2023-1-EL01-KA220-HED-000161272

GUESS: Gamified User-centered Environmental Sustainability for university Students

The Learning Environment	
Intellectual Output or Activity Number	NA
Short Description	Development of a learning environment that will offer a gamification-based approach in which school student will engage in environmental activities and ultimately help them raise their environmental awareness.
Authors (per partner, if more than one partner provide it together)	Dimosthenis Minas (UPatras) Panagiotis Vrettis (UPatras)
Status (D: Draft RD: Revised Draft F: Final)	F
Dissemination Level (P: Public C: Confidential, only for members of the consortium, National	P

Agency, Commission services and project reviewers)			
File Name	**GUESS_Learning_Environment_v1.0**		
Date	ver0.1	20/06/2024	Draft version of the design of the environment based on the proposal
	ver0.2	10/10/2024	Initial version of the first gamified use case
	ver0.3	21/03/2025	Addition of the first half of the iteration of gamified use cases
	ver0.4	16/07/2025	Addition of Language Selection room
	ver0.5	13/10/2025	Final Implementation Changes
	Ver1.0	26/11/2025	Finalisation of Learning Environment Report

Table of Contents

Executive Summary

The 3rd work package deliverable involves the design and development of a customizable 3D Virtual World Learning Environment (VWLE). An iterative development is used, with 4 important milestones: Basic Infrastructure (M8), Demonstrative Gamified Use Case (M12), First Iteration of Half of the Use Cases (M17), and the Final Version of all Use Cases (M22). Development is done by UPatras, guided by HOU and monitored by all the partners that supported the scenario development (ISMAI, MENDELU and UPCT). It is important to mention that in the original proposal, it was decided that each partner would develop 1 gamified use case. However, due to the importance of the project's goal and to make the project more inclusive, we decided to create 9 use cases in total. In the current version of the 3D Virtual World, 4 of the use cases have been created and implemented.

Versions of the document and explanation for the changes:

List of Figures

1. Introduction

The WP3 objective is to design and develop a gamified environment that combines learning-by-doing with virtual world technology, focused on educating university students about environmental sustainability. This aims to develop a tool for environmental sustainability education that will foster a deeper understanding of these issues among university students, contributing to sustainable futures.

The 3D VWLE is a multidisciplinary and friendly gamified learning environment, allowing the technically non-expert users to deal easily with the in-world gamified learning activities about environment.

The 3D VWLE developed for the GUESS project provides users with a gaming environment. The special characteristics and distinct possibilities of the 3D Virtual Worlds make them a powerful technological tool towards enhancing the learning experience. This is one of the main reasons for selecting a 3D VW platform for the development of an advanced, interactive, and motivating tool for the GUESS project, as teaching and training applications in 3D Virtual Worlds seem to offer remarkable benefits to students. 3D Virtual Worlds are ideally placed to support pedagogies that aim at moving away from chalk-and-talk learning and focus on more real-world learning styles such as learning through action, cooperation, gaming, problem-solving, etc.

A key goal of utilising a virtual environment that involves students in simulated scenarios, especially those designed to address environmental challenges, is to achieve deep immersion. Without such immersion, students might not achieve the intended learning outcomes, thereby diminishing the simulation's realism.

We should begin by establishing conditions and requirements for agent-based approaches. While a network of manufacturing services might be considered a network of agents, this type of application is not generally considered an agent-based simulation, so we should first seek out common qualities of agents. Historically, several qualities have been attributed to agents. Macal and North present a two-part tutorial that identifies agent qualities, a subset of which are included here:

- Identity: a discrete individual with a set of characteristics
- Interaction: the agent is situated in an environment where there is interaction with other agents
- Goal-directed: the agent seeks an objective
- Autonomy: the agent is autonomous

When considered at a glance, these qualities are those of living organisms. Based on this observation, the following might be regarded as valid agents: a virus, a bacterium, a cow, or a human. However, to be flexible in interpretation and to allow for the maximal use of the agent concept, we can extend the definition of an agent to include any adaptive object with organic, autonomous qualities. In this extended use of “agent”, a software module may be considered to be an agent since it may be adaptive, highly reconfigurable, and mobile. For example, an internet worm or virus is an agent according to this definition.

The agent concept is indeed a powerful one and relevant to many domains, from modeling communities within an ecological niche to modeling the interaction between software components. A panel was recently held to consider different ways agents can be used in defense and civilian simulation use cases. There is a compendium of knowledge about the state of the art from modeling humans from individuals to societies or ecosystems. The concept of *scale* is a key concern: how can we model agents at different levels of abstraction and aggregation while maintaining semantic interoperability across levels? Different scales allow us to study the system as we scale along the *axis of agent population*: from one to many.

However, as interest in agents and populations grows, a missing component remains: the scale associated with human reference. When simulating agents, it may be assumed that the person running the simulation maintains a relative distance from the model and its behaviors. For example, if one executes a cellular automaton, one sees cell colonies growing, dying, and transforming, creating an array of patterns and sometimes, self-organization. The relative position of the human running this automaton is similar to that of a scientist running an experiment, perhaps through a microscope or in situ by studying the agents “from the outside.” It is as if we can maintain that we understand how populations undergo their dynamics without becoming members of those populations. Is this possible? To a large extent, this is true–we routinely do science in this fashion, but primarily because we are unable to see the environment through the eyes of the agent. Even though we may gather helpful information about population behavior in this manner, we may be collectively missing the opportunity to act out roles, thereby enabling agents to better understand how an agent functions within a specific context. One way to summarize the issue of “human becoming the agent” is to observe that agents have bodies, and the point of the body is that one employs its affordances through role-playing.

1.1 Current state of the Learning Environment

By the 8th month, we had unveiled the basic infrastructure of the 3D VWLE, which included the reception room and the selection of the language.

By month 12, all scenarios were available exclusively in English.

By month 16, allscenarios were available at partners languages.

By month 22 after the second piloting during the summer, the final version of the virtual world was created.

Finally, 9 out of the 9 scenarios are available, in all partners languages English, Greek, Spanish, Czech and Portuguese. Other functionalities have been added in the virtual world as well, such as the chat with other players, customisation of the avatar, view of the world's map, and player's inventory. Figures 1 and 2 offer a visual summary of the 3D VWLE's progress to date.

Figure 1: Overview of the 8th month of the GUESS virtual world

Figure 2: Another view of the 8th month development of the GUESS 3D Virtual World

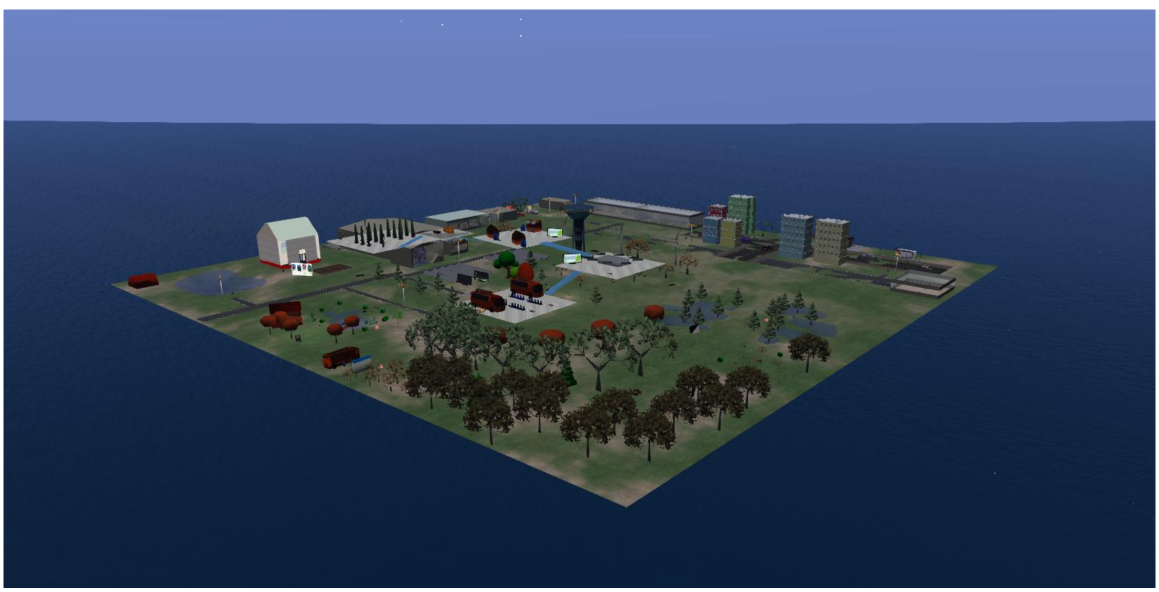

Figure 3: Final version of GUESS virtual world

2. Basic Infrastructure

The purpose of this chapter is to provide an analysis of the parameters and consideration utilised to architect a scalable, open-source, virtual world grid for use in various delivery scenarios. This case focuses on the detail leading up to deployment of the solution and includes discussion regarding solution selection and incorporation of virtualisation technologies to maximise institutional hardware resources based on established functional need.

The core working space in OpenSimulator is the region which is similar to a sim in Second Life. A region is what the end-user sees when they log into OpenSimulator; it is the visible virtual working space where the avatars interact. The region is a square piece of virtual landscape that can be further developed to contain such topologies as deserts, mountains, roads, buildings, classroom spaces, vast oceans, and other virtual space. A collection of multiple regions then forms what is typicaly referred to as a grid. Fundamentally, a grid provides organisation structure to the many regions by managing the relative position of each region within the virtual. The grid also manages such services as permissions, inventory and user access.

The current deployment of OpenSimulator utilises a server shell called ROBUST which runs a collection of virtual world management services. There are currently 14 services that are managed collectively by the ROBUST server that include: Assets Service, Authentication Service, Authorisation Service, Avatar Service, FreeSwitch Service, Friends Service, Gatekeeper Service, Grid Service, Grid user Service, Inventory Service, Login Service, presence Service, user Accounts Service, and the User Agents Service. Based on the functional criteria, it was determined that OpenSimulator solution coupled with additional open source support applications not only met most of the established criteria but also provided a platform similar in design to Second Life, thus reducing some of the in-world learning curve required by both students and grid managers. Overall similarities in end-user viewers, in-world content design, and account management made it a viable option to move forward with.

2.1 Virtual World Architectural Solution

At its basic level, today's virtual world solutions include four key components:

- A simulated environment or virtual world solution
- An end-user client or viewer
- A collection of collaborative resources available from within the virtual world environment
- A network infrastructure that encapsulates and supports the virtual world solution.

For computing and network resources this case utilised a virtualised infrastructure. A virtualised infrastructure allows users to transform hardware resources into a more flexible software-based resource and more specifically, provides for the ability to compile and then redistribute multiple hardware resources such as processors, memory, storage, and network controllers to create one or more fully functional virtual machines. These virtual machines or VM's can support their own operating system and applications – thus reproducing the same capabilities of one or more singular physical computing platforms.

In developing a scalable virtual world infrastructure, virtualisation provides us with the ability to both isolate and encapsulate application activity. Since VM's can share the physical resources of one or more computers, they are essentially isolated from each other just as if they were separate physical platforms. A VM is essentially a software container that packages or encapsulates a complete set of virtual hardware resources inside a software solution. The concept of encapsulation makes VM's incredibly scalable and much easier to manage.

The physical resources required for this virtual world solution (including computing, networking and storage) are all consolidated and managed under a single virtualised computing umbrella; computing capacity is then subdivided into multiple VM's as required for the solution. In describing the architecture for a scalable virtual world grid, a layered model is being presented. Case analysis in this instance provides for a solution architectural model in four distinct layers: a client layer; a robust layer; a collaborative layer; and a regional layer.

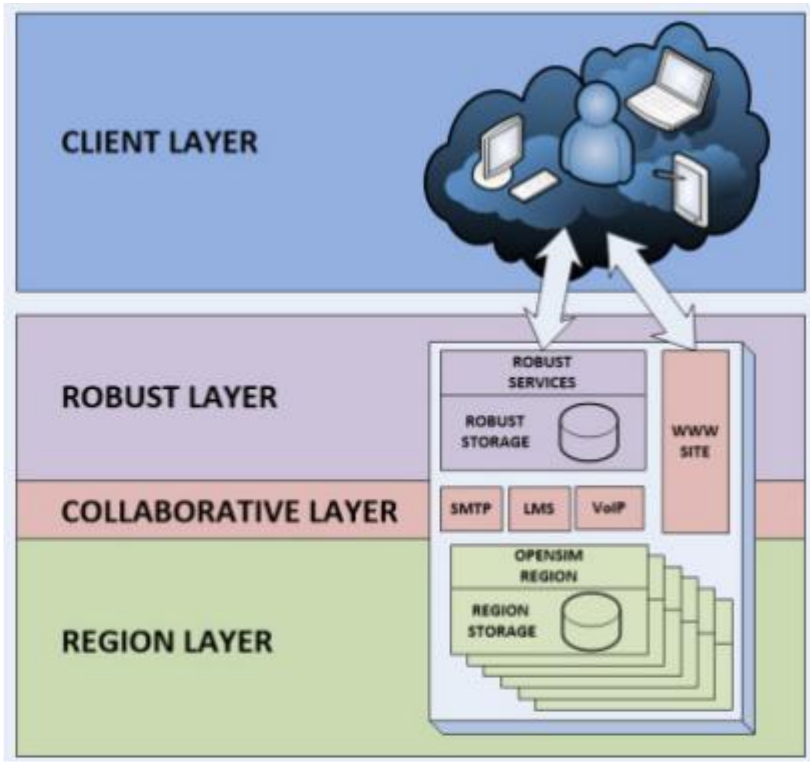

Figure 4: Solution Architectural Layers

Client Layer: The client layer encompasses the end-user's virtual world experience; this consists primarily of the client viewer that is loaded on the end-users computer. The client viewer is the application that provides the end-user with a window that allows them to experience the virtual environment; essentially, it frames the end-user experience and defines their interactions. At present, the OpenSimulator application distribution does not come with its own client viewer; however, several client viewers have been developed to date that support viewing OpenSimulator virtual world spaces.

While evaluating the solution options, several client viewers were identified and tested for functionality with the Firestorm Viewer retaining the functionalities that meet established requirements. Other than their ability to view OpenSimulator virtual world content, each client viewer was evaluated for their ability to:

1. Access key collaborative functions including voice chat and media viewing in-world
2. Present detailed virtual content including mesh support
3. Import, export and manipulate virtual content while interacting within the virtual world

It should be noted here that two browser-based experimental clients were evaluated but failed to meet either stability or minimal functionality requirements.

