://ambient.gumroad.com/l/stepper.

[23] Epic Games, *Unreal engine 5.3 release notes*, Accessed: 2024-07-10, Sep. 2023. [Online]. Available: https://dev.epicgames.com/documentation/en-us/unreal-engine/unreal-engine-5.3-release-notes.