LINA: a virtual rea*L*Ity user *IN*ter*A*ction framework for intelligent greenhouses

Stavros Dagkalakis

Thesis submitted in partial fulfillment of the requirements for the Masters' of Science degree in Computer Science and Engineering University of Crete School of Sciences and Engineering Computer Science Department Voutes University Campus, 700 13 Heraklion, Crete, Greece

Thesis Advisor: **Constantine Stephanidis**

This work has been performed at the University of Crete, School of Sciences and Engineering, Computer Science Department.
The work has been supported by the Foundation for Research and Technology - Hellas (FORTH), Institute of Computer Science (ICS).

University of Crete
Computer Science Department

LINA: a virtual rea***L***Ity user ***IN***ter***A***ction framework for intelligent greenhouses

Thesis submitted by
Stavros Dagkalakis
In partial fulfillment of the requirements for the
Masters' of Science degree in Computer Science

THESIS APPROVAL

Author: ____________________________
Stavros Dagkalakis

Committee approvals: ____________________________
Constantine Stephanidis
Professor, Thesis Supervisor

____
George Papagiannakis
Assistant Professor, Committee Member

M. Antona
____
Margherita Antona
Principal Researcher, Committee Member

Departmental approval: ____________________________
Antonis Argyros
Professor, Director of Graduate Studies

Heraklion, August 2019

Abstract

Intelligent Environments (IE) are spaces with embedded systems and information and communication technologies (ICTs) that bring computation into the physical world in order to enhance occupants' experiences and activities. Recent advances in ICTs have resulted in new ways for remote management and monitoring capabilities of Intelligent Environments in almost all domains of everyday life.

One such novel approach is through the use of Virtual Reality (VR) technologies. In this context, Intelligent Environments (IE) can be simulated by a computer-generated world allowing users to be fully immersed. Thus, users can freely interact with the artificial environment while, at the same time, their actions have an immediate effect in the real world. This hybrid approach creates new opportunities for alternative means of natural interaction, as well as new interaction challenges.

The purpose of this work is to design and implement a technological framework to support user interaction for remote management and monitoring of Intelligent Greenhouses using VR technologies. The technological framework provides users with personalized interaction techniques configuration and adaptive user interface elements within Virtual Reality environments. In this way, users will have seamless access to real-time information in an unobtrusive manner, as well as intuitive control over the actuators of the Intelligent Greenhouse. The Intelligent Greenhouse contains plethora of different sensors used to measure environmental parameters and actuators arranged to build a controlled environment, thus highlighting the contribution of the proposed framework for creating seamless, simple and intuitive to use VR techniques for remote management and monitoring of Intelligent Greenhouses.

The aforementioned technological framework has been deployed within the Intelligent Greenhouse simulation space of the ICS-FORTH's Ambient Intelligence Facility, and two rounds of heuristic evaluation have been conducted in order to identify potential usability barriers prior to its practical exploitation.

Keywords: Virtual Reality, Intelligent Environments, Interaction within Virtual Environments, Intelligent Greenhouse, Remote Management and Monitoring, Gamification.

ΛΙΝΑ: ένα τεχνολογικό πλαίσιο αλληλεπίδρασης εικονικής πραγματικότητας με έξυπνα θερμοκήπια

Περίληψη

Τα έξυπνα περιβάλλοντα είναι χώροι εξοπλισμένοι με Τεχνολογίες Πληροφορικής και Επικοινωνιών (ΤΠΕ), που προσθέτουν διάχυτη υπολογιστική ικανότητα σε χώρους με σκοπό να επαυξήσουν τις δυνατότητες και τις εμπειρίες που μπορεί να προσφέρει. Παράλληλα η πρόοδος των ΤΠΕ, έχουν οδηγήσει σε νέους τρόπους απομακρυσμένης παρακολούθησης και διαχείρισης έξυπνων περιβαλλόντων, σε όλους σχεδόν τους τομείς της καθημερινής ζωής.

Ένας τέτοιος νέος τρόπος είναι μέσω της χρήσης τεχνολογιών Εικονικής Πραγματικότητας, όπου χρησιμοποιούνται εικονικά περιβάλλοντα τα οποία δημιουργούνται από ένα υπολογιστή, για να αντικατοπτρίσουν τα έξυπνα περιβάλλοντα και να ενβυθίσουν τον χρήστη σε μια εικονική εμπειρία. Με τον τρόπο αυτό, ο χρήστης μπορεί να αλληλοεπιδρά με το εικονικό περιβάλλον, κάνοντας τους χειρισμούς που επιθυμεί και παράλληλα οι χειρισμοί που κάνει να μεταφέρονται στο έξυπνο θερμοκήπιο, πίσω στον πραγματικό κόσμο. Αυτός ο νέος τρόπος αλληλεπίδρασης δημιούργει ένα φυσικό μέσο αλληλεπίδρασης, ενώ παράλληλα δημιουργεί νέες προκλήσεις.

Στόχος τη παρούσας εργασίας είναι ο σχεδιασμός και η υλοποίηση ενός τεχνολογικού πλαισίου το οποίο θα παρέχει στο χρήστη τη δυνατότητα να επιτηρεί και να διαχειρίζεται ένα έξυπνο θερμοκήπιο με τη χρήση τεχνολογιών εικονικής πραγματικότητας. Το τεχνολογικό πλαίσιο παρέχει στο χρήστη εξατομικευμένες και προσαρμοζόμενες τεχνικές αλληλεπίδρασης με έξυπνες διεπαφές μέσα σε ένα εικονικό περιβάλλον. Με τον τρόπο αυτόν ο χρήστης μπορεί να έχει πρόσβαση στη πληροφορία που τον ενδιαφέρει, εύκολα, φυσικά και σε πραγματικό χρόνο, καθώς και στο χειρισμό των εγκατεστημένων συστημάτων του έξυπνου θερμοκηπίου. Το έξυπνο θερμοκήπιο εμπεριέχει πολλούς και ποικίλους αισθητήρες και ενεργοποιητές που χρησιμοποιούνται για να δημιουργήσουν ένα ελεγχόμενο περιβάλλον, αναδεικνύοντας την καταλληλόλητα του προτεινόμενου τεχνολογικού πλαισίου για την απομακρυσμένη επιτήρηση και διαχείρισή του μέσω ενός συστήματος εικονικής πραγματικότητας.

Το προαναφερθέν τεχνολογικό πλαίσιο αναπτύχθηκε και συνδυάστηκε με το χώρο προσομοίωσης του έξυπνου θερμοκηπίου του κτιρίου Διάχυτης Νοημοσύνης του Ινστιτούτου Πληροφορικής του ΙΤΕ, όπου διεξάχθηκαν δύο κύκλοι ευρετικών αξιολογήσεων (heuristic evaluation) προκειμένου να εντοπιστούν πιθανά εμπόδια στη λειτουργικότητα του συστήματος.

Λέξεις κλειδιά: Εικονική Πραγματικότητα, Έξυπνα Περιβάλλοντα, Αλληλεπίδραση Χρηστών με Εικονικά Περιβάλλοντα, Έξυπνο Θερμοκήπιο, Απομακρυσμένη Διαχείριση και Παρακολούθηση, Gamification.

Acknowledgments

I would first like to thank my thesis advisor Professor Constantine Stephanidis, who gave me the opportunity to become a member of the Human Computer Interaction (HCI) Laboratory FORTH - ICS and provided me with the necessary technological means and resources that facilitated my research towards this thesis.

I would also like to thank Dr. Emmanouil Zidianakis, who guided me during the design and implementation process and was always eager to assist with any troubling matters.

I would also like to thank the experts who were involved in the design and evaluation phases for this thesis: Ilia Adami, Chryssi Birliraki, Manousos Bouloukakis, Stavroula Doa, Danai Ioannidi and Vasileios Kouroumalis. Without their participation and input, the development and the heuristic evaluation of the proposed framework could not have been completed.

Moreover, I would like to acknowledge Professor George Papagiannakis and Dr. Margherita Antona members of the Human - Computer Interaction Laboratory of ICS-FORTH for their very valuable comments on this thesis.

Finally, I must express my very profound gratitude to my family and friends for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Stavros Dagkalakis

Table of Contents

Table of Figures

1 Introduction

The purpose of this work is to design and implement a technological framework to support user interaction for remote management and monitoring of Intelligent Greenhouses using VR technologies.

Intelligent Environments (IEs) are spaces with embedded systems and communication technologies (ICTs) that bring computation into the physical world in order to enhance occupants' experiences and activities [7]. Recent advances in ICTs have resulted in new ways for remote management and monitoring capabilities of IEs in various domains, such as business, health care, environmental monitoring, etc., through applications created for this purpose.

A field where the introduction of IEs technologies greatly enhances management is agriculture. In this field, the technological advancements lead in reduction of cost (profitability), time, and resources required for crops cultivation while improving productivity and crop quality [65]. In this context, an area of increasing interest is the application of IEs technologies in agricultural greenhouse farming. The greenhouse is designed for the protection of tender or out-of-season plants by providing a controlled environment, adapted to the needs of particular plants. It allows the growing of crops independently of the outside climate, since its interior temperature and humidity can be controlled. Greenhouses vary in size and complexity from small home or hobby structures to large commercial units covering an acre or more of land. Large greenhouses are important in agriculture and horticulture and for botanical science, while hobbyists, collectors, and home gardeners commonly use smaller structures. Greenhouses are equipped with automation systems capable of regulating the growing environment by controlling systems for irrigation, ventilation, shading, cooling, heating, misting, nutrient dosing and pest control. The use of IE's technologies in building the greenhouse allows the creation of a self-regulating, microclimate, suitable for plant growth using sensors, actuators, and monitoring and control systems that optimise growth conditions and automate the growing process.

In this work an Intelligent Greenhouse (IGH) is defined as the product of combining traditional greenhouse technology with intelligent environment technologies such as sensors, actuators and communication technologies. IGHs enable users (e.g. farmers and agronomists) to review (monitor) contextual information and actuator status, manage resources and operate actuators remotely, which in turn enables efficient time, resources and cost regulation. However, users need to able to effectively interact with IGHs to take advantage of their remote capabilities and remotely monitor, manage and operate them.

Over the last years a paradigm shift is taking place with systems providing mobile GUI applications and smart device Interfaces. As the industry is exploring new platforms and alternative means of interaction, opportunities are presented for progress in the field of agriculture and IGHs and the means through which the users interact with them.

In the search, for alternative platforms to exploit this paradigm shift and provide interaction with IGHs, Virtual Reality appears to be of great interest. Virtual Reality technologies have reached the productivity plateau as defined by Gartner's Hype curve [44], and the industry produces cheaper and more advanced platforms for developers to take advantage of for their applications. These platforms reduce the negative side effects of virtual reality systems (i.e. nausea, motion sickness etc.) created by older hardware [62] and enhance immersion by captivating the user in a 3D Virtual Environment they can explore and interact.

Virtual Environments (VEs) are artificial computer generated 3D worlds interacting in a natural and unobtrusive manner by using special equipment such as a head- mounted display, controllers, gloves or gestures [13]. Virtual Reality (VR) is the experience of full immersion in a 3D world. Designers create VR experiences transporting users to 3D environments where they freely move and interact to perform predetermined tasks and attain goals e.g. crop growing.

This thesis work integrates Virtual Environments with IGH environments by taking advantage of VR interaction techniques offering users an interesting, immersive and natural way to interact with their IGH.

The LINA framework provides users with an automatically generated virtual environment. An Application Programming Interface (API) in real time connects the generated VE to the IGH to provide real-time a) contextual information about the IGH, using data of the installed sensors and b) actuator operation through the integrated actuators.

The primary user groups of the LINA framework are, a) the "*Owner/farmer*", that might own or cultivate the greenhouse*,* b) the "*Agronomist or scientific consultant*", whose role is to supervise the crop growing, provide instructions, and handling issues related to the IGH, and c) "*Guest",* users that could visit the IGH for purposes other than crop growing e.g. learning.

The LINA framework will enable the users to effectively monitor, manage and operate their IGH using the generated VE. Through the interaction with the VE the users will be able to acquire information about cultivation parameters of the IGH, such as, air temperature, soil moisture, etc. as well as, information on the installed actuator status (e.g. window is open). The users will be also able to alter the state of the installed actuators through their interaction to affect cultivation parameters of the IGH. The Virtual Greenhouse (VGH) was separated in cultivation zones to allow for easier categorisation of the provided information and to match the FORTH's IGH architecture, which was used as case study for this work. This will enable the user to a) select the zone they wish to review, b) move to its location, c) review the information provided by the API using the implemented 3D GUIS and d) if deemed necessary alter the state of the installed actuators for the selected zone to regulate the cultivation parameters.

The users will interact with the generated VE by means of interaction techniques implemented as part of this work and their actions will enable them to monitor, manage and operate their IGH. The selected interaction techniques enable the user to, a) *navigate* the virtual environment, b) *select and manipulate* the aspects of the environment (e.g. 3D GUIs), and c) *control the LINA* framework (system control). These are the three main tasks a user attempts to complete, when using a virtual reality system, as defined by Bowman [11, 12]. These tasks are mapped as follows for the users of the LINA framework.

All user groups need to *navigate* the generated virtual environment in order to explore it, identify areas of interest and visit them. The farmer and agronomist users need to be able to effectively move and quickly identify and reach areas that require attention since they are responsible for the wellbeing of the Intelligent Greenhouse (IGH). The guest groups needs to be able to explore the environment and reach points of interests to interact with the provided information.

For enabling navigation, the implemented interaction techniques are *a*) a *point* and *click* to move technique using ray-casting, *b*) a thumb stick based technique using controllers thumb sticks and c) a *fast travel* choose to select and instantly transferred to a location.

However, movement in the virtual environment increases visual- reality sickness, like nausea, motion sickness and disorientation. In this work, nausea is addressed by having the user seated. For addressing the disorientation side effect created by *point* and *click to move* and *fast travel* techniques, the "blink" mechanism was designed and implemented.

For the *interaction/manipulation* of the virtual environment elements, the selected interaction technique is *ray-casting* (i.e. point to interact) which allows the user using the controller to point and interact with objects and *manipulate* them using the index trigger.

Finally, in order to enable the user to have access in functionality beyond the scope of monitoring, managing, operating the IGH and provide them with *system control* over the LINA framework, a menu was implemented that is allowing to control the system by using a combination of a gesture-based selection menu and ray-casting 3D-GUI interaction.

This gesture-based selection menu was developed to allow direct access to the system controls. The grip trigger activates a menu option; rotation of user's wrist (clock wise or counter-clock wise) highlights the selected menu element and the release of the trigger activates the element. Upon activating, the users can interact with the activated 3D- GUI element using the ray-cast technique.

In this work, 3D- GUIs are used as means to enable the users to, a) *access* the information retrieved by IGH's API and b) *manipulate* (alter the state of) the installed actuator as the user interacts with the 3D GUIs which are spread in the dynamically generated environment. The users require this information in order to ensure the cultivation parameters are optimal for each crop. For example, soil humidity parameters can be regulated by activating remotely the irrigation system. 3D-GUIs are also used in system control to enable the user to, a) access their dashboard from where they can quickly review and enter different virtual greenhouses, b) access the fast travel option and move to their desired location(s) and c) have access to the system's managing tools (e.g. calendar).

This document is organized as follows. Chapter 2 presents the literature review, definitions of terms, related work and a short discussion on this work's place on Intelligent Environments, Intelligent Greenhouses, Virtual Reality and Gamification.

Chapter 3 defines Lina framework's user groups (UGs), the developed scenarios of use (SoU), the user and interaction requirements elicitation process as well as the description of the high-level system entities. Chapter 4 describes the Implementation of the LINA framework, system's architecture, and framework components as well as how the user interaction issues were addressed by LINA framework. The case study of this work was based on "ICS-FORTH's

Intelligent Greenhouse”, which is presented in Chapter 5, while in Chapter 6, the evaluation process and its findings are presented. Finally, Chapter 7 describes the conclusions and the future work of LINA.

2 Background and Related Work

This chapter contains the theoretical background and relevant literature review of this work, that include the definition of Intelligent Environments, their components, fields of application and remote capabilities. Selected published works, as examples of IE's applications that take advantage of IE's capabilities, as well as the means by which users interact with these systems are also presented. The topic of Virtual Reality, VR interaction and how VR systems are evaluated according to literature, as well as example VR systems for remote monitoring management and operation of IEs, are also presented.
Since the focus of this works is to integrate VE with IGH environments by taking advantage of VR interaction techniques, the advantages and impact of Gamification in engaging the user in the task and introducing the user in the systems controls, also have been reviewed. Other works implementing and/or proposing systems for IGH remote monitoring, management and operation are also presented, and finally the discussion for demonstrating this work's contribution in the field of IGHs.

2.1 Intelligent Environments

There are multiple definitions for the extended idea of ubiquitous computing and its implementation in various environments in order to enhance the provided functionality and the occupant's user experiences. For instance, the term **Ambient intelligence** is used to describe environments embedding a wide variety of technologies (i.e. sensors, automated devices, computational devices) in order to create an ergonomic space for the occupant user [63]. The term **Intelligent Environment** describes an environment that consists of numerous networked controllers, each one of which controls a different aspect of the environment (e.g. automated windows, heat system, kitchen appliances), collaborating in such a way as to create an interactive holistic functionality that enhances occupant's experiences [7]. Finally, the term **Smart environment** promotes the idea of "a physical world that is richly and invisibly interwoven with sensors, actuators, displays, and computational elements, embedded seamlessly in the everyday objects of our lives, and connected through a continuous network" [18]. In this thesis, when referring to these enhanced environments, the term Intelligent Environment (IE) is used.

IEs contain a set of key components that are crucial for the accomplishment of their purpose. These components are: a) the distributed smart devices and appliances installed in the IE, b) an internal network, sensors and actuators, and finally, c) an external network (see Figure 1) [16]. Smart devices and appliances (i.e. smart kitchen appliances, AC unit, lighting system, etc.) are distributed in the IEs, providing their own individual APIs and controls, while sensors provide the IE with contextual information which can be easily categorized. For example, the user can review all sensor's values installed in the living room. Information about location (where), activity (what), time (when), and identity (who) [56] is easily accessible through an internal network which allows the IE to use contextual information and act in an autonomous way or allow manual configuration, so that the system complies with users' preferences. Through an external network, the IE's close network connects with the "outside". These components create a holistic system, however they may come in different forms depending on the type of IE's design; an Intelligent home may provide actuators for lighting regulation, AC, or motorised windows. In an IGH, the installed actuators may be the motorised windows and the irrigation system that will allow the user to regulate the cultivation parameters.

One of the first environments that was enhanced using ambient technologies in order to improve the occupant's quality of life and improve the day to day user experience, while increasing the environments functionality and usability, was the user's home [13]. Apart from this, there are a wide variety of fields that benefit from the introduction of smart devices and the implementation of intelligent environment techniques. For example, health-related applications, such as in hospitals, can increase the efficiency of their services by monitoring patients' health and progress by performing automatic analysis of activities in their rooms. Public transportation could efficiently monitor traffic using GPS-based location estimations and redirect it if necessary, using remotely controlled traffic lights. Education services at universities use smart card technologies to permit access to their estates, and the same technologies could be used to monitor attendance. Emergency services, safety related services such as ambulance and fire brigades could improve their reaction time by using automatically calculated routes to reach their destinations. Lastly, such environment could benefit industry and production, which is the field this work focuses on. These areas can greatly improve their productivity and minimize costs by collecting data during the production phase, evaluating the production process and aiding the personnel to improve their performance, while keeping detailed reports about production [5].

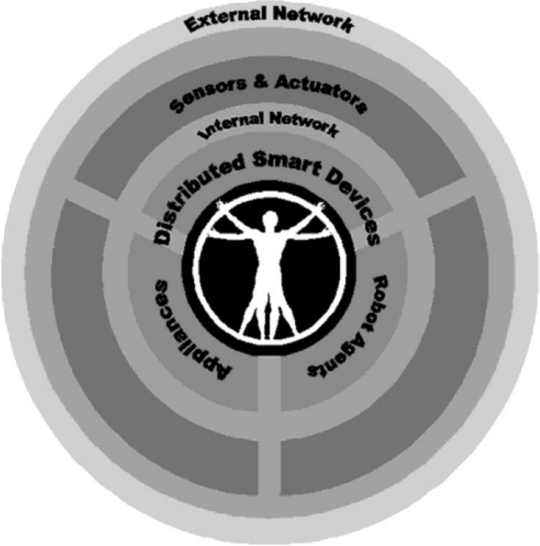


Figure 1: Intelligent Environments [7]

IEs heavily depend on networks (i.e., external/internal network components) and their reliability to function properly. This happens because IEs in order to relay information and issue commands connect to a Control Unit, which is the link between sensors and actuators. Centralized and Distributed Control are two major approaches, when designing controls for IEs [15]. In *centralized control* approaches, a single centralized controller based on global information makes decisions, e.g. the IGH "senses" the temperature does not match ideal cultivation values and activates the fan unit. In *distributed control* models, distributed actuators made the decisions based on local information; the fan unit, changes airflow to alter the room temperature, to match the desired temperature.

When designing IEs, it is important to decide which approach best suits the IE's needs, because different IEs have different needs, requirements, and risk for failure. For example, in Intelligent Cities [8], commands need to be issued quickly and at great distances. Therefore, in this case distributed control units should be deployed in order to avoid delays, minimize failure risk, and avoid creating discomfort to the users. On the other hand, in an intelligent house centralized control appears to be a better option since it removes the need for large computational power in actuators and simplifies management and maintenance by providing a "central" super-agent, which is responsible for the installed sensors and actuators. Lastly, in some cases a mixture of both approaches may be required. This can be due to the fact the scale of the IE is too large for one centralised control unit, but the computational power required for the decision-making is too great. In these cases, one or more "hubs" are created acting as agents for a designated area, while being controlled by a central super-agent, which manages and supervises these hubs. In this work's IGH case study, a centralised hub is used for collecting all the information from sensors and actuators. Subsequently, the hub relays the information to a central processing agent over the cloud to decide the appropriate course of action.

Context is essential when managing IEs and the ability to sense the environmental parameters of the IE is achieved with sensors deployed inside the IE.
Sensors enable the Intelligent Environment to identify the context by providing categorized information such as location, activity, time, identity [67]. An internal network is created between sensors, actuators and control unit(s), thus distributing crucial context information to the interested parties. This ability to sense context and issue commands through a network in conjunction with wireless network technologies has created a growing trend in automation industry. To employ wireless technologies to reduce production cost, provide easy access to context information about devices despite their location and enable wireless control applications, this trend led to the creation of Wireless Sensor and Actuator Networks (WSANs).

WSANs provide an infrastructure capable of supporting the distributed communication needed in highly dynamic environments. For example, an Intelligent Environment-based telemonitoring system for a dependency scenario has been realized, increasing mobility, flexibility, and efficiency, since resources can be accessed regardless their physical location [4, 19]. WASNs when compared to wired networks provide greater flexibility and require less infrastructural support [3]. However, this flexibility comes with a cost. Disadvantages of Wireless networks when compared to wired networks are delay, fairness and robustness [34]. Lastly, WSANs are less open to integration of new devices from different technologies in a single network [17, 41]. Due to the advantages and disadvantages of both technologies when designing Intelligent Environments, most of the times, designers tend to use a combination of both wired and wireless technologies when the field of application allows for it.

In IEs, networks connect everyday life artefacts, enhance their functionality and incorporate them to the concept of Internet of Things (IoTs). IoTs is a worldwide network of uniquely addressable interconnected objects, based on standard communication protocols [46].
Internet of Things (IoTs) for Intelligent Environments can be defined as: "Interconnection of sensing and actuating devices providing the ability to share information across platforms through a unified framework, developing a common operating picture for enabling innovative

applications. This is achieved by seamless ubiquitous sensing, data analytics and information representation with Cloud computing as the unifying framework" [26].

The success of IoT has enabled the creation of more complex Intelligent Environments at a greater scale than ever before, introducing concepts like "Intelligent cities".
As a by-product of population growth and urbanization, many cities' infrastructure and resources are stretched, this has led city councils to a search for means to effectively manage these cities and provide services. To accomplish these tasks cities' councils, employ Intelligent Environments and IoTs' technologies in Transportation, Healthcare, Education, Public safety and security, City administration, Waste management, and Building management [8].

Intelligent Cities are an example for the application of IEs' techniques being applied to a large-scale environment using IoTs. In this context, IEs' techniques enable effective *remote monitoring* (e.g. in transportation the traffic light moderators monitor traffic to avoid congestion), *teleoperation* (e.g. in city administration activates the lampposts at noontime) and resources management (e.g. detect water reserve shortage and issue guidelines for less water consumption by the citizens). IGH implementations using IoT's technologies already exist for a single IGH [69]. However, by applying the centralised hubs techniques proposed in the Intelligent Cities example, the IGH IoT's implementation could be applied to a large number of IGHs. A centralised super-agent containing the information for all the currently cultivated plants and automatically regulating the IGHs could ensure an optimal cultivation outcome.

2.2 IEs remote monitoring, management and operation

Recent advances in ICTs have resulted in new ways for remote management and monitoring capabilities of Intelligent Environments in almost all domains of everyday life. Some examples of related research are:

- *environmental state monitoring*: [38], this paper proposes the use of sensors to remotely monitor runoffs and land erosion while providing accurate predictions to enable measures to be taken in advance [30]. This system provides real-time remote monitoring soil moisture.
- *marine biology:* [83], this paper provides a comprehensive review of the state-of-the-art technologies in the field of marine environment monitoring using wireless sensor networks.
- *health:* [19], this paper presents a distributed telemonitoring system, aimed at improving healthcare and assistance to dependent people at their homes. [23], this paper proposes the use of Wi-Fi technologies to create Remote Patient Monitoring and Body Sensor Networks that will allow managing patients in the community. [1], this study Wearable and Proactive computing, as a means to anticipate the user health state using wearable devices and provide medical care.
- *industry and production:* [16], this paper proposes and evaluates a new distributed estimation and collaborative control scheme for industrial control systems with WSANs. Also discusses how dynamically extend the scale of a WSAN with only local adjustments of sensors and actuators.

- *education:* [82], this work describes the application of Intelligent Environment technologies in a smart Classroom to provide Tele-education capabilities.

These systems highlight the effectiveness of IE's technologies in monitoring and managing different environments. To enable users to act on the provided information systems provide operation capabilities to affect aspects of the IEs through, using actuators or machinery (i.e. in industry). The primary focus of this work, the domain of agriculture, will be discussed, in section "2.5 Intelligent Greenhouses".

"Remote operation" or "Teleoperation" refers to the ability to operate a piece of equipment remotely. In IEs this translates into issuing commands towards the installed actuators (e.g. in an Intelligent Greenhouse operating the motorized windows) and being able to effectively operate the equipment remotely. The system should enable the user to review (monitor) the environmental context and equipment status to allow for optimal resources management and provide the best possible conditions for the operator. Effective teleoperation is intertwined with monitoring, since environmental context is required for precise operation, and management of the system's resources ensures the system's stability and reliability [32]. The first implementation of teleoperation stemmed from the need to operate vehicles in extreme environments where a human operator would not be able to survive. Human performance issues involved in teleoperating unmanned systems generally fall in two categories, i.e., remote perception (telemonitoring) and remote manipulation (resources management/ operation), which includes both navigation and manipulation tasks [32]. In the case of IGHs, teleoperation could refer to altering the state of the actuators remotely using the provided API.

From a control theoretic point of view, the main goals of teleoperation are twofold: *Stability*, maintain stability of the closed-loop system irrespective of the behaviour of the operator or the environment, and *Telepresence,* provide the human operator with a sense of telepresence, with the latter regarded as transparency of the system between the environment and the operator.

Stability is achieved through designing robust systems and implementing safety and fall-back plans for cases of failure (design for failure). Remote management systems tend to use APIs that consume web services in order to provide the user with control and real time interaction with the IEs. In this manner, systems ensure the closed loop between user and environment is maintained, while allowing the user to operate and interact with the IE [73].

Telepresence with respect to stability is harder to achieve, as well as more complex to maintain. Mollen and Wilson (2010) define telepresence as "the psychological state of 'being there' in a computer-mediated environment, augmented by focused attention." [49]. Drawing on these latest descriptions, telepresence is described as the perception of direct product experience simulated through a medium [43, 73].

Telepresence is intertwined with environmental status and context of the Intelligent Environment the user wants to operate. Intelligent Environments use several sensors and actuators strategically distributed in the environment to provide the systems with context-aware capabilities (*monitoring*) and enable changes over the system's state and behaviour automatically (*management*) while providing the user with control for manual configuration over the installed actuators (*operation*) [19]. In an IGH, the context consists of the installed sensors, separated in general sensors measuring the ambient IGH metrics and cultivation zone specific sensors measuring zone specific metrics. By operating the installed actuators, the users can effectively manage the environmental variables to match the desired values.