Robust Layer: Currently, an OpenSimulator grid deployment provides for a ROBUST (Redesigned OpenSimulator Basic Universal Server Technology) service module that manages fourteen frontend, client-facing services. For scalability, ROBUST can run as a single application managing all its related services or can be separated into several ROBUST server instances, each running one or more services. In this case, all ROBUST services were initially maintained on a single VM instance utilising a separate robust storage VM for data management; however, initial penetration testing highlighted the need to move all but the login service behind the firewall within the solution's own private IP domain, thus splitting the ROBUST services into two separate VM's. Future consideration was made to further separate out the grid and asset services into separate VM's once the grid began to build-out content and began to realise an increased concurrent user level.

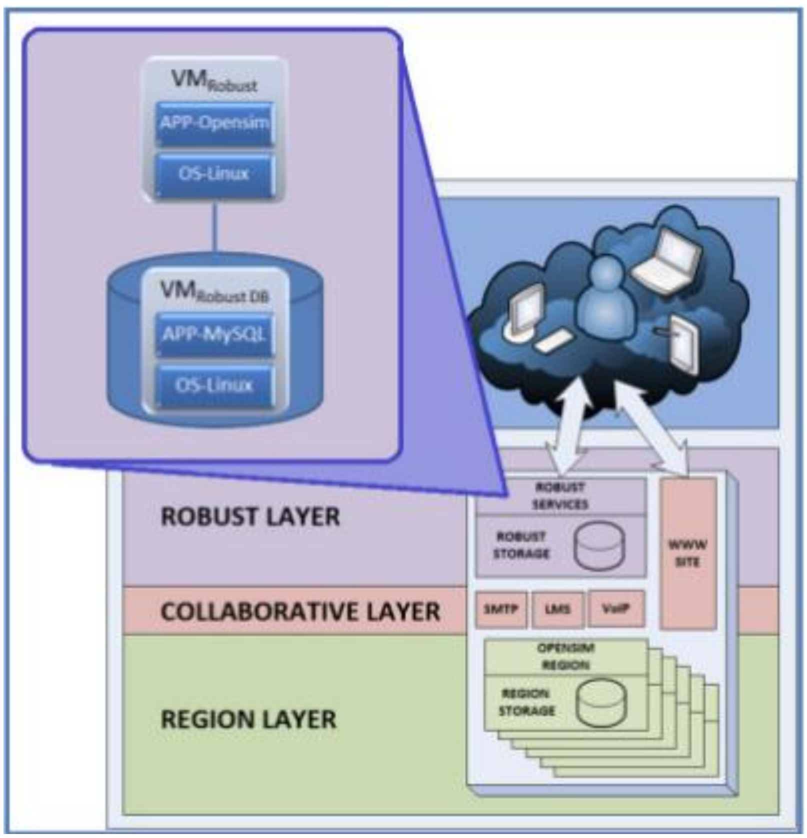

Figure 5: ROBUST layer

The bulk of the solution security concerns are managed at the ROBUST layer. Identity management, access rights management and content ownership management are all initialiised at the ROBUST layer. Identoty management within a given avatar session is based on the capability concept where each entity (avatar, content object, texture, etc.) receives a unique URL for each given session of avatars' interaction; these capabilities are dynamically generated UUID based URLs that act as session tokens for a given period, based on the activity. Access rights management or Role Based Access Control (RBAC) is considered using various privilege user categories (i.e. 'God' (highest role), 'Region Owner', 'Parcel Owner', etc.). Finally, content ownership management is based on the established and use of two distinct in-world roles: the Creator and the Owner.

Collaborative Layer: There are several collaborative service capabilities that, by default, are not included with the OpenSimulator distribution. Based on the functional requirements identified, there are four collaborative services deemed essential to the case virtual world solution; email (SMTP); learning management solution (LMS); voice chat (VoIP); the website services. For the case solution, each of the four collaborative services receives their own VM. The following open-source solutions were selected to meet these collaborative requirements:

- Website Service: Aside from the need to provide a basic frontend to present grid information to the public (i.e. grid location and address, grid activity, end-user setup, grid support, etc.) the website provides the grid with a platform for handling various end-user functions that are not included with the OpenSimulator distribution including: account creation and updates; password recovery; end-user inventory management; welcome page for logging in; and online administrator site management tools.

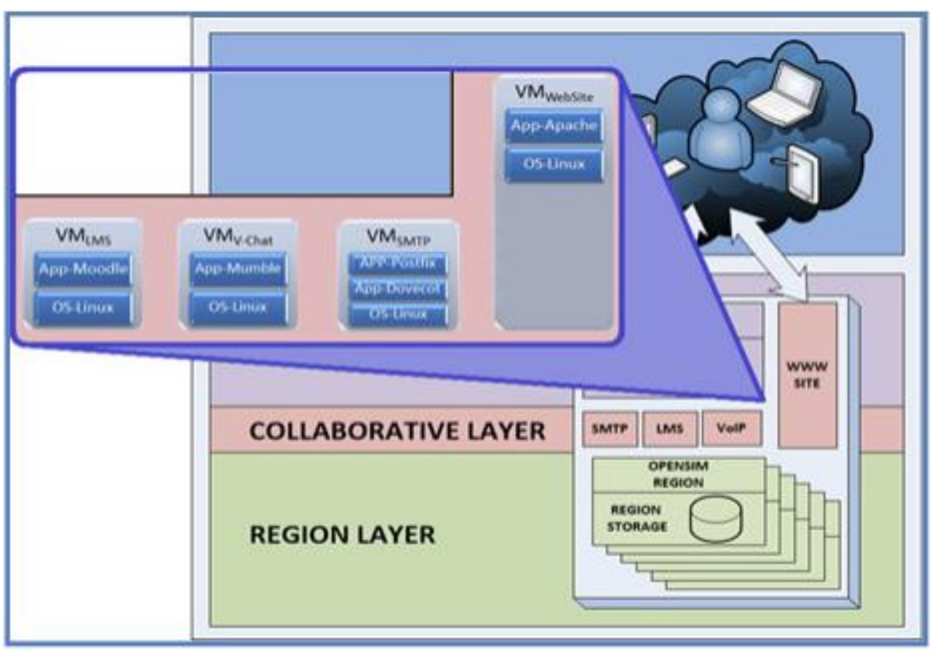

Figure 6: Collaborative Layer

Since the operating environment selected for this case is Linux-based, a default apache web service was installed, and WiFi was selected to provide a basic web account management front-end. WiFi is an add-on module that provides an embedded Web application for handling user registrations. It supports web-based functionalities to include Account creation; configurable default avatars for new accounts; account updates by both users and administrator; account deletion; password recovery via email; and basic management of user inventory.

- Learning Management Service (LMS): The LMS functionality and integration with OpenSimulator are supported by Moodle and SLOODLE respectively. Moodle is an e-learning management solution designed to assist academics in creating online course offerings with a focus on interaction and collaborative construction of content. The SLOODLE (Simulation Linked Object Oriented Dynamic Learning Environment) project provides the integration of OpenSimulator with the Moodle learning management solution.
- Email Service (SMTP): Simple Mail Transport Protocol Services (SMTP) is required for several of the grid service needs including password recovery, in-world email and SLOODLE collaboration. Several viable Linux-based Mail Transfer Agent (MTA) solutions are available, however for this case Postfix was selected.
- Voice-over-IP Service (VoIP) or Voice Chat: The open-source application Whisper that is based on Mumbe was selected as the voice-over-IP (VoIP) application to provide voice chat functionality to the virtual world solution. The voice chat solution consists of an OpenSimulator region module and and Murmur which is the Mumble voice server. The region module, referred to as MurmurVoiceModule, communicates with murmur to open channels and to register users. Although the Mumble solution is currently limited to a select number of client viewers it was selected over other open-source options for its avatar lip-sync support and speaker indication functions.

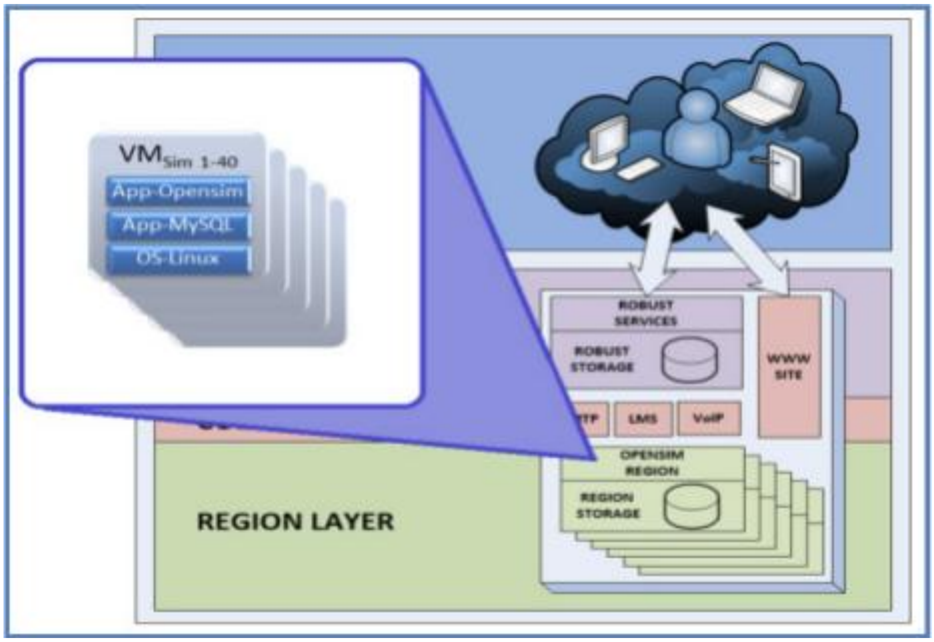

Figure 7: Region Layer

Region Layer: As noted earlier, OpenSimulator supports one or more regions. A region in OpenSimulator is the virtual space that is seen when the end-user logs in; it is the environment where all virtual activity occurs and may contain a flat piece of land, an island, mountains, a plain, buildings, a combination of all of these, or simply be a vast ocean space. For this case solution, each region and its data store were encapsulated within their own VM.

By default, each region has a virtual land surface of 256 meters squared; however, OpenSimulator can merge multiple regions into a single region that is referred to as a mega-region. Larger scale virtual workspaces consisting of multiple OpenSimulator regions are referred to as grid. The content within each region can contain upwards of 15,000 primitive (referred to as prims) objects.

Regions are organised by way of a two-dimensional coordinate system that allows for the precise locating of each region. Scalability of the solution is further demonstrated by the user of two of OpenSimulator's built-in archive functions. At the region level, OpenSimulator supports the archiving of specific content/inventory for the entire region. The OpenSimulator Archive (OAR) function saves an entire region's content to a single file so that it can be later reloaded to any select region. The Inventory Archive (IAR) function saves selected content from an avatars inventory to a single file so that it can be later reloaded either to the same avatars account or loaded to another avatar's inventory account

2. Game Mechanics

2.1 Interaction between Non-player characters and players

Players can engage with non-player characters (NPCs) by clicking on them. Following this action, a conversation box pops up, presenting various choices or responses for the player to select as shown in Figure 8. Choosing one of these responses advances the dialogue.

Figure 8: Example of interaction between the player and an NPC

2.2 NPCs responses

NPCs can respond to a click or a dialogue option by:

- Displaying a new dialogue
- Giving a notecard
- Changing their position (see limitations on 2.3)
- Performing an animation (see limitations on 2.3) or a combination of the above.

2.3 Limitations of NPCs responses

Technical constraints necessitate consideration of the following limitations in scenario creation:

- Dialogue screens can accommodate up to 256 characters of text. This restriction is specific to the Greek language, yet adopting it universally across all languages is advisable to ensure a uniform experience.

- Notecards function similarly to tangible letters held in the player's inventory (illustrated in Figure 4). They can comprise multiple pages of text and are suited for delivering educational content, although lengthy texts are discouraged.

- While NPCs can change their location, it's advised to minimize their movements due to technical limitations. NPCs can move within the same building but not to different buildings associated with other scenarios. Their movement around obstacles can be problematic, and they are not designed to follow the player. Instead, their movements are pre-set (e.g., moving to point x, then to point y, and so on).

- NPCs can perform animations, though the selection is limited, and it's suggested to use as few animations as necessary because of technical constraints. NPCs are unable to execute more than one animation at a time; thus, actions such as sitting and waving must be treated as separate animations. During an NPC's animation, interaction through dialogue with the player is temporarily halted, meaning players cannot engage with the NPC until the animation concludes. Moreover, NPCs are restricted from moving and animating at the same time.

2.4 Interaction between the player and the virtual environment

Players can start an interaction with the environment by:

- Clicking on an object

- Passing by a certain place (e.g. through a door)

2.5 Environment responses

The environment can respond in the following ways:

- Displaying a dialogue

- Giving a notecard

The same limitations as NPC’s responses apply.

2.6 A few more things to consider when writing a scenario

In crafting a scenario, developers are required to integrate the previously discussed features and account for the inherent technical constraints of the platform. This entails meticulously planning out the outcomes of player-NPC interactions, detailing the consequences of each player action. For instance, selecting the second dialogue option should trigger both a new message and a specific sound effect, while choosing the third option prompts the NPC to move.