User interfaces (UIs) associated with teleoperation are designed in order to provide tools enabling the users to perceive the remote environment, make decisions, and generate commands [24]. Although most of the applications provide a traditional Desktop GUIs, a paradigm shift is taking place with systems providing mobile GUI applications along with Desktop GUI, and in some cases smart devices Interfaces (e.g. smart speakers). This shift reaches new frontiers with the introduction of VR technologies, in fields like:

- *health:* [78], this system uses a gamified VR environment, where the goals of the game have been designed in a way as to represent physical exercises (e.g. the user has to visually follow a moving target and by doing this exercises his/her neck.)
- *entertainment:* [57], PIEdeck serves as an experimental environment to explore novel game concepts and other forms of co-located interaction in a hybrid-reality (vr and physical world) context, combining real physical movement with virtual content in a large-scale setting.
- *assisted living:* [29], the main objective of this paper is how to create and to implement VR Smart Home prototype as a platform to increase user contribution in the earliest phases of design.
- agriculture: [37] this work implements a visualisation tool for the purposes of allowing researches and breeders to see, how their plants grow in the greenhouse, compare to previous years and share phenotypic data with their colleagues.

However, the transition from traditional Desktop GUIs to new alternative forms of interaction (e.g. mobile devices apps), in the case of IEs is not a simple task. When designing IEs, a major challenge concerns user interaction, since it is fundamentally different from traditional PC's GUIs. In multimodal interaction in conventional PC applications, the user sits in front of his desktop with one or more screens, a keyboard and mouse. The user interacts with the application using said mouse and keyboard and feedback about is presented through the means provided by the PC (e.g. display monitor, speakers).
On the other hand, in an IE, the user is not assumed to be in front of a desk and the means of interaction extend from the traditional keyboard and mouse to a variety of means such as, gestures, speech, presence etc. In addition, the user can physically interact with devices and artefacts in the IE.

Another challenge, is that the set of devices is dynamic, providing the user with multiple ways of interaction. It is hard to distinguish his/ her focal point, fact that is crucial when deciding where the information the user is interested in is presented. For example, the user requests information for the current ambient temperature of an intelligent greenhouse, the system needs to identify the user's focal point in order to display the requested information. As opposed to traditional PC applications, the focal point was the user's monitor and information was presented there for the user to consume. Finally, an additional challenge is that multiple users can interact with the IE simultaneously [67].

Therefore, when selecting the means of interaction for IEs, it is important to take into consideration the challenges, limitations and opportunities that arise by each "mean of interaction". Moreover, applying human centred design principles results to selection of the best-suited interaction technique for the users to accomplish their desired goals.

2.3 Virtual Reality

Virtual Reality is an artificial computer generated environment, with the purpose of providing the user with a medium for real time interaction with the virtual world. In the case of an IGH, a remote management system will represent the real world greenhouse, which the user attempts to monitor, manage and operate through the interaction with its virtual counterpart. The user interacts with the VE through a combination of immersive hardware (e.g. headsets, controllers, gestures etc.) and software [30, 84]. Designers create VR experiences (e.g. virtual museums, virtual tours, and tutorials) in order to transport the users in 3D environments and worlds where they can freely move and interact to perform predetermined tasks and attain goals (e.g. education, entertainment, remote monitoring and operate) [5, 84].

A practical way to categorise VR technologies is by their purpose: (1) *expression technology* is targeted to stimulating human sensory systems, (2*) interaction technology* aims at interfacing reality with VR, (3) *authoring technology* supports developing VR content, and (4) *collaboration technology* networks multiple participants within VR [17]. This categorization provides insights regarding VR technologies' capabilities.

Virtual reality technologies have already been applied in a variety of fields, due to their immersiveness and interaction through the user's natural movement (e.g. head, hand, etc), which enable the user to be easily engaged in the task at hand. This aspect of VR technologies is very useful when creating applications that require the users' constant focus and engagement, such as educative applications. Research in the domain of education has focussed on the impact of the immersive nature of the VR, how to attract users' attention and help maintaining focus by excluding the real world, along with the distractions that come with it [55, 69, 59]. Also, by exploiting gamification techniques under the label of "serious games" [48] researchers have explored the benefits of VR for training, simulation, and education [9]. Studies regarding systems in military [9, 42] health and doctor training [56, 66] have proved that VR training systems provide efficient, convenient and risk free training sessions. For example, even if a soldier fails the training mission or the trainee fails the OR simulation, they are able to restart the virtual session and try again, something which is not possible in real life. Systems for virtual tours engage the users in an interactive, immersive experience enabling them to visit places of interest like museums [81], oceanariums [25] and other.

Another area of application of VR, which has been extensively explored, is industry automation and the paradigm shift from "real" to "virtual" production, which sparked the interest of researchers since the early 90s. According to [35], systems explored included among others Virtual Manufacturing (VMfg), Virtual Machine Tools (VMT), Virtual Machining (VMach), Virtual Assembly (VA), Virtual Tooling (VTo), and Virtual Prototyping (VP). The above have demonstrated benefits in cost, production time, and productivity in remotely operated industrial equipment and monitoring of the industrial environment. Their work aims to further exploit these benefits by combining VR technologies with IGHs towards improving the quality and quantity of cultivated crops, reducing costs and enabling efficient management of IGHs resources.

Finally yet importantly, a field with very wide impact is entertainment industry, which has produced a wide variety of virtual reality games and applications. In the race to create interesting immersive experiences, game designers are faced with the challenges to create immersive storytelling to intrigue the users and provide user interfaces which are intuitive, fun and require little effort and time for the user to be accustomed with [81]. The gamification and storytelling techniques used by them could be applied in "non-gaming" systems to further enchase user engagement, reduce the learning curve, and create a more pleasant experience for the user [27]. Gaming in general requires accurate and easy to learn controls. For this reason, in this work interaction techniques have been selected which are widely used in the gaming industry (e.g. raycasting). Such techniques have been transferred and configured to match the LINA system's needs.

VR systems are based on 3D representations with complex interaction techniques, usually multimodal, where input and output can be very complex to manage due to the number of potential devices (data gloves, eye trackers, 3D mouse or trackball, force-feedback devices, stereovision, etc.). Designing or implementing VR applications requires addressing several issues such as immersion, 3D visualization, handling of multiple input and output devices, complex dialogue design, etc. Usually, interaction techniques for VR applications are described informally sequentially, presenting the actions the users can perform within the VE and their results in terms of triggered events and modifications of objects appearance and/or location. Most immersive VR systems have developed interaction techniques in an "ad hoc fashion" or "to meet the requirements "of each individual Virtual Environment; i.e. design and development techniques that complement the system and are best suited to complete the system's tasks. Even if such techniques are useful, they need to be evaluated formally [11].

In order to evaluate VR systems, the literature suggests that the evaluation process should focus on a small number of universal tasks such as travel and/or movement, object selection and manipulation [12].

Finally, regarding usability and interaction with the user interface, general guidelines are mainly available. Each system is evaluated by the completion of a number of predefined tasks, which stem from the system provided functionality. These tasks could not be used to evaluate different systems [11, 12].

In an attempt to define user interaction techniques for the universal tasks mentioned above (i.e. navigation, object manipulation, interface interaction), this work adopts the approach by Bowman et al. (2010) for defining the tasks the users attempt to complete when interacting with the VR system [11].

Navigation: The users "need" to be able to *travel* in the virtual environment and the system should provide means for *wayfinding*. These abilities are used to complete three types of tasks, *exploration, search,* and *maneuvering,* which in LINA framework case, covers the users need to explore the VE and reach areas of interest (i.e. cultivation zones). There are five common metaphors for travel interaction techniques, and most published interaction techniques for travel fit into one of these five categories:

- *Physical movement*. The system tracks the user's position and transfers movement in the system. Such techniques are appropriate when an enhanced sense of presence is required or when the application requires the user to experience physical exertion when traveling [11].
- *Manual viewpoint manipulation***.** The system provides the user with the ability to "move towards" a selected point using a system provided trigger (e.g. pull a virtual rope to move towards the direction the rope is pointing at). These techniques can be efficient and easy to learn, but they can also cause fatigue [11].
- *Steering*. Steering is the continuous specification of the direction of motion. This metaphor is primarily used with gaze-direction (moving in the direction the user is looking at) or pointing (moving towards the direction the user is pointing at). Steering techniques are general and efficient [11].
- *Target-based travel***.** The user specifies a destination and the system handles the actual movement. This can be done in many different ways like "teleportation" (user jumps to the desired location) or through a transitional move (the user's avatar move between starting and ending point). Target-based techniques are very simple from the user's point of view [11].
- *Route planning***.** The user specifies the path that should be taken through the environment and the system handles the movement (e.g. the user "draws" the path on a map and the system moves him over the path). These techniques allow the user to control travel while retaining the ability to perform other tasks during motion [11].

Selection and Manipulation: Interaction techniques for 3-D manipulation in virtual reality systems and environments should provide the means to accomplish at least one of three basic tasks: *object selection, object positioning,* and *object rotation*.

Designers tend to use the classical approach, which is to provide the user with a *virtual hand* - a 3D cursor- often shaped like a human hand, whose movements correspond to the movements of the hand tracker. Selection and manipulation of an object involve touching the object, then positioning and orienting the virtual hand within the virtual environment.
However, even though the hand technique is intuitive and easy to use, it limits the user's area of effect to that of his hand reach. To address this problem, a number of solutions have been proposed such as, for example, a nonlinear extension of the virtual hand (the virtual hand extends "further" after a certain threshold).

Another common way to select and manipulate objects is to use the ***"point to interact"*** *- ray casting* -technique, in which the virtual ray emanates from the virtual hand, which is one of the interaction techniques used in this work. This technique has been selected as it removes the need to move closer and "reach" the element the users wants to interact with.
A variation of the ray casting technique is the *spotlight* technique, which uses a cone to select small or distant objects. The *aperture technique* uses a combination of eye-gaze (head

location provided by headset) and hand gestures to control the size and / or distance of the selected object. Another similar technique is the *image plane family of interaction techniques,* which most users are accustomed with due to the mobile phones use [11]. Last, the *World-in-Miniature (WIM) technique* allows the user to indirectly manipulate virtual objects by interacting with their representations in the WIM.

None of the above techniques can be identified as optimal; their performance is task and environment dependent. Because all manipulation techniques have particular strengths and weaknesses, a common practice is to attempt to integrate and combine their best features in order to create the interaction technique that best matches the system. Often, non-realistic techniques have better performance than those based on the real world [11].

System control: When designers refer to system control in a VR system, they mean a task through which the user can change either the state of the system or the mode of interaction. Choosing an element or set of elements from a set, issues this change. In VR systems, control can be provided through *graphical menus* (visual representations of commands), *voice commands* (menus accessed via voice), *gestural interaction* (command sets accessed via gesture)*,* and *tools* (virtual objects with an implicit function or mode) [11]. This work enables system control through a combination of a gesture based selection menu and raycasting on 3D-GUIs (i.e. GUIs placed on 3D space).

Design strategies for designing user interfaces in a 3D virtual world (3D UIs) recommend using *real-world metaphors* to intuitively guide the user to the correct actions *by using physical props or physics-constraints* to lessen precision requirements. In addition, *applying principles of aesthetics and visual design and basing UI design on formal taxonomies* of devices or interaction techniques allow the user to interact with a pleasing interface and transfer knowledge from other devices and interfaces (e.g. desktop GUIs). The magic element in VR allows the user to go beyond the physical world limitation and capabilities and *intentionally violate assumptions* about the real and virtual world (e.g. raycast to interact with an object out of the users reach) [12, 13].

When implementing virtual reality systems, the designer usually has to address hard to tackle issues such as parallelism of actions, actions sequencing or synchronization, fusion of information gathered from different input devices, as well as combination or separation of information (fission mechanism) to be directed to different devices.
These issues make modelling and implementation of VR systems vary, mainly because it is difficult to describe and model how such different input events are connect to the application [50]. Thus, the designer should attempt to tackle these issues and focus on creating systems, which create virtual experiences incorporating features (e.g. images and sound) that introduce the user in a realistic 3D environment. To create such a VE, this work generates a virtual greenhouse (VGH) to provide users with a familiar environment, while distinguishably separates the cultivation zones to allow information categorization (i.e. each zones information is attached over its representation in the VGH), as well as reduce cognitive load.

Such environments should be *interactive* employing intuitive interaction techniques and removing any outside-world interference. Moreover, these environments should be *explorable,* by allowing the user to move freely and discover the “reality” offered. To this end, by combining the above factors, the designer can create an *immersive* experience and transfer the user's presence to the VE [79].

2.4 Gamification

VE designers attempt to attract the users' attention and captivate them in an immersive virtual experience [51]. To this end, VE designers employ meaningful gamification techniques by adding gameful and playful layers to help a user find personal connections that motivate engagement [51].

Gamification has been defined as the process of enhancing services with (motivational) affordances in order to invoke gameful experiences and further behavioural outcomes [27, 33]. Gamification involves three main aspects: a) the *implemented motivational affordances*, b) the *resulting psychological outcomes*, and c) the *further behavioural outcomes* [28].

Motivational affordances. There has been a variety of motivational affordances used in gamification. For categorizing these motivational affordances into larger groups by their purpose, we remove the theme element i.e. using a basketball game for training hand- eye coordination, where basketball is the theme, points are progress reward as the motivational affordance.
Such categorization classifies the different motivational affordances found in various studies into 10 different motivational affordance categories, based on the terminology used: 1) Points, 2) Leaderboards, 3) Achievements / Badges, 4) Levels, 5) Story / Theme, 6) Clear goals, 7) Feedback, 8) Rewards, 9) Progress, 10) Challenge [28]

Resulting psychological and further behavioural outcomes. Studies investigating psychological and behavioural outcomes are focused on aspects such as *motivation*, *attitude* and *enjoyment*. These studies use either evaluative interviews or questionnaires. All of the studies in educational contexts considered the learning outcomes of gamification as mostly positive. Gamification is claimed to increase motivation and engagement in the learning tasks as well as enjoyment [51].

According to the User Centred Design, the designers of gamification systems take into consideration who the players are and determine what outcomes they want to achieve with the system. The designers set the short-term goals of the system (e.g. enchase user experience, attract users' attention, motivate the user to explore the system etc.), and apply user reward-based gamification to achieve these goals, such as a "fun" tutorial on system's controls. However, for long term goals meaningful gamification should be used, as means to assist the user to explore the system and introduce him to system tasks, goals and controls. Meaningful gamification is the use of gameful and playful layers to help a user find personal connections that motivate engagement with a specific context for long-term change. [51].

While both reward-based gamification and meaningful gamification can be tools to get someone engaged in a context, they are only starting points. To create true long-term change, the entire gamification system should be designed to come to an end for an individual player. As the player gets more involved in the system, he or she should be spending more time engaged directly with the "real system" and less time engaged with the gamification system [51].

2.5 Intelligent Greenhouses

Previous work in the domain of IGHS primarily focuses on the remote capabilities of IGHs. Remote management, monitoring and operation of IGHs is currently achieved through traditional Desktop GUI applications, mobile application and IoT's technologies.

The authors of [58] propose environmental monitoring and greenhouse control using a sensor network. The hardware implementation shows periodic monitoring and control of greenhouse gases in an enhanced manner. Future work is concentrated in application of the same mechanism using wireless sensor network. A solution to the existing drawbacks can be found out from this proposed model. The proposed model is implemented in hardware and tested. The results show an excellent improvement in the sensing parameters when compared to previous approaches to environmental monitoring and greenhouse control models. Sensor arrays like temperature sensor, light sensor, humidity sensor and vibration sensors are incorporated in the board. The sensed data is processed by the micro controller and displayed on an LCD display. Wireless transmission of the parameters is accomplished through a ZigBee module that periodically sends information to the remote monitoring station. To control and monitor environmental variables, sensors and actuators capable of measuring and controlling the values inside the greenhouse are necessary. Generally, a greenhouse control is implemented just by approximating a measured cost to a reference or ideal cost. Due to cost considerations, the proposed model uses sensor network instead of wireless sensor network. The sensed data is forwarded to the gateway. The gateway then forwards the data to the remote monitoring base station. The base station is a remotely located *software configured computer*, (i.e. desktop GUI application) where the monitored details are periodically visualized to carry out further control actions.

The work presented in [2], takes full consideration of cost, practicability and other factors, combining IoT with fuzzy control methods using GPRS to remote control, and thus designing a smart greenhouse monitoring system with better performance, simple structure and easy extensibility. There is also controlling action taken automatically, as greenhouse windows/ doors roll are open or closed based on the soil moisture levels. Thus, the system will help the farmers to avoid physical visit to the field, and increase the yield with the maintenance of precise parameters such as CO^2, soil moisture, temperature, and light in the greenhouse with the help of IoT. The project is carried out with the help of an IoT kit and Internet connection. The results are analyzed for greenhouse parameters such as CO^2, soil moisture, temperature, and light for bell pepper plant with the help of graphical representation based on the practical values taken by the IoT kit. The comparative result shows the effectiveness of the proposed approach. In conclusion, this paper proposes a use of IoT technologies to effectively monitor, manage and automate a smart greenhouse (IGH).

In [54], the development of a remote environmental monitoring and control framework is presented. The framework consists of environmental monitoring and control node as the local management subsystem (LMS), and the web data providing and system management as the global management subsystem (GMS) to establish a simple and flexible remote environmental monitoring and control based on a cloud platform. The supported features are online and offline environmental monitoring, synchronization of system configuration, actuation, and offline management.

The review presented in [61] highlights some of the most recent advances in greenhouse technology and controlled environment agriculture (CEA) in order to raise awareness for technology transfer and adaptation, a necessity for a successful transition to urban agriculture. Based on the reviewed literature, this study concludes that computer simulation models and adaptive analysis software are already available for greenhouses and controlled environment agriculture (CEA) systems can be extended and modified for this purpose. Recent trends in greenhouse environmental monitoring show that research and development in this field are shifting from offline systems to wireless and cloud-based data collection architectures. Various data acquisition platforms, either prototype or commercial, are used for improving the performance of greenhouse production. Some of the most recent examples include web-based, cloud-based (i.e. web based applications), IoT communication and control, wireless sensor networks, field-server based monitoring, field router systems, and distributed data acquisition with local control management [61].

A list of commercial smart greenhouse systems, greenhouse monitoring software, DIY (do it yourself) solutions and equipment, providing remote management, monitoring and equipment control (operating) of IGHs using IOS and Android devices (mobile) can be found in [70]. This also identifies the components required to "*sense*" environmental parameters, control the installed equipment, remotely manage the IGH and review the analytics gathered by these systems [Figure 2].

The Hackster's community website [71] provides a DIY guide to create an IGH using Amazon components and software (see Figure 3). The sensors and actuators are connected to a commercial micro controller. The micro controller send data and receive commands from a control center hosted in the cloud. Users can interact with the Intelligent Greenhouse through a dashboard or a tablet application. Users can also issue voice commands to the greenhouse using a smart speaker.

The Hackster website also proposes the use of IoT's technologies to monitor manage and automate an IGH [69]. The aforementioned work uses a microcontroller connected with sensors to constantly monitor the temperature, soil moisture, humidity and light levels. After *sensing* the environmental parameters, the data are stored on a cloud server and evaluated. The micro controller is also connected to a set of actuators who are instructed by the cloud to regulate the environmental values to match the ideal cultivation parameters of the currently cultivated plant, this work uses as case study Red Bell Peppers.

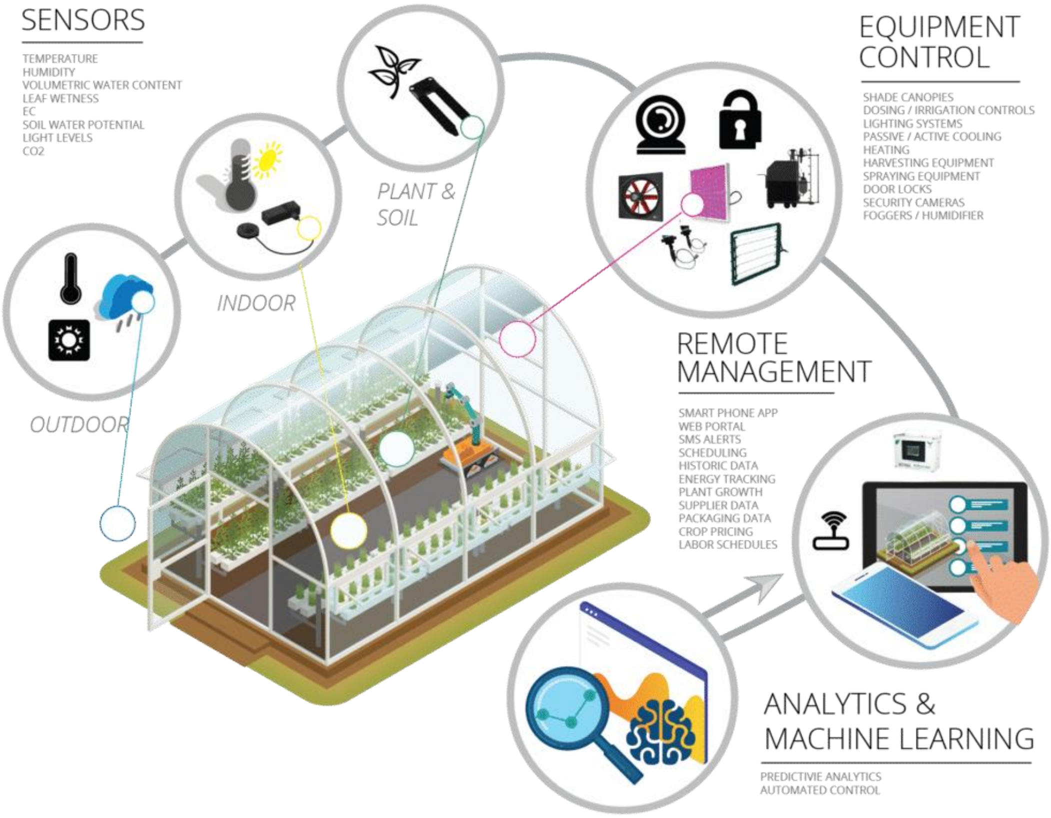


Figure 2: PostScapes Smart greenhouses [71]

The owner can intervene at any time if deemed necessary using the internet. This work focuses on reducing cost, effort and create a precisely controlled environment.

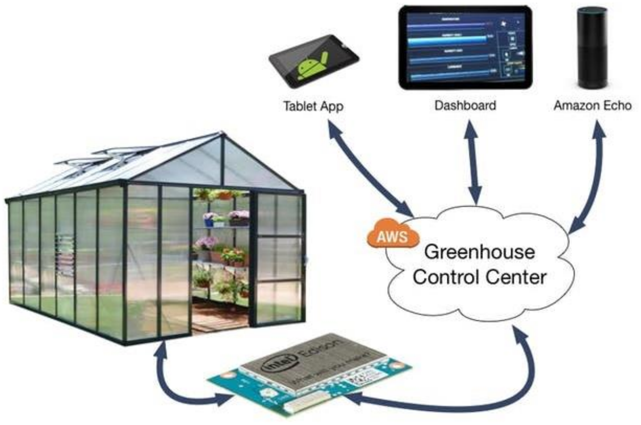


Figure 3: DIY IGH using Amazon Services and products [69]

The authors of [76] present the structure of greenhouse control system composed of an Arduino microcontroller and an Android device. The management system has been implemented physically and tested, and is characterized by low price.

The authors of [31] propose an efficient automatic irrigation system based on computing various changes necessary in the greenhouse using wireless sensor network (WSN), as well as a server and client web service for control and monitoring (i.e. web base application). Their model has two main characteristics, which are reduced computational power, and control and monitoring of the proposed irrigation system over long distances.

So far, however, none of the discussed systems appears to have integrated Virtual Reality with intelligent environments or to have focussed on the provision of suitable interaction techniques for smart greenhouse monitoring and control.

2.6 Discussion

Closed-field agriculture is experiencing a breakthrough transition driven by advances in precision technology, data processing and intelligent farming. Protected cultivations have changed from simple covered greenhouse structures to high-tech plant facilities that optimize the productivity of the plants and human labour [68]. Facilities like [30, 59, 68, 69, 70, 75], as briefly presented in section 2.5, provide users with capabilities for remotely monitoring, managing and operating their IGHs, as well as automation in crops care, cultivation parameters regulation, resources management, growth monitoring, quality certifications, etc.

By observing the paradigm shift taking place with systems providing mobile GUI applications on top of Desktop GUI, and in some cases smart devices Interfaces (e.g. smart speakers), it is evident that the industry is exploring new platforms and alternative means of interaction for IGHs.

As briefly presented in section 2.5, remote monitoring, management and operation of IGHs is currently achieved with the use of web based applications on mobile and desktop devices, and in certain cases with IoT devices (e.g. smart speakers, automated actuator systems) [69, 68, 2]. However, to the best of our knowledge, no attempt has been conducted so far to exploit VR technologies as an option to enable users to interact remotely with IGHs.

VR systems provide a variety of controls and interaction techniques (i.e. gestures, controllers, audio, visual, etc.) and are immersive by nature that greatly enhances telepresence. As an immersive response whereby users perceive the artificial environment, telepresence is targeted to provide the necessary cognitive and sensory input equivalent to that of the more concrete real environment. Telepresence positively affects instrumental and experiential value, which in turn, influences user’s attitude towards the system, while studies have shown it to greatly enhance user’s remote monitoring, operating and maintenance capabilities [49].

VR technologies are already used in different IE’s implementation in fields such as health [78], entertainment [57], assisted living [29] and the industry [79]. This work bridges the gap between VR and IGHs with the intent to further extend the means of remotely managing, monitoring and operating an IGH by using the VR technologies.

Additionally, the application of gamification techniques in the LINA framework, such as *challenges* i.e. during the movement configuration the users are presented with some check points they need to reach, to test and familiarise with their selected preferences. *Leader boards* and *points i.e.* during the cultivation the users need to monitor their “*score”* which is presented with the “*health value”* and can evaluate their effort by reviewing previous cultivations scores. The affordances increases the users’ engagement in the *task at hand*. The framework steadily introduces the users to the system’s functionality and capabilities using gamification techniques thus ensuring the cognitive load of the system’s learning curve will be low and unobtrusive [51].

To this end, this thesis aims to create a Gamified Virtual Reality Framework providing the end-users with remote management capabilities of an IGH. IGHs incorporate a great variety of sensors (e.g. temperature, humidity etc.) and actuators (e.g. motorised windows, irrigation system etc.), that monitor and alter the environmental parameters of the greenhouse. The system present the users with a virtual representation of the IGH with the included sensor and actuators, as well as a set of tools and VR techniques to accomplish the desired tasks (i.e. navigation, object manipulation, interface interaction) within the automatically constructed virtual environment.

3 User Groups and Requirements

This section aims to specify the user groups that are addressed in this research work, which have beenselected by identifying who would benefit the most by the remote capabilities of an IGH, as well as presenting the conducted collection and analysis of user requirements.

Identifying user groups is critical to the success of any system. Cooper (1999) suggests the use of user archetypes, called personas, to help define the product by replacing the notion of the abstract, elastic user [73]. The persona is a precise description of a hypothetical user and his or her goals, and it represents a group of users throughout the entire design process. When describing a persona, details about the user's lifestyle, interests, needs, limitations, patterns of behaviour and other, are included. The engaging persona method is directed at creating scenarios that describe solutions [74].

In this work, we describe three personas as representatives of the user groups of LINA system use. The primary user group of the system, namely "*Owner/Farmer*", includes users that operate the HCI and might have economically invested in an agriculture project. They need to constantly monitor, manage and operate the cultivation processes. The second user group includes users such as the *agronomist*(s) or the *scientific consultant*(s), whose role is to supervise the cultivation and facilitate the owner's work, mainly by providing instructions, advice and occasionally directly acting for handling specific situations, (e.g. adjust humidity). Furthermore, a third group of users is introduced, named "Guests". This group refers to individuals that could visit the IGH for other purposes. For example, an educational visit for a remote agronomist lesson or a virtual tour in a botanic park, or a virtual showcase for a potential investor. Each group has distinctive tasks to complete and goals to achieve through the use of the system.

3.1 User Groups (UGs)

3.1.1 UG1 - Greenhouse Owner / Farmer

This user group consists of the Intelligent Greenhouse's owners and personnel, and is referred as the primary user group of the system. The user of this group has invested time and money for the IGH's infrastructure (i.e. hardware and software), operates the cultivation and expects to financially benefit from the harvesting. It includes all the users that have access to managing and operating the IGH, and are the more interested in the success of the cultivation, considering that their profit depends on it. These users are the gardeners - caregivers for the plants and they must be able to manage their time effectively and create a schedule with all "chores" that need to be done or tasks to be accomplished such as watering the plants, light exposure, temperature adjustments, etc. The provision of means for remotely scheduling and operating such tasks will assist them to minimize their daily engagement in the greenhouse, and maximize their productivity. To this end, it is essential for them to easily access information about the overall state of the greenhouse through the sensor readings, (e.g. air temperature, humidity), as well as plants' specific metrics to ensure the best possible conditions for their plants.

Finally, these users need to be able to receive status information for the all installed devices (actuators) in the IGH, such as motorised windows, irrigation system, fan, etc., as well as be able to operate them in response to the greenhouse's current conditions and /or plants' needs.

3.1.2 UG2 - Agronomic consultants

This user group consists of agronomic consultants whose main interest is to facilitate their clients (owners) by providing them with specialised knowledge such as ideal cultivation conditions for plant species, advice on cultivation methods and techniques, as well as solutions for any plant's health issue that may occur during the cultivation period.

Due to the nature of their work, this type of users would most likely have to supervise a large number of crops and greenhouses. To this end, they need to be able to quickly review the overall health status of a greenhouse and be notified if any condition requires attention. In addition, in order to be able to provide proper, individualized advice for various crops to different owners, they need access to a great variety of information in a suitable and efficient manner.

3.1.3 UG3 - Guest

This user group, compared to the above, appears to have a variety of interest according to their scope for visiting the IGH that may be in terms of educational, financial or other reasons. This heterogeneous user group should have limited access to the greenhouse functions/ services, since their interest lies only in reviewing the greenhouse's status rather than taking action for improving or maintaining it. This user group should be able to obtain information about the plants' status and the environmental parameters of the greenhouse to review the greenhouse's state in general or the inhabitant plants and their current condition. The users should not influence the greenhouse operation in any way, and thus they should have limited access privileges, which will be of the "view only" type.

3.2 Scenarios of use (SoU)

According to [75], a scenario of use is a narrative written in a natural language that focuses on a user using the system. The goal of the scenario is to explore design solutions. The engaging persona is a description of the user in a scenario. The engaging persona can be the user in one or more scenarios depending on goals and the situations of use. The objective of using the engaging persona is to enable the designers to engage in users that differ from the designers in terms of sex, culture and/or age. Scenarios are arguably the starting point for all modelling and design, and contribute to several parts of the design process. In HCI, scenario-based methods have become an accepted approach for requirements elicitation in the preliminary design phase [74].

This section presents the narrative descriptions of interactive processes, including user and system actions and dialogues. More specifically, a set of realistic examples will be given regarding what tasks the users is able to carry out in the specified context of the LINA framework.

3.2.1 SoU1 - Greenhouse Owner

Giannis is a 48-year-old farmer, with a special interest and expertise in fruit and vegetable cultivations that is his full-time job. Giannis has been working as a farmer for over 30 years, during which he has gained a fair share of knowledge and experience. He owns three (3)

greenhouses, one in Sitia for growing peppers, one in southern Heraklion for growing bananas and one near his residence in Agios Nikolaos where pumpkins and eggplants are the seasonal crops. Giannis recently focuses his job interest on developing optimal cultivation conditions for herbs and aromatic plants as he intends to enter the local market next year.
He uses the Intelligent Greenhouse Assistant System on a daily basis, which enables him to review and manage all his greenhouses remotely and in an automated way. His daily routine includes a quick review of the greenhouses by using the system in order to identify the chores he has to complete in each greenhouse. The Lina framework enables him to effectively manage his time and create a schedule with tasks he needs to complete. Giannis wears his VR gear and starts the system to initiate his revision in order to create his "*todo-list*" for the day.

Upon entering the virtual system, he is presented with the virtual representations of his greenhouses. Using his controllers, he chooses the one labelled **"***Herbs and Aromatic plants"*. Upon entering, Giannis is presented with the central control panel where the ambient greenhouse variables are displayed along with the general actuators (e.g. fan, windows, etc). While reviewing the greenhouse's ambient variables, Giannis notices that one of the sections is marked "orange" by the system. This color encoding represents the need for further attention; the particular cultivation zone's overall health is lower that 75%, but over 50% (between 75% - 50%). Giannis moves closer to the section and he is now able to read that the section's overall health is 70%. After a quick review of the displayed sensor values, he can easily identify the reason why the system marked orange the section, which is that the soil moisture is lower than the proper value.

Giannis decides to fix the soil moister and he uses his controllers to select the section's actuator control, subsequently selecting the irrigation control. Giannis activates the section's irrigation and sets the timer to deactivate it after 10 minutes. Giannis moves on to review the rest of the greenhouse's sections and ensure that everything is in the best possible condition.

After ensuring everything is good enough, and checking the greenhouse's notes to make sure there is nothing left on his "*todo-list*" for today, he exits the virtual greenhouse and moves to the next one.

3.2.2 SoU2 – Agronomist

Marios is 40 years old and holds a degree in agricultural science. He runs a mid-size agricultural firm since over 10 years. His client portfolio consists of at least 100 clients, both hobbyists and professionals. Since Marios focuses on being very competitive and productive, he has adopted a state-of-the-art Intelligent Greenhouse Assistant System.

On a regular day, Marios enters his office holding a coffee and greets his trainees, who analyse data gathered from his clients' greenhouses, extracted from the Intelligent Greenhouse system. Marios places his laptop on his desk, takes a sip from his coffee and logs into his workstation. His daily routine begins by reviewing the status of all his clients' greenhouses.

Upon logging in the Lina framework Marios is greeted by the system and is presented with all the greenhouses he is granted access to from his clients. The system presents Marios with labelled and colour encoded information about each greenhouse, thus enabling him to identify the greenhouses that require his immediate attention at a glance. Marios notices one of the greenhouses has a message attached on it, an indication provided by the system. Marios selects the greenhouse labelled with the owner's name, and he is now able to enter "*Giorgo's greenhouse*".

Once inside, Marios accesses the central control panel for a quick review of environmental variables before moving on to the section with the attached message. Marios reviews the general actuators control schedule in conjunction with the environmental variables to ensure the automated actions will not affect the environmental variables in a negative manner.

Marios then moves to the section with the attached message indication, he selects the message using his controllers and reads it. Giorgos says "*Mario I have noticed a small deformation on the dorsal and ventral sides of the tomato leaves, even though the humidity indication is high. Any suggestions?"*

Marios selects the humidity sensor indication, which is colored "green" by the system since the humidity levels are good. The system presents Marios with graphs detailing the sensor values over time. Upon further inspection Marios concludes that Giorgos is irrigating the plant late at night and this creates issues with the humidity, so he selects the message tool and replies to Giorgos informing him what he should do to correct the irrigation plan.

Marios reviews the rest of the sections, and exits the virtual greenhouse to move on to the next one.

3.2.3 SoU3 - Guest

Helena is a recently retired woman, aged 68 years old. She lives in a cosy flat in Athens and she wants to build a small garden at her veranda, although her knowledge about flowers is limited. For that reason, Helena, who always liked flowers and now has plenty of free time, has decided to attend online floriculture courses from the Department of Agricultural Science of the Technological Educational Institute.

Helena wakes up in the morning, gets ready and pours herself a nice warm cup of coffee before attending today's online lecture. Helena puts her VR headset on and launches the Intelligent Greenhouses application. Using her touch controllers, Helena enters the lobby virtually located in front of the Floriculture Course's greenhouse as assigned by professor Serchio Makina.

Professor Makina uses the Intelligent Greenhouse's *Tour Functionality* as a mean to deliver his lectures. Upon entering the virtual greenhouse, Helena sees the virtual representations of the sections being cultivated at the time in the Floriculture Course's greenhouse as part of the course. Professor Makina has attached questions for every section, in the form of attached messages.

Helena uses her controllers to explore the virtual greenhouse and select the first section, with the attached message indication. She reads the message with the professor's question and then proceeds to select the information tool to read information about the plant in order to answer the question. Some of the questions require Helena to identify potential issues by reading sensor values from the installed sensors and proposing solutions, (e.g. temperature is high -> open window.)

Helena answers all the questions, and completes this lecture's assignment. Then she takes her time moving around and reviews her answers and the greenhouse's state before exiting the virtual greenhouse and logging off the system.
Professor Makina will review her answers attached on the sections and will add her grades as attached messages.

3.3 User requirements

User requirements analysis provides precise descriptions of the content, functionality and quality demanded by prospective users [45]. There is a wide variety of methods and techniques for user requirements elicitation, mostly originating from the social sciences, psychology, organizational theory, creativity and arts, as well as from practical experience. Many of these techniques are based on the direct participation of users or user representatives in the process of formulating their own technological needs [45]. The methods used for the elicitation of user requirements were interview, focus groups, brainstorming and use cases, which were selected as the most suitable for the user groups UG1, UG2, and UG3.

Interviewing is a commonly used technique where users, stakeholders and domain experts are questioned to gain information about their needs or requirements in relation to the new system. Scenarios and use cases give detailed realistic examples of how users may carry out their tasks in a specified context with the future system. In focus groups, the general idea is that each participant can act to stimulate ideas in the other people present, and that by a process of discussion, the collective view becomes established which is greater than the individual parts [77]. A fifteen minutes interview with the experts of FORTH's Intelligent Greenhouse (IGH) was conducted and their opinions about the IGH and the capabilities it should provide were discussed. The definition of user groups and the creation of use cases was the following phase. The requirements elicitation continued during a two hours long brainstorming session and finally during a focus group meeting the requirements were established. The participants of the focus group were experts involved in the development of the FORTH's Intelligent Greenhouse.

The paragraphs 3.3.1 and 3.3.2., present the functional and non-functional requirements for the aforementioned stakeholders (UG1, UG2 and UG3) towards the creation of a framework, containing the virtual environment, the authoring tools, and the monitoring and operation applications, to support user's VR interaction with IE.

4 Functional Requirements

Functional requirements capture the functionality required by the system towards effectively supporting (an agreed upon set of) user tasks. In other words, functional requirements define what a system is supposed to do, i.e., "user should be able to <requirement>". For the purposes of this thesis, the word "system" represents the entire proposed framework to support user's Virtual Reality interaction with Intelligent Environments.

4.1 UG1 - Greenhouse Owner / Farmer

Monitoring and operating activities are essential for the greenhouses' owners and farmers. These activities influence the greenhouses' environmental state, and enhance the users' ability to effect cultivation variables (i.e. soil humidity, air temperature), while enabling real-time monitoring of the greenhouses' environmental variables (e.g. ambient air temperature) and hardware status, such as the if the motorized roof window is open or closed**.**

According to "Greenhouse Owner" scenario of use, these users are interested in improving production while minimizing the labour and expenses required for harvesting the crop. To this end, users will use the system to remotely review and monitor their crops. They will be enabled to enter the system, navigate and enter the desired VE to review, navigate towards the desired virtual cultivation zone presented with the information collected by the installed sensors. The user can either review the *current values* of all the sensors for an overview of the state of the cultivation zone or select one sensor and review previous values collected for the selected sensor (e.g. select the soil moisture sensor and review the values for the last 24 hours). If the user deems necessary to regulate the state of the currently reviewed zone, they can access the actuators and operate them to match the plant's ideal cultivation values or the users' preferences (e.g. in the soil moisture example to activate or deactivate the irrigation system). Finally, these users will use the system to manage their chores, by accessing the "chores list" for each zone and scheduling or removing chores that need be done or were already done (e.g. the user need to visit the IGH to remove the weeds from cultivation zones 1 and 3) .

The User Group 1 functional requirements (UG1-R) are:

- **UG1-R1:** The user should be able to enter a VE that is a representation of the user's IGH.
- **UG1-R2:** The user should be able to effectively explore the VE using the provided interaction techniques. The user should be able to enter the VE and move around to reach points of interest, such as virtual greenhouse, cultivation zones etc.
- **UG1-R3:** The user should be able to identify the areas of interest (i.e. cultivation zones that require attention) and easily access corresponding specific information.
- **UG1-R4:** The user should be presented with real-time information in the VE, to ensure that the user acts upon the latest or current greenhouse's status information and not on an outdated version of it.

- **UG1-R5:** The user should be able to receive status information for all installed devices (actuators) in the IGH, such as motorised windows, irrigation system, fan, and effectively operate them in response to the IGH's current conditions and/ or plant's needs.
- **UG1-R6:** The users should be able to access the history of the cultivation zone's sensor collected data and review dated on values.
- **UG1-R7:** The user should be able to evaluate the cultivation progress.
- **UG1-R8:** The user should be able to create or access the automatically created schedule with the chores each zone needs and when completed to mark them as such.
- **UG1-R9:** The user should be able to access information about the overall state of the greenhouse through the sensor readings, (e.g. air temperature, humidity).
- **UG1-R10:** The user should be able to access plant (zone) specific information through the installed sensors (e.g. soil temperature, air temperature).

4.2 UG2 – Agronomic Consultants

An agronomic consultant is a type of user who acts as a mentor/advisor for the farmer user. The agronomic consultant might be an agronomist or any other expert in the field and is most likely associated with more than one IGH owners. For these users, daily routine includes inspections of different greenhouses and crops in order to provide their clients with tailor made advice and guidelines.

These users use the LINA system regularly and for extensive periods to gather as much information as possible. An agronomic consultant will need to go over the current values of the sensors and review crop's overall health. If any problem appears, he would refer to the previous values information in order to propose the optimal suggestion to resolve the problem and ensure harvesting. The agronomic consultant could also review the chores the farmer has completed and propose other according to the plant's needs (e.g. the ladybugs are active this season, make sure to protect your tomatoes) or recommend changes on the schedule the IGH owner has created for the cultivation zone.

The User Group 2 functional requirements (UG2-R) are:

- **UG2-R1:** The user should be able to access the virtual environments that are representations of the respective client's IGH.
- **UG2-R2:** The user should be provided with detailed information that corresponds to environmental parameters of the currently reviewed IGH.
- **UG2-R3:** Information should be up to date - in real-time - to ensure that the instructions given to the IGH's owner by the user, match the IGH's current needs.
- **UG2-R4:** The agronomic consultant should be able to review the current values as well as the history of the provided information from each sensor.
- **UG2-R5:** The user should be able to review the actuator status and operate them.

4.3 UG3 – Guest

This user group has restricted interaction privileges that are limited to review rights, meaning they will not be able to access information beyond the current situation and control the greenhouse's actuators (e.g. irrigation system). These users can only have temporary access, and they would not be able to access the virtual greenhouse after the visit session ends. The creation of this group facilitates the study of how the first time users interact with a virtual environment, how they use the system to explore the new environment and how they interact with the different sections and sensors included in the environment.

These users are enabled to explore the VE and review the currently cultivated crops. To this end, a guest would like to explore the VE using the provided interaction techniques, review the cultivation instructions and plant's information and observe the cultivated plant's health and sensor metrics.

The User Group 3 functional requirements (UG3-R) are:

- **UG3-R1:** The user should be able to access information about each cultivation zone and the included sensors (i.e. cultivated plant's information, plant's name, included sensors, cultivation instructions).
- **UG3-R2:** These users need to be able to review the current state of each cultivation zone (e.g. health sensor current value).
- **UG3-R3:** The user should be able to explore the virtual environment, move around and select the cultivation zone they wish to review.

4.4 Non-Functional Requirements

As presented in [86], non-functional requirements (NF-R) are constraints on various attributes of the system's functions or tasks associated with the users' goals. In other words, non-functional requirements specify the required behaviour of the system towards supporting the user goals efficiently, i.e., the user might end up disliking the system if some of his/her goals (e.g., to perform administrative tasks as quickly as possible; to stay away from big mistakes, to be consistent) are violated while using the system. The most relevant non-functional requirements supporting the aforementioned user groups are the following:

- **NF-R1: Learnability:** It is essential for the system to be intuitive and require minimum learning from the user. To this end, the provision of easy to use, self-descriptive and intuitive user interfaces constitutes a fundamental requirement.
- **NF-R2: Satisfaction:** Offering satisfaction to the user by accessing an interactive application is considered a primary requirement. Satisfaction is achieved through a variety of ways, like providing with a smooth experience, operating through intuitive and usable UI, customization to match one's preferences and needs, etc.
- **NF-R3: Robustness:** The application must be extremely robust against all kinds of misuse and errors. Wrong inputs must not lead to a system malfunction or crash.
- **NF-R4: Availability:** The system must remain functional even in the presence of hardware component crashes (e.g. the irrigation system malfunctions, the user should be to operate the other actuators).
- **NF-R5: Safety:** Safety requirements are the 'shall not' requirements that exclude unsafe situations from the possible solution space of the system. In systems that use Virtual Reality devices, safety is a major issue since the user is immersed in the virtual

environment while losing track of the physical world, which in turn may lead to injury or discomfort if designers do not implement proper measures to ensure user's safety.

- **NF-R6: Security:** Since the user's interaction is constantly monitored, privacy issues must take into consideration and rules must be precise and verified.
- **NF-R7: Timeliness:** Most services in an IE system have to be carried out in real time.
- **NF-R8: Resource Efficiency:** The available resources, i.e., processing power, memory, communication bandwidth and energy, have to be utilized as efficiently as possible in order to allow: a) an affordable price of the systems, and b) the realization of highly integrated, autonomous sensor nodes with a high endurance. Resource efficiency also ensures the scalability of the system to larger greenhouses with multiple sensors and actuators.
- **NF-R9: Natural, Anticipatory Human-Computer Interaction:** Virtual Intelligent environments have to provide user interaction techniques for the target users. Each group has different requirements for interacting with the system. Multimodal interaction paradigms that combine several modes are a powerful approach to enhance usability. Anticipatory interfaces, which proactively engage users in interaction in certain situations, are considered mandatory.
- **NF-R10: Adaptivity:** The system is able to adapt itself at runtime and provide real time contextual information regarding the IGH.
- **NF-R11: Extensibility:** The system should be extensible, meaning it should easily support the installation of new devices (e.g. sensors, actuators), new (virtual) plants/crops, as well as system components and functionality.
- **NF-R12:** The VE should provide the user with a non-stressful, immersive, intuitive experience, allowing him/ her to get the most out of the virtual environment and be able to effectively monitor and operate his/ hers Intelligent Greenhouse remotely.
- **NF-R13:** The process of reviewing/inspecting each greenhouse should be as little time consuming as possible, and the system should attempt to reduce cognitive load as much as possible to ensure the effective use of the system.
- **NF-R14:** Since the agronomist will extensively use the system as a tool for reviewing the green houses, the system should provide with a pleasant experience and interaction methods that suit agronomist's requirements (e.g. need to use the system for eight (8) hours, the system shouldn't require physical effort).

4.5 Interaction Requirements

Due to the nature of virtual reality systems, the users virtually navigate in an environment where their physical movements are mirrored through a virtual body created by the system (i.e. an avatar). However, in many cases, the lack of physical feedback may confuse users and break the immersive experience; when they "touch" something in the virtual world, the sensory feedback is absent.

This create challenges for the designers to design and implement interaction methods that maintain the immersion. A number of interaction techniques were identified while reviewing the literature [12], and this thesis aims to identify strengths and weaknesses of the selected interaction techniques.

The first need is to identify the tasks that the users attempt to complete when using the system, and figure out which techniques are best suited for the created framework. As stated by Bowman [12], the tasks users are interested in, in a 3-D virtual environment can divided in three categories, **navigation, selection/manipulation**, and **system control.**

4.5.1 Task 1: Navigation - Requirements (Task1–R)

The users need to be able to navigate through the virtual environment, identify, and reach destinations of interest. Human centered design indicates that the system should be designed for user errors, meaning that in case of user error, the system should enable the user to recover and prevent discomfort and uncertainty. The design process of the greenhouse navigation techniques should also take into consideration a learning period in which the user will familiarize with the environment and is prone to mistakes.

- **Task 1-R1:** All users should be able to explore the virtual greenhouse.
- **Task 1-R2:** The system should provide indication(s) for all possible user destinations (e.g. Cultivation zone name, exit)
- **Task 1-R3:** The system should create the virtual greenhouse in a manner that will avert user errors, meaning the system should create boundaries. However, the system should enable the user to access areas of interest easily.
- **Task 1-R4:** Unnecessary user movements (e.g., gestures) should be minimized in order to avoid fatigue.
- **Task 1-R5:** The system should provide configuration over the implemented navigation techniques to enable the users to configure the navigation parameters to fit their preferences.

4.5.2 Task 2: Selection/manipulation – Requirements (Task2-R)

Users need to be able to interact with the virtual environment and manipulate objects placed in it. In this work, this translates in the user's ability to select cultivation zone, interact with 3D GUI elements and access controls that will alter the IGH's state through the interaction with their virtual representation.

- **Task 2-R1:** The system should provide the users with the ability to "touch" the interactive elements of the virtual greenhouse.
- **Task 2-R2:** The system should provide appropriate feedback so that the users can see the effect of their actions in the virtual world.
- **Task 2-R3:** The system should be consistent with feedback, meaning specific user actions should always have the same outcome (e.g. clicking the same button 3 times should repeat the same task 3 times).
- **Task 2-R4:** The selected interaction techniques should be intuitive, as well as easy to learn and use.
- **Task 2-R5:** The system should be flexible and provide assistance when users attempt to select an object, since accuracy tends to be an issue in virtual environments.

4.5.3 Task 3: System control – Requirements (Task3-R)

System control refers to the task of altering the system's state to match the user's preference or needs. For the case being reported here, this includes interaction techniques, information presentation, and variables for the selected techniques.

- **Task 3-R1:** Each user should be able to access the system controls and interact with them.
- **Task 3-R2:** Users should be able to easily identify and select what they are looking for and for that, the controls need to be properly labelled and organized.
- **Task 3-R3:** Users preferences are stored in the user's account and the system loads them automatically when the user successfully logs in.
- **Task 3-R4:** Users should be able to review values when adjusting them and be informed about the impact they have on the system (e.g. when changing the movement speed, from 1 to 10, they should be able to move around and "Feel"/sense the difference in velocity.
- **Task 3-R5:** The system should not overload the user with unnecessary configurations; when designing a system, designers should resist the temptation of adding unnecessary features.

4.6 System Design

4.6.1 Virtual Environment

The Virtual Environment (VE) is the virtual world the user interacts with in order to remotely monitor, manage and operate the IGH. The VE component is generated by the Generator component using the IGH's information, which is received through the Data Manager (DM) component.

The system consists of two (2) scenes, a *dashboard scene*, which allows the user to select which of virtual greenhouses she wishes to enter and to enable quick revision of the user's IGHs, and a *virtual greenhouse* scene for the purposes of monitoring, management and operation of the selected IGHs. The virtual greenhouse (scene) is generated in three (3) parts, which are: a) the *scenery*, a generic scene used as a base for the environment, b) the *greenhouse* asset, an asset created to house the cultivation zones and act as the greenhouse representation, c) the *cultivation zone* assets, which acts as the zone representation inside the IGH.

4.6.2 3D Graphical User Interfaces (3D GUIs)

The 3D GUIs are the means through which the user can receive information from the system and issue commands towards the system. 3D GUIs are GUIs placed on a world space instead of screen space. The screen space GUI is usually used for 2D simulations, in which the UI objects are not necessarily associated with any particular objects in the scene. However, when it comes to 3D-GUI objects attached on scene objects in a simulation, it is preferable to use the World Space (canvas) in order to create the metaphor of real world information signs. The user can directly interact with the information provided though the GUIs, which are automatically generated by the system and placed in the virtual environment during the generation process. Each 3D GUI appears above a cultivation zone and the displayed information is relevant to the installed sensors and actuators for this particular zone.

The human vision span or perceptual span is the angular span (vertically and horizontally) within which the human eye has sharp enough vision to perform an action accurately (e.g. reading or face recognition). The human visual field (see Figure 4) spans approximately 120 degrees of arc where most of that arc is peripheral vision. The brain creates an illusion of having a greater visual span by automatically and unconsciously moving the centre of vision into any area of interest in the field of view [77]. Eye accommodation is the process by which the eye changes focusing power to maintain focus on an object as its distance varies. In eye accommodation, distances vary for individuals from the far point (NPA)—the maximum distance from the eye for which a clear image of an object can be seen, to the *near point (NPC)*—the minimum distance for a clear image [80].

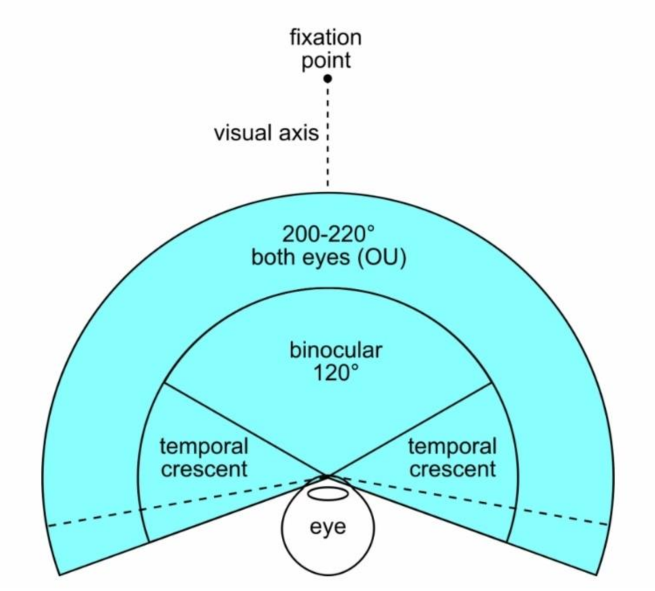


Figure 4: Human field of view[1]

In [62] the factors NPA, NPC and "eye pain" found to be associated with visual fatigue, while monitors with various radii of curvature (flat to 1000R curve) were used. Results show that the curvature of the monitor is related to the visual fatigue and the score of "eye pain" was significantly higher for the flat monitor versus the 1000R curved monitor after the visual tasks (p = 0.034).

Which is why this work implements curved 3D-GUIs to display cultivation zone's information, in order to reduce user fatigue while reviewing and interacting with such information.

4.6.1.2 Interaction Techniques

Interaction techniques are the means through which the user interacts with the virtual environment (greenhouse). Bowman in [11, 12] explains that in VR environments there are three (3) primary tasks the user wishes to complete. These tasks are *navigation, selection / manipulation* and *system control.* In this work, the interaction techniques requirements are described in section (3.5). What follows are the interaction techniques through which the LINA framework covers/addresses these requirements.

The user uses controllers to achieve the following tasks in collaboration with the VR head set.

[1] https://www.pnglot.com/i/oTihmb_fov-both-eyes-binocular-field-of-view-humans/

4.6.2.2 Navigation

The user needs to be able effectively and effortlessly navigate the virtual environment. However, movement in virtual environments may further increase visual reality sickness. [62]. Also, since the system is intended for use over extended periods of time, the user should be able to move into the virtual environment with minimal effort. In [47] the results indicate that commercial console video game systems can induce motion sickness when presented via a head-mounted display.

To tackle these negative effects while minimizing the physical effort required for the navigation task, the *navigation mechanisms* of the system were designed to allow the user navigate the environment seated. As stated during the first heuristic evaluation of the system, those mechanisms had a positive impact on avoiding dizziness and increasing usability.

In the gaming industry, the *"WASD" keyboard combination* is commonly used to move the *"player"* (e.g. an avatar), into *front – back – side* directions and the mouse for rotating the camera and change the player's orientation. Even playing tasks differ, such as operating a vehicle instead of a character the navigation techniques are often the same, i.e. WASD keys ant the mouse.

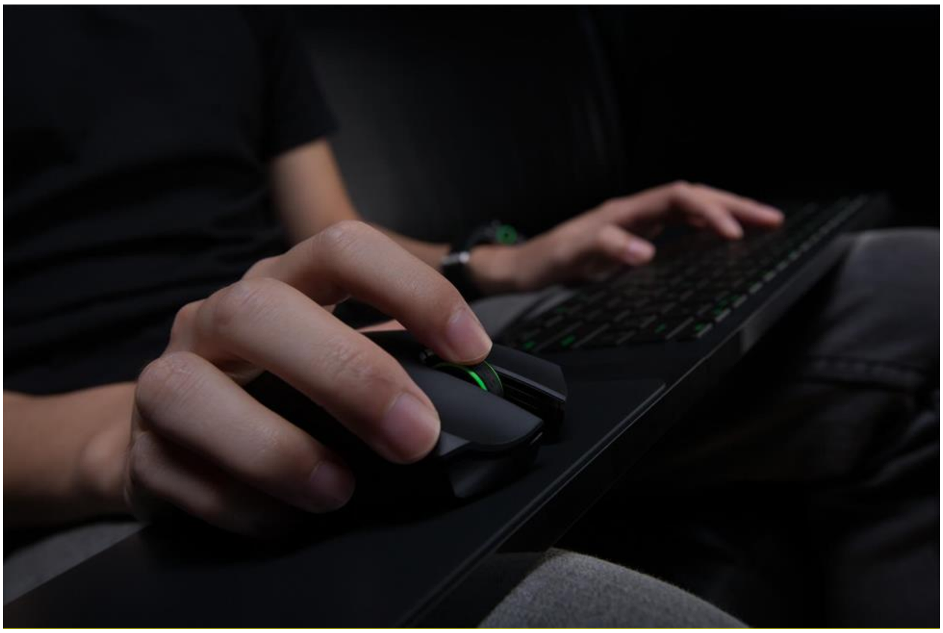


Figure 5: WASD and Mouse hand placement[2]

In console games (e.g. Playstation, xBox) the controllers' current form include two – joysticks, which are placed in such a manner that interaction with the game is made with user's thumb (see Figure 6).

[2] https://www.techadvisor.co.uk/how-to/game/use-keyboard-mouse-on-xbox-one-3683145/

Figure 6: Xbox and PS4 controllers[3]

This work imitates and uses these navigation techniques in the virtual reality environment using the controllers provided by the Oculus Rift (see Figure 7).