2.7 Player Statistics

To further improve the players experience, as well as the outcomes of the project, we have implemented the gathering of the players' statistics inside the game. As can be seen in Figure 9, we are gathering the user's ID, the scenario they're playing, the time when they entered and when they completed the scenario, as well as additional comments. The times are collected to examine how long each participant takes to complete each scenario, offering us results such as the effectiveness of each scenario and the comparison between short and longer scenarios, which can be used as a baseline for scenario creation in the future. An example of additional comments is the time the player takes to find each token in the "Glimpse into the Future" scenario, which can be used to adjust the difficulty of the scenario.

```
MariaDB [divaopensim]> SELECT * FROM user_scenarios
    -> ;
+--------------------------------------+-----------+-----------------+---------------------+---------------------+---------+----+
| user_id                              | user_name | scenario_number | timestamp_entered   | timestamp_completed | comment | id |
+--------------------------------------+-----------+-----------------+---------------------+---------------------+---------+----+
| 94df2de4-4223-47f1-b249-4c3ecf2d0a45 | Admin VL  | 4               | 2024-10-23 11:32:16 | 2024-10-23 11:32:16 | NULL    |  1 |
| 94df2de4-4223-47f1-b249-4c3ecf2d0a45 | Admin VL  | 4               | 2024-10-23 11:34:57 | 2024-10-23 11:35:09 | NULL    |  2 |
| 94df2de4-4223-47f1-b249-4c3ecf2d0a45 | Admin VL  | 1               | 2024-10-23 12:45:40 | 2024-10-23 12:45:40 | NULL    |  3 |
| 94df2de4-4223-47f1-b249-4c3ecf2d0a45 | Admin VL  | 1               | 2024-10-23 12:47:14 | 2024-10-23 12:48:49 | NULL    |  4 |
| 94df2de4-4223-47f1-b249-4c3ecf2d0a45 | Admin VL  | 4               | 2024-10-25 15:09:19 | 2024-10-25 15:12:19 | NULL    |  5 |
| 94df2de4-4223-47f1-b249-4c3ecf2d0a45 | Admin VL  | 5               | 2024-10-28 22:39:53 | 2024-10-28 22:40:10 | NULL    |  6 |
| 94df2de4-4223-47f1-b249-4c3ecf2d0a45 | Admin VL  | 5               | 2024-10-28 22:40:56 | 2024-10-28 22:41:55 | NULL    |  7 |
| 94df2de4-4223-47f1-b249-4c3ecf2d0a45 | Admin VL  | 5               | 2024-10-28 22:42:37 | 2024-10-28 22:42:37 | NULL    |  8 |
| 94df2de4-4223-47f1-b249-4c3ecf2d0a45 | Admin VL  | 5               | 2024-10-28 22:45:06 | 2024-10-28 22:47:29 | NULL    |  9 |
| 94df2de4-4223-47f1-b249-4c3ecf2d0a45 | Admin VL  | 5               | 2024-10-28 22:49:18 | 2024-10-28 22:49:45 | NULL    | 10 |
| 94df2de4-4223-47f1-b249-4c3ecf2d0a45 | Admin VL  | 5               | 2024-10-28 22:51:21 | 2024-10-28 22:51:48 | NULL    | 11 |
| 94df2de4-4223-47f1-b249-4c3ecf2d0a45 | Admin VL  | 5               | 2024-10-28 22:54:49 | 2024-10-28 22:55:17 | NULL    | 12 |
| 94df2de4-4223-47f1-b249-4c3ecf2d0a45 | Admin VL  | 5               | 2024-10-28 23:01:22 | 2024-10-28 23:01:58 | NULL    | 13 |
| 94df2de4-4223-47f1-b249-4c3ecf2d0a45 | Admin VL  | 5               | 2024-10-28 23:02:41 | 2024-10-28 23:03:18 | NULL    | 14 |
| 94df2de4-4223-47f1-b249-4c3ecf2d0a45 | Admin VL  | 5               | 2024-10-28 23:05:17 | 2024-10-28 23:05:38 | NULL    | 15 |
| 94df2de4-4223-47f1-b249-4c3ecf2d0a45 | Admin VL  | 5               | 2024-10-28 23:06:22 | 2024-10-28 23:06:42 | NULL    | 16 |
| 94df2de4-4223-47f1-b249-4c3ecf2d0a45 | Admin VL  | 5               | 2024-10-28 23:08:05 | 2024-10-28 23:08:17 | NULL    | 17 |
| 94df2de4-4223-47f1-b249-4c3ecf2d0a45 | Admin VL  | 5               | 2024-10-28 23:09:03 | 2024-10-28 23:10:44 | NULL    | 18 |
```

Figure 9: Example of the statistics gathered in the Virtual World

3. Demonstrative Use Cases

The initial version of the 3D Virtual World Learning Environment (VWLE) involves crafting a multiuser 3D immersive Virtual World, designed to mimic real-life curriculum activities. This virtual world was constructed based on the specifications outlined in the Scenarios deliverables, where it encapsulates the scenarios into the learning platform. It features a comprehensive setup with buildings, avatars, non-player characters (NPCs), interactive elements, multimedia educational resources about environmental problems, informational boards detailing user instructions, and various landscapes. All interactions within this world are facilitated by custom scripts developed specifically for the GUESS project, ensuring the virtual world aligns with the learning goals, scenario-based specifications, and educational frameworks considered during the planning stage.

3.1 Tutorial

To adequately prepare the player for the mechanics and actions available in our virtual world, a carefully designed introductory scenario was created. This tutorial is where new players start their venture in the virtual world. First, the player learns how to navigate around the virtual world. A set of instructions show the player the key bindings for avatar movement, before they are teleported to a short level, where they can test their navigation skills by navigating through a set of obstacles (Figure 11). Second, they are informed about the basic camera controls, so that they can better explore the virtual world (Figure 12). The player then learns how to enter scenarios using the buttons found outside of each of them (Figure 13). They also learn that if the scenario is occupied, they will be informed. Finally, the player learns how to interact with NPCs through a set of instructions and talking to Jane NPC to try it out (Figure 14). The player is now ready to explore and experience all that the virtual world has to offer.

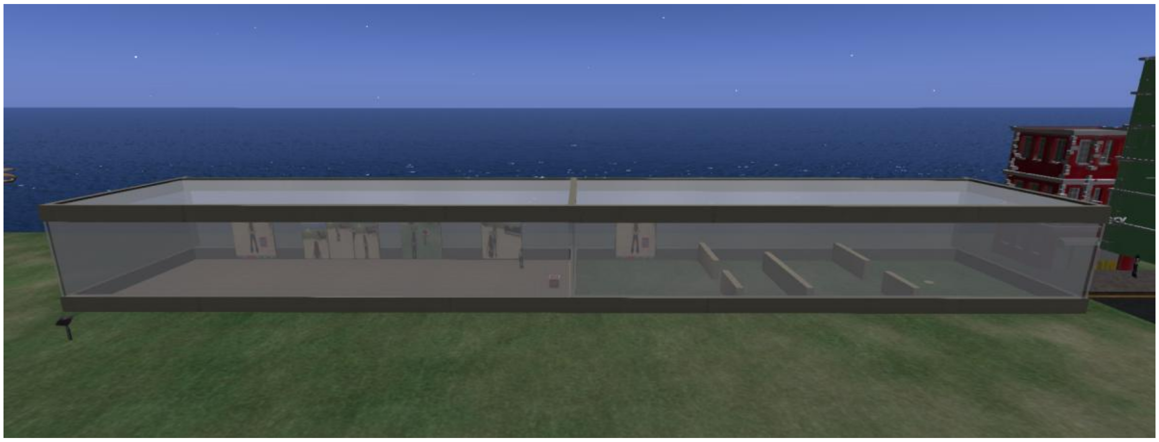

Figure 10: Tutorial room overview

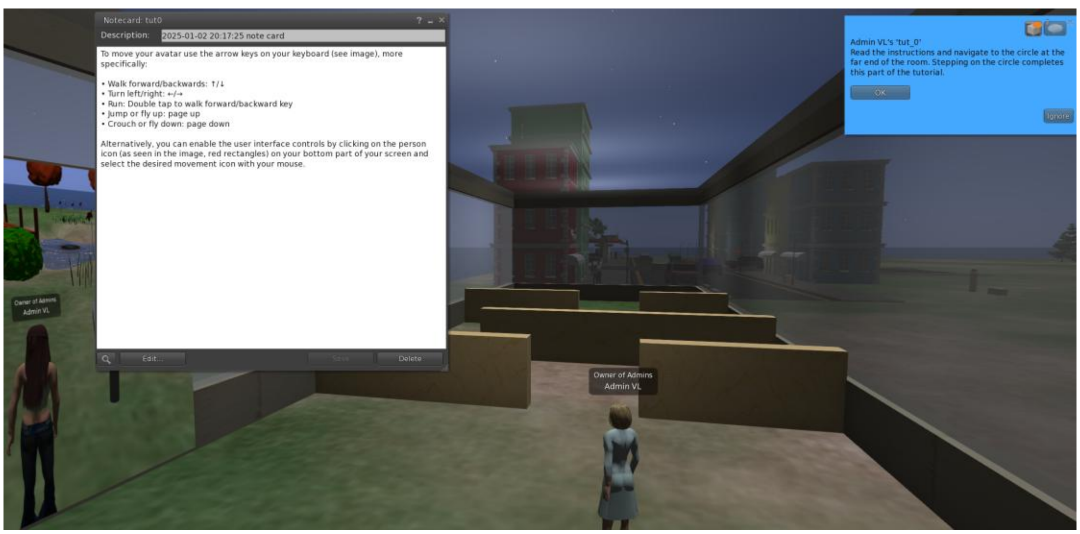

Figure 11: The player inside the interactive navigation part of the tutorial

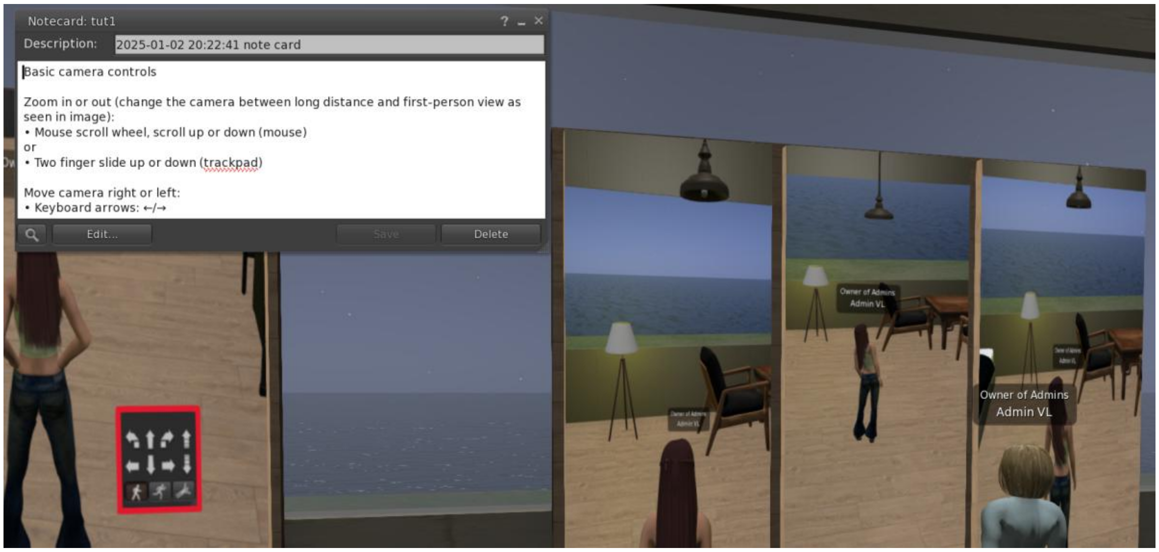

Figure 12: The player learning about the camera controls in the tutorial

Figure 13: The player learning how to enter scenarios in the tutorial

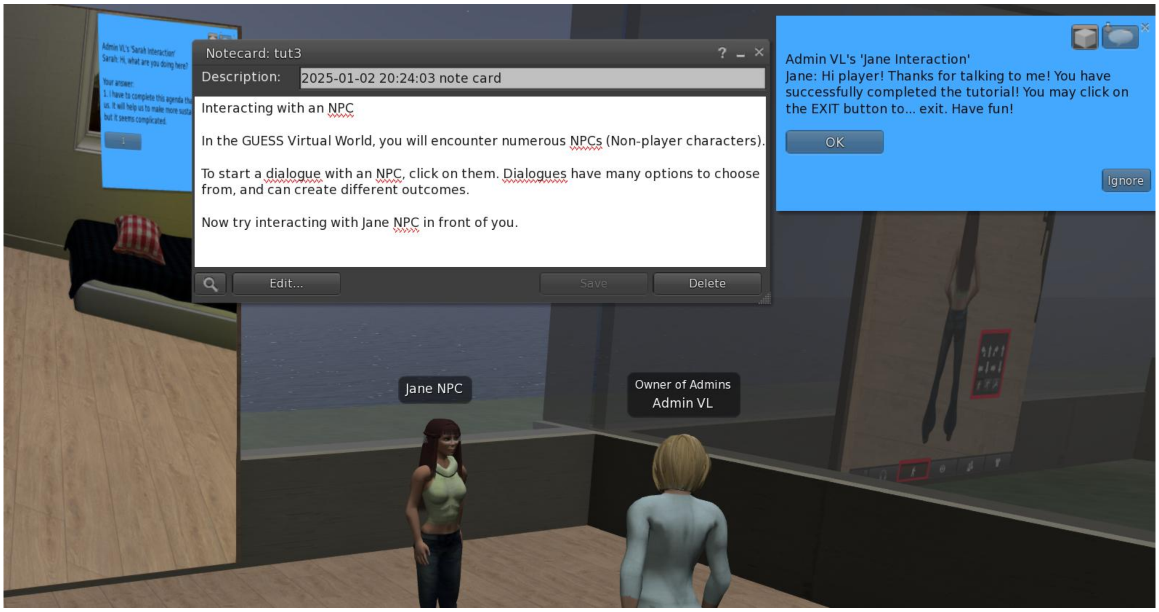

Figure 14: The player learning how to interact with NPCs

3.2 Observation Tower

The observation tower serves as the current reception room. All the use cases are offered in all partner languages. The player can select to visit the observation tower, from where they have the view of the whole world of GUESS. Figure 15 shows the view the player has when they are on the observation tower.

Figure 15: The view of the GUESS world from the observation tower

3.3 Glimpse into the future Scenario

The “Glimpse into the future” scenario is set at a hiking trail near a lake. The player encounters an NPC, Hiker Nick, sitting on bench admiring the natural scenery. The hiker reveals that he is greatly worried about climate change and its detrimental effects on the environment. To inform the youngster and sensitize him to the issue he paints a bleak picture of what this scenic view could look like in the future. The hiker asks some basic questions about climate change in the form of a short quiz. Then, as the hiker explains what could happen if climate change is left unchecked, the view transforms form a lush landscape to a withered, barren land. The player is then asked to explore and find tokens scattered and hidden around the area. Each token gives access to information about climate change in the form of an in-game note card. After collecting all tokens, a board appears near the hiker. The items on the board must be sorted into three categories relating to climate change: causes, effects, and solutions. The player can use their knowledge and the information they have acquired to sort the items correctly and complete the game. The educational goal of this game scenario is to improve the student’s understanding of the concept of climate change, as they will be presented with information about some key causes and effects of climate change, as well as some ways to combat the phenomenon.