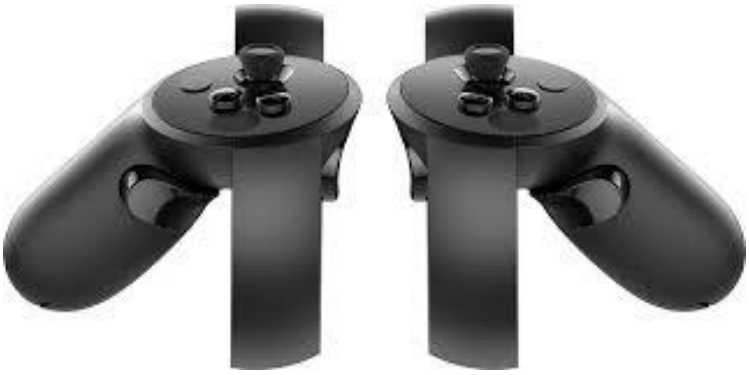


Figure 7: Oculus Rift controllers[4]

4.6.3.2 Selection manipulation and system control

The LINA framework uses 3D GUIs in order to provide the user with context information and actuator control, presented above each corresponding cultivation zone asset or at the entrance of the virtual greenhouse.

To enable the user to interact with 3D GUIs, this work primarily focused on using the *raycasting* technique, as presented by Bowman in [12]. Raycasting is a familiar to the users "point to interact" technique, known from desktop applications, where users using a pointer for selection and, or point and press on a graphical object with the mobile device's applications (see Figure 7). In addition, in an attempt to explore a different way of interaction, a *gesture based selection mechanism* was developed.

[3] https://www.sccpre.cat/show/hTToxTi_ps-controllers-psd-ps4-and-xbox-controller-transparent/

[4] https://www.vr-freaks.de/oculus-touch-controller-nun-einzeln-verfuegbar/

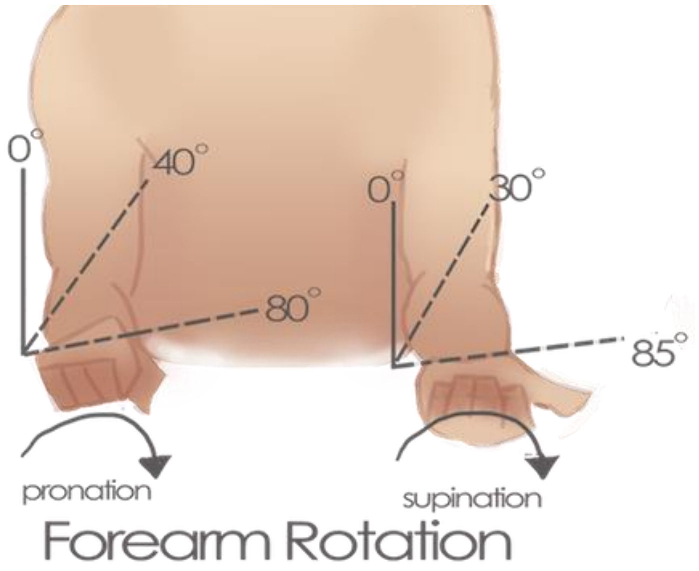


Figure 8: Forearm Rotation[5]

The gesture used as a basis for this selection mechanism is hand/controller rotation movement. The *radial shaped menu* used for system control in this work and gesture mechanism were designed in a manner that ensures, no matter the number of items in the menu, that the range of the gesture (number of degrees required to go over all items) will remain the same. This was crucial because the human forearm has a range of rotation shorter than 170° when the hand is resting on a surface, and beyond that range, the motion is less comfortable (see Figure 8).

4.6.4.2 User configuration

As stated in [12, 13], *navigation, manipulation/ selection,* and *system control* are of the greater value because they affect the three (3) primary tasks the user wishes to achieve in a virtual environment. The user should be able to configure the system to meet individual preferences; the system provides this ability through a settings menu. During the design phase of the framework's settings, the settings that appeared to hold greater value where the interaction techniques representation and navigation variables**.**

To enable the system to remember the user's settings a user's profile was created. However, the user has to run a first time configuration and tutorial scenario during the first login, to select their preferences. By skipping the first configuration, the system defaults to predefined options.

[5] http://www.militarydisabilitymadeeasy.com/elbowandforearm.html

5 Implementation

This chapter describes the implementation of the developed framework. The implementation was based on the high-level architecture of the framework as illustrated in Figure 9.

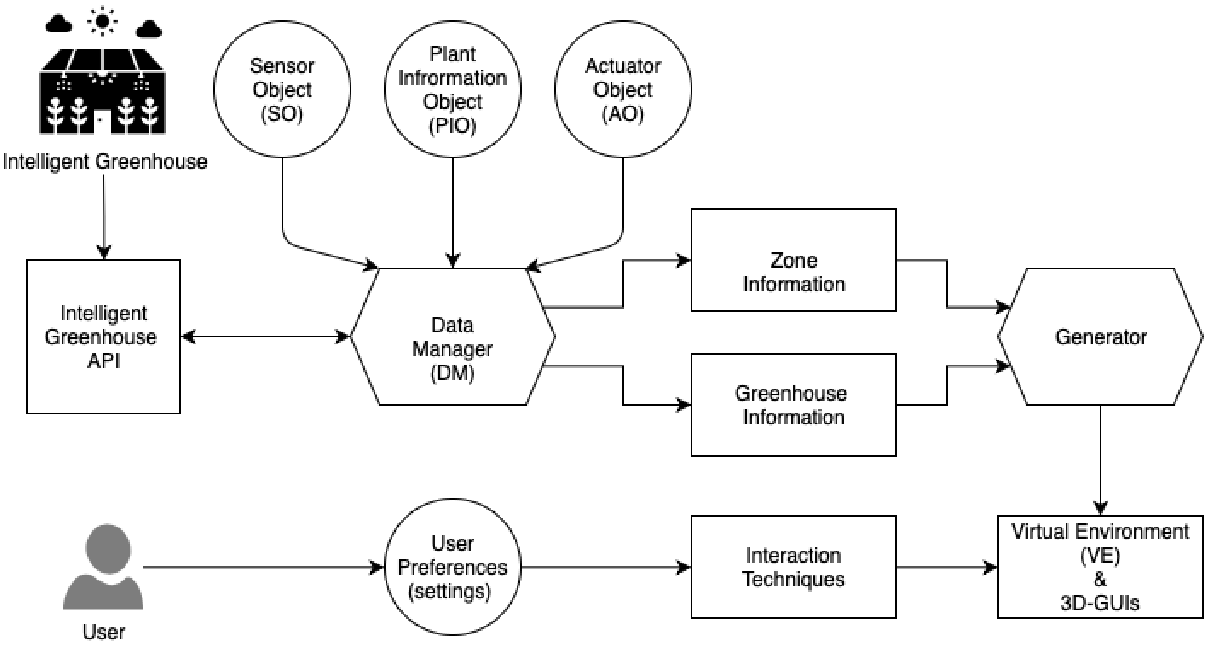


Figure 9: High level architecture

Using the Intelligent Greenhouse API, the Data Manager component retrieves the Greenhouse Information, such as included zones, ambient environmental sensor values and general actuators. Zone Information such as zone-specific sensor values and actuators the DM component, uses the System Entities (SO, PIO, AO) to represent the retrieved information and then feeds this information to the Generator component to generate the Virtual Environment. The LINA framework configures the implemented Interaction techniques using the User's preferences to enable the user to interact with the generated VE.

5.1 System Architecture

The LINA framework was developed to create a virtual representation of an IGH and provide the user with the ability to interact with the Virtual Greenhouse (VGH). In order to create the virtual representation, the LINA framework requests the IGH information through the IGH's API, using the *Data Manager* component that deserializes the requested data to *System Entities.* The *Generator* component uses them to create the *Virtual Environments* and the *3D-GUIs*. Afterwards, the user is able to interact with the generated Virtual Environment and 3D-GUIs through the provided interaction techniques. As for controlling the IGH' actuators, the user generates commands when interacting with their virtual counterparts. Finally, the DM component serializes the commands and relays them towards the IGH, using the IGH's API.

5.1.1 Data Manager

The produced information from the IGH is available to third party applications through a restful API. Information exposed through the API include **sensor data** (e.g. values, timestamps and history) and **actuator data** (e.g. current state, history). The IGH also provides **actuator**

control services such as “operate window”, “operate irrigation system”**,** which the LINA framework consumes in order to relay user commands towards the IGH.

Through the generation process run by the Generator component, the Data Manager (DM) receives the data from the IGH’s API and transforms it in proper form so that the LINA framework is able to effectively create the VGH and the 3D Graphical User Interfaces (3D GUIs). The DM component is also responsible for translating user-actuating commands into valid API calls, through the HIGHs API and thus relaying these commands toward the IGH.

5.1.2 System Entities

In order to effectively represent all the entities included in the IGH in the proposed framework, dedicated objects were created to house information and enable easy access and manipulation.

To this end, the entities were designed and developed, in collaboration with an agronomist, a technical advisor and the designers of the FORTH’s AmI Greenhouse, which is used as the case study for this work. These objects are the **Sensor Object (SO)**, the **Plant Information Object (PIO)** and the **Actuator Object (AO)**.

The SO hosts information about the installed sensors into the IGH, acting as means to monitor cultivation parameters and provide context information to the user. The PIO is designed and implemented for presenting information corresponding to a specific cultivation zone into the IGH. Finally, the AO has been implemented to host tool and actuator information and to provide interaction and operation for the included actuators.

5.1.1.2 Generator

The *Generator* component is responsible for receiving the data from the DM component and generating the virtual environments and the 3D-GUIs of the Greenhouse, through which the user interacts and remotely manage the IGH.

5.1.3 Virtual Environment

The virtual environment (VE) consists of the “world” and all the “assets”. A virtual world is a computer-based simulated environment and an asset is a representation of any item that the user can interact with into the virtual world. Within the LINA framework, the user is enabled to monitor, manage and operate the IGH. The VE is generated automatically by consuming the IGH data provided by DM component.

5.1.4 3D Graphical User Interfaces

GUIs are generated automatically by the Generator component to represent the information and controls received by the Data Manager component. Based on HCI and VR techniques, the Greenhouse’ GUIs were designed and developed.

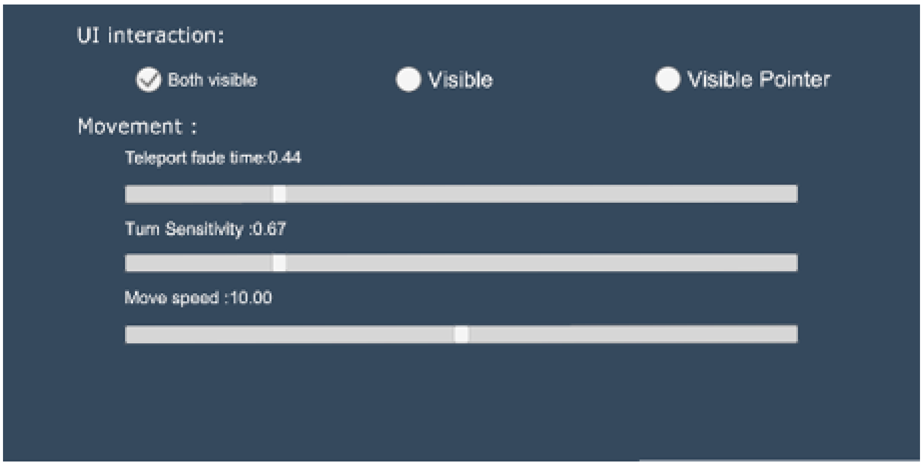
In an effort to take advantage of the potential user familiarity with traditional 2D GUIs, design patterns (user centered design, prototyping, mock-ups etc.) were used and accordingly extended with 3D capabilities in order to place them in the 3D virtual world.

5.1.5 Interaction Techniques

The Interaction Techniques component includes all the implemented interaction techniques that the users might use in order to complete the main tasks they are interested in while interacting with a VE, namely navigation, selection/ manipulation, system control [11, 12].
To enable the user to complete the *navigation* task, two mechanisms were implement: a) point and click to move, and b) thumb stich based movement. As for *selection/ manipulation*, the raycasting technique was implemented and is activated from both controllers. Finally, system control is enabled through gesture-based selection of a menu, and raycasting over User Interfaces.

5.1.1.5 User Configuration and Tutorial

The LINA framework aims to provide the user with means of interaction within a VE enabling to monitor, manage and operate an IGH remotely. When the user logs in for the first time, the system guides him/her through the first configuration, where desired preferences are selected, such as rotation sensitivity, speed, etc. A short tutorial of how to configure the system is provided to the user and in case that the user skips the tutorial, the system will enable the default configuration (see Figure 10). The user can alter any of the preferences at any moment from the settings menu. User's preferences are automatically loaded from the user's profile every time they log into the system.



UI interaction:

☑ Both visible ○ Visible ○ Visible Pointer

Movement :

Teleport fade time:0.44

Turn Sensitivity :0.67

Move speed :10.00

Figure 10: Default configurations

5.2 Framework Components

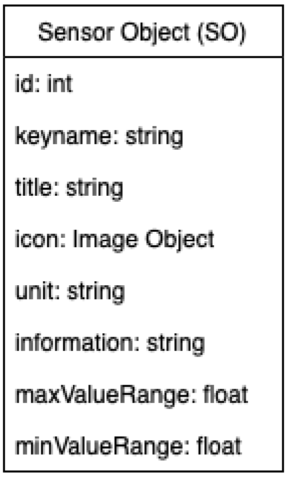
This section describes the components of the system (FC) and the technologies that have been used for their implementation. For the development of the components, **Unity** and **C#** were used to create the virtual environment's assets and provide functionality. **SteamVR 2.0** was used to connect the Oculus Rift (headset, controllers) with the virtual environment.

5.2.1 FC1 System entities

This section reviews the system entities in detail.

5.2.1.1 Sensor Object

Sensor Objects (SO) contain information from the sensors installed in the IGH and enable the system to present information received by the DM component. SOs include the following information: a unique sensor *identifier "Id"*, the name of the sensor to be displayed *"title"*, an *“icon”* or *“unit”* (e.g. temperature sensor: **°C**), the unique per sensor name *“keyname”* , the text *“information”* (e.g. “Air temperature inside of the greenhouse”), and the minimum and maximum range of value *“Values Max Range”* and *“Values Min Range”* (see Figure 11).



Sensor Object (SO)

id: int

keyname: string

title: string

icon: Image Object

unit: string

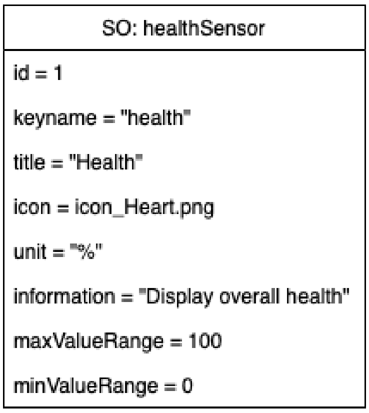
information: string

maxValueRange: float

minValueRange: float

Figure 11: Sensor Object

Each sensor included in the cultivation zone of an IGH is represented in the system by filling in these fields of the specific sensor. This in turn enables the system to identify the sensor’s object using the sensor’s **keyname** and add it to the respective **virtual cultivation zone**. In case that the sensor is not included in the system, the system will add an ***“Invalid sensor”*** with the **keyname** value set to the requested sensor’s **keyname.** The system can be easily extended to include new sensors using an integrated application.



SO: healthSensor

id = 1

keyname = "health"

title = "Health"

icon = icon_Heart.png

unit = "%"

information = "Display overall health"

maxValueRange = 100

minValueRange = 0

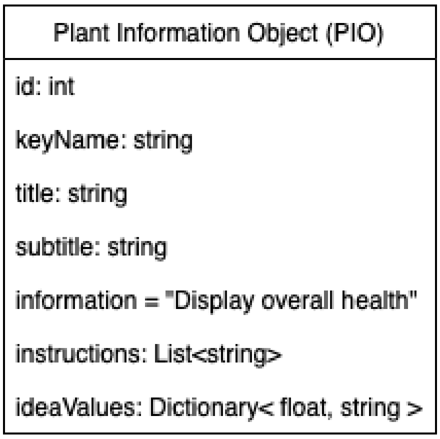
Figure 12: Sensor Object example: “Health sensor”

5.2.2.1 Plant Information Object

Plant Information Object contains plant species information, such as taxonomy, variety, common/scientific name, soil demands, planting and other relevant information (wiki). The PIO includes cultivation instructions, as well as ideal values and their respective ranges as they are defined by the consultant agronomist, categorized as normal - green, warning - orange and critical-red for the included sensors, e.g. Health ranges ***Normal***: 100, **Warning**: ideal -25, **Critical**: warning – 25 (see Figure 12).

The above information included in a Plant Object allows to: a) create the zone *titles* and *subtitles* using the plant’s name, variety and family, b) provide easy access to each plant’s general *information*, c) create a *“Todo”* list for the user including cultivation instructions (see Figure 13).

Finally, by keeping the ideal values in a C# dictionary, the framework is able to provide **color encoding** for each value represented on the **3D GUIs** to enable the user to easily identify if values are within normal range or they need attention.



Plant Information Object (PIO)

id: int

keyName: string

title: string

subtitle: string

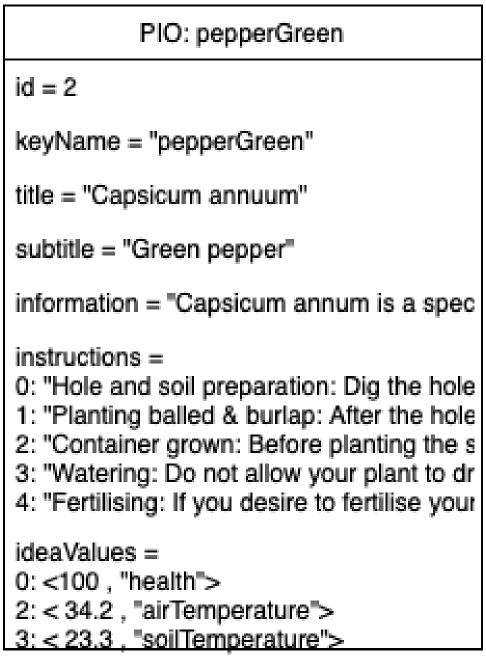
information = "Display overall health"

instructions: List<string>

ideaValues: Dictionary< float, string >

Figure 13: Plant Object

The included fields in **PIO** are: the unique identifier of each specie *“Id”*, the name of the plant *“title”* and family name as *“subtitle”*, the unique key Name *“keyName”,* the general information *“information”*, the list of cultivation instructions *“Instructions”,* and the “*idealValues”* dictionary (key: sensorKeyname , value: array [0:ideal value, 1: warning range,2: critical Range]) (see Figure 14).



PIO: pepperGreen

id = 2

keyName = "pepperGreen"

title = "Capsicum annuum"

subtitle = "Green pepper"

information = "Capsicum annum is a spec

instructions =
0: "Hole and soil preparation: Dig the hole
1: "Planting balled & burlap: After the hole
2: "Container grown: Before planting the s
3: "Watering: Do not allow your plant to dr
4: "Fertilising: If you desire to fertilise your

ideaValues =
0: <100 , "health">
2: < 34.2 , "airTemperature">
3: < 23.3 , "soilTemperature">

Figure 14: Plant object: GreenPepperPlant example

5.2.3.1 Actuator Object

There is a variety of tools and actuators that control mechanisms and systems necessary for the crops, such as the irrigation system or the motorised windows. The Actuator Object (AO) allows for actuator operation, such as “on/off” function, and for state information i.e. “range ±”. In addition, a short description in the form of general information has been added for quick tool tips or additional information about a particular actuator.

In more details, the AO key member fields are: a unique sensor identifier *“Id”*, the name of the sensor “*Title”*, the actuator’s icon **“***icon”*, the unique actuator name **“***keyName***”**, the field of general information/ tooltip information *“Information”*, and the operation state *“State”*, values: 0 when is off, >0 when is on (see Figure 15).

During the VGH generation, the system uses the AO in order to form each cultivation zone and attach the data from the particular actuators installed in this zone. The AO is also used for general actuator control (e.g. *close all windows*), which are attached at the VGH’s ambient information 3D GUI, along with ambient information sensors (e.g. external Air Temperature sensor).

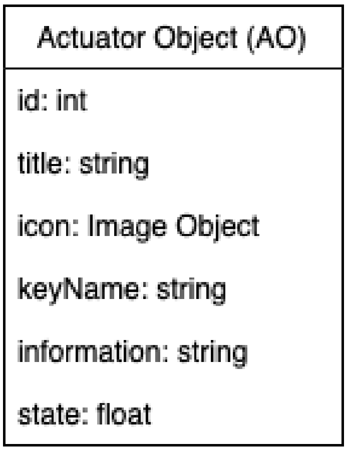


Figure 15: Actuator Object

5.2.2 FC2 Data Manager

The FC2 Data Manager component is responsible for getting the data from the IGH, and propagating user’s commands towards the IGH. The IGH provides an API that produces rest-full services for data acquisition and actuator control. The Lina Framework uses this API at the beginning of a session to consume these services using C# scripts and instantiate prefabs**,** i.e. assets created beforehand which are filled in with the appropriate data, and generates the Virtual Environment (Figure 16).

During a session, the DM component is responsible for updating the sensors values as well as actuator status (e.g. irrigation system is now open), and ensuring the values are up to date.

The DM component is also responsible for managing user-generated data in the form of commands towards the IGH. The DM compiles the data produced from the user’s actions and transforms it in an appropriate form to pass as argument in the IGH’s API service, which in turn will inform the IGH to execute the user’s command; e.g. the user “clicks” the update button to request the latest graph values for a sensor and the DM does what, which in turn, updates all the corersponding 3D GUIs.

Finally, the DM is responsible for the “translation” of the data received from the API, meaning that it receives the data form the API in a pre-agreed form and converts them in a type of LINA object, e.g. sensor object; they are then consumed by the DG to create the virtual environment.

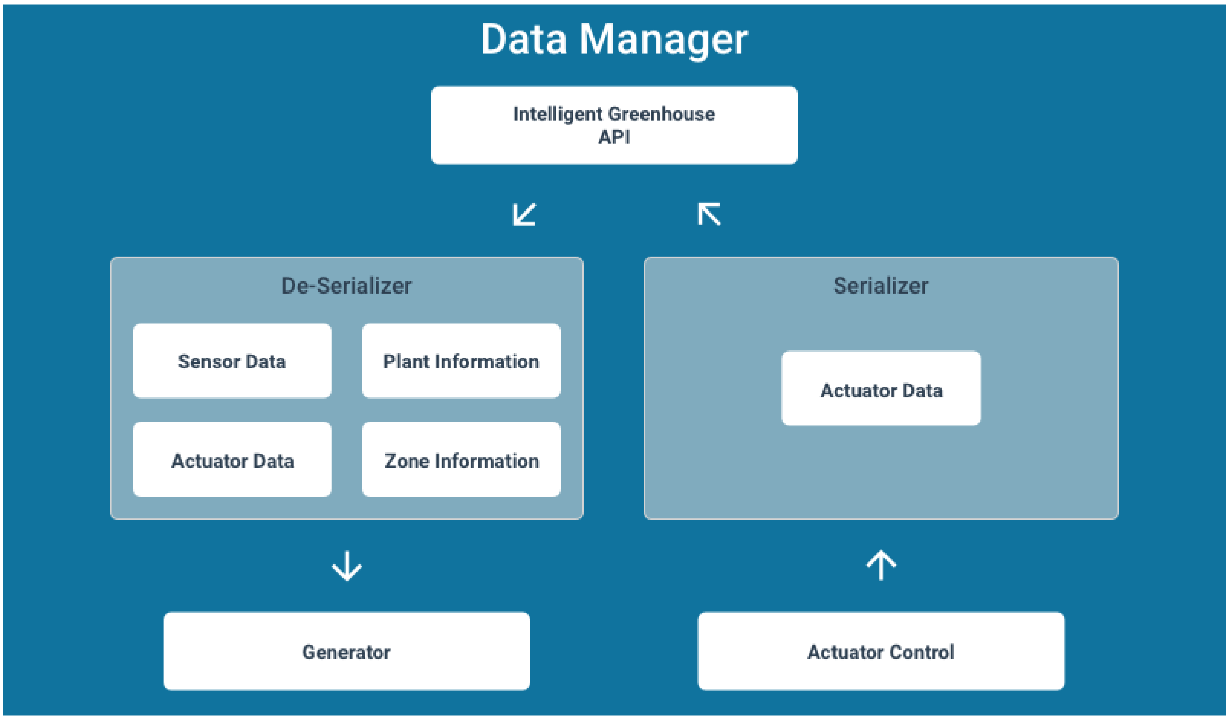


Figure 16: Data Manager

The DM component is responsible for consuming the IGH’s API services in order to acquire context information about a) the IGH’s cultivation zones and b) the installed sensors and actuators.

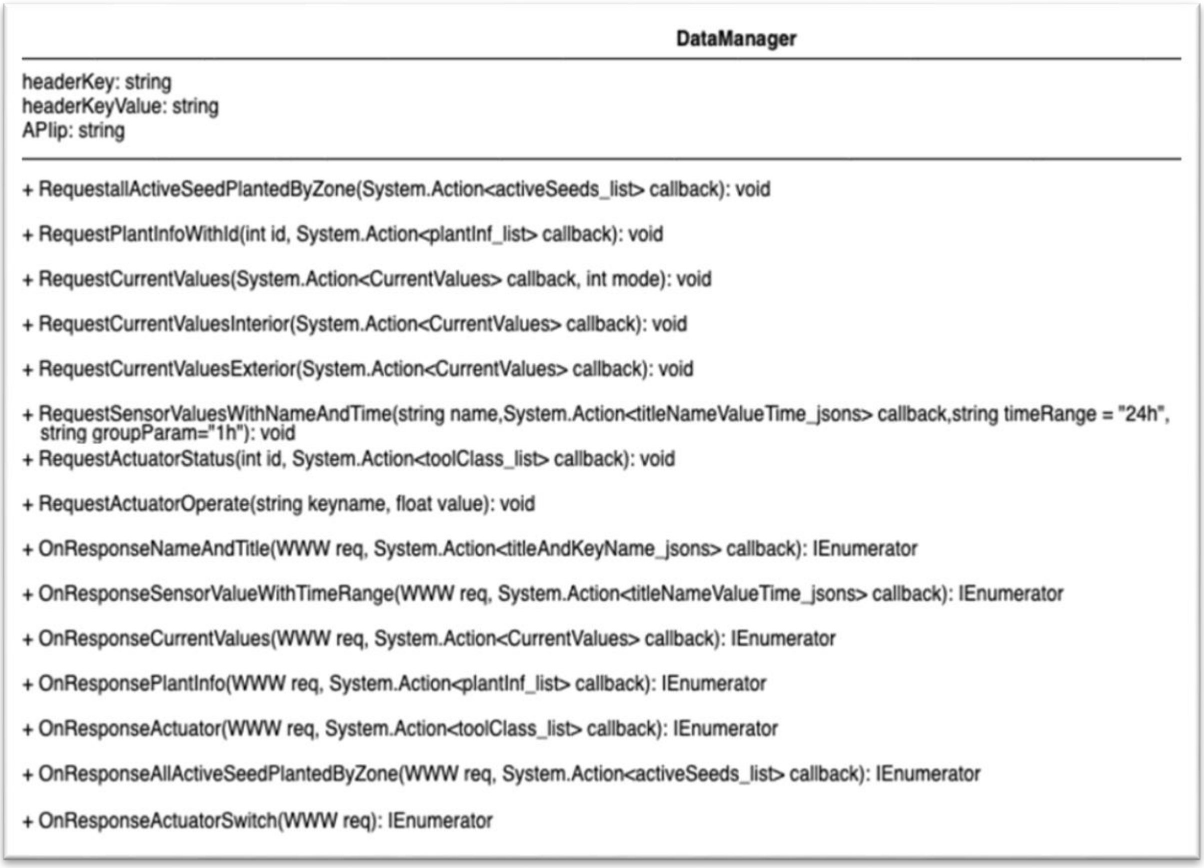
The DM is responsible for compiling the received data and translating it in an appropriate form, using the LINA objects, namely sensor, actuator, plant information object, thus enabling the system to create the IGH’s virtual representation.

The DM component was designed in collaboration with the IGH’s team, who provided a valid contract (documentation) of the API on which the DM component was based.
This component was created to provide a single link between the **API** and **Lina Framework**, which in turn allows changes to be easily applied, meaning that when the Intelligent Greenhouse’s API changes, the Lina framework maintainer has to review ONLY the **data manager** component and apply the changes.

The LINA framework communicates with the IGH API by consuming HTTP services. However, due to the nature of HTTP services, the IGH’s API may take some time to respond, or return the requested data in an unordered fashion (in the case of multiple requests).

To this end, the LINA framework needs to be able to receive the data asynchronously, meaning the system will request one data set using a function (e.g.

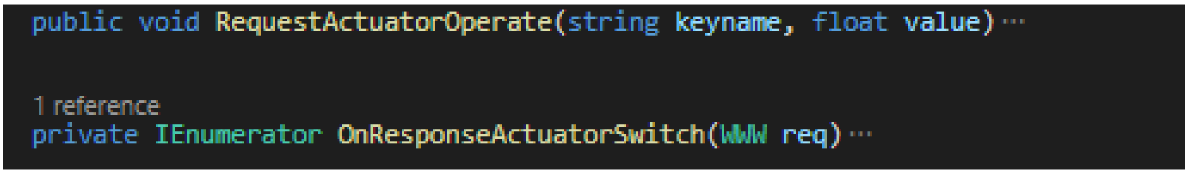
RequestSensorValuesExterior) and then complete function's execution when the requested data set is returned.



```
DataManager
-------------------------------------------------------------
headerKey: string
headerKeyValue: string
APIip: string
-------------------------------------------------------------
+ RequestallActiveSeedPlantedByZone(System.Action<activeSeeds_list> callback): void

+ RequestPlantInfoWithId(int id, System.Action<plantInf_list> callback): void

+ RequestCurrentValues(System.Action<CurrentValues> callback, int mode): void

+ RequestCurrentValuesInterior(System.Action<CurrentValues> callback): void

+ RequestCurrentValuesExterior(System.Action<CurrentValues> callback): void

+ RequestSensorValuesWithNameAndTime(string name,System.Action<titleNameValueTime_jsons> callback,string timeRange = "24h",
  string groupParam="1h"): void
+ RequestActuatorStatus(int id, System.Action<toolClass_list> callback): void

+ RequestActuatorOperate(string keyname, float value): void

+ OnResponseNameAndTitle(WWW req, System.Action<titleAndKeyName_jsons> callback): IEnumerator

+ OnResponseSensorValueWithTimeRange(WWW req, System.Action<titleNameValueTime_jsons> callback): IEnumerator

+ OnResponseCurrentValues(WWW req, System.Action<CurrentValues> callback): IEnumerator

+ OnResponsePlantInfo(WWW req, System.Action<plantInf_list> callback): IEnumerator

+ OnResponseActuator(WWW req, System.Action<toolClass_list> callback): IEnumerator

+ OnResponseAllActiveSeedPlantedByZone(WWW req, System.Action<activeSeeds_list> callback): IEnumerator

+ OnResponseActuatorSwitch(WWW req): IEnumerator
```

Figure 17: Data manager script

To achieve this the system uses *"callback" functions* which are commonly used to handle HTTP requests. This type of function is called by the system automatically. The callback function is called with the requested data set as an argument, when the API's response arrives with the requested data set. To create these callback functions in C#, "*System.actions"* were used. The created callback functions are used as data handlers functions, and are responsible for handling the data retrieved from the API *on Response (when the IGH API returns the requested data set)*. The IGH API returns a *JSON file* on response that the LINA framework has to de-serialise and transform into system objects using the data handlers and entities mentioned above.



```
public void RequestActuatorOperate(string keyname, float value)···


1 reference
private IEnumerator OnResponseActuatorSwitch(WWW req)···
```

Figure 18: RequestActuatorOperate

When the user needs to issue a command towards the IGH, the system invokes a Request operate function using the actuator's *keyname*, and the value the user wishes to set the actuator to. For example: [User closes window in virtual environment]-> System:

roofWindow, 0 -> **closes roof window in** IGH) then, when the API returns a **success** message, the system alters the actuator's state inside the Virtual Greenhouse to match the user issued command. If there is an **error,** the system informs the user that it was unable to complete the request] (see Figure 18).

5.2.1.2 FC2 - Generator

The Generator component is responsible for generating the Virtual Environment (VE) that will enable the user to remotely monitor, operate and manage the IGH. The Generator component is responsible for a) automatically generating the VE, and b) generating the 3D GUIs that the user will use during interaction. The LINA framework generates the VE as soon as the user select the IGH she wishes to visit from a dashboard.

Upon selection, the LINA framework invokes the Virtual Greenhouse Generator (VGHG) (see Figure 19) component which is responsible: a) for handling and displaying ambient interior and exterior information, b) for **actuator control** for the entirety of the greenhouse (e.g. open all windows), and c) for managing and monitoring general information for the IGH (e.g. notes, species currently cultivated etc.). Towards this end, the VGHG pulls data from the **data manager (**DM**)** component and uses **data handlers** (ambient, cultivation zone, greenhouse information) to generate the VGHG asset and invoke sub-generators to "fill" the VGHG with the IGH's data.

This component is also responsible for handling and displaying **cultivation zones** and creating the grid on which the cultivation zones' representations will be placed. Upon creating the grid and **Zone assets**, the **Zone Generator** is called to complete the process of displaying the data (**sensors**, **actuators/ tools**, **information**) for each zone and creating the **3D GUIs** and **controls (tools)**.

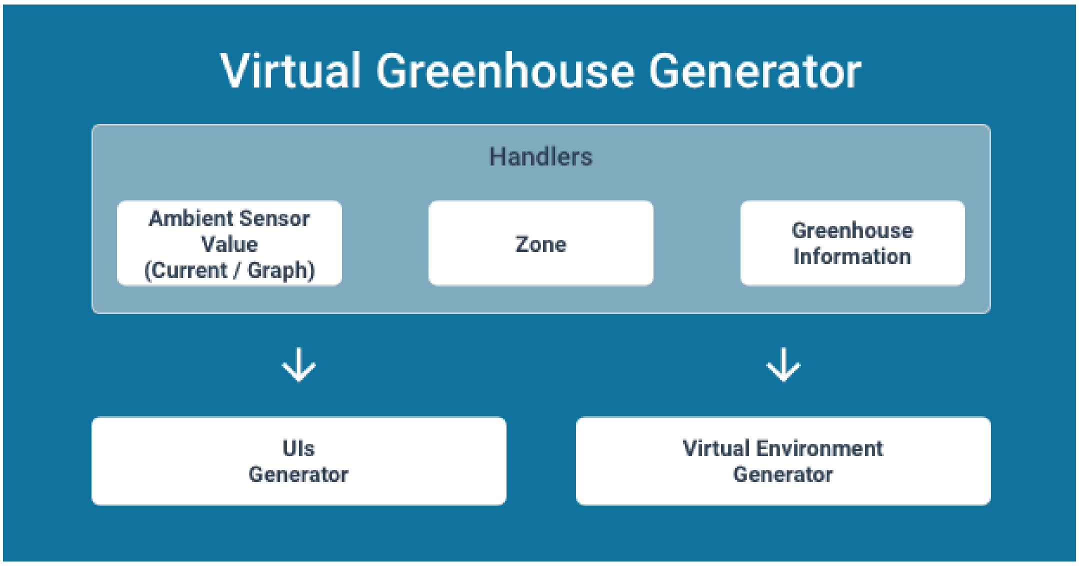

Virtual Greenhouse Generator
Handlers
Ambient Sensor Value (Current / Graph)
Zone
Greenhouse Information
UIs Generator
Virtual Environment Generator

Figure 19: Virtual Greenhouse Generator Architecture

As depicted in Figure 20, the **zone generator** is responsible for displaying each cultivation zone's **information** and **control GUIs.** The generator pulls data using **data manager** (using the VGHG handler) from the Intelligent Greenhouse's API and the Zone's ID, which was acquired from the VGHG in the previous phase of the generation.

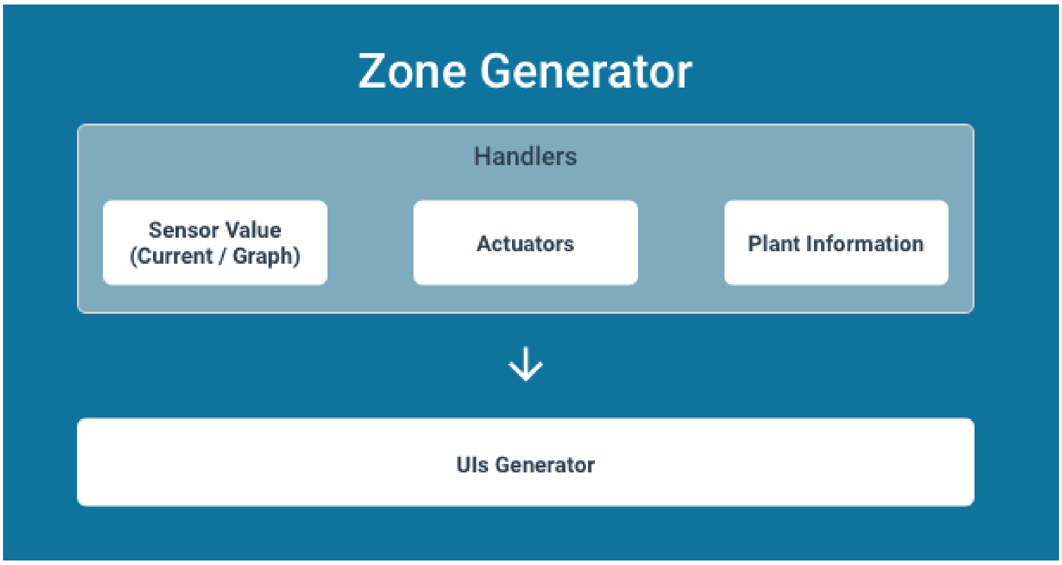


Figure 20: Zone Generator Architecture

During the **Cultivation zone** generation, the system uses the sensor object to fill sensor GUI templates in order to create the *current value panel* (see Figure 21) representation, which is then attached to the zone's sensor section.

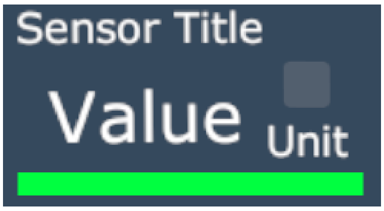


Figure 21: Sensor template

For each sensor installed in the IGH's zones, the system creates a *sensor panel*, which includes sensor's *current value, unit, icon and name,* as well as colour-encoded values that inform the user if they comply with the ideal cultivation variables for this sensor for the plant currently being cultivated in this zone (see Figure 22).

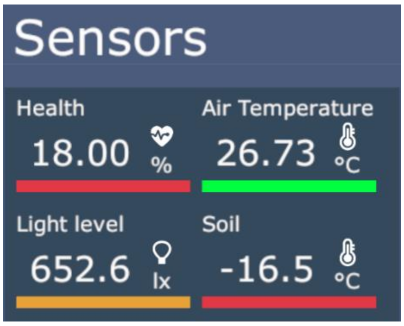
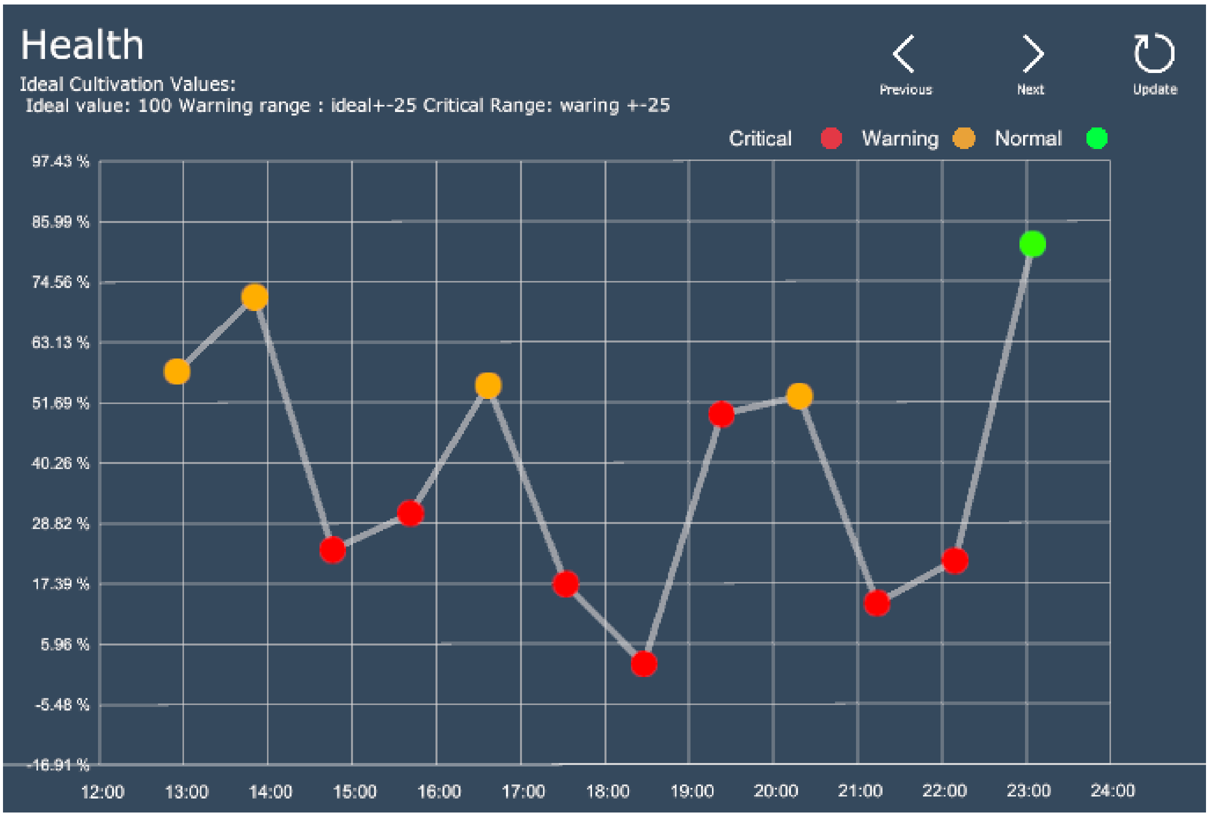


Figure 22: Sensor's current values with colour encoding

Data history of each installed sensor is depicted in a graph window that displays colour-encoded values for a specific period of time (see Figure 23). The user can access the graph by selecting the *sensor panel*, use the graph window's controls to move further **back** and **forth** (until present day) in time, or **update** to receive the latest values (values are also automatically update every minute).

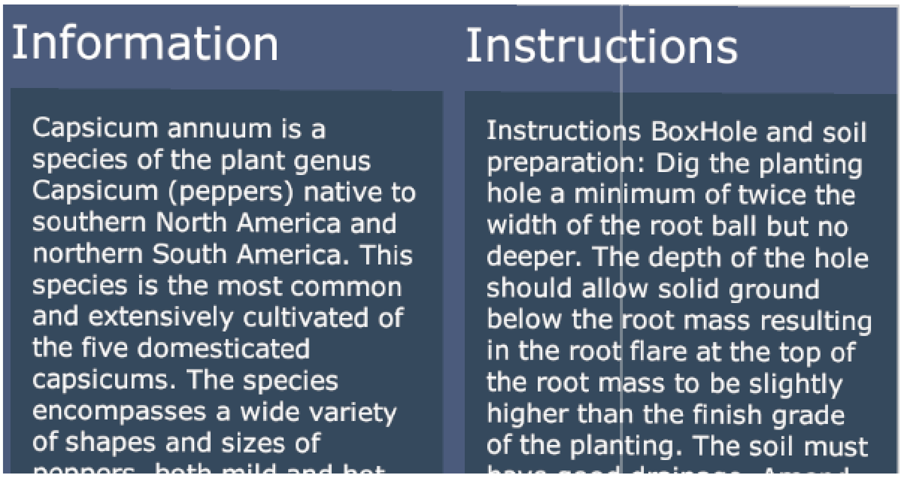


Health

Ideal Cultivation Values:
Ideal value: 100 Warning range : ideal+-25 Critical Range: waring +-25

Figure 23: Health graph window example

In order to decide the colour of each represented value, the system uses the sensor's *keyname* field (e.g. airTemp), to look up the ideal cultivation values in the *plant information ideal values dictionary* of the P*lant Information* object (the PIO is attached on the asset by the *data hander* during the generation process). Upon retrieving, the ideal values array, the generator uses it to colour encode the values that were provided by the DM component. Color encoding is also used for the values of the graph window that is generated for every installed sensor.



Information

Capsicum annuum is a species of the plant genus Capsicum (peppers) native to southern North America and northern South America. This species is the most common and extensively cultivated of the five domesticated capsicums. The species encompasses a wide variety of shapes and sizes of peppers, both mild and hot

Instructions

Instructions BoxHole and soil preparation: Dig the planting hole a minimum of twice the width of the root ball but no deeper. The depth of the hole should allow solid ground below the root mass resulting in the root flare at the top of the root mass to be slightly higher than the finish grade of the planting. The soil must have good drainage. Amend

Figure 24: Information and Planting Instructions

The system then attaches the PIO to the cultivation zone's *Information* section, and proceeds to create the list of *Cultivation Instructions* which is also attached to the zone 3D GUIs.

5.2.3 FC3 - Virtual Environment

The Virtual Environment (VE) component as explained in 3.7.1 is separated in two scenes: a) *Dashboard scene* the user uses to access the VGH, and b) the VGH environment (scene). The VGH scene is generated in three (3) parts: i) the *scenery,* ii) the *greenhouse asset,* and iii) the *cultivation zone asset.*

In more detail, the **scenery** (see Figure 25**)** is created using **Unity** world building tools. It aims to provide a natural environment for the user to interact with, which in turn enhances immersion. A fence was added to avert the user from unintentionally wandering off. Through the teleportation mechanism, the user can explore the scenery (which most users did during the evaluation).

The **scenery** could be easily enriched, by adding small or larger assets, which will make it more natural and further, enhance immersion. However, in this version of the framework, the scenery was made as simple as possible to avoid creating distractions and clearly evaluate the user's interaction with the VGH.

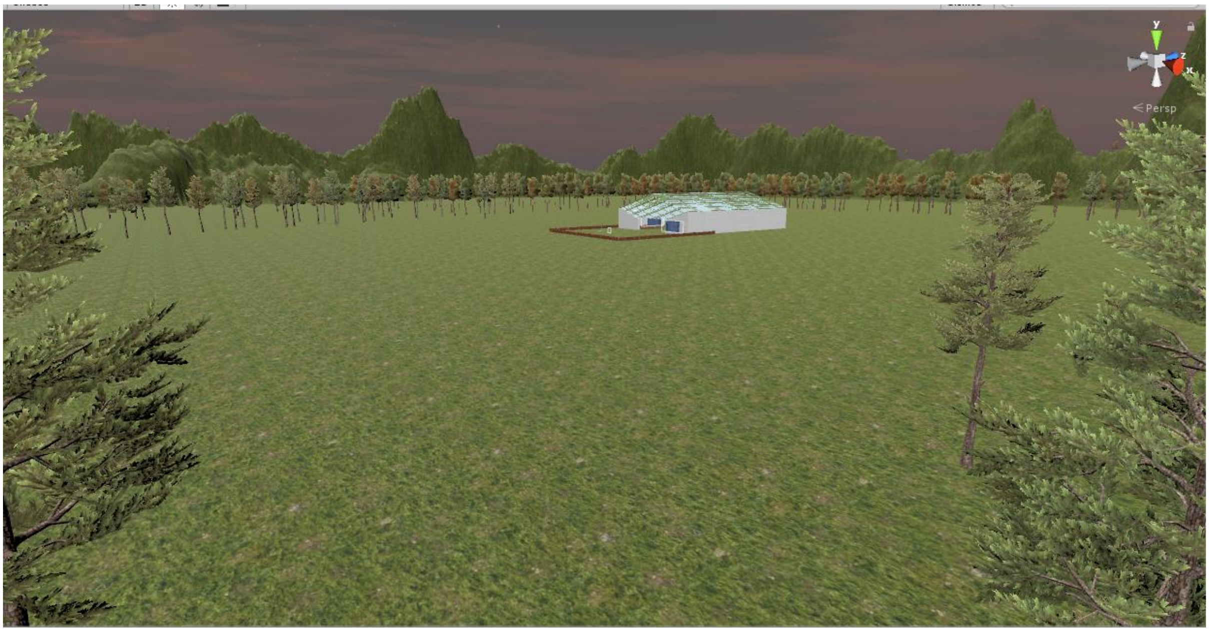


Figure 25: Virtual environment scenery

The **greenhouse** asset was created to house the **cultivation zones** and provide a **virtual representation** of the **IGH** (Figure 26). The asset changes size depending on the **cultivation zones grid** to ensure all **zones** are housed inside the Virtual Greenhouse.

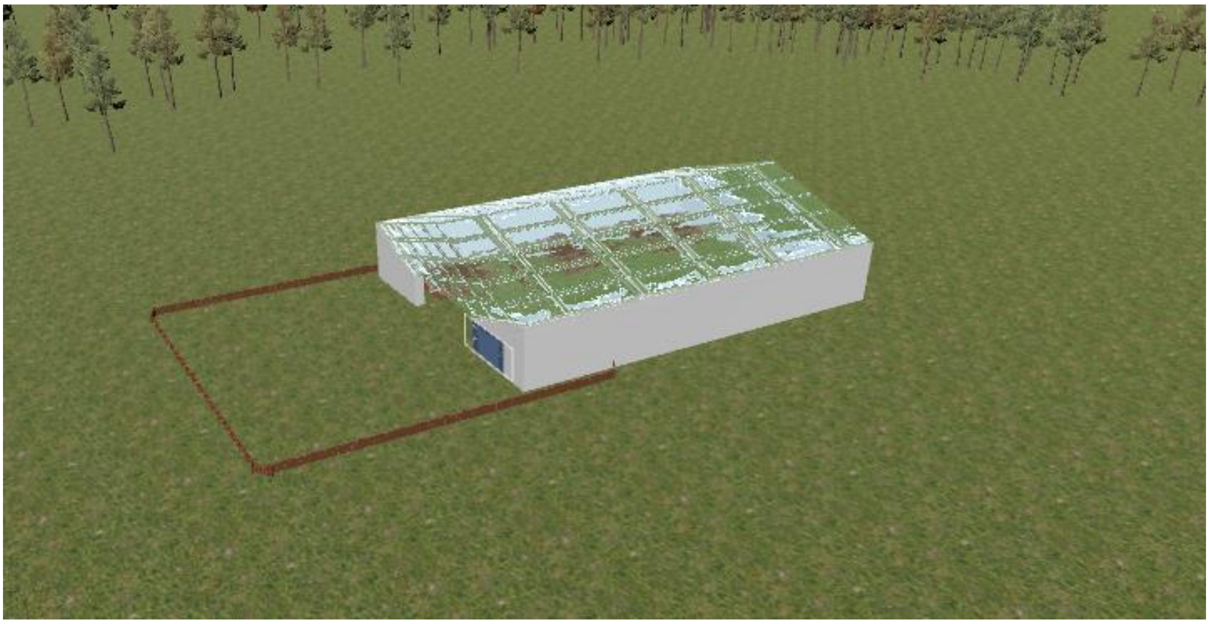


Figure 26: Virtual Greenhouse asset

The grid size is decided by the **generator** component; the **cultivation zones** are added on top of each grid placeholder. The **cultivation zones** are placed inside the greenhouse asset in the order retrieved by the **data manager** component, meaning the **generator** receives a list of zones and places them in the greenhouse. The grid (Figure 27) starts from the VGH's entrance, meaning that first cultivation zone is placed on the **right** of the entrance, the second on the left, etc.

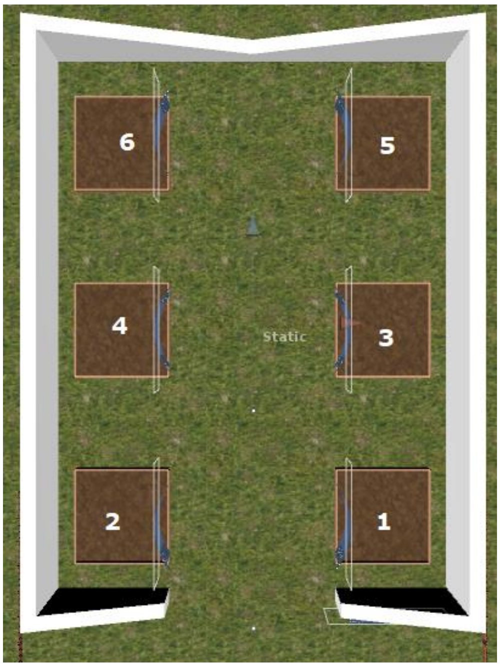


Figure 27: Grid Layout example

The **cultivation zone** asset (see Figure 28) represents the cultivation zones inside the IGH. Each cultivation zone, as explained above (plant information), represents one type of plant cultivation. The size of the cultivation zone asset is not related to the actual size of the cultivation zone inside the IGH. This was done intentionally to allow the user to monitor larger cultivation zones with the same effort as smaller ones, and to ensure the system will require minimal effort from the user regardless of the physical size of the zone.

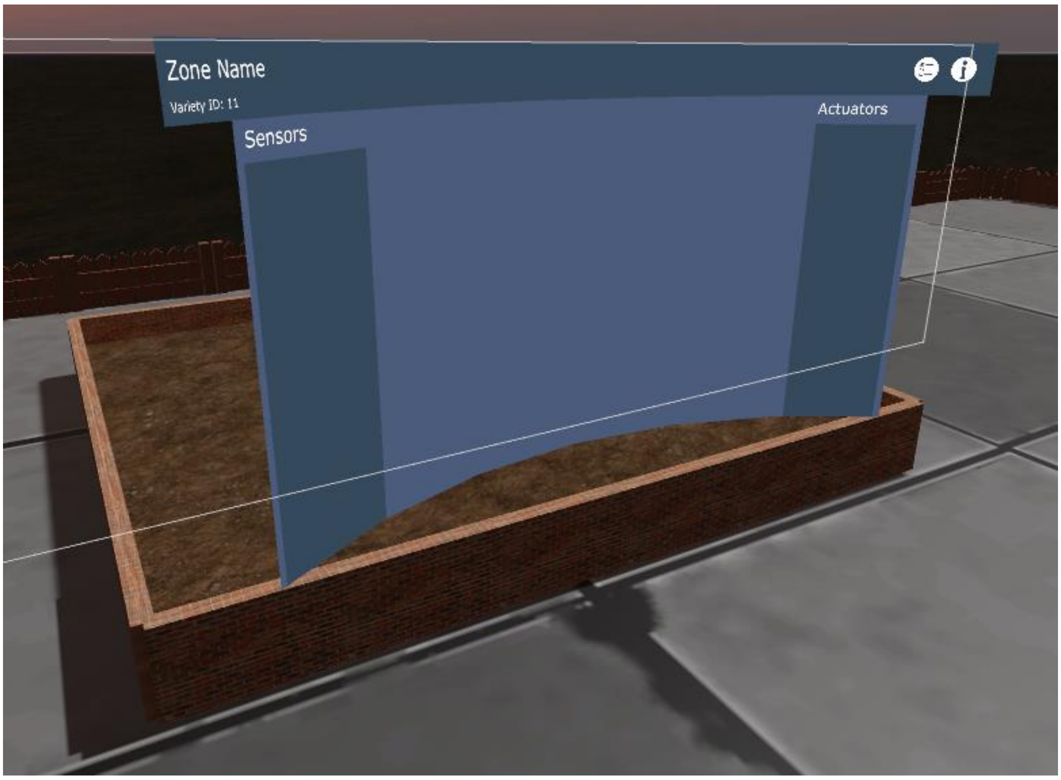


Figure 28: Cultivation zone asset

Each **cultivation zone** is accompanied by the cultivation zones **3D GUIs,** which display the zone's information and controls. In future versions of the system, plant models for each type of plant will be created and added in the zone representation, which will grow as the time progresses; at the beginning of the crop, seedlings, after a week green sprouts and so on, as for example presented in the Farmville [22].

Finally, the **dashboard scene** allows the user to quickly review general information about each greenhouse (e.g. **health state, included zones**) and to enter the Virtual Greenhouse of her choice (e.g. Figure 29).

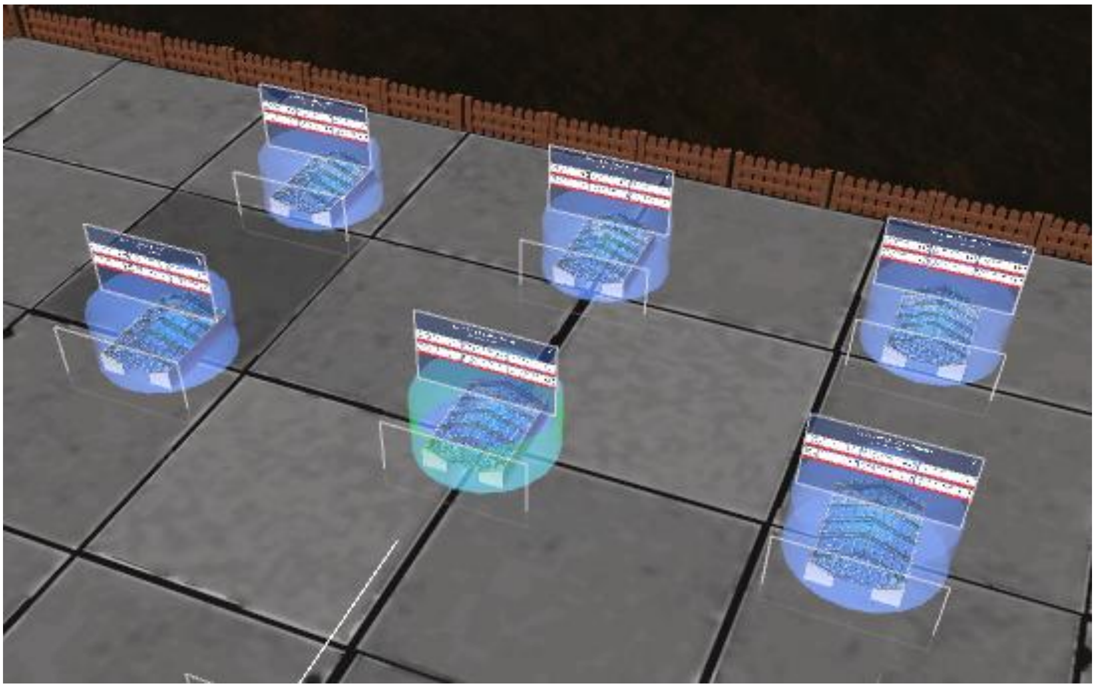


Figure 29: Dashboard scene example

The dashboard scene is also used for the "*first use scenario*", which is triggered on the first access by the the user in order to provide a short tutorial of the system controls and an initial configuration that the user can change at any time.