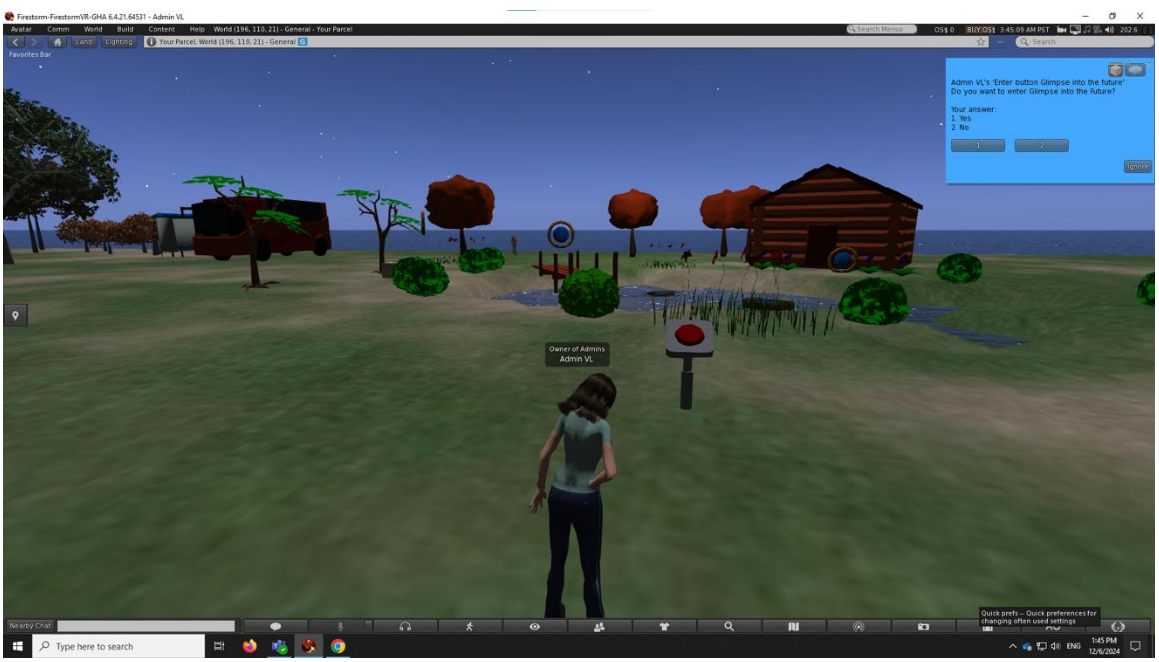

Figure 16: Entering the "Glimpse into the future" Scenario

Figure 17: The player interacting with an NPC in the scenario

Figure 18: Example of a quiz question and an information card

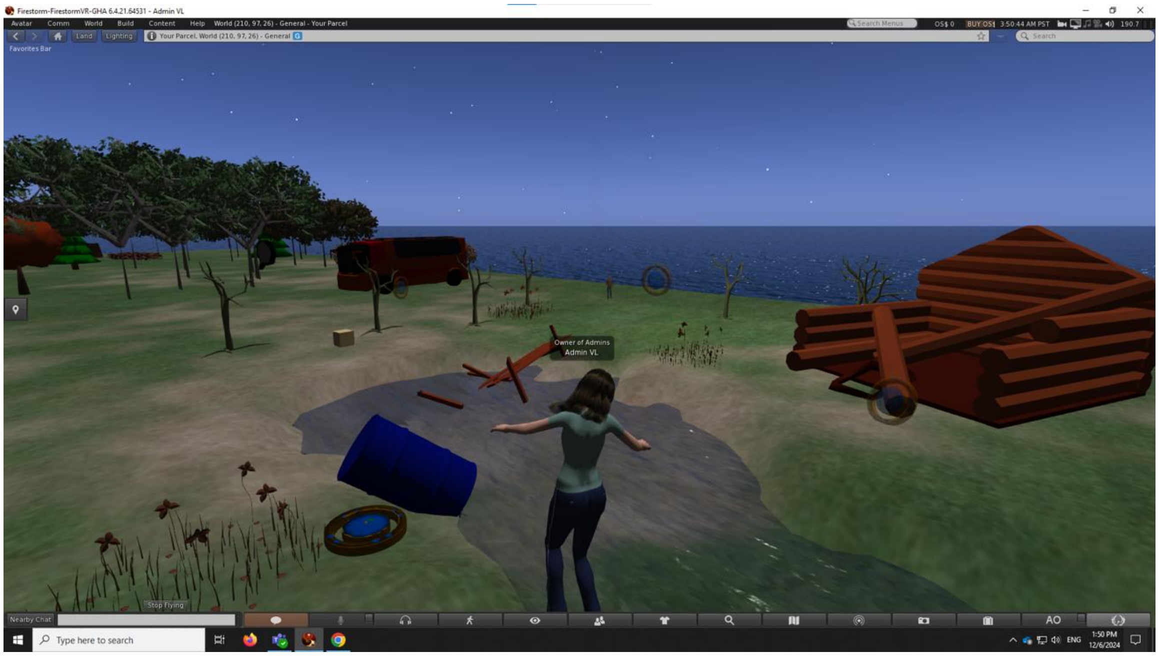

Figure 19: The player navigates in the scenario

3.4 Lake Lifeline Scenario

The "Lake Lifeline" scenario is set near a polluted lake. The player is a scientist, working to prevent global warming and minimize its effects. They visit a lake whose waters have been warming up the past months. They do some measurements and compare data from previous years, seeing that the situation in the lake is bad, and making actions to change it. They visit the town's Mayor to inform him about the situation and together they organize a gathering in the town hall to inform the residents of the town about the problem. They plan an afforestation near the lake and in the end, they visit it again and see that the situation has improved. The educational content of this scenario is the rising temperature of lakes due to global warming, the reasons behind it and the ways it can be prevented or reversed. Inside the scenario, the player has the chance to answer a short quiz to understand the reasons of global warming, as well as see how it can be prevented, and also play a mini game of afforestation, where they can read some interesting facts about each tree they want to plant.

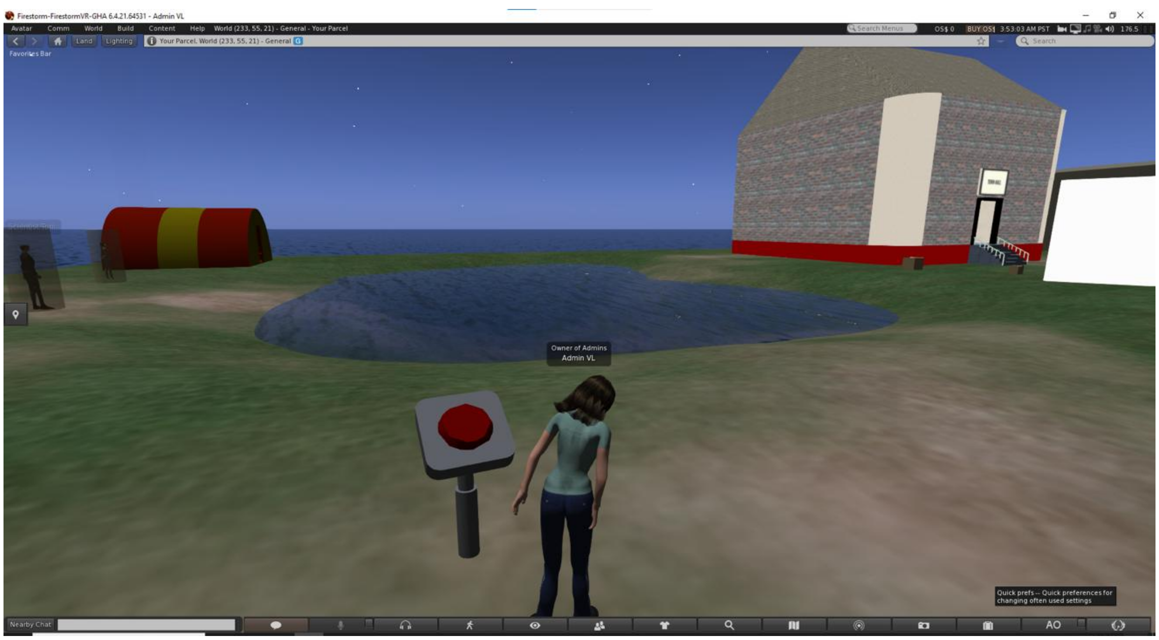

Figure 20: The player ready to enter the Lake Lifeline scenario

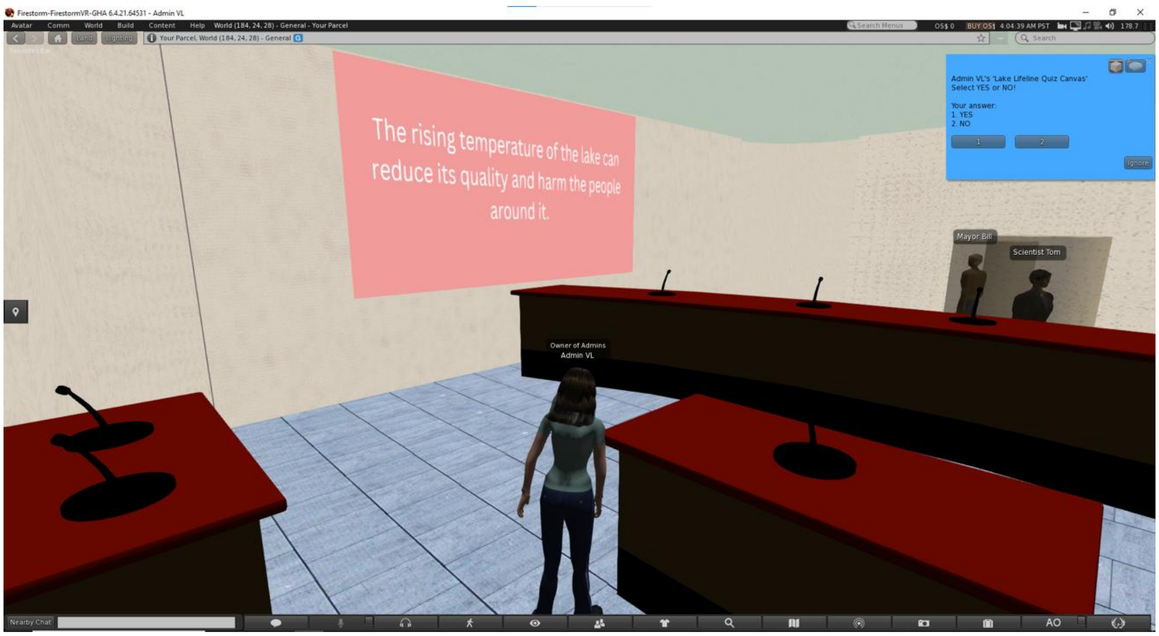

Figure 21: Example of the player interacting with the quiz and the NPC players in the scenario

Figure 22: Example during the afforestation

3.5 Sustainability Awareness Scenario

The “Sustainability Awareness” scenario takes place in the room of a student. The student (i.e., the player) is at their house and have to make a school project. The task of their project is to complete an agenda of things that can make their life more sustainable, so that they can then start to implement these activities in their everyday life. During the scenario, the player, with the help of their “sister” Sarah, has the chance to check if their common everyday actions are sustainable or not, as well as see useful tips on how they can change their habits to be more sustainable. Those activities revolve around actions that should be present in most households, like closing the tap while brushing your teeth, and preferring reusable products (e.g., multi-use water bottles and shopping bags). For each action that is sustainable, the player receives +1 to their score, which is displayed to them towards the end of the scenario. After they receive their score, NPC Sarah provides them with further information on how to continue to be sustainable, and it is recommended to the player to visit the 17 UN Sustainable Development Goals (SDGs) webpage to learn more. The educational content in this scenario is the creation of self-awareness on personal contribution to environmental sustainability.

Figure 23: Example of the player in the Sustainability Awareness scenario

Figure 24: Example of the facts shown to the player in the scenario

3.6 Desalination Factory Scenario

The “Desalination Factory” scenario takes place inside a factory. The player is invited to visit a Desalination plant. Their aim is to explore the desalination plant and interact with the workers, gaining knowledge of the entire water desalination process. During the scenario, the player has the chance to play a game to see the steps of desalination in a factory, as shown in Figure 25, as well as interact and learn from the workers, as shown in Figure 26. In the first game, the player is shown a board shown the different stages of the desalination factory, and next to it cards with small descriptions. The player needs to drag each description to the related stage of the factory to see the steps followed for the desalination. After that, the player is presented a mini game where chemical drums are falling off the shelves and they need to dodge them and put them back in the right place. In the third mini game the player is presented with some chemical symbols, and they have to match the chemical symbol to the description. When the player makes the match, an explanation appears explaining the usefulness of the chemical in water treatment. Then, the player is presented with a quiz to test the knowledge they gained. Lastly, the player has to complement the map they drew at the start of the game with the components of the process in each of the steps, and then answer another quiz. The educational contents to be embedded in this scenario are the problematic of potable water scarcity for a large part of the population, the desalination as one of the processes for obtaining potable water for coastal arid countries, stages, processes, technologies and components of a desalinization plant, the environmental impacts from the desalinization process, water distribution activity, and good practices for water management.