5.2.4 FC4 - 3D Graphical User Interfaces

In order to curve the 3D GUIs canvases, an asset was modified and used from the Unity asset store. A C# script was created, on top of this asset, to allow for interaction with the curved 3D GUIs, using the implemented interaction techniques.

However, in order to avoid difficulties in interacting with **canvases** (screens), the curved option was only applied when necessary, i.e., in cases where the canvases where "too wide", which would in turn force the user to move while in front of the canvas in order to interact with the canvas.

Finally, the 3D GUIs act as forms to be filled in, by the objects the Lina framework uses to display information (see Figure 30), as explained above in the generator component (4.2.3). This enables the system to display the information without the need to "know" what is being displayed, and provides the system with the ability to modify the GUIs with minimal effort (change the template to change the GUIs).

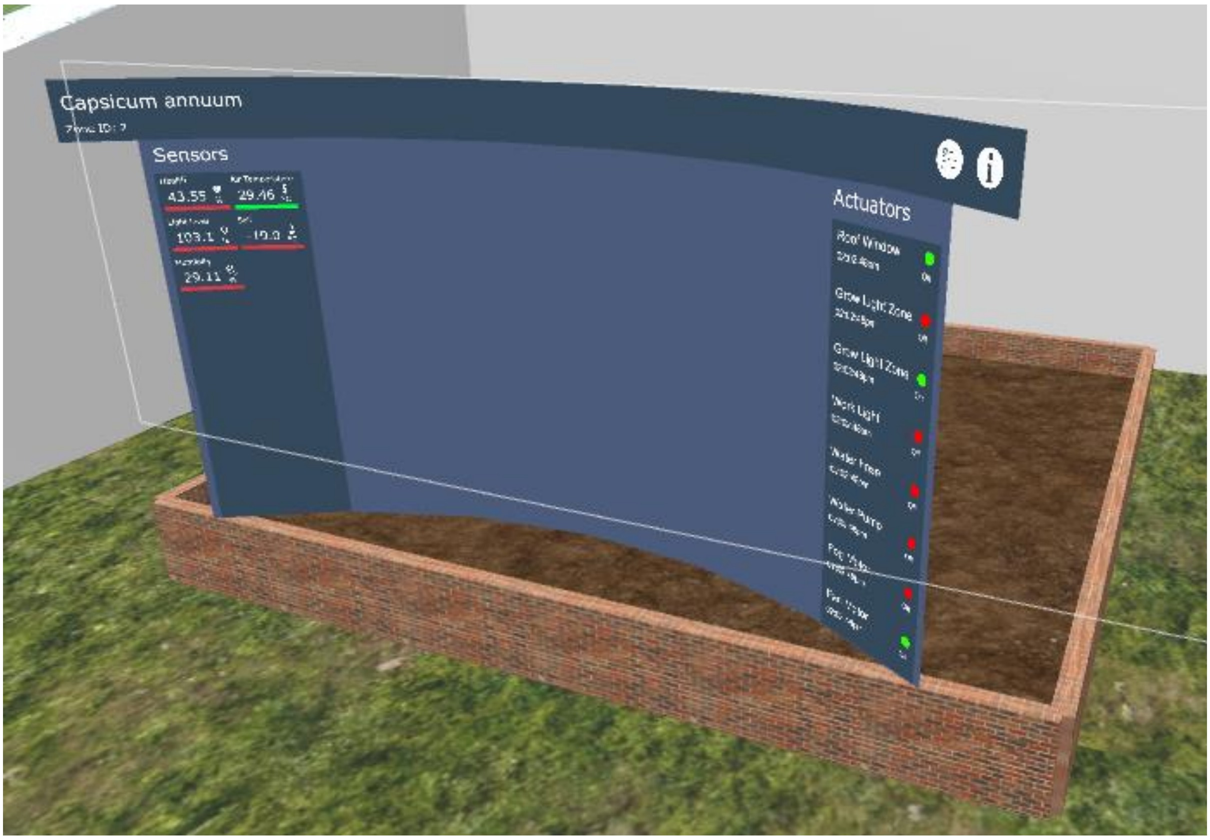


Figure 30: Cultivation zone with 3D GUIs

5.2.5 FC5 - Interaction Techniques

To enable the user to interact with the system, the interactions techniques were selected and implemented as described below. The user primarily use the *Oculus* controllers to interact with the VE and control the system, while receiving feedback through the VR- headset.

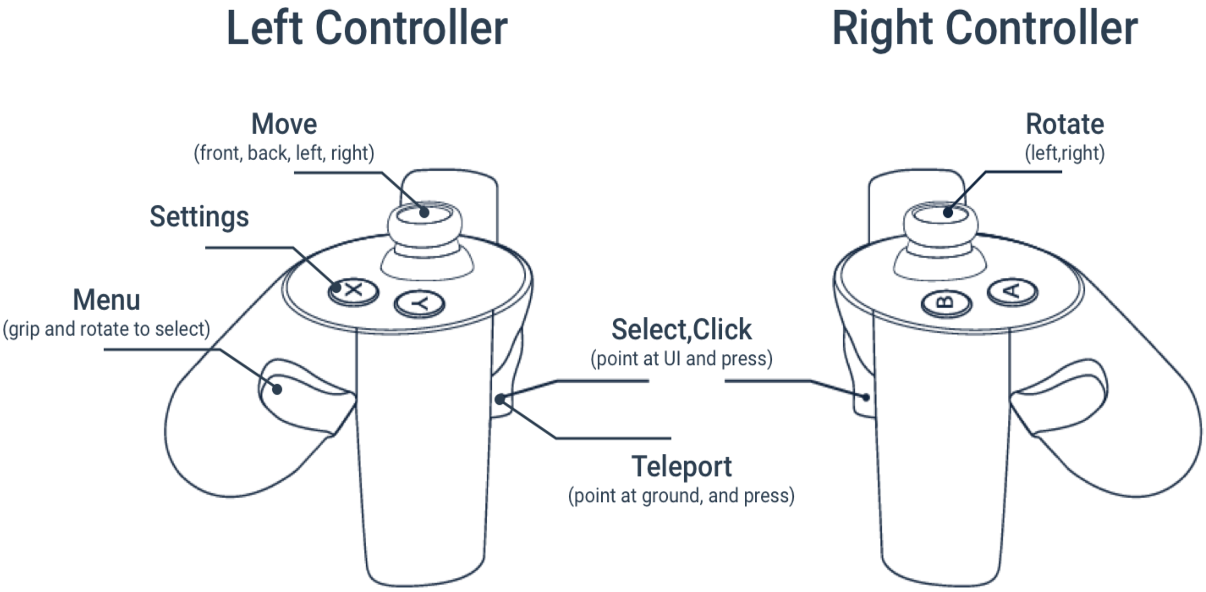


Figure 31: System controls

5.2.1.5 Navigation

The system provides the user with two (2) ways to move within the virtual environment: a) a combination of **both** thumb sticks *left to move* and *right to steer*(**left**/**right** controller) mechanism, **and** b) a **point** and **click** mechanism.

The dual thumb stick combination was introduced to mirror movement as it is being commonly handled in gaming industry over the past two decades. In computer gaming, "*wasd*" movement is a standard nowadays, while orientation and rotation are handled using the mouse. The majority of users use their left hand to navigate using the *wasd* letters on the keyboard and their right hand to use the mouse [36], no matter which one their dominant hand is. It is important to note that, even though most games use as default settings these methods, users are able to remap (in most cases) these settings / buttons to their personal preferences.

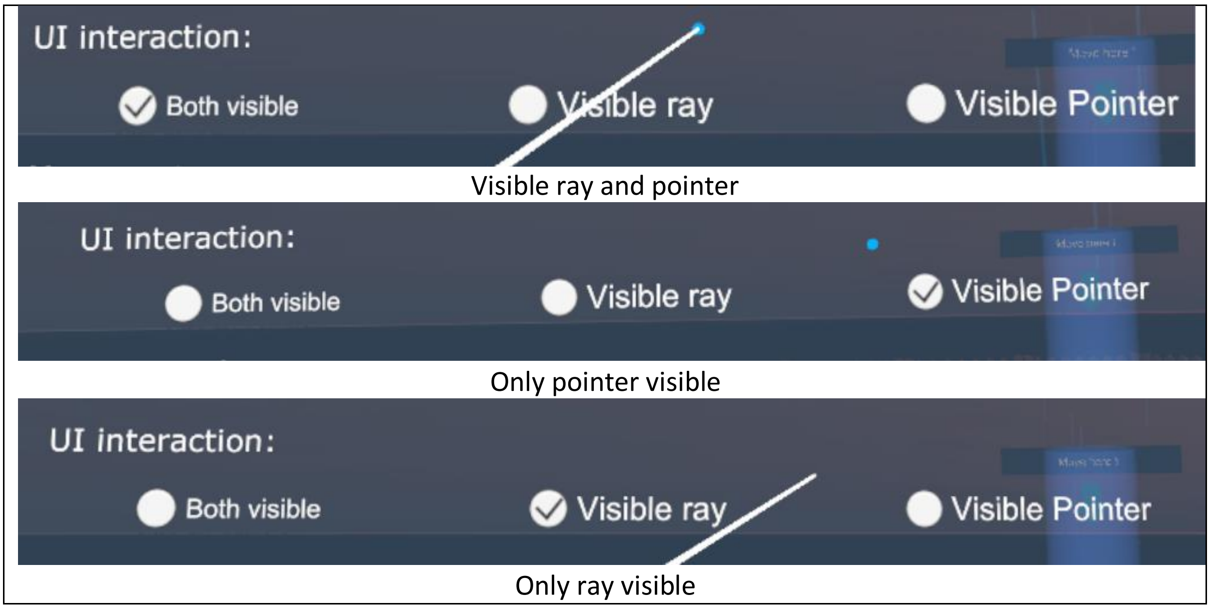
Console games usually, use the **left thumb-stick** to move **front** – **back** – **side** the player, and the **right thumb-stick** to change the player's orientation or move the camera. That kind of games require precise movement and a small learning curve to be interesting and fun. The LINA framework "transfers" the movement mechanisms used in console games and attempts to *gamify* a portion of the movement mechanisms, in order to **decrease** the learning curve, and make the movement more **interesting** and **precise** in order to enable the user to explore the environment in a **fun** way (as per the advantages of gamification described in [51]).

The point and click movement mechanism, or *click to teleport***,** is the technique used by the **Oculus Rift ©** platform, as well as the **SteamVR** plugin that the LINA framework uses, to connect the VR head set and controllers to the system.
This technique uses the **left-controller** and **pointing at** the location the user wants to move, and the index trigger to click and instantly transfer. A small pointer provides feedback to the user where the location that is currently being pointed at is (click and transfer*)*.
To ensure that the sudden movement does not affect the user in a negative way, i.e. create nausea or motion sickness or disorientation [62], the LINA framework implements a "*blink*" effect. When the transfer initiates, the system's camera *fades to black* over a selected period, the user is transferred while the camera is dark and then the camera *fades back to normal*. This creates the illusion of blinking (**close** eyes to black and **open** to normal view), due to the fact the user is transferred while blinking, and the feeling of disorientation is reduced. The user is able to modify the time the camera takes to fade (**teleport fade time** setting) from her settings to make the transition as **slow** or **fast** as desired.

5.2.2.5 Selection and manipulation and system control

To enable the user to interact with **3D GUIs**, this work primarily focused on using the *raycasting* technique, as presented by Bowman in [12]. Raycasting is a "point to interact" technique which enables the user to point at an area in the VE by using his/hers left or right controller and select it by using the *index triggers* on the selecting controller. The virtual pointers provide the same functionality as the traditional 2D desktop pointers, i.e. hold to drag and press trigger to click, in order to facilitate a smaller learning curve as knowledge from the user's background is applied to a new task.

To provide the user with visual feedback of where and what she is “pointing at”, there are three (3) options for the user to select: a) display a ray extending from the controller towards the object the user is pointing at, b) display a pointer on top of the pointed object, c) display both ray and pointer. The user can change the interaction feedback from settings at any time (Figure 32.



UI interaction:

☑ Both visible ○ Visible ray ○ Visible Pointer

Visible ray and pointer

UI interaction:

○ Both visible ○ Visible ray ☑ Visible Pointer

Only pointer visible

UI interaction:

○ Both visible ☑ Visible ray ○ Visible Pointer

Only ray visible

Figure 32: Visible Ray and Pointer

To deploy the controller rotation gesture selector and access the radial menu, the user holds the **grip trigger** on the left controller, which in turn activates a radial shaped menu. Then the user rotates his/her controller, clock or counter clock wise, to highlight the desired menu item, and when she **releases** the grip trigger the selection is performed. The user goes through the process of **hold** - **select** (rotate) – **release** to select the item of her choice (see Figure 33) to interact the system’s menu. When the item is selected, the GUI is presented to the user who can then proceed to interact with the selected interface using raycasting.

For example in Figure 33, the selected item is the center button, and the user can **rotate clock wise** to select the next items (e.g. calendar) or **counter- clock wise** to select the previous ones (e.g. tools), then the user can release the **grip trigger** to “click” the selected item on the menu.

However, during the first heuristic evaluation, the evaluators stated that “*some menu items were hard to reach*”. Following the first heuristic evaluation, to address this comment and upon consulting field experts, the gesture was re-mapped to allow for a full rundown of all items within a margin of 170° forearm rotation. The field experts also proposed some additional modifications of the gesture, which were included in the design. Thus, the final gesture based **radial menu** selector mechanism took the form of **hold** grip trigger to activate the menu, **select** rotate left controller within the range of **-20°** (20° counter clock wise) **to 150°** (clock wise) from neutral hand position thumb facing upwards, and **release** grip trigger to select “**click**” the highlighted button/ item.

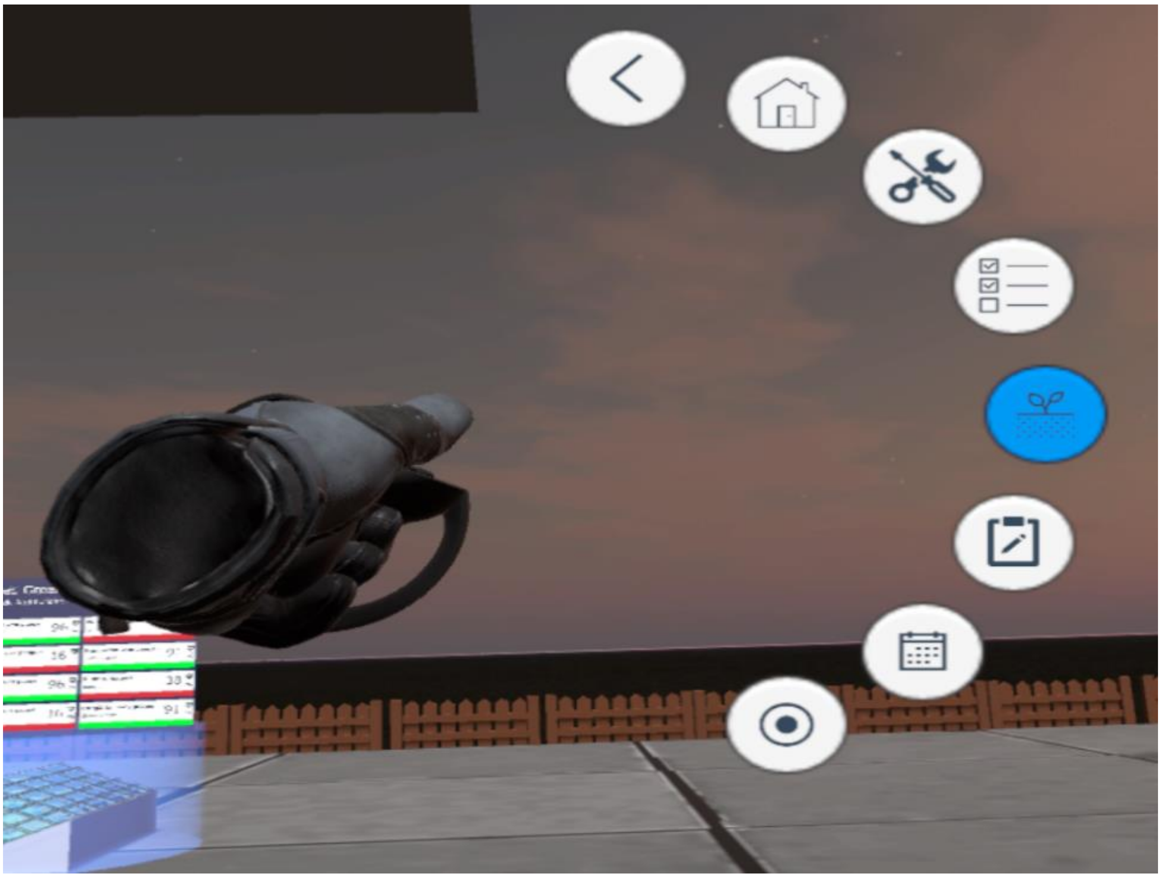
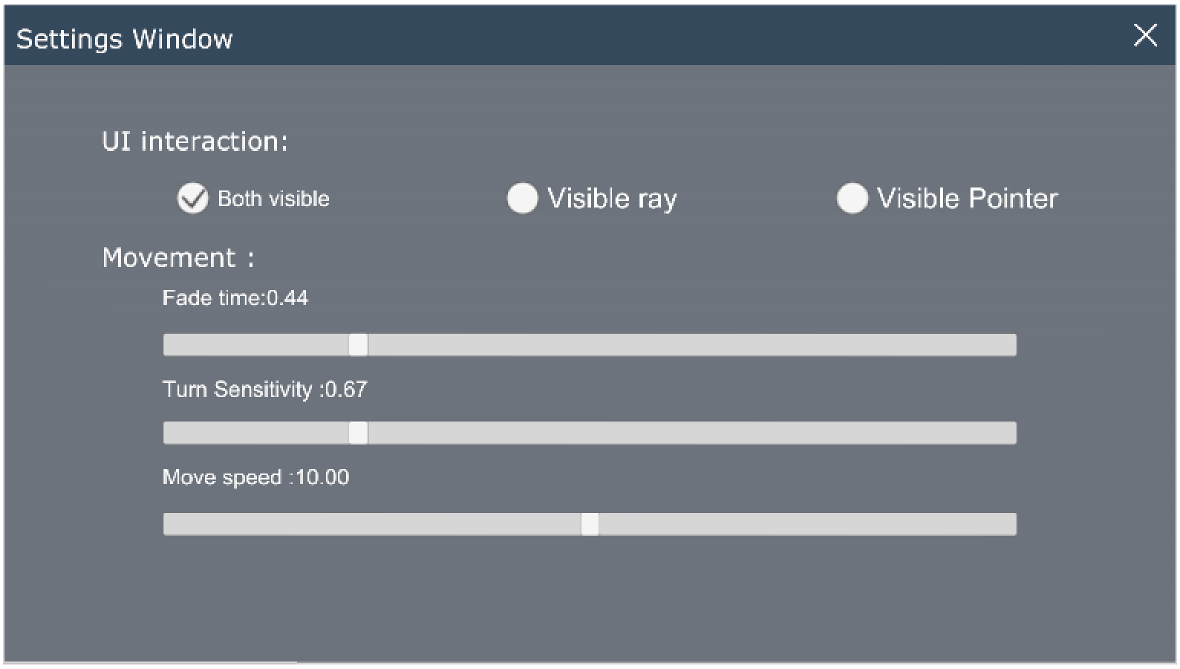


Figure 33: Radial menu - gesture example

5.2.6 FC6 - User configuration

From the “*Settings* Window” (see Figure 34), the user can adjust settings and receive real time feedback, without the need first to perform “save”.



Settings Window ×

UI interaction:

☑ Both visible ○ Visible ray ○ Visible Pointer

Movement :

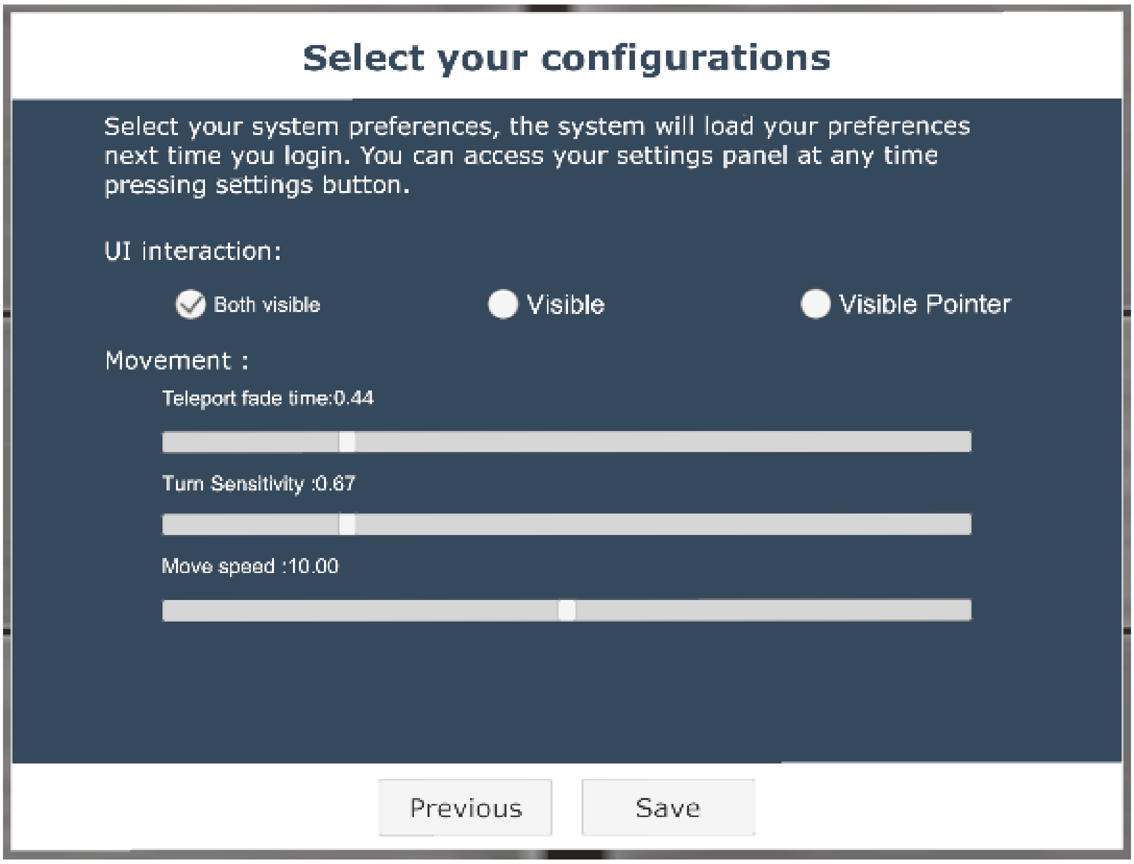
Fade time:0.44

Turn Sensitivity :0.67

Move speed :10.00

Figure 34: Settings window

This design choice was primarily made in order to enable the user to receive feedback about the configuration while making changes, and better tailor the settings to her preferences. The system saves the preferences automatically when the user closes the settings window.



Select your configurations

Select your system preferences, the system will load your preferences next time you login. You can access your settings panel at any time pressing settings button.

UI interaction:

☑ Both visible ○ Visible ○ Visible Pointer

Movement :

Teleport fade time:0.44

Turn Sensitivity :0.67

Move speed :10.00

Figure 35: Quick tutorial screen

The first time the user access the system, the “*Quick Tutorial”* screen is displayed (see Figure 35), which is a quick guide of the systems controls per Controller**.** The user can skip it if desired or continue to the first configuration screen.

The first configuration screen pre-selected default configurations variables (move speed, turn sensitivity, teleport fade time, etc.), and the user can alter these variables during the first configuration **or** later from their settings (see Figure 36).

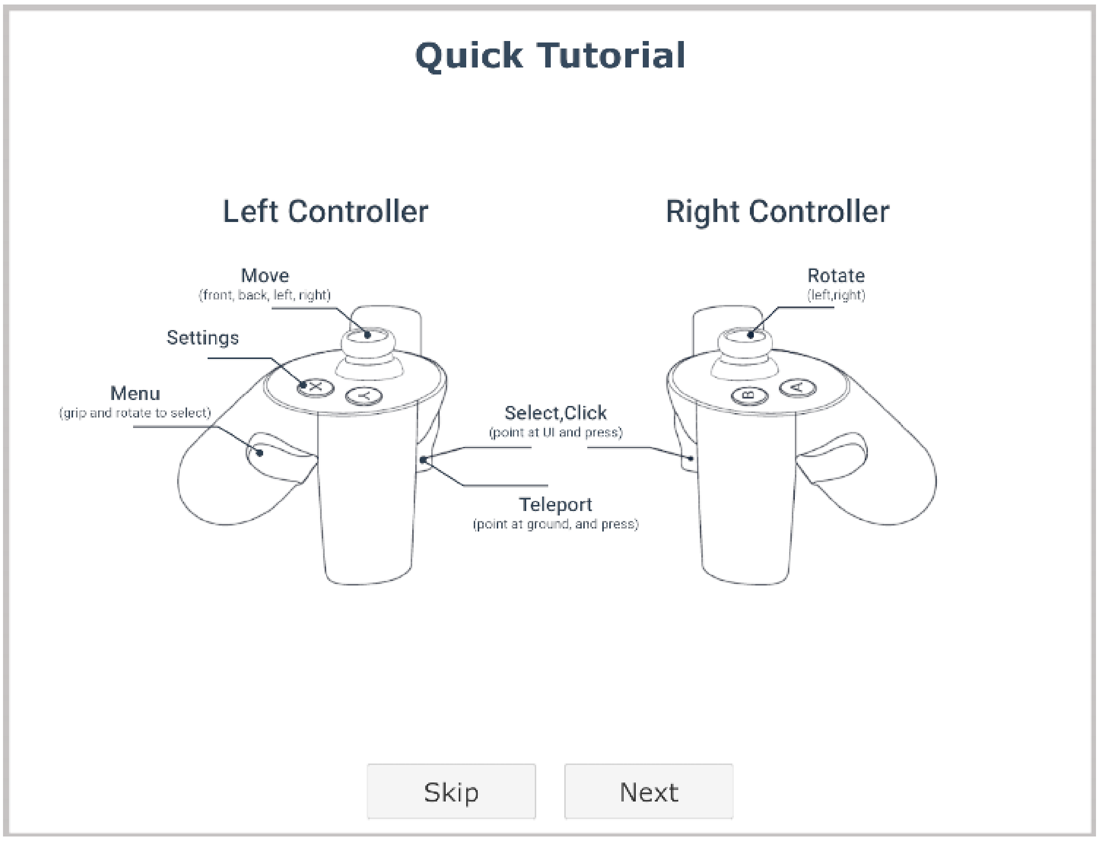


Figure 36: First configuration screen

6 FORTH's Intelligent Greenhouse as Case Study

ICS-FORTH has recently initiated AmI-Garden, a smart farming project in the framework of its Ambient Intelligence Research Programme [10]. A small experimental IoT greenhouse has been constructed and equipped with polycarbonate cover sheets and all the necessary infrastructure and hardware (automatic window-roof opening/closing, sliding door, fan installation for heating/cooling, vegetable breeding lamps etc.). Inside the greenhouse, a network of wireless sensors is used to measure environmental conditions and parameters, such as air/soil temperature and moisture, sunlight level, soil conductivity, quality and level of chemical ions in irrigation water, etc. The sensors communicate through IoT gateways to the greenhouse's data centre for storage and post-processing.

The system comes with pre-installed agricultural scenarios, a set of activity flows based on environmental conditions that are ideal for each plant species and are monitored in the greenhouse. The scenarios currently contain parameters to predict common diseases of the plants, as well as unexpected changes in the greenhouse's microclimate. For example, the irrigation process is built as an agricultural scenario using data from current plant status and past data in order to establish the optimal amount of water to irrigate. The parameters of this scenario are based on specific plant breed and environmental variables.

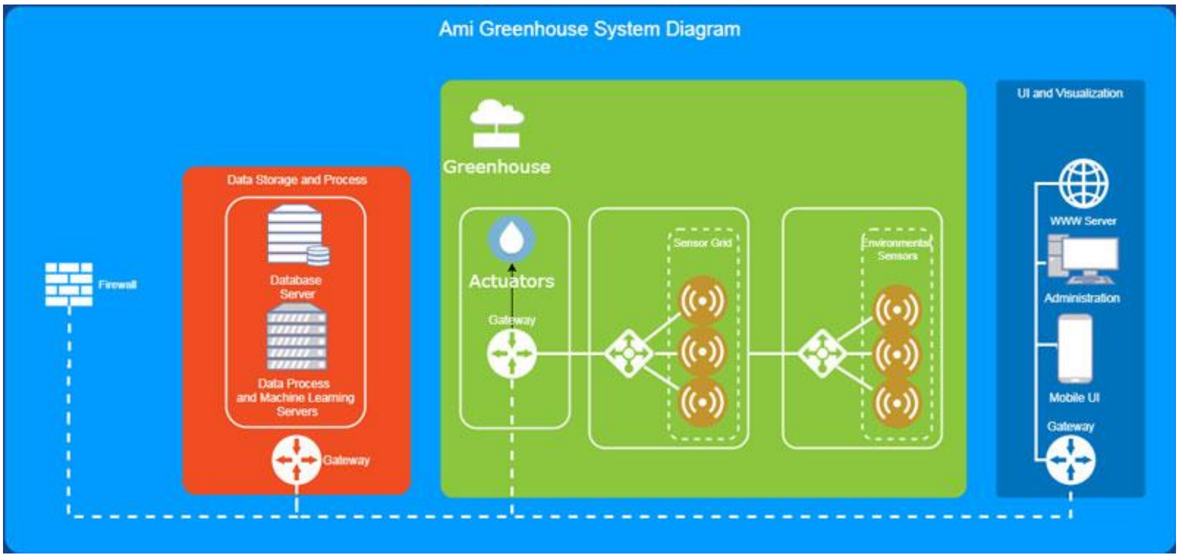


Figure 37: AmI Garden Diagram [10]

To enable ideal cultivation environmental parameters monitoring and regulation for each plant species, the greenhouse is separated in cultivation zones. Each zone contains one plant species, as well as a grid of sensors, monitoring the environmental parameters enabling automated environmental parameter regulation though the installed actuators. The AmI Garden provides access to the installed actuators, to cultivation zone's data, and ambient environmental parameter's sensors over RESTful HTTP services (see Figure 37).
In order to validate the LINA framework, a case study has been developed based on AmI-Garden.

6.1 AmI’s Greenhouse API communication process

The data collected by the AmI Greenhouse system are available through a Gateway, which connects the database with WWW server through an Administration hub as explained in Figure 37. This hub provides an API with a set of Restfull HTTP services in order to allow external users to access the IGH information and relay commands towards the IGH.

The process of requesting information from the API initiates with requesting all user's IGHs, by using the user’s “*userId*” currently stored at the AmI’s Greenhouse database, and then the API returns a JSON file containing an array with all the IGHs (see Table 1).

Table 1: fieldsByUser

Route	Response
fieldsByUser?userID=”userId”	array : **{** **Id :**bigint **name:**string **address:**string **latitude:**string **longtitude:**string **}**

Upon retrieving all the IGHs owned by the user, the party interested in the information, selects the IGH of interest that the AmI system calls *“fields”*. In the conducted case study, it is the LINA’s frameworks Data Manager Component. The interested party then uses the IGH’s “*id”* to request all information for the selected IGH and the API returns a JSON file containing an array with all the cultivation zones currently stored for this IGH as depicted in Table 2.

Table 2: zonesByField

Route	Response
zonesByField?fieldID=”fieldId”	array : **{** **id :**bigint **name:**string **health:**string **fieldName:**string }

At this point the interested party can also request all internal and external environmental variables by using the “allSensorsValuesInterior” and “allSensorsValuesExterior” methods. These methods return JSON files containing arrays with the interior and exterior sensors. The “switchActuatorsStatus” can also be invoked at this point to request actuator information and returns a JSON file containing actuators keynames, their status (value) and a timestamp of “when” the information was requested (see Table 3).

Table 3: allSensorsValuesInterior/Exterior

Route	Response

allSensorsValuesInterior	array : {
	"title":string
	"categoryID":int,
	"categoryName":string,
	"categoryKeyName": string,
	"locationID":int,
	"locationName":string,
	"locationKeyName":string,
	"keyName":string,
	"value":int,
	"time:datetime
	}

allSensorsValuesExterior	array : {
	"title":string
	"categoryID":int,
	"categoryName":string,
	"categoryKeyName": string,
	"locationID":int,
	"locationName":string,
	"locationKeyName":string,
	"keyName":string,
	"value":int,
	"time:datetime
	}

switchActuatorsStatus	array : {
	"title": string,
	"keyName": string ,
	"value": boolean,
	"time": datetime
	}

At this point, the collected information includes the general information of an IGH, such as environmental parameters, provided by the installed sensors (e.g. air temperature, air humidity) and the installed actuators (e.g. roofWindow=open), as well as, zone information such as name, health and id. To retrieve more specific information about the cultivation zones, more requests are required to retrieve the installed sensors, their current and previous values and the actuators installed.

To retrieve information for a specific cultivation zone, the interested party requests information for a) the zone data to retrieve information about the cultivated plant name (common name) health, etc. , b) plant specific information for the cultivated plant, such as id, ideal values, description, cultivation instructions etc., c) installed sensors to retrieve their “keyName”, and current values d) previous values for each “keyName”, for a specific time range (e.g. last 24h), and e) the installed actuators to retrieve an array with all installed actuators and their status. To achieve this, the interested party must request the data using the cultivation zones “fieldId” and specific API requests for each information separately, and process the retrieved JSON files to create image of the current state of the cultivation zone (see Table 4).

Table 4: Zone Information provided by the API (Routes/Reponses)

/zoneData?zoneID=”zoneId”
array : **{** "id":bigint "plant":string, "zoneName":string, "health":string, "datePlanted":datetime, "planted_units": string, "crops_amount": string, "crops_volume": string }
/allPlantInfo?plantID=”plantID”
array : **{** `id` bigint(20) , `name` varchar(300) , `description` text , `varietyID` bigint(20) , `plantICON` varchar(45) , `minSoilTempDay` double , `maxSoilTempDay` double , `averageSoilTempDay` double , `minSoilTempNight` double , `maxSoilTempNight` double , `averageSoilTempNight` double , `minSoilMoistDay` double , `maxSoilMoistDay` double , `averageSoilMoistDay` double , `minSoilMoistNight` double , `maxSoilMoistNight` double , `averageSoilMoistNight` double , `minAirTempDay` double , `maxAirTempDay` double , `averageAirTempDay` double , `minAirTempNight` double , `maxAirTempNight` double , `averageAirTempNight` double , `minAirHumidityDay` double , `maxAirHumidityDay` double , `averageAirHumidityDay` double , `minAirHumidityNight` double , `maxAirHumidityNight` double , `averageAirHumidityNight` double , `lightLevelDay` double , `lightLevelNight` double , `averageBarPressure` double , `windSpeed` double , `WindDirection` double , `rainLevel` double , `sunLevel` double , `plantingInstructions` longtext, `plantCaring` longtext, `plantFullName` varchar(145) , `planTips` longtext }
/sensorsValueByCatIDLocID?sensorCatID=1&sensorLocID=”zoneId”
array : **{** "title":string "keyName": string, "value":int, "time":datetime **}**

/sensorAllValuesByKeyNameTime **?sensorKeyName=”sensorKeyName”&sensorTimeRange=”TimeRange”**
array : **{** "title":string "keyName": string, "value":int, "time":datetime **}**
/switchActuatorsStatus?sensorKeyName=roofWindow **Or** /switchActuatorsStatus (without parameter return all the switch Actuators Status)
array : **{** "title": string, "keyName": string , "value": boolean, "time": datetime }

Finally, upon collecting all this information, an external source can relay information towards the IGH and alter the status of the installed actuators using the “switchActuatorAction” command with actuators “keyName” (see Table 5).

Table 5: Switch Actuator Status

Route	Response
switchActuatorAction?sensorKeyName=”keyName”	“200ok”

6.2 Integration issues

The primary integration issue and in order to access the zone information, stemmed from the AmI’s Greenhouse API structure - was the required series of functions calls, which was explained in the communication process in section 5.1.

For example, the LINA framework provides the users with colour encoding of sensor values, to allow them to easily identify if a value is within the cultivated plants requirements. As the user reviews the previous values for the *humidity sensor* of a specific cultivation zone, the LINA framework uses colors to inform them if the values displayed are as ***Normal:*** the ideal humidity level for this plant, **Warning**: ideal –the warning range, **Critical**: warning – the critical range.

The series of function calls that required are: a) retrieve all cultivation zones and their id, b) use the plant’s id to retrieve the plant’s information, c) retrieve all sensors installed in the zone and their keyNames, d) use the keyName of the sensor of the interest (e.g. humidity) to retrieve the sensor’s values from the database, e) create the correct keyword using the sensor’s keyName and manipulate it to search for the ideal values (e.g. “averageAirHumidityNight”), and f) use retrieved value for colour encoding of all values. However, due to the nature of asynchronous programming and HTTP-services each of these requests could be delayed or dropped, meaning the rest of the calls could not be continued, or that there would be delays in the information the user requested leading to the user not being able to access the information of interest when requested.

LINA framework uses the PIO at the beginning of the session and cashes the data for later use. The data are updated when the cultivation is changed or the plant's information is updated in the AmI's database. However, the previous values cannot be cashed from previous sessions. To this end, when the user enters the VGH, the LINA framework retrieves the previous values for the last 24h per hour, for each sensor in each cultivation zone, to avoid delays during the users' review. In this way, the LINA framework "hides" the delays caused by the server, by requesting the data and preparing them for the users' review. The user can always use the update button to request the latest values.

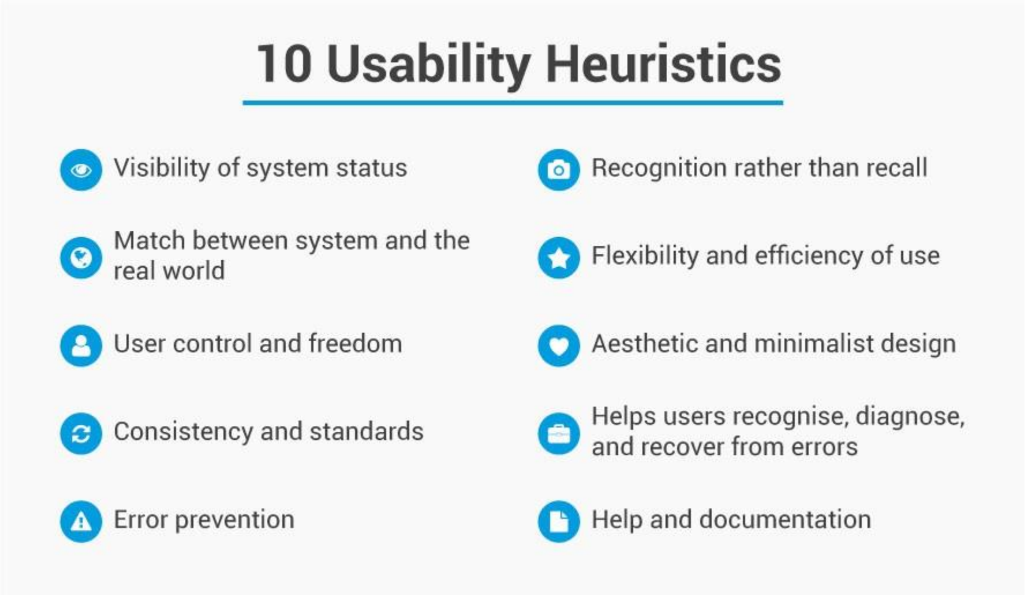
Finally, during the second heuristic evaluation of the LINA framework, no plants were currently cultivated in the IGH, meaning there was no content for the users to review. To overcome the lack of data, a generator was developed that produces data dynamically. The generator, when generating the data, respects the FORTH's IGH architecture and thus allows user evaluation even if the IGH is currently empty. The evaluation of the LINA framework is presented in the following section.

7 Expert Based Evaluation

To evaluate the LINA framework, two (2) rounds of **heuristic evaluation** were conducted involving experts from different fields of expertise (UI/UX experts, software engineers, accessibility experts).

7.1 Evaluation setup

Jakob Nielsen's 10 general principles for interaction design are called "heuristics" because they are broad rules of thumb and not specific usability guidelines [75, 52]. The set of 10 Usability Heuristics for User Interface Design are one of the most used heuristics for User Interface Design and it has been used for conducting the LINA framework's assessment (see Figure 38).



10 Usability Heuristics

- Visibility of system status
- Match between system and the real world
- User control and freedom
- Consistency and standards
- Error prevention
- Recognition rather than recall
- Flexibility and efficiency of use
- Aesthetic and minimalist design
- Helps users recognise, diagnose, and recover from errors
- Help and documentation

Figure 38: Nielsen's Usability Heuristics[6]

Due to the nature of the system (virtual reality) and the inability of the experts to write down their own comments (they were wearing the VR-headset), an assistant was keeping notes of these comments during the session. At the end of the session, the experts reviewed the notes and were asked to add or remove comments, to ensure the comments were true to the evaluators opinion. All comments from all evaluators were added in a document and the evaluators were asked to grade **all** comments (including their own) with a score as follows:

0 = I don't agree that this is a usability problem at all,
1 = Cosmetic problem only: need not be fixed unless extra time is available on project,
2 = Minor usability problem: fixing this should be given low priority,
3 = Major usability problem: important to fix, so should be given high priority,
4 = Usability catastrophe: imperative to fix this before product can be released**.**

The comments then were evaluated and changes were included in the design process to improve the system and correct the design issues pointed out by the evaluators.

[6] https://www.interaction-design.org/literature/article/how-to-conduct-a-heuristic-evaluation-for-usability-in-hci-and-information-visualization

The first heuristic evaluation was conducted on a “beta” version of the system. The system was fully functional and contained three (3) parts:

a) The **quick tutorial / first use scene**: At the start of this scene, the user was greeted by the quick tutorial screen and was placed in a white room to avoid distractions. The evaluators would **skip** or continue with the **configuration.** The scene then opened to allow the user to familiarise with the movement techniques and finalise the configuration (if she did not skip). At the end of the tutorial, the user was transferred to the **dashboard scene** from where she would be able to access the virtual greenhouse, which was created for the evaluation using data from the Intelligent Greenhouse API.

b) The **dashboard scene:** This scene enables the user to **quickly review** his/ her Intelligent greenhouses and perform a quick **overall** evaluation of **the greenhouse** as well as of **zone health information** through the use of colour encoding. The user then can then use the “**intelligent greenhouse representation**” to **enter** the virtual greenhouse.

c) Finally, the **virtual greenhouse scene:** the virtual greenhouse scene is the scene that includes the context information of the Intelligent Greenhouse represented.

7.1.1 First round of heuristic evaluation

The first heuristic evaluation was conducted by four (4) experts, three (3) **UI/UX** and one (1) Virtual Reality engineer. Comments are following in the next subsection.

7.1.1.1 First round comments and tables

The comments are grouped to provide better readability and categorization.

Table 6: System controls and general comments

Comments	**Average Score**
The radial menu buttons were difficult to reach. The users had a hard time rotating their controller over 140° left or right.	3.2
The users would like to be able to select radial menu, buttons with their pointer and not only through controller rotation gesture.	3
The users did not feel confident about understanding what each radial menu icon represented. (e.g. home, tools, settings, etc.)	2.2
The system did not support single controller movement	3
The system did not support swapping the primary controller (e.g. teleport with right controller/ rotate with left thumbstick)	2.6
Settings could not be accessed from the radial menu.	2.1
Pointer (blue dot on UI elements) visibility was low.	2.4

Table 7: Scene 1- Tutorial and first configuration

Comments	**Average Score**
Sliders’ handlers were hard to see.	2.5
Tutorial assets remained present in the scene after tutorial, which was confusing.	1.6
Enter button on greenhouse dashboard looked like an exit button.	2.3
On greenhouse dashboard zone’s colour encoding was not displayed. Every zone was healthy (green) (health > =75%->green, health < 75% -> orange , <50% -> red, etc.)	2.9
The checked health icon had ambiguous meaning. (Does it mean that the plant is healthy?)	2.6
The user had to go back to first quick tutorial screen to skip configuration.	2.1

Table 8: Scene 2 - Virtual GreenhouseGraph

Comments	**Average Score**
White placeholder, near graph window title, serves no purpose.	2.3
Points on graph were too hard to hover.	2.9
The user was not able to identify the ideal cultivation values (min, max, average).	3.3
Colour encoding of value points on the graph was not clear (What was the meaning of red, green, orange)	3
Y,X axis values were not present (e.g. °C/ hour).	2.8
Y axis text overflows graph window.	2.4
Buttons (previous, next, update) highlight colour was not easy to see (too faint).	2.4
Previous and next buttons were enabled but clicking them did nothing (after all values were displayed).	2.7
Users were not confident about previous/ next buttons functionality.	2.7

Table 9: Ambient information and general controls (actuator control)

Comments	**Average Score**
Color encoding on controls was confusing. Green and red have different meaning on graphs.	3.3
Control names were not self-explanatory; the user was not sure what he was turning on/off.	3.3
Interior, exterior tabs do not look like tabs.	2.9
Interior / exterior tabs "on hover" colour is hard to read (white letters on grey background)	2.8
Sensors were not categorized/grouped.	2.8
Current values of sensors were not colour encoded, which made it hard for users to understand if the current value was normal, alarming or critical. (green, orange, red)	3.3
UI sections were not easily distinguishable (e.g. Sensor tabs, Actuator control over controls tab)	3

Table 10: General comments on virtual greenhouse

Comments	**Average Score**
Shadows on walls make the walls appear black.	2
3D object interaction was not provided as a means to operate actuators, alongside UI actuator operation (E.g. inside the zone I "use" the 3D fan object to activate the fan over this zone.)	2
Plant growth was not visualized in the virtual environment. (Farmville style)	3
Add greenhouse map, with zone location (with fast travel capabilities)	2.9

7.1.2 System revision and implemented comments

After the first evaluation round, some aspects of the system were re-designed and changes implemented, using the findings (**comments** and **scores**) of the evaluation as input. These changes were focused on correcting flaws that came to light during the first round of evaluation, and then evaluated by experts (who took part in the first round), during the second round.

The comments that were selected to be addressed from the sum of all evaluator comments, were selected by their **score** (severity), **impact on the user experience** and the **system's usability**. After the implementation of the improvements, the second heuristic evaluation took place on the improved version of the system.

7.1.3 Improvements

The comments and related improvements that were implemented are presented in the following tables.

Table 11: System Improvements

Comments	Improvements
The radial menu buttons were difficult to reach. The users had a hard time rotating their controller over 140° left or right	The radial selection gesture was redesigned to allow for selection of all menu items using 170°. The degree's curve was designed to allow selection from -20° (rotate counter clockwise) from default hand position (thumb pointing upwards) to 170° degrees (clock wise).
Pointer (blue dot on UI elements) visibility was low.	Increased pointer size by 70% to provide better visibility
To access a greenhouse or zone information the user had to "find" the zone or greenhouse in the virtual environment.	Designed and implemented a "quick access" screen, which provided a list of all greenhouses or zones. The user can receive further information by selecting the list items, which will in turn transport the user (fast travel) in front of the location (zone or greenhouse) to receive further information. The user can access this lift from his/ her menu
Content was hard to read due to the small font	Increased 3D GUI screens' scale by almost 200% to provide better visibility
Inconsistency. Content was presented in multiple languages (Greek, English).	All content was translated in English which is the primary language of the system
Sliders' handlers were hard to see	Increased 3D GUI screens' scale by almost 200% to provide better visibility
Tutorial assets remained present in the scene after tutorial, which was confusing	After the tutorial and configuration scenario, all tutorial assets were removed to avoid confusion
Enter button on greenhouse dashboard looked like an exit button	Replaced button icon with better icon
On greenhouse dashboard zone's color, encoding was not displayed. Every zone was healthy (green). (health > =75%->green, health < 75% -> orange, <50% -> red, etc.)	Created content generators using random data, to provide more content, by providing more content the desired outcome (display more color encoded zones) was achieved
The checked health icon had ambiguous meaning. (Does it mean that the plant is healthy?)	Replaced health icon with a more suitable one
White placeholder, near graph window title, serves no purpose	Removed un-used placeholder
Points on graph were too hard to hover	Increased point's size
Color encoding of value points on the graph was not clear (What was the meaning of red, green, orange)	Introduced color legends explaining the colors on the graph
The user was not able to identify the ideal cultivation values (min, max, average).	Added label over ideal cultivation values, explaining what the values below mean

Comments	Improvements
Y axis text overflows graph window	Recreated window graph asset to avoid text overflowing off the graph window
Previous and next buttons were enabled but clicking them did nothing (after all values were displayed)	Redesigned the "Unable to receive values" screen, and disabled buttons if no values were displayed
Color encoding on controls (actuator control) was confusing. Green and red have different meaning on graphs.	Redesigned and implanted actuator control buttons adding on/ off labels to provide better status information
Control names were not self-explanatory; the user was not sure what he was turning on/off.	Redesigned control object to provided better title and short description for on hover information
Interior, exterior tabs do not look like tabs. Interior / exterior tabs "on hover" color is hard to read (white letters on grey background). UI sections were not easily distinguishable (e.g. Sensor tabs, Actuator control over controls tab).	Redesigned ambient information and general control screen
Shadows on walls make the walls appear black	Added more light in the virtual greenhouse
Add greenhouse map, with zone location (with fast travel capabilities)	Implement lift of zones screen (accessible from radial menu) with fast travel (click to transfer to location) capabilities. The list changes to display the zones of the greenhouse the user is currently in

All improvements were reviewed by two (2) evaluators, in order to ensure that the comments addressed and the changes made to the system were effective. The comments that were not addressed and the changes that were not as effective as expected were added to the comments received from the second evaluation for scoring.

7.2 Second round of heuristic evaluation

During the **second** evaluation round, the evaluators were asked to provide feedback on the system for all three (3) scenes of the system. The second heuristic evaluation was conducted by four (4) experts two (2) **UI/UX** (who took part in the first round), **one** (1) Software engineer (responsible for the Intelligent Greenhouse API) and **one** (1) occupational therapist.

7.2.1 Second round comments and tables

The comments are grouped to provide better readability and categorization.

Table 12: System controls and general comments

Comments	Score
Accidentally pressing "return to dashboard" radial menu item lead to exiting the virtual greenhouse	2.6
The users would like to be able to select radial menu, buttons with their pointer and not only through controller rotation gesture	1.8
Rotating over the last selected menu item on radial menu selected the first item and vice versa, which was confusing	2

The system did not support single controller movement	2.1
The system did not support swap primary controller functionality (left, right) (e.g. teleport with right controller/ rotate with left thumb stick)	2.3
Settings could not be accessed from radial menu	1.5
On cultivation zone's screen, default health graph selection to avoid empty space on first interaction	1.8
Health value generation was not clear, the user was not sure what was the reason the health was low (e.g. low temperature)	2.5
On the greenhouse dashboard zone's health color encoding was not explained (health > =75%->green, health < 75% -> orange , <50% -> red, etc)	1.8
On the graph window, the user was confused about previous and next buttons functionality (i.e. he / she though that it would select the next or previous sensor and display its graph window)	2.3

Table 13: Scene 1 - Tutorial and first configuration

Comments	Score
Sliders' handlers were hard to see	1.8

Table 14: Scene 2: Virtual Greenhouse

Comments	Score
There is no "panic button" option in the virtual greenhouse to disable all actuators (kill switch)	2.5
The user was not able to access ambient information and general actuator control screen from inside the virtual greenhouse, and had to move outside (near the door) to access that information	2.8
Sensors were not categorised/ group. (on ambient sensor screen)	2.6
Actuators didn't have icons	2.8

Table 15: Scene 2: Virtual Greenhouse

Y,X axis values were not present (e.g. °C/ hour)	2.8
Cultivation zones with critically low health were not highlighted or indicated in a clear manner	3.1
In the virtual greenhouse, the user was not confident which greenhouse he/ she was reviewing due to the absence of title	3.3
Previous and next buttons were enabled but clicking them did nothing (after all values were displayed)	1.9
Users were not confident about previous/ next buttons functionality	2.4

Table 16: General comments on virtual greenhouse

Comments	**Score**
Shadows on walls make the walls appear black	1.3
3D object interaction was not provided as a means to operate actuators, alongside UI actuator operation (E.g. inside the zone “use” the fan 3D object to activate the fan over this zone)	2.5
Plant growth was not represented in the virtual environment (farmville style)	2.9

The second heuristic evaluation’s comments (along with the comments **not** addressed from the first round) are kept for future work.

7.3 User feedback

During both rounds of heuristic evaluation, all evaluators offered their opinion on the system’s controls and overall feeling of the user experience, beyond the scope of their expertise. These comments were not added in the comment’s list, since they represented the evaluator’s subjective opinion on the system and could not be categorised using Nielsen’s rules for heuristic evaluation.

7.3.1 Movement

All evaluators (all six (6) of them) admitted to have played **little** to **none** games using controllers, but were able to easily familiarise with the movement and explore the system. After familiarising with the system, they were able to use it to explore the virtual environment. Only **one** (1) of the evaluators was not able to use the system for an extended period of time (due to motion sickness), but admitted of not being able to operate VR systems in general. However, he/ she agreed with the rest of the evaluators that the choice of the sitting position greatly **reduced** motion sickness.

The evaluators also were able to effectively configure the system’s movement controls (**speed**, **rotation sensitivity**, **fade time**) and calibrate them to their preferences. This was also endorsed by the fact that all evaluators (except the one mentioned above) could be immersed in the exploration and felt confident enough to exit the “safe zone” (fence inside the virtual greenhouse) and **explore** the virtual terrain. Lastly, the evaluators admitted that they lost track of time while using the system, and didn’t realized they reached the end of the session (20 minutes) which further comments positively on the immersiveness of the system and the reduced cognitive load the system forces on the user. However, this claim need further investigation (user testing with more users), which will be conducted as part of the future work.

Finally, one of the evaluators commented on the decision of the office (rotating) chair as recommended operation position, that it simplifies the navigation by minimizing the effort to change orientation inside the virtual environment (by rotating the chair in with the same direction of the controller input).

7.3.2 Interaction and system control

All evaluators were able to effectively operate the system and receive the desired information. However, they all pointed out that the 3D GUIs could use some input from a UI designer to further improve the “look” and “feel” of the GUIs.

The evaluators were able to effectively use the **raycast technique**, and commented that it “**felt natural**” and even though at the start of the session some commented that raycasting using both controllers was not necessary (could instead use **single hand navigation** / **interaction**), by the end of the session they were also using both controllers to interact with the virtual environment.

7.3.3 Virtual environment

Finally, the evaluators commented on the virtual greenhouse scene, that even though it was simple, which was a good thing because it was **not** confusing, for later versions, they would like more content to further enhance the immersion and engagement with the system.

Finally, they pointed out (which was included in the comments) that 3D assets would complement the functionality and usability of the system by creating alternative means for the user to operate the virtual environment.

7.3.4 Conclusion

All users were able to interact with the IGH though the virtual environment. They were able to effectively and effortlessly identify, review and act on information of their interest.

All users, even the ones with different preferences from the implemented navigation techniques, were able to familiarise themselves with the system in a short span of time and completed a twenty-five (25) minute long session without interruptions due to dizziness or motion sickness. They were also able to move to their desired location and effectively explore the virtual environment.

One important note here is the fact that all users admitted that the seated position enabled them to enjoy the navigation and drastically reduced the negative side effects from moving in the VE. The evaluators were also able to use the settings menu effectively and configure the system settings to match their preferences.

In conclusion, the system was able to complete successfully the Virtual Reality primary tasks (navigation, selection/ manipulation, system control) and successfully provide remote, monitoring, management and operation capabilities for the users, while also creating an interesting, immersive and fun virtual experience.

8 Conclusion and Future Work

This chapter summarizes the results of this thesis and describes future work.

8.1 Conclusion

The LINA Framework enables user interaction by utilizing Virtual Reality (VR) technologies for remote monitoring, management and operation of an Intelligent Greenhouse (IGH). An IGH contains sensors used to measure environmental parameters and actuators arranged to build a controlled environment. Based on that, the LINA Virtual Interaction Framework simulates the physical world by creating a virtual world containing proper assets that can facilitate and enhance user's cultivation task performance and overall experience. The user is fully immersed in the virtual environment, interacts and performs tasks that have real time effect in the IGH.

The LINA framework allows the user to select, monitor, manage and operate a wanted IGH, among many, via the *Dashboard* and Virtual *Greenhouse* scenes. The 3D GUIs enable the user to acquire contextual information about the IGH and interact with the system as well as with the IGH through the *Greenhouse* and *Cultivation Zone* assets.

Proper *Interaction Techniques* allow the user to perform the three primary tasks that a user wishes to complete namely navigation, selection / manipulation and system control [11,12]. Navigation is achieved using a *Point and click to move* and/ or a combination of both *thumb stick*s *to move*. Selection/ manipulation and system control are based on gestures to access and select menu items and *raycasting* while interacting with the 3D GUIs.

The LINA framework provides users with personalized *Interaction Techniques* configuration, which allows users to have access to real-time information as well as system control for altering the system's state to match their preferences.