Figure 25: The player enters the Desalination Factory scenario

Figure 26: Example of the player interacting with a worker NPC

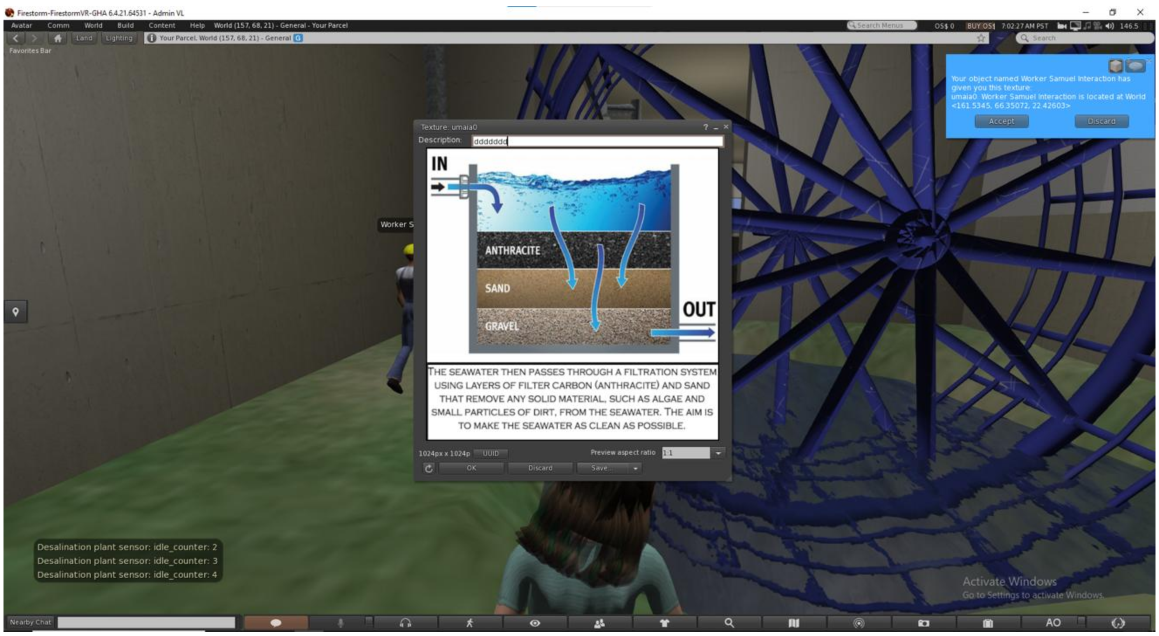

Figure 27: Example of the facts shown to the player during the scenario

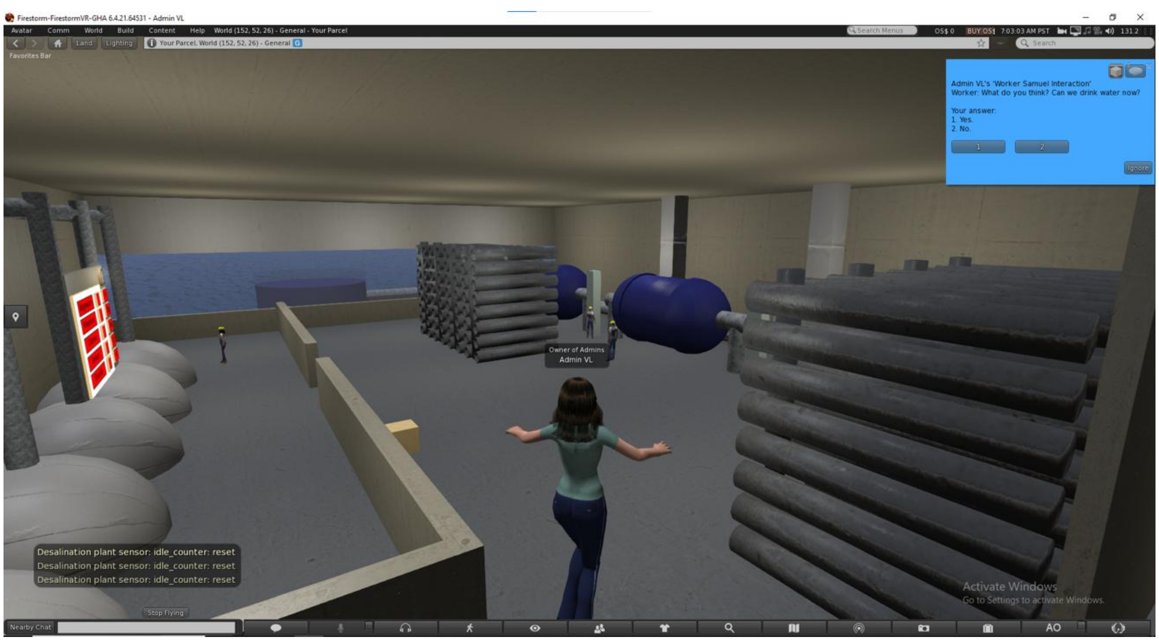

Figure 28: Example of the player navigating in the factory

3.7 Learning about Composting Scenario

The “Composting” scenario takes place at a local park promoting composting. The player is attending along with their friend Kate, to learn more about how to make their own compost, and what benefits this process offers. The NPC Employee Autumn guides them through the definition and process of composting, as well as the scientific details behind it. During the scenario, the player learns what materials can and cannot be used in composting. They will then evaluate their knowledge by playing a minigame where the select the materials that are compostable to make a compost (Figure 30 and Figure 31). The player is then explained the procedure of composting, along with the order of operations that have to take place to compost successfully. They evaluate this knowledge by interacting with a minigame that has them correctly sort the actions required to build a compost structure (Figure 32 and Figure 33). After completing the minigame, NPC Employee Autumn provides the player with tips on what to be careful on when building your own compost. Finally, the player learns what the uses of a compost are, together with the benefits of building one at home. The educational content in this scenario is the benefits of a made-at-home compost, how to make such a compost, and what its overall contribution is to environmental sustainability.

Figure 29: The player enters the "Composting" scenario

Figure 30: The player interacting with the compostable/non compostable materials minigame, with no panel selected

Figure 31: The player interacting with the compostable/non compostable materials minigame, with a few panels selected

Figure 32: The player interacting with the compost order minigame

Figure 33: The player having completed the compost order minigame

3.8 Proper Recycling Scenario

The "Proper Recycling" scenario begins at a classroom. The player is attending a lecture, along with their friend Nick, about recycling and common misconceptions associated with it. The NPC Teacher Tom asks the player a few questions about recycling at home. NPC Nick indicates that it is common to deem everything that is made of paper, plastic, glass, or and metal to be recyclable. The player plays a minigame where they are asked to indicate the items that are recyclable and those that are not (Figure 35 and Figure 36). Despite most of the selectable materials being commonly associated with recycling, the player is explained that only two of the seven are indeed recyclable. Teacher Tom provides them with a list of everyday items that are often sources of contamination of recycling batches, since they are not themselves recyclable (e.g., pizza boxes, takeaway coffee cups, and biodegradable plastic). The player is then led to their home, where they are tasked with teaching their family how to recycle properly and help break the habits of contaminating the recyclables bin with non-recyclable items. They teach their dad about how much recyclable material is wasted in landfills because it was in a contaminated batch. Finally, they are tasked to answer “Yes” or “No” to questions about achieving proper recycling (Figure 37). The educational content in this scenario is the severity of the issues caused by improper recycling, when is an item recyclable, and what the contribution of proper recycling is to environmental sustainability.

Figure 34: The player enters the "Proper Recycling" scenario

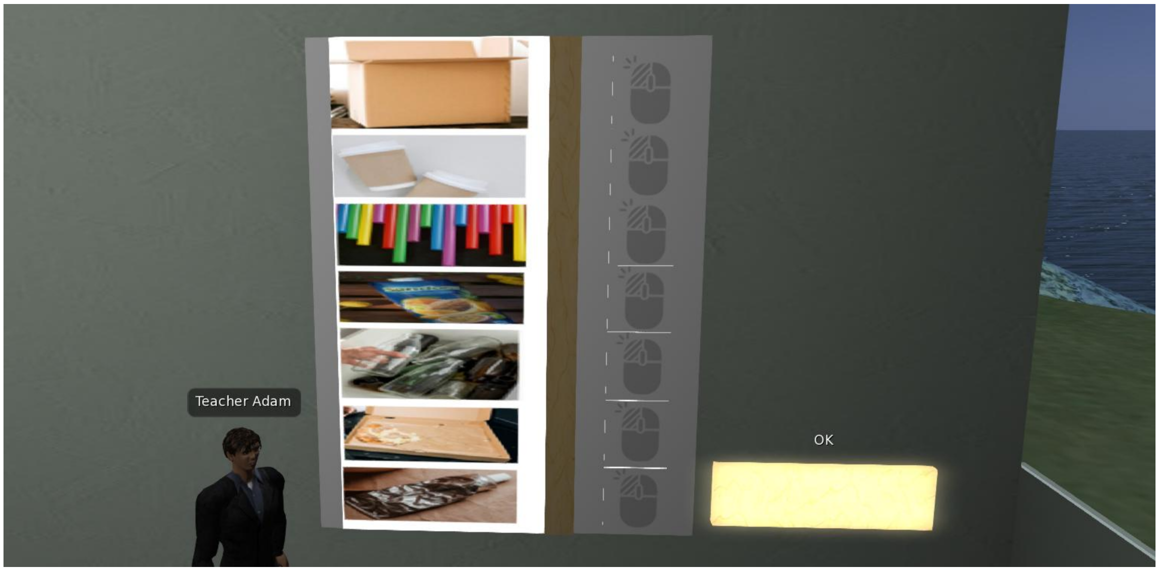

Figure 35: The player interacting with the recyclable or not materials minigame

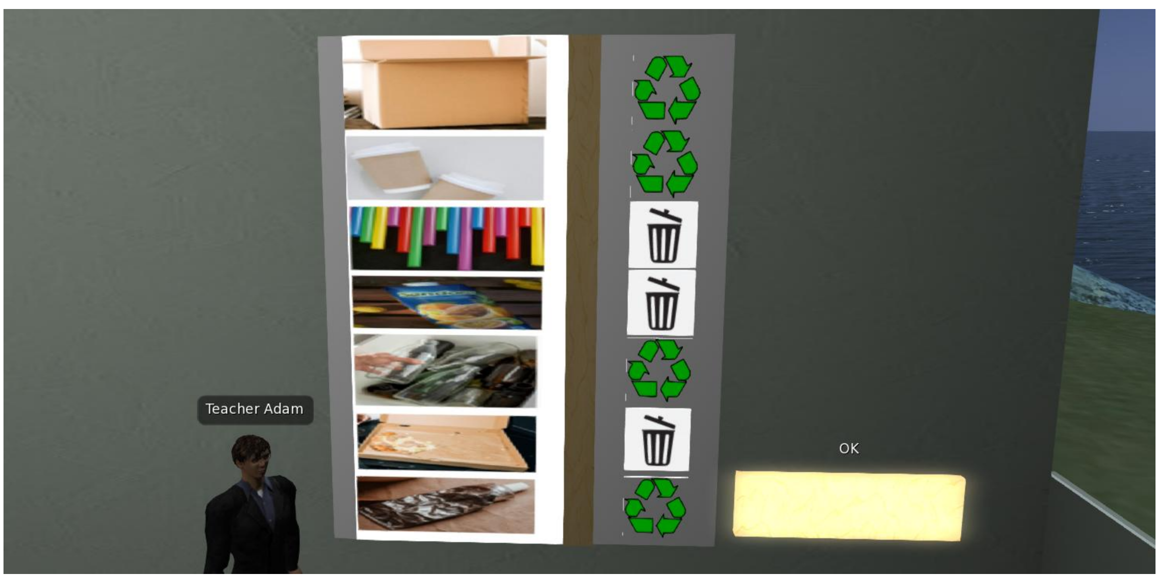

Figure 36: The player having separated the recyclable and not materials in the respective minigame

Figure 37: The player answers the quiz on what to do and what not to do to achieve proper recycling

3.9 Circular Economy Scenario

The “Circular Economy” scenario starts outside a clothing store on a busy street. The player is taking a walk around the shops along with their friend Jack. Outside a clothing store, they encounter a group of protesters. The player talks to the leader of the protest group, NPC Panam, to find out that they are protesting fast fashion, and are attempting to inform the public on the importance of buying clothes from recycled fabrics, and how that enhances the circular economy. The player learns about the environmental impact of the textile industry. They are then presented with possible solutions to alleviate this impact, primarily centred around circular fashion and slow fashion. For each of these solutions, they are given detailed definitions and what their effects can be on environmental sustainability. Furthermore, e-waste is mentioned, along with food waste and packaging waste. Finally, a comparison between linear economy and circular economy is presented to the player. They are then presented with a quiz that goes through all the relevant aspects of circular economy they just learned about (e.g., textiles, linear versus circular economy, etc.). The educational content in this scenario is the definition of circular economy and all its aspects, why it is important to switch from a linear economy to circular, and what effects of this change would be to environmental sustainability.

Figure 38: The player enters the "Circular Economy" scenario

3.10 Polluted City Scenario

The “Polluted City” scenario starts outside the library on a busy city street. The player, along with their friends Julia and Jonas, decided to take a walk to get some fresh air after working on a group project about pollution. As the group walks through the city, they begin to notice symptoms of severe pollution (e.g., people coughing and smog). Through the conversation, the player learns about the primary contributors to urban pollution. They then play a minigame that teaches them about how the harmful gasses emitted by cars can be reduced (Figure 40 and Figure 41). The group is then moved to a doctor (NPC Doctor Jhanna) because they notice NPC Julia getting sick from the smog and pollution. The doctor provides them with a comprehensive list of all the symptoms and health risks of exposure to urban pollution, in detail. Furthermore, the player tests their knowledge on the subject, by completing a quiz given by the doctor. Finally, the group can decide upon a way to reduce pollution, primarily by using and encouraging the use of public transport, or using bicycles where able to get to their destination. A few decisions about how to reduce pollution when a private vehicle must be used are also made by the player (e.g., keeping the car serviced and avoiding idling). The educational content in this scenario is urban pollution, its causes and effects on human health, and how it can be combatted to pursue environmental sustainability.

Figure 39: The player enters the "Polluted City" scenario

Figure 40: The player interacting with the catalytic converter minigame

Figure 41: The catalytic converter minigame solved

3.11 Forest Scenario

This "Walk in the Forest" scenario guides learners through a virtual woodland where each stop along the path reveals a new lesson about how forests work and why they matter. As players move from one environment to another—say, from a sparse sandstone grove into a lush, nutrient-rich stand—they encounter simple quizzes and prompts that ask them to notice details like tree species, soil color, or signs of decay. These moments of exploration and interaction are designed to help students connect what they see on screen with real-world ideas about how soil, water, and biodiversity come together to support healthy forests. Beyond just pointing out facts, the scenario encourages reflection on the broader benefits forests provide—everything from clean water and timber to habitat for wildlife and places for people to enjoy. By weaving short questions and dialogue with a virtual companion into the walk, the experience keeps learners engaged and prompts them to think critically about topics like native versus non-native species, the role of disturbances like fallen logs or insect outbreaks, and the balance between human use and ecological health. In the end, it's not just about facts on a page, but about building an appreciation for the unseen processes that sustain forests and the choices we make in managing them.

Figure 42: Intro of Forest scenario

Figure 43: Educational banner regarding the forests

Figure 44: Tree minigame trying to distinguish native and non-native forests

Figure 45: Informative banner regarding the native forests

Figure 46: Pile of woods

Figure 47: Diagram regarding the forest ecosystem functions

3.12 Language Selection Room

The final version of the 3D VWLE is multilingual and is offered in all partners' languages. To this end, a language selection area was created (Figure 49), where students can choose one of the available languages (English, Portuguese, Czech, Spanish and Greek) for the environment. To select their language, users must click on the corresponding flag and then on the exit to begin.