The aforementioned technological framework has been deployed within the Intelligent Greenhouse simulation space of the ICS-FORTH's Ambient Intelligence Facility, and two rounds of heuristic evaluation have been conducted. The findings have shown that all users were able to effectively explore the VE and effortlessly identify, review and act on information of their interest. They were also able to familiarize themselves with the system in a short span of time and complete their sessions without experiencing side effects caused by VR. Moreover, all users admitted that the seated position enabled them to enjoy navigation in the VE and to configure the system settings to match their preferences.

8.2 Future Work

8.2.1 Virtual 3D assets

One aspect that all evaluators commented on was the absence of virtual 3D assets from the representation of the user's Intelligent Greenhouse. The envisioned virtual 3D asset library will include the two (2) **types** of 3D assets described below.

8.2.1.1 Actuator control assets

Actuator control assets will be modelled after the actuators they represent/ operate. When the virtual greenhouse provides control over an Intelligent greenhouse actuator (e.g.

irrigation system for a cultivation zone), the virtual greenhouse will select a 3D asset from the implemented asset's list, associated with this type of actuator (e.g. for irrigation control a hose) and included it in the cultivation zone's representation. The system will then enable the user to interact with such 3D asset (hose) to operate the actuator (i.e. in this example by "selecting" the 3D virtual hose asset, the user can open or close the irrigation system for this cultivation zone.

This option will be added on top of the 3D GUI actuator control and will provide the user with alternative means of operating the intelligent greenhouse actuators. In addition, it will provide interaction that is more natural for the user and increase immersion, by using objects associated with actions from the Intelligent Greenhouse context (i.e. the 3D assets will be modelled after real greenhouse tools).

8.2.2.1 Plant models for growth monitoring

Each 3D plant models will include three (3) different models of the same plant. The first will be a representation of the plant as seedling (this could be universal for all plants), the second model will represent the intermediate phase of the plant's growth, and finally the third mode will represent the plant's full-grown form.

These assets will then be placed inside the virtual cultivation zone's representation (inside to soil bed of the zone) and the model will change over a time period spanning from the date the plant was planted to the expected ripe date. This can be achieved by comparing the planted date marked by the creation of the cultivation zone, the current date and the expected ripe time, from the plant information (i.e. the plant's information includes information about the time the plant needs to be fully grown and ripe for gathering.)

8.2.2 3D GUIs

In the next phase of the 3D GUIs iteration, the UI/UX experts' comments will be addressed. During the two (2) rounds of heuristic evaluation, the UI/UX experts proposed changes to ensure the UI are usable and intuitive. Based on these comments, mock-ups will be created, and after the UI/UX experts evaluation these mock-ups will be implement and added in the system.

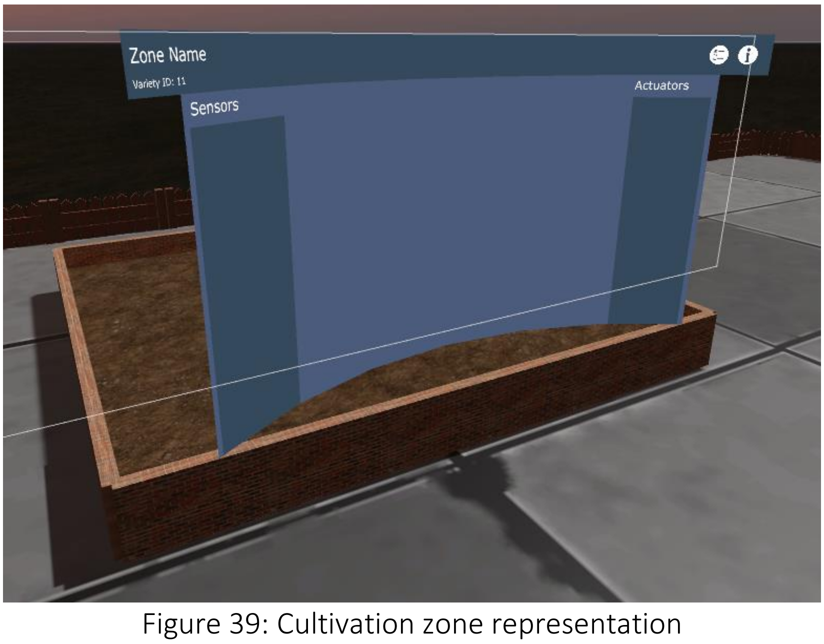


Figure 39: Cultivation zone representation

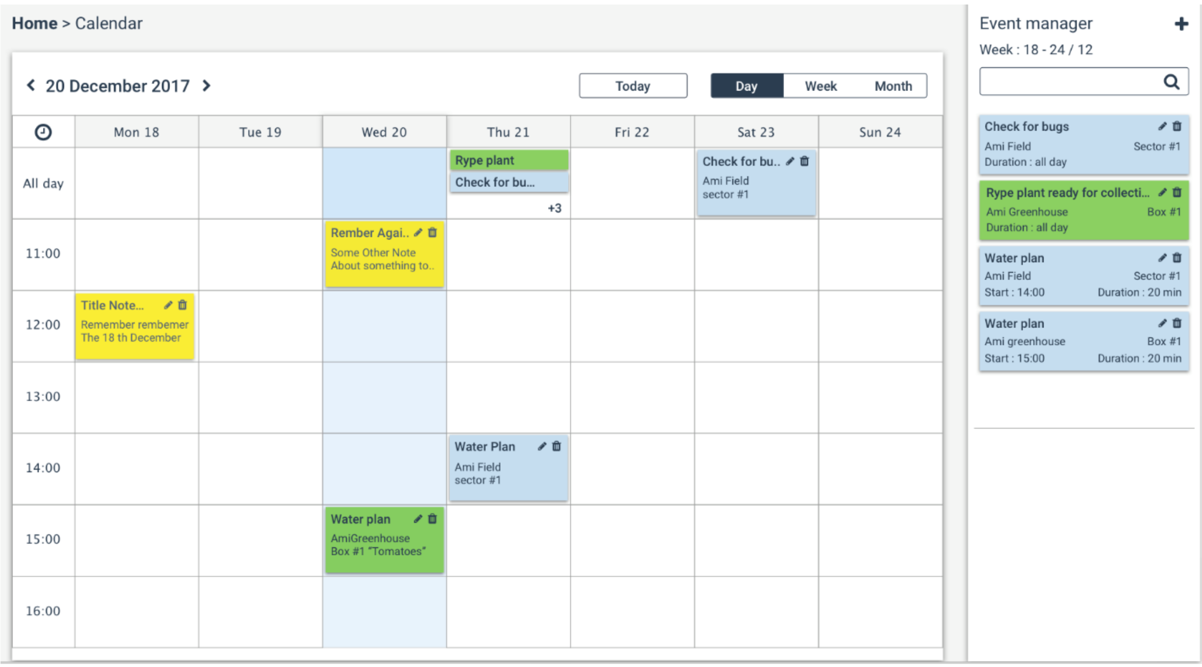
8.2.3 Additional functionality and management tools

The purpose of the presented framework is to provide remote monitoring, management and operation of Intelligent greenhouses. The provided functionality could be extended with the automation of certain tasks: a) the environmental parameter regulation (e.g. soil moisture, air temperature etc.), b) the organisation of required cultivation chores in the users' agenda (e.g. weed removal, pest control etc.), c) the notification mechanism to inform the user of the IGHs state and conditions that require immediate attention and intervention and d) the implementation of an Intelligent Assistant to provide the users with a means for verbal interaction with the system

8.2.1.3 Organizing and notification tools

As described in paragraph 3.1.1. SoU 1, "LINA enables the user to effectively manage his time and create a schedule with tasks he needs to complete", a functionality that has not been implemented in current version. In the future version of the LINA framework, a set of tools to optimise time management will be added.

In order to facilitate users' time and labour resources, the system will include an auto-updated **Agenda** (see Figure 40) and could be connected to the user's email or calendar (e.g. google calendar). This agenda will include important dates, such as when a cultivation zone planted, when harvest, when irrigate, etc. This information will be pulled from the plant information object (PIO). By implementing the agenda component, the system will enable users to schedule actuator control, meaning the users will be able to select or create operation plans for actuator control. A plan for a specific actuator is an automatic operation the user can set for this actuator to be executed a) in a specific date/ time (e.g. close "roof window" at 12.07.2019 - 18:00), b) using a timer (e.g. activate "fan" for 10 minutes) c) repeatedly at a specific time (e.g. activate "irrigation system for 10 minutes" at "10:00" until "10.07.2019")



	Mon 18	Tue 19	Wed 20	Thu 21	Fri 22	Sat 23	Sun 24
All day				Rype plant Check for bu... +3		Check for bu.. Ami Field sector #1	
11:00			Rember Agai.. Some Other Note About something to..				
12:00	Title Note... Remember rembemer The 18 th December						
13:00							
14:00				Water Plan Ami Field sector #1			
15:00			Water plan AmiGreenhouse Box #1 "Tomatoes"				
16:00							

Week : 18 - 24 / 12

Check for bugs
Ami Field Sector #1
Duration : all day

Rype plant ready for collecti...
Ami Greenhouse Box #1
Duration : all day

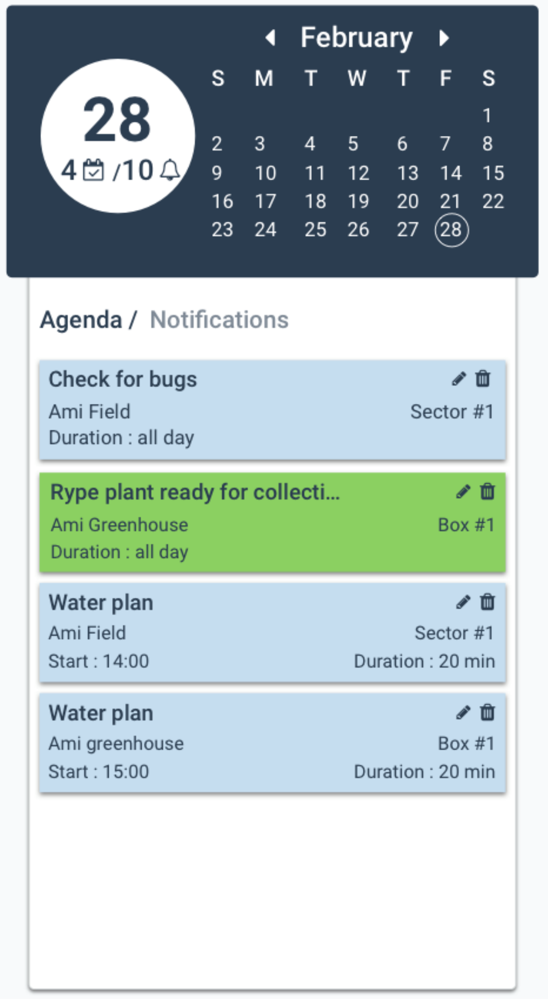
Water plan
Ami Field Sector #1
Start : 14:00 Duration : 20 min

Water plan
Ami greenhouse Box #1
Start : 15:00 Duration : 20 min

Figure 40: Agenda mock-up screen

The user needs to be notified by the system for a variety of things: a) events on the agenda these notifications could refer to zone (plant) information, such as that the plant will be ripe for gathering and the user should take action, notes the user has attached on the cultivation zone and actuator plans the user has scheduled, b) system assistive notifications, the system should provide system notifications to assist the user with the cultivation of the plants, meaning the system should automatically notify the user for a situation that requires his/ her immediate attention (e.g. “the air temperature of the AmI greenhouse has reached critical level, you should activate the fan”) and provide quick actions (in the previous example “accept to activate”) from the notification’s dialog box, and c) system notification such as error and system messages (e.g. “your system has been updated”)

To this end, a notification system will be implemented to allow the user to receive and review the notifications while using the system, as well as to be notified when he is not logged in.



Check for bugs
Ami Field Sector #1
Duration : all day

Rype plant ready for collecti...
Ami Greenhouse Box #1
Duration : all day

Water plan
Ami Field Sector #1
Start : 14:00 Duration : 20 min

Water plan
Ami greenhouse Box #1
Start : 15:00 Duration : 20 min

Figure 41: Agenda widget mock-up

8.2.2.3 Intelligent Assistant (Lina)

The intelligent assistant will be created as a plugin to enable the user to interact with the system through verbal commands. The user will use the keyword **Lina** to activate the assistant and then verbally speak what he/ she wishes the system to do (e.g. “**Lina,** activate fan of tomatoes zone”). To achieve this objective the plugin will use the “Google Assistant SDK - Google Developers”, which enables developers to use the googles voice recognition technologies to create custom command and trigger services using an SDK.

This will in turn add to the overall experience of using the system and provide a layer of interaction with the system through natural verbal speech.

This plugin could be further improved by adding an AI agent in the system, which will learn the user’s system operation patterns and create tailored quick actions for the user, along with cultivation recommendations

References

1. A. D. Jurik and A. C. Weaver, "Remote Medical Monitoring," in Computer, vol. 41, no. 4, pp. 96-99, April 2008. doi: 10.1109/MC.2008.133
2. Agnal, S., Kapoor, K., Musfik, M., & Sharma, R. (2018). Automated Smart Greenhouse Environment Using IoT.
3. Ahmed, A., Ali, J., Raza, A., & Abbas, G. (2006, November). Wired vs wireless deployment support for wireless sensor networks. In TENCON 2006-2006 IEEE Region 10 Conference (pp. 1-3). IEEE.
4. Akerberg, J., Gidlund, M., & Björkman, M. (2011, July). Future research challenges in wireless sensor and actuator networks targeting industrial automation. In 2011 9th IEEE International Conference on Industrial Informatics (pp. 410-415). IEEE.
5. Aleotti, J., & Caselli, S. (2011). Physics-based virtual reality for task learning and intelligent disassembly planning. Virtual reality, 15(1), 41-54.
6. Augusto, J. C., & McCullagh, P. J. (2007). Ambient intelligence: Concepts and applications. Comput. Sci. Inf. Syst., 4(1), 1-27.
7. Augusto, J. C., Callaghan, V., Cook, D., Kameas, A. & Satoh, I. (2013). Intelligent environments: a manifesto. Human-Centric Computing and Information Sciences, 3(1), 12.
8. Bélissent, J. (2010). Getting clever about smart cities: New opportunities require new business models. Cambridge, Massachusetts, USA.
9. Bhagat, K. K., Liou, W. K., & Chang, C. Y. (2016). A cost-effective interactive 3D virtual reality system applied to military live firing training. Virtual Reality, 20(2), 127-140.
10. Bouloukakis, M., Stratakis, C., & Stephanidis, C. (2018). AmI Garden: building an Iot Infrastructure for Precision Agriculture. ERCIM NEWS, (113), 18-19.
11. Bowman, D. A., Johnson, D. B., & Hodges, L. F. (2001). Testbed evaluation of virtual environment interaction techniques. Presence: Teleoperators & Virtual Environments, 10(1), 75-95.
12. Bowman, D. A., Kruijff, E., LaViola Jr, J. J., & Poupyrev, I. (2001). An introduction to 3-D user interface design. Presence: Teleoperators & Virtual Environments, 10(1), 96-108.
13. Bowman, D., Kruijff, E., LaViola Jr, J. J., & Poupyrev, I. P. (2004). 3D User interfaces: theory and practice, CourseSmart eTextbook. Addison-Wesley.
14. Brumitt, B., Meyers, B., Krumm, J., Kern, A., & Shafer, S. (2000, September). Easyliving: Technologies for intelligent environments. In International Symposium on Handheld and Ubiquitous Computing (pp. 12-29). Springer, Berlin, Heidelberg.
15. Cao, X., Chen, J., Xiao, Y., & Sun, Y. (2010). Building-environment control with wireless sensor and actuator networks: Centralized versus distributed. IEEE Transactions on Industrial Electronics, 57(11), 3596-3605.
16. Chen, J., Cao, X., Cheng, P., Xiao, Y., & Sun, Y. (2010). Distributed collaborative control for industrial automation with wireless sensor and actuator networks. IEEE Transactions on Industrial Electronics, 57(12), 4219-4230
17. Choi, S., Jung, K., & Noh, S. D. (2015). Virtual reality applications in manufacturing industries:Past research, present findings, and future directions. Concurrent Engineering, 23(1), 40-63.

18. Cook, D., & Das, S. K. (2004). Smart environments: Technology, protocols and applications (Vol. 43). John Wiley & Sons.
19. Corchado, J. M., Bajo, J., Tapia, D. I., & Abraham, A. (2010). Using heterogeneous wireless sensor networks in a telemonitoring system for healthcare. IEEE transactions on information technology in biomedicine, 14(2), 234-240.
20. Dam, R., & Siang, T. (2018). Personas–A Simple Introduction. Interaction Design Foundation [verkkojulkaisu]. Saatavuus:< https://www. interaction-design. org/literature/article/personas-why-and-how-you-should-use-them>(luettu 26.9. 2018).
21. Di Gironimo, G., & Leoncini, P. (2005). Realistic interaction for maintainability tests in virtual reality. In Proceedings of Virtual Concept.
22. Farmville @ https://www.zynga.com/games/farmville/ (31/05/2019)
23. Fass, L. (2007). Patient-centric healthcare
24. Fong, T., & Thorpe, C. (2001). Vehicle teleoperation interfaces. Autonomous robots, 11(1), 9-18.
25. FROHLICH, T. (2000, July). THE VIRTUAL OCEANARIUM. Communications of the ACM, 43(7), 94. https://link.galegroup.com/apps/doc/A63635267/AONE?u=googlescholar&sid=AONE&xid=29bbdc90
26. Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. Future generation computer systems, 29(7), 1645-1660.
27. Hamari, J. (2013). Transforming homo economicus into homo ludens: A field experiment on gamification in a utilitarian peer-to-peer trading service. Electronic commerce research and applications, 12(4), 236-245.
28. Hamari, J., Koivisto, J., & Sarsa, H. (2014, January). Does Gamification Work?-A Literature Review of Empirical Studies on Gamification. In HICSS (Vol. 14, No. 2014, pp. 3025-3034).
29. Heidari Jozam M., Allameh E., De Vries B., Timmermans H., Masoud M. (2012) VR-Smart Home. In: Andreev S., Balandin S., Koucheryavy Y. (eds) Internet of Things, Smart Spaces, and Next Generation Networking. ruSMART 2012, NEW2AN 2012. Lecture Notes in Computer Science, vol 7469. Springer, Berlin, Heidelberg
30. Herkelrath, W. N., Hamburg, S. P., & Murphy, F. (1991). Automatic, real-time monitoring of soil moisture in a remote field area with time domain reflectometry. Water Resources Research, 27(5), 857-864.
31. Hind Fadhil Abbaset al, International Journal of Computer Science and Mobile Computing, Vol.4 Issue.8, August-2015, pg. 335-34
32. Hokayem, P. F., & Spong, M. W. (2006). Bilateral teleoperation: An historical survey. Automatica, 42(12), 2035-2057.
33. Huotari, K., & Hamari, J. (2012, October). Defining gamification: a service marketing perspective. In Proceeding of the 16th international academic MindTrek conference (pp. 17-22). ACM.
34. Ilyas, M., & Mahgoub, I. (Eds.). (2004). Handbook of sensor networks: compact wireless and wired sensing systems. CRC press.
35. Kadir, A. A., Xu, X., & Hämmerle, E. (2011). Virtual machine tools and virtual machining—a technological review. Robotics and computer-integrated manufacturing, 27(3), 494-508.

36. Ken Hinckley Hugh McLoone (2003) Proceedings of the XVth Triennial Congress of the International Ergonomics Association and the 7th Joint Conference of Ergonomics Society of Korea [and] Japan Ergonomics Society
37. KEYGENE Nederlands (2017), The Virtual Reality Breeding Tool, https://www.worldfoodinnovations.com/innovation/the-virtual-reality-breeding-tool
38. King, C., Baghdadi, N., Lecomte, V., & Cerdan, O. (2005). The application of remote-sensing data to monitoring and modelling of soil erosion. Catena, 62(2-3), 79-93.
39. Kujala, S., & Kauppinen, M. (2004, October). Identifying and selecting users for user-centered design. In Proceedings of the third Nordic conference on Human-computer interaction (pp. 297-303). ACM.
40. Lee HJ, Kim S-J (2016) Factors Associated with Visual Fatigue from Curved Monitor Use: A Prospective Study of Healthy Subjects. PLoSONE 11(10):e0164022.doi: 10.1371/journal.pone.0164022
41. Lei, S., Xu, H., Xiaoling, W., Lin, Z., Cho, J., & Lee, S. (2006, August). VIP bridge: Integrating several sensor networks into one virtual sensor network. In International Conference on Internet Surveillance and Protection (ICISP'06) (pp. 2-2). IEEE.
42. Lim, C. W., & Jung, H. W. (2013). A study on the military Serious Game. Advanced Science and Technology Letters, 39, 73-77.
43. Lim, J., & Ayyagari, R. (2018). Investigating the determinants of telepresence in the e-commerce setting. Computers in Human Behavior, 85, 360-371.
44. Linden, A., & Fenn, J. (2003). Understanding Gartner's hype cycles. Strategic Analysis Report № R-20-1971. Gartner, Inc
45. Maguire, M., & Bevan, N. (2002, August). User requirements analysis. In IFIP World Computer Congress, TC 13 (pp. 133-148). Springer, Boston, MA.
46. Mainetti, L., Patrono, L., & Vilei, A. (2011, September). Evolution of wireless sensor networks towards the internet of things: A survey. In SoftCOM 2011, 19th International Conference on Software, Telecommunications and Computer Networks (pp. 1-6). IEEE.
47. Merhi, O., Faugloire, E., Flanagan, M., & Stoffregen, T. A. (2007). Motion Sickness, Console Video Games, and Head-Mounted Displays. Human Factors, 49(5), 920–934
48. Michael, D. R., & Chen, S. L. (2005). Serious games: Games that educate, train, and inform. Muska & Lipman/Premier-Trade
49. Mollen, A., & Wilson, H. (2010). Engagement, telepresence and interactivity in online consumer experience: Reconciling scholastic and managerial perspectives. Journal of business research, 63(9-10), 919-925
50. Navarre, D., Palanque, P., Bastide, R., Schyn, A., Winckler, M., Nedel, L. P., & Freitas, C. M. (2005, September). A formal description of multimodal interaction techniques for immersive virtual reality applications. In IFIP Conference on Human-Computer Interaction (pp. 170-183). Springer, Berlin, Heidelberg.
51. Nicholson, S. (2015). A recipe for meaningful gamification. In Gamification in education and business (pp. 1-20). Springer, Cham.
52. Nielsen, J., and Molich, R. (1990). Heuristic evaluation of user interfaces, Proc. ACM CHI'90 Conf. (Seattle, WA, 1-5 April), 249-256.
53. Nielsen, L. (2004). Engaging personas and narrative scenarios. *PhD series*, *17*.
54. Nugroho, A. P., Okayasu, T., Hoshi, T., Inoue, E., Hirai, Y., Mitsuoka, M., & Sutiarso, L. (2016). Development of a remote environmental monitoring and control framework

for tropical horticulture and verification of its validity under unstable network connection in rural area. Computers and Electronics in Agriculture, 124, 325-339.

55. Pan, Z., Cheok, A. D., Yang, H., Zhu, J., & Shi, J. (2006). Virtual reality and mixed reality for virtual learning environments. Computers & graphics, 30(1), 20-28.
56. Papagiannakis, G., Lydatakis, N., Kateros, S., Georgiou, S., & Zikas, P. (2018, December). Transforming medical education and training with VR using MAGES. In SIGGRAPH Asia 2018 Posters (p. 83). ACM.
57. PIEdeck (system) https://pie-lab.at/projects/piedeck/
58. Prabhu, S. B., Dhasharathi, C. V., Prabhakaran, R., Kumar, M. R., Feroze, S. W., & Sophia, S. (2014). Environmental monitoring and greenhouse control by distributed sensor Network. *International Journal of Advanced Networking and Applications*, *5*(5), 2060.
59. Psotka, J. (1995). Immersive training systems: Virtual reality and education and training. Instructional science, 23(5-6), 405-431.
60. Qiu, W., Dong, L., Wang, F., & Yan, H. (2014, April). Design of intelligent greenhouse environment monitoring system based on ZigBee and embedded technology. In 2014 IEEE International Conference on Consumer Electronics-China (pp. 1-3). IEEE.
61. R Shamshiri, R., Kalantari, F., Ting, K. C., Thorp, K. R., Hameed, I. A., Weltzien, C., ... & Shad, Z. M. (2018). Advances in greenhouse automation and controlled environment agriculture: A transition to plant factories and urban agriculture
62. Regan, C. (1995). An investigation into nausea and other side-effects of head-coupled immersive virtual reality. Virtual Reality, 1(1), 17-31
63. Remagnino, P., Foresti, G. L., & Ellis, T. (Eds.). (2005). Ambient intelligence: a novel paradigm (p. 240). New York, NY, USA:: Springer.
64. Rizzo, A. A., Buckwalter, J. G., Bowerly, T., Van Der Zaag, C., Humphrey, L., Neumann, U., ... & Sisemore, D. (2000). The virtual classroom: a virtual reality environment for the assessment and rehabilitation of attention deficits. CyberPsychology & Behavior, 3(3), 483-499.
65. Runkle E., Cutting Edge Greenhouse Technology, > https://gpnmag.com/article/cutting-edge-greenhouse-technology/https://gpnmag.com/article/cutting-edge-greenhouse-technology/
66. Seymour, N. E., Gallagher, A. G., Roman, S. A., O'brien, M. K., Bansal, V. K., Andersen, D. K., & Satava, R. M. (2002). Virtual reality training improves operating room performance: results of a randomized, double-blinded study. Annals of surgery, 236(4), 458.
67. Shafer, S. A., Brumitt, B., & Cadiz, J. J. (2001). Interaction issues in context-aware intelligent environments. Human-Computer Interaction, 16(2), 363-378.
68. Shamshiri R R, Kalantari F, Ting K C, Thorp K R, Hameed I A, Weltzien C, et al. Advances in greenhouse automation and controlled environment agriculture: A transition to plant factories and urban agriculture. Int J Agric & Biol Eng, 2018; 11(1): 1–22.
69. Shukla, A. J., Panchal, M. V., & Patel, M. S. (2015). Intelligent Greenhouse Design based on Internet of Things (IoT). International Journal of Emerging Trends in Electrical and Electronics, 11(2), 78-86.
70. Smart Greenhouse Remote Monitoring Systems https://www.postscapes.com/smart-greenhouses/
71. Smart Greenhouse: The future of agriculture https://www.hackster.io/synergy-flynn-9ffb33/smart-greenhouse-the-future-of-agriculture-5d0e68

72. Steinfeld, A., Fong, T., Kaber, D., Lewis, M., Scholtz, J., Schultz, A., & Goodrich, M. (2006, March). Common metrics for human-robot interaction. In Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction (pp. 33-40). ACM.
73. Steuer, J. (1992). Defining virtual reality: Dimensions determining telepresence. Journal of communication, 42(4), 73-93.
74. Sutcliffe, A. RE 2003 Mini-tutorial: Scenario-based Requirements Engineering. *IEEE Xplore Digital Library.*
75. Ten Usability Heuristics by Jakob Nielsen. http://www.useit.com/papers /heuristic/heuristic_list.htm
76. Teslyuk, T., Denysyuk, P., Kernytskyy, A., & Teslyuk, V. (2015, September). Automated control system for arduino and android based intelligent greenhouse. In 2015 XI International Conference on Perspective Technologies and Methods in MEMS Design (MEMSTECH) (pp. 7-10). IEEE.
77. Vision Span https://en.wikipedia.org/wiki/Vision_span
78. VR health (systems) https://www.xr.health/
79. What is virtual reality, https://www.interaction-design.org/literature/topics/virtual-reality
80. Wikipedia contributors. (2019). Accommodation (eye). In Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Accommodation_(eye)&oldid=897393243
81. Wojciechowski, R., Walczak, K., White, M., & Cellary, W. (2004, April). Building virtual and augmented reality museum exhibitions. In Proceedings of the ninth international conference on 3D Web technology (pp. 135-144). ACM.
82. Xie, W., Shi, Y., Xu, G., & Xie, D. (2001, October). Smart classroom-an intelligent environment for tele-education. In Pacific-Rim Conference on Multimedia (pp. 662-668). Springer, Berlin, Heidelberg.
83. Xu, G., Shen, W., & Wang, X. (2014). Applications of wireless sensor networks in marine environment monitoring: A survey. Sensors, 14(9), 16932-16954.
84. Your constantly-updated definition of Virtual Reality and collection of topical content and literature (date: 02/2019) https://www.interaction-design.org/literature/topics/virtual-reality
85. Yue, J., Lei, T., Li, C., & Zhu, J. (2012). The application of unmanned aerial vehicle remote sensing in quickly monitoring crop pests. Intelligent Automation & Soft Computing, 18(8), 1043-1052.
86. Zidianakis, E. (2015). *Supporting young children in ambient intelligence environments* (Doctoral dissertation, Πανεπιστήμιο Κρήτης. Σχολή Θετικών και Τεχνολογικών Επιστημών. Τμήμα Επιστήμης Υπολογιστών).

    Zyda, M. (2005). From visual simulation to virtual reality to games. Computer, 38(9), 25-32.