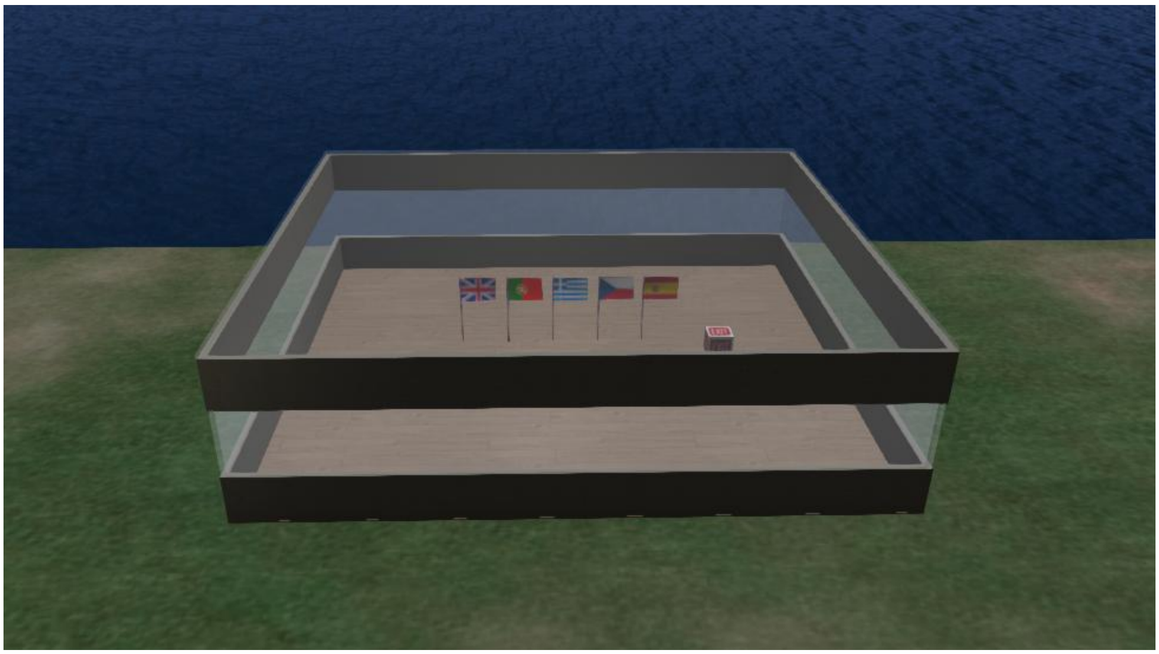

Figure 48: Language Selection Room Overview

Figure 49: Flag Selection for Language set.

4. Testing Periods

To validate the design and impact of the GUESS virtual world developed in OpenSim, a structured, three-phase testing process was implemented. This process aimed to evaluate both the pedagogical and technical aspects of the platform, ensuring that the learning experience was not only functional and user-friendly but also practical in raising environmental awareness among university students. The testing periods served as critical checkpoints for gathering data, refining the scenarios, and ensuring that the virtual environment was well-aligned with the project's educational goals. Each phase engaged different stakeholders: experts in educational gaming, project partners, and students, creating a comprehensive feedback loop from diverse perspectives.

The first testing phase took place with the participation of four educational game experts, each with experience in designing or evaluating digital learning tools. These individuals were given access to the early version of the GUESS environment and encouraged to explore and interact with the educational scenarios freely. Following this guided session, we conducted semi-structured interviews with each expert, during which they shared open-ended insights on aspects such as usability, clarity of objectives, educational potential, and visual design. Their feedback revealed several essential themes: while the platform was generally praised for its concept and structure, improvements were suggested in terms of guiding the player, reducing cognitive load in specific areas, and refining visual storytelling elements to convey the learning objectives better.

Following this, the second phase of testing was conducted in January 2025 during the 11th online project meeting, which brought together all partner organisations involved in the GUESS project. During this session, all partners joined an online walkthrough of the virtual world, led by the University of Patras. This live demonstration allowed each partner to directly test the scenarios while receiving instructions and context from the developers. After the guided walkthrough, participants were asked to complete a structured evaluation form similar to that used in the first phase. The responses showed a generally positive reception: partners found the scenarios to be user-friendly and visually engaging, and they believed the objectives were met, mainly. However, they also provided valuable suggestions such as the need for more precise instructions during tasks, more interactivity in the minigames (especially the tree-planting and token collection activities), and the inclusion of context-sensitive guidance to help users progress through the scenarios with less confusion.

Several comments from partners specifically addressed technical or instructional issues, which were used to inform the next development sprint. Some participants noted difficulties in understanding what to do next without explicit prompts. In contrast, others mentioned that the text boxes and feedback windows were sometimes inconveniently placed or challenging to read. In terms of scenario length, feedback was mixed: while some found them to be well-paced, others recommended trimming or restructuring the scenarios to maintain engagement. In the tree planting minigame, for example, a few testers found the learning content to become repetitive when the same species of tree provided identical messages upon planting. Suggestions such as displaying unique facts or unlocking more advanced content based on player choices were noted for implementation. Moreover, some reviewers proposed adding a checklist or map at the beginning of each scenario to help users track their progress and avoid missing key interactions.

To further enhance the educational dimension, partners recommended embedding more contextual information into the scenarios, such as the environmental significance of different trees or recycling practices in various countries. Several comments highlighted the need for localisation and adaptation, especially when scenarios referred to region-specific systems. Participants emphasized the importance of generalizing these models or allowing for regional customization to ensure the scenarios were relatable to students from all partner countries. Overall, the partner evaluation phase confirmed that the GUESS virtual world had strong potential as a learning tool but could benefit from iterative refinements based on user feedback. This phase was especially valuable as it represented the perspective of those responsible for local implementation and dissemination of the platform.

Before reaching the final release, a beta testing phase was conducted with a total of 121 students, allowing the consortium to evaluate system performance, identify weaknesses, and refine functionality prior to the creation of the final version of the virtual environment. Participation in this phase included 24 students from the Hellenic Open University, 35 from the University of Patras, 23 from UPCT, 15 from the University of MAIA, and 24 from Mendel University in Brno. The purpose of this intermediate testing stage was to assess the effectiveness of the initial beta features, collect user feedback, and apply necessary improvements, ultimately shaping the polished final version that was later validated in the concluding evaluation round.

The final phase of testing was conducted with students from the participating universities. A total of 126 students from all five partner institutions explored the final version of the GUESS virtual environment. Specifically, 24 students participated from the Hellenic Open University (HOU), 29 students from the University of Patras, 17 from the University of MAIA (UMAIA), 31 from Mendel University in Brno, and 25 from the Polytechnic University of Cartagena (UPCT). These students were asked to engage with the scenarios, interact with minigames, and reflect on the messages conveyed by each activity. After completing their exploration, they filled out evaluation questionnaires designed to capture their experience in terms of usability, engagement, educational content, and aesthetic quality. This phase was crucial in determining how the intended audience, university students, actually interacted with the platform and whether it effectively achieved its learning goals.

The feedback gathered from students reflected a broad spectrum of experiences and opinions, with a general trend that was overwhelmingly positive. The majority of students expressed satisfaction with the GUESS virtual environment, highlighting features such as the intuitive navigation, the immersive design of the virtual world, and the engaging nature of the environmental themes. Many participants found the scenarios enjoyable and easy to follow, praising the realistic visuals, the natural flow of activities, and the interactive elements like tree planting, recycling, and token discovery. According to the questionnaires, most students felt that the platform successfully raised their awareness about environmental issues and helped them connect theoretical knowledge with practical, relatable experiences. Several respondents explicitly noted that the virtual format made learning feel less like a lecture and more like an exploration, which contributed positively to their motivation and engagement.

Nonetheless, as with the earlier testing phases, a few areas for improvement were identified through the student evaluations. A few students commented that the minigames, although educational, could become repetitive, particularly when the same feedback was displayed after identical actions, such as planting multiple trees of the same species. Others suggested reducing the amount of on-screen text or dividing it into smaller, more digestible sections accompanied by visuals.

4.1 Expert Evaluation Session (October 2024)

The first structured testing of the GUESS virtual world took place remotely during October 2024 and involved four educational game experts with prior experience in using the OpenSim platform. These experts were granted individual access to the environment and were instructed to explore all available scenarios at their own pace freely. Although no facilitator was present during their navigation, clear guidelines were shared on how to log in, move within the environment, and enter each scenario. Their prior familiarity with OpenSim allowed them to navigate the platform confidently without requiring additional onboarding or support.

The duration of each expert's testing session ranged from approximately 52 minutes to 1 hour and 7 minutes, with the average playthrough lasting around 1 hour. During this time, the experts were encouraged to keep personal notes on anything they considered necessary, ranging from usability and clarity of instruction to the educational value and technical aspects of each scenario. These notes later served as valuable anchors for the feedback collection phase. Rather than administering structured questionnaires, we opted for a semi-structured interview format. The interviews began with a set of broad questions addressing general impressions, but the conversation then flowed naturally based on each expert's observations and notes. This allowed us to delve deeper into each expert's unique insights and uncover scenario-specific recommendations. The result was a rich pool of qualitative data that guided the next development sprint and helped us identify both design strengths and pain points early in the refinement process.

4.2 Partner Evaluation Session (January 2025)

The second phase of testing occurred in January 2025 and involved all partner institutions during the 11th online project meeting. This testing session was conducted entirely online and was carefully organised to ensure a structured yet collaborative experience. To facilitate effective participation, all partners were divided into two distinct virtual rooms within the GUESS platform. Each room was led by one expert from the University of Patras, who had previously been involved in the development and internal evaluation of the scenarios.

Before gameplay began, both UPatras experts provided a concise briefing on the basic functionalities of the virtual world, including avatar movement, interaction mechanics, and instructions on entering and navigating through the scenarios. Following the briefing, partners were invited to test each scenario, taking active notes independently. This allowed participants to form their impressions without guided narration, ensuring more authentic feedback. Throughout the session, the UPatras experts remained in their assigned virtual rooms, available to address any questions in real-time. This live support mechanism helped maintain engagement and minimise confusion. Once all partners completed the testing, each room hosted a collaborative discussion session. These wrap-up conversations were informal but constructive, focusing on how the virtual world could be improved and how the feedback should guide the development of the next iteration.

The feedback collected from this phase—summarised in the tables previously presented—was especially rich in constructive observations. Partners highlighted the intuitive nature of the scenarios and affirmed that the educational goals were met. However, they also suggested refinements such as improving instruction clarity, enhancing the interactivity of minigames like the tree planting and recycling tasks, and adapting scenario content to be more regionally inclusive and culturally relevant. This feedback served as a direct input to the next round of development, ensuring that the project reflected both pedagogical soundness and international usability.

Figure 50: Familiarization phase of partners during partner testing session

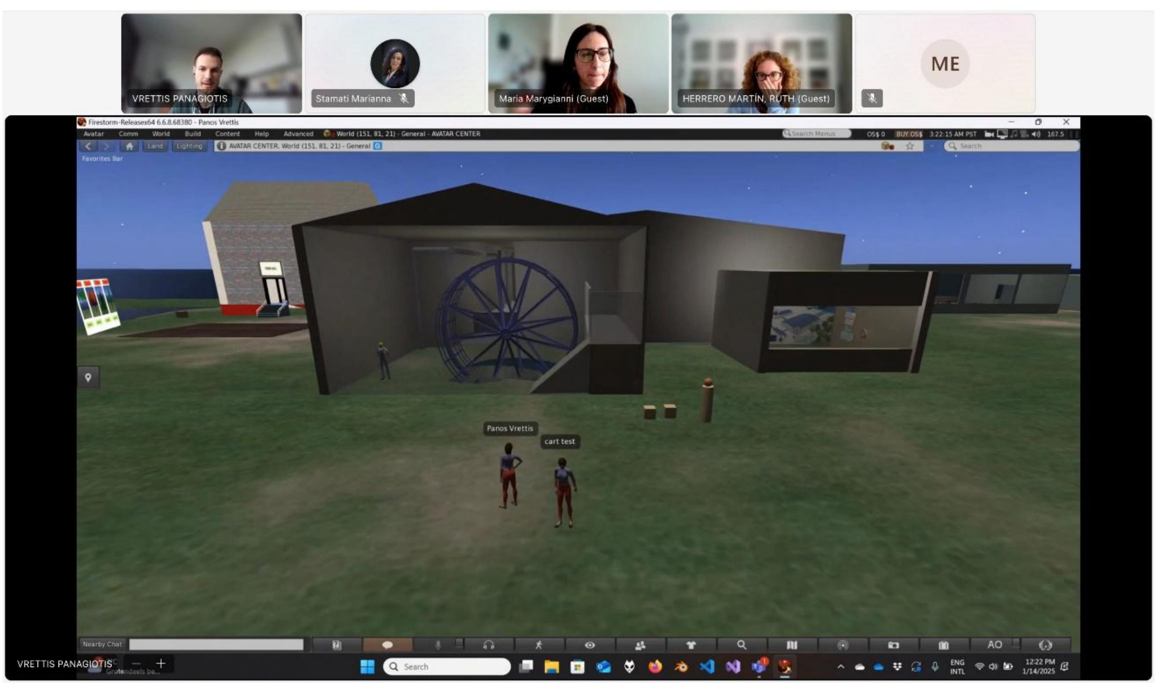

Figure 51: Process of partner testing session at the desalination factory scenario

4.3 Student Testing Session (May 2025-July 2025)

The final and most comprehensive testing session was conducted with actual university students, specifically targeting the intended end-users of the GUESS platform. The University of Patras organised this session in a controlled lab environment, where 35 students participated under the supervision of a trained facilitator. This physical presence enabled the facilitator to ensure a smooth operation, assist as needed, and monitor the students' overall experience without compromising their autonomy. The students were introduced to the virtual environment and were then asked to explore and complete the scenarios independently.

Upon finishing their in-world experience, each student completed a structured evaluation questionnaire. The questionnaire was designed to capture a wide range of feedback, including the user-friendliness of the interface, clarity of instructions, enjoyment, visual design, and most importantly, whether the educational objectives were achieved. The results of the questionnaire revealed that a significant majority of the students enjoyed the experience, found the scenarios easy to navigate, and felt they had learned something valuable about environmental topics. Many students reported that the immersive format helped them better understand real-world environmental issues by experiencing them in a simulated setting. Features such as the tree-planting minigame, scenario-specific missions, and visual storytelling elements were consistently highlighted as practical tools for engaging learners.

Students also provided feedback on areas of potential improvement. A few mentioned that the interface could be more responsive and that some textual elements could be better positioned or broken down into more readable segments. However, the tone of the feedback remained overwhelmingly positive, and many students expressed interest in exploring similar educational platforms in the future. The student testing session provided key validation of the platform's accessibility and impact on actual learners. It demonstrated that the GUESS virtual environment could not only deliver its intended content but also foster curiosity and motivation among university students.

Finally, from the user_statistics table, we were also able to extract detailed data on the session times for each student in each scenario. This quantitative insight enabled us to assess the average time spent per scenario, identify which parts required more time or led to delays, and gain a better understanding of user behavior across the different components of the platform.

The related data were retrieved from the user_scenarios table in the database, which stores user interaction metrics for each scenario. This table includes the following fields: an auto-incrementing id serving as the primary key, a unique user_id and user_name identifying each participant, a scenario_number indicating which virtual scenario the data correspond to, and two timestamps (timestamp_entered and timestamp_completed) marking when each user entered and completed a given scenario. The comment field allows for optional notes or structured JSON data that can be used for storing additional contextual information, such as performance metrics or task-specific feedback. This table structure ensures precise tracking of user progress and enables the extraction of detailed behavioural analytics.

To calculate the average duration spent in each scenario, the following SQL command was executed on the MariaDB database:

```
SELECT
   scenario_number,
   SEC_TO_TIME(AVG(TIME_TO_SEC(TIMEDIFF(timestamp_completed,     timestamp_entered))))
AS avg_time_spent
FROM
   user_scenarios
WHERE
   scenario_number BETWEEN 1 AND 4
   AND timestamp_entered BETWEEN '2025-05-01' AND '2025-07-31'
   AND timestamp_completed IS NOT NULL
GROUP BY
   scenario_number
ORDER BY
   scenario_number;
```

The analysis of the session duration data revealed that the average time students spent per scenario ranged between approximately three and eight minutes, indicating consistent engagement throughout the learning experience. As show in , the average time spent in Scenario 1 was 3 minutes and 48 seconds, Scenario 2 3 minutes and 55 seconds, Scenario 3 required the longest interaction with an average of 7 minutes and 54 seconds, while Scenario 4 was completed in about 3 minutes and 17 seconds. These results suggest that while most scenarios were completed in a relatively uniform time frame, Scenario 3 stood out as being significantly more demanding in terms of user interaction and task completion time.

This extended duration in Scenario 3 likely reflects its higher complexity or greater number of interactive elements compared to the others. It may also indicate that students required additional time to understand the objectives or perform the assigned activities within that scenario. In contrast, the shorter durations in Scenarios 1, 2 and 4 suggest a smoother flow and possibly more intuitive design, where users could progress with minimal confusion or interruption. The consistency in average times across these three scenarios demonstrates that the GUESS platform maintained a balanced level of engagement without overwhelming participants.

Overall, the time-distribution data complements the qualitative feedback gathered from the questionnaires, confirming that students were actively engaged and that the scenarios were appropriately challenging without being excessively time-consuming. These findings help validate the platform's educational pacing and usability while highlighting Scenario 3 as a potential area for refinement in future iterations. Adjustments such as clearer instructions, improved navigation cues, or minor simplifications could further optimize the learning flow and ensure a more uniform experience across all scenarios.

Table 1: Average playing time per scenario

Scenario number	Average Time Spent
1	3 minutes and 48 seconds
2	3 minutes and 55 seconds
3	7 minutes and 54 seconds
4	3 minutes and 17 seconds

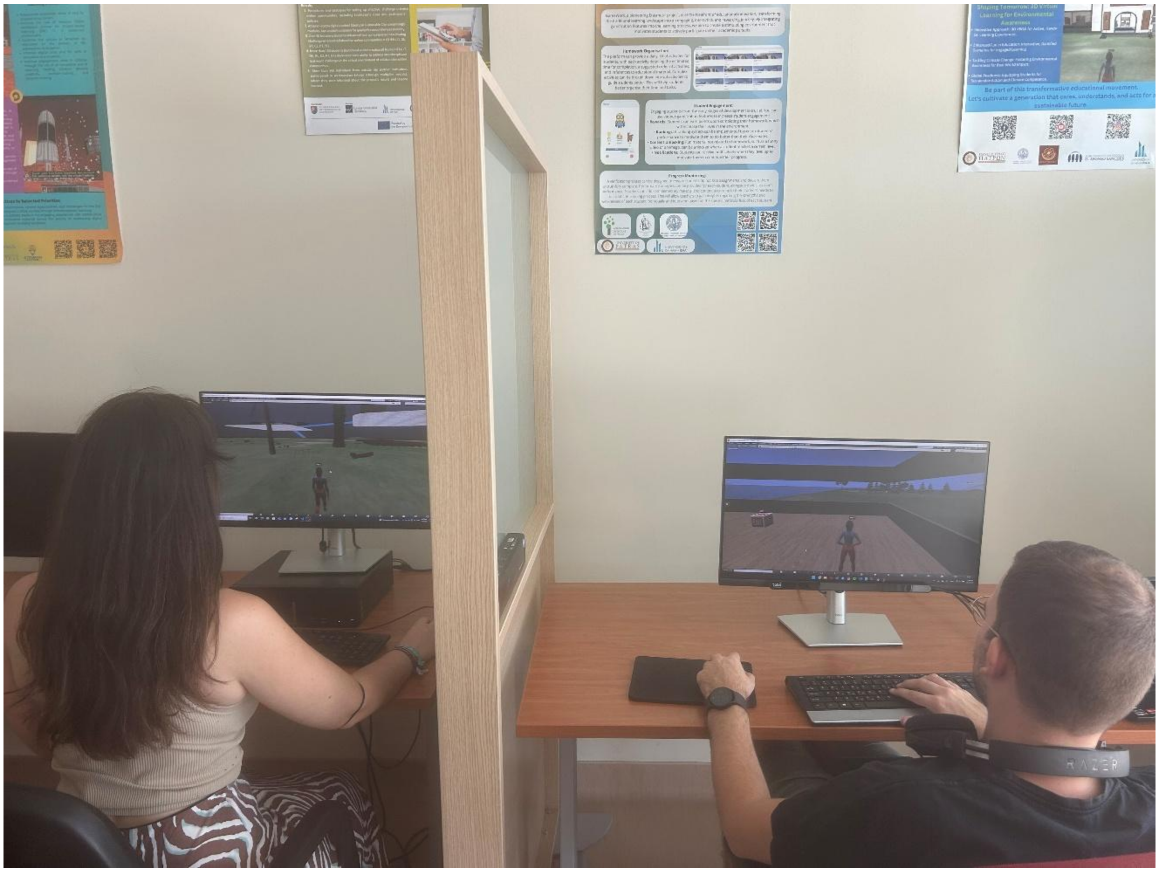

Figure 52: University of Patras testing phase

4.4 Version Changes

The final version of the GUESS 3D Virtual World Learning Environment (VWLE) incorporated several improvements following the feedback gathered from educational game experts, project partners, and over **120** university students during the structured testing phases. These changes aimed to improve scenario flow, enhance educational clarity, and optimise the overall user experience. The most significant changes introduced in the final version include:

1. Scenario refinement based on structured testing feedback. All four scenarios were re-evaluated and adjusted to improve clarity, flow, and alignment with the educational goals. Particular focus was placed on:
    - Improving the guidance and navigation system by adding subtle visual cues (e.g. directional arrows, animated helpers, or messages) that help users understand what to do next.

- Adjusting question and task design to make them more accessible without reducing educational depth. Optional prompts and tiered hints were added for more flexibility.
- Integrating more branching dialogue and informative hotspots, especially in tree planting and recycling minigames, to reduce repetitive content and keep engagement, the guidance and navigation system by adding subtle visual cues (e.g. directional arrows, animated helpers, or messages) that help users understand what to do next.

2. User interface enhancements and better onboarding support:

    - A startup orientation zone was introduced where players can practice movement and interactions before entering actual scenarios.
    - All NPCs (non-player characters) were revised to have more distinctive outfits and gestures, making interactions more intuitive and character roles easier to identify.

3. Visual and environmental updates to enhance immersion and readability:

    - Objects that were unnecessarily blocking avatar movement (such as bins, and fences) were made phantom, allowing smoother navigation for beginner users.
    - Key interactive items, like trees, bins, doors and information panels, were given animated glows or highlights to make them easier to spot.
    - Several scenarios had their lighting and texture colors adjusted based on student feedback that the original palette felt too heavy. A more vibrant and natural color scheme was applied to improve the atmosphere.
    - All NPCs (non-player characters) were revised to have more distinctive outfits and gestures, making interactions more intuitive and character roles easier to identify.

4. Scenario accessibility improvement and flow control:

    - An "Exit" cube was added to every scenario, allowing students to leave at any point and return to the central area—especially useful for those who entered by mistake or wanted to reset the experience.
    - Scenarios now feature **progress checkpoints** or reminders if players remain idle for a long time, minimising unintended session timeouts or confusion during questionnaire completion.

- Scenario descriptions and “Play” buttons were added directly outside each scenario's entrance, clearly explaining the topic and goals of each learning experience.

5. Expanded language support and partner localization:

    - The entire platform was translated into all three project languages: English, Greek, Portuguese, Spanish and Czech
    - A language selection area was added at the main hub, where users choose their preferred language by clicking on the corresponding national flag before entering the experience.
    - Country-specific references were revised to adopt a more universal framing, allowing students from all partner countries to relate to the content.

6. Central hub redesign and added social learning elements:

    - A new central area was introduced between scenarios, featuring lakes, plants, and benches, acting as a relaxed environment for exploration or group discussion.
    - More student-centric design choices were applied across the experience, including motivational banners, custom teleport areas, and post-scenario feedback spaces.

The final version of the GUESS 3D VWLE was finalised following the full round of testing and was officially released in October 2025. The following sections present the updated versions of the scenarios, minigames, and supporting spaces that together form the GUESS learning environment.

4.5 Student Testing Scenario (October 2025)

Following the main testing phase, a final validation session was carried out during October 2025, focusing on confirming platform readiness and ensuring that the GUESS virtual environment functioned reliably in its near-final state. A total of 126 university students participated across all partner institutions, providing final impressions on interaction flow, learning clarity, and overall usability of the environment. The testing followed a similar structure to previous sessions, with students entering the platform, completing the environmental scenarios independently, and submitting structured feedback through a post-use questionnaire.

The results indicated that users found the experience smooth, stable, and well-paced, demonstrating that the improvements introduced after the beta phase successfully enhanced clarity and navigation without altering the core structure of the scenarios. Students again highlighted the environmental simulations as enjoyable and educational, while several noted that scenario transitions felt cleaner and easier to follow compared to earlier versions. Comments such as *"the world felt more polished now,"* and *"the interaction was natural and easy to continue through,"* reflected a general sense of functional maturity. Visual guidance enhancements—though subtle—were recognised as helpful by a number of participants, particularly in tasks requiring step-by-step decision-making.

Finally, from the user_statistics table, we were also able to extract detailed data on the session times for each student in each scenario. This quantitative insight enabled us to assess the average time spent per scenario, identify which parts required more time or led to delays, and gain a better understanding of user behavior across the different components of the platform.

The related data were retrieved from the user_scenarios table in the database, which stores user interaction metrics for each scenario. This table includes the following fields: an auto-incrementing id serving as the primary key, a unique user_id and user_name identifying each participant, a scenario_number indicating which virtual scenario the data correspond to, and two timestamps (timestamp_entered and timestamp_completed) marking when each user entered and completed a given scenario. The comment field allows for optional notes or structured JSON data that can be used for storing additional contextual information, such as performance metrics or task-specific feedback. This table structure ensures precise tracking of user progress and enables the extraction of detailed behavioural analytics.

To calculate the average duration spent in each scenario, the following SQL command was executed on the MariaDB database:

```
SELECT
  scenario_number,
  SEC_TO_TIME(AVG(TIME_TO_SEC(TIMEDIFF(timestamp_completed, timestamp_entered)))) AS avg_time_spent
FROM
  user_scenarios
WHERE
  scenario_number BETWEEN 1 AND 9
  AND timestamp_entered BETWEEN '2025-10-01' AND '2025-10-31'
```

```
   AND timestamp_completed IS NOT NULL
GROUP BY
   scenario_number
ORDER BY
   scenario_number;
```

Table 2: User statistics during final testing

Scenario number	Average Time Spent
1	4 minutes and 23 seconds
2	5 minutes and 15 seconds
3	8 minutes and 44 seconds
4	4 minutes and 34 seconds
5	5 minutes and 26 seconds
6	6 minutes and 37 seconds
7	3 minutes and 28 seconds
8	3 minutes and 39 seconds
9	6 minutes and 3 seconds

4.6 Version Changes

The October 2025 release of the GUESS 3D Virtual World Learning Environment included a small number of refinements inspired by the last student testing session. The changes introduced in the final version include:

1. Improved interaction visibility and task clarity:

- Key interactive objects were given slightly stronger visual markers, ensuring that students could identify actionable items more quickly during each scenario.
- Subtle highlights and soft glow effects were applied to doors, plants, and in-scenario prompts to help users recognize their next objective without breaking immersion.

2. Light onboarding text simplification:
    - The introductory instructions at the starting hub were shortened and made more concise, allowing new users to begin the experience faster without extended reading time.
    - Non-essential text was reduced and structured into clearer, shorter segments, improving first-time comprehension and overall flow.

5 Adding Language

The GUESS virtual world learning environment is designed to support localisation into multiple languages. Since dialogue forms the foundation of every scenario, all textual content must be translated into the target audience's language. At present, the scenarios are available in four languages, English, Portuguese, Greek, Czech and Spanish.

Upon entering the environment, players are taken to a language selection room, where they can choose their preferred language before proceeding further. Each language corresponds to a numeric identifier from 1 to 5.

To add a new language, you must first modify the language selection room so players can select the new option. This selection should trigger a request to the database that sets the player's language index (e.g., 6 for a sixth language). Next, translate all scenario content, primarily dialogue, into the new language and include it in the relevant script files.

Each piece of dialogue is stored in a list where the index position corresponds to the language index minus one. Therefore, to add a new translation, simply append your text to these lists.

For notecards and images, the translated versions should be placed in the content tab of the corresponding in-world object. Their filenames must end with the appropriate language index (for example, notecard_forest4). The current language index is then appended to the script line that delivers the notecard or image to the player.

Finally, ensure that all interactive buttons are also translated.

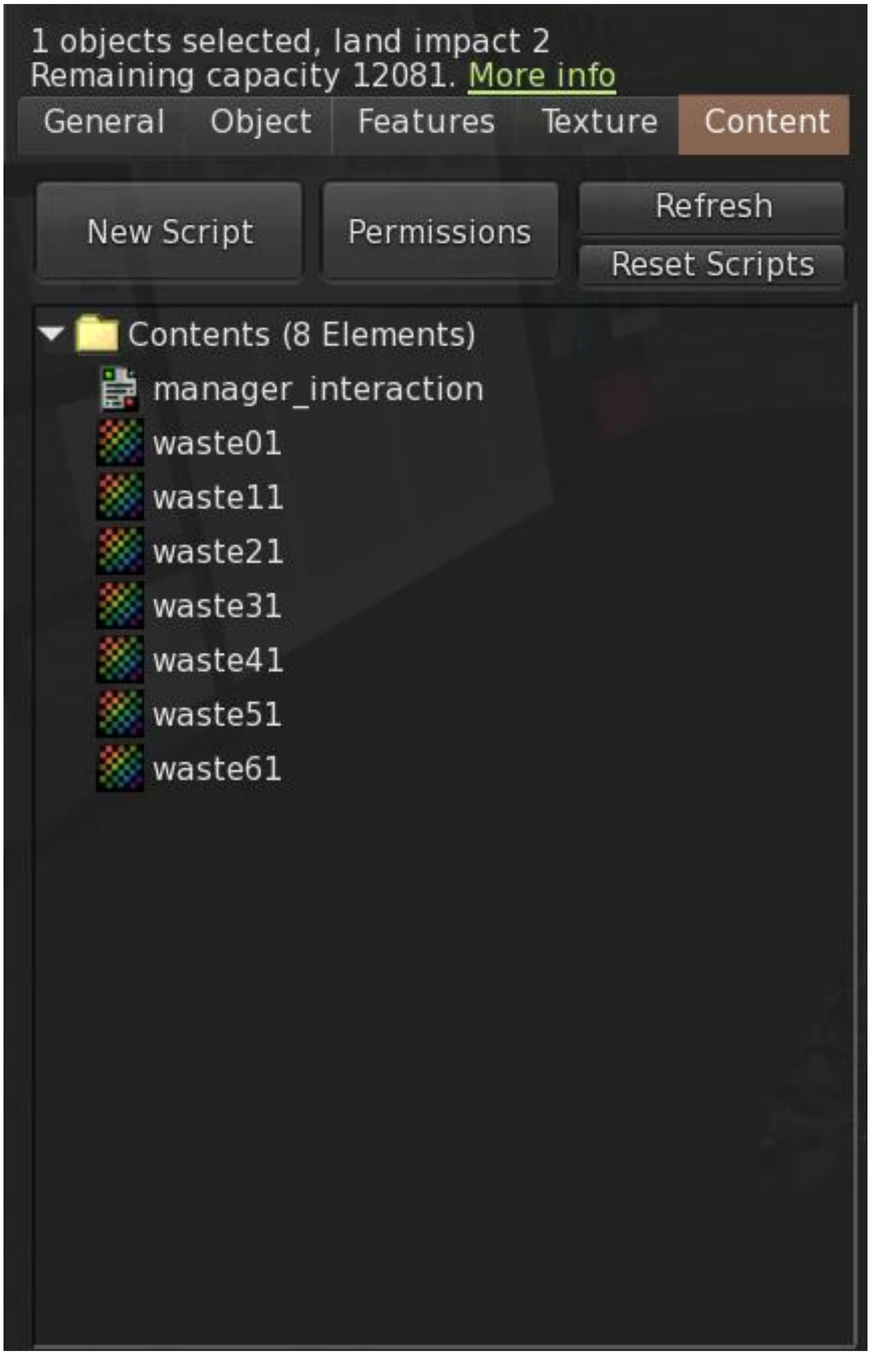

Figure 53: Images in a localised format

5.1 Languages

OpenSimulator runs on a MySQL database and has been customised to meet the specific needs of the GUESS web server. This server uses PHP scripts to manage three key variables — lang, room, and role — which are passed through an initiator script. These variables are used to store and update each user's language preference, role, and scenario progress.

They are defined as follows:

- Lang: user's language preference
- Room: used to store the user progress within each scenario

```
<?php
$servername = "mariadb";
$username = "root";
$password = "mariadb";
$dbname = "divaopensim";

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

$sqllang = "ALTER TABLE UserAccounts ADD COLUMN IF NOT EXISTS lang TINYINT(4) NOT NULL DEFAULT 0";
$sqlroom = "ALTER TABLE UserAccounts ADD COLUMN IF NOT EXISTS room INT(11) NOT NULL DEFAULT 0";
$sqlrole = "ALTER TABLE UserAccounts ADD COLUMN IF NOT EXISTS role_id TINYINT(4) NOT NULL DEFAULT 0";
$sql1 = "ALTER TABLE UserAccounts ADD Trophies VARCHAR(30) NOT NULL DEFAULT '0:0:0:0:0:0:0:0:0:0:0'";
// $sql_selected_item = "ALTER TABLE prims ADD COLUMN IF NOT EXISTS end_House_explorer TINYINT(4) NOT NULL DEFAULT 0";

if (mysqli_query($conn, $sqllang)) {
  echo "Table UserAccounts created successfully the column lang";
  } else {
  echo "Error creating table: " . mysqli_error($conn);
  }

if (mysqli_query($conn, $sqlroom)) {
  echo "Table UserAccounts created successfully the column room";
```

```
    } else {
    echo "Error creating table: " . mysqli_error($conn);
    }

  if (mysqli_query($conn, $sqlrole)) {
    echo "Table UserAccounts created successfully the column role_id";
    } else {
    echo "Error creating table: " . mysqli_error($conn);
    }

  if (mysqli_query($conn, $sql1)) {
    echo "Field Trophies added successfully";
    } else {
    echo "Error creating table: " . mysqli_error($conn);
    }
  // if (mysqli_query($conn, $sql_selected_item)) {
  //   echo "Table prims created successfully the column
  selected_item_flag";
  // } else {
  //   echo "Error creating table: " .mysqli_error($conn);
  // }

  mysqli_close($conn);

  ?>
```

In short, the osInitiate.php script updates and extends the default OpenSim UserAccounts table to include new attributes. Once the structure is set, the osLang.php script manages the retrieval and modification of each user's language preference.

While osInitiate.php prepares the database by adding the lang column, osLang.php handles user-specific updates. It processes requests containing the avatar's ID and chosen language via a URL, then updates the corresponding database record or retrieves the saved preference. This mechanism ensures that each user's language settings are dynamically maintained throughout their interactions in the GUESS environment.

```
Database changed
MariaDB [divaopensim]> SHOW TABLES;
+--------------------------+
| Tables_in_divaopensim    |
+--------------------------+
| Avatars                  |
| Friends                  |
| GridUser                 |
| Presence                 |
| UserAccounts             |
| assets                   |
| auth                     |
| bakedterrain             |
| classifieds              |
| estate_groups            |
| estate_managers          |
| estate_map               |
| estate_settings          |
| estate_users             |
| estateban                |
| hg_traveling_data        |
| im_offline               |
| inventoryfolders         |
| inventoryitems           |
| land                     |
| landaccesslist           |
| migrations               |
| os_groups_groups         |
| os_groups_invites        |
| os_groups_membership     |
| os_groups_notices        |
| os_groups_principals     |
| os_groups_rolemembership |
| os_groups_roles          |
| primitems                |
| prims                    |
| primshapes               |
| regionban                |
| regionenvironment        |
| regionextra              |
| regions                  |
| regionsettings           |
| regionwindlight          |
| spawn_points             |
| terrain                  |
| tokens                   |
| user_scenarios           |
| userdata                 |
| usernotes                |
| userpicks                |
| userprofile              |
| usersettings             |
+--------------------------+
47 rows in set (0.00 sec)
```

Figure 54: MySQL tables of divaopenm database

```
MariaDB [divaopensim]> DESCRIBE UserAccounts;
+-------------------+--------------+------+-----+-----------------------+-------+
| Field             | Type         | Null | Key | Default               | Extra |
+-------------------+--------------+------+-----+-----------------------+-------+
| PrincipalID       | char(36)     | NO   | PRI | NULL                  |       |
| ScopeID           | char(36)     | NO   |     | NULL                  |       |
| FirstName         | varchar(64)  | NO   | MUL | NULL                  |       |
| LastName          | varchar(64)  | NO   | MUL | NULL                  |       |
| Email             | varchar(64)  | YES  | MUL | NULL                  |       |
| ServiceURLs       | text         | YES  |     | NULL                  |       |
| Created           | int(11)      | YES  |     | NULL                  |       |
| UserLevel         | int(11)      | NO   |     | 0                     |       |
| UserFlags         | int(11)      | NO   |     | 0                     |       |
| UserTitle         | varchar(64)  | NO   |     |                       |       |
| active            | int(11)      | NO   |     | 1                     |       |
| FinishedScenarios | varchar(20)  | NO   |     | 0:0:0:0:0:0:0:0:0:0   |       |
| lang              | tinyint(4)   | NO   |     | 0                     |       |
| room              | int(11)      | NO   |     | 0                     |       |
| Trophies          | varchar(30)  | NO   |     | 0:0:0:0:0:0:0:0:0:0:0 |       |
+-------------------+--------------+------+-----+-----------------------+-------+
15 rows in set (0.00 sec)
```

Figure 55: Attributes of updated UserAccounts table

Below is an example of the PHP script (osLang.php) that returns the user language or sets the user language:

```php
<?php
$servername = "mariadb";
$username = "root";
$password = "mariadb";
$dbname = "divaopensim";

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

//stores url properties in variables
$idAvatar =  $_GET["idAvatar"];
$lang = $_GET["lang"];

//$room = $_GET["room"];
```

```
// if id and lang are set find id
if(isset($idAvatar ) && isset($lang))
{
   // find id in db
   $myQuery = "SELECT * FROM UserAccounts WHERE
`PrincipalID`='$idAvatar'";
   $result = mysqli_query($conn,$myQuery) or die('Error searching the
avatar');

   if (mysqli_num_rows($result) == 1)
     {
   // Puts, updates the lang in the table
   $myQuery ="UPDATE UserAccounts SET lang ='$lang' WHERE PrincipalID
='$idAvatar'";
     if ($conn->query($myQuery) === false)
     {
     echo "Error updating record: " . $conn->error;
     } else if($conn->query($myQuery)=== true &&  $lang === "01"){
           echo "You have successfully chosen English as your language
Click!";
         } else if($conn->query($myQuery)=== true &&  $lang === "02"){
           echo "Έχετε επιλέξει τα Ελληνικά ως την γλώσσα επιλογή σας
Πατήστε!";
         } else if($conn->query($myQuery)=== true &&  $lang === "03"){
          echo "Avete scelto con successo l'italiano come lingua
          Cliccare su";
         } else if($conn->query($myQuery)=== true &&  $lang === "04"){
           echo "Você escolheu com sucesso o português como seu idioma.
           Clique";
         }
     }

  /*
  if (mysqli_num_rows($result) == 0)
  {

  // Inserts id and lang in the table
  $myQuery = "INSERT INTO useraccounts (PrincipalID, lang)
  VALUES ('$idAvatar', '$lang')";
  if (mysqli_query($conn, $myQuery))
  {
```

```
  }

  else
  {
    echo "Error: " . $myQuery . "<br>" . mysqli_error($conn);
  }
  }
  */

}

// Returns the lang value stored in the table
if(isset($idAvatar ) && !isset($lang))
  {
      $myQuery = "SELECT * FROM UserAccounts WHERE
`PrincipalID`='$idAvatar'";
      $result = mysqli_query($conn,$myQuery) or die('Error getting the
language for  this avatar');
      $row = mysqli_fetch_assoc($result);
      echo $row["lang"];
    }

mysqli_close($conn);
?>
```

Similarly, different script save the user's progress and retrieve mechanisms.

6. Adding Scenarios

The GUESS Virtual World Learning Environment utilizes a framework specifically designed to facilitate the integration of new scenarios, if such a need were to arise. This establishes the scalability and extendibility of our project, allowing it to be improved and built upon for future use cases.

The addition of new scenarios can be achieved without the need for any special changes to the backend of the environment (i.e., the database), allowing for parties with less technical/programming skills to develop their own scenarios. With our use of the OpenSimulator platform, we allow the quick creation of 3D objects inside the Virtual World, or the upload of models created using external 3D modelling software in the form of .dae files. Scripting the functionalities necessary for the realization of the scenario is made possible using the .lsl scripting language used by the OpenSimulator platform. Our scenario scripts are engineered to be easy to understand and extend, as well as transfer into new scenarios. Their logic is simple to follow and can be translated to work for any concept a new scenario follows.

The list of items that can be modified or transferred from existing scenarios include:

- Static objects
- Interactive objects
- Non-Player Characters
- Animations
- Sounds
- Educational material

The OpenSimulator platform provided all the necessary animations for most of the common NPC actions (e.g., walking, waving hello, speaking, etc.). Sounds can be easily uploaded and played from any object using a simple script. Animations work the same way. NPCs can be customized by the designer. To do this, the user customizes their own character with the appearance they desire (e.g., avatar customization and clothing worn) and they can use the ***create_npc.lsl*** scripts to name and create the NPC. Educational material can be provided in the form of posters (given to the player as 2D textures), notecards (editable text), or -more organically- in the form of dialogue during conversations with NPCs. The latter requires some scripting support. In the subsections that follow, we provide an overview of how the scripting should be structured for a scenario to function.

6.1 Scripting support

Our ***dialog.lsl*** scripts constitute concrete examples of how the OpenSimulator platform can be used with the GUESS framework to realize functioning and interactive scenarios. All scenarios states are represented with an integer value stored in the **room** variable. Scenarios start with a different value (0 to 9 in our case) and are incremented depending on player interactions. We follow a design where the various state for a scenario with initial state *st* is incremented in the form of *st * 100* (e.g., for 2, we store states 200, 201, 202, etc.). These are particularly important when reacting to player choices in dialogue or otherwise.

6.1.1 Playing animations

To play an existing animation, we need the name of the animation (can be found here) and the target NPC that should play that animation. The **PlayAnimation** function (Figure 56) does exactly that.

```
playAnimation(string animation, key npc)
{
    osNpcStopAnimation(npc, currentAnimation);
    currentAnimation = animation;
    osNpcPlayAnimation(npc, animation);
}
```

Figure 56: The animation wrapper.

6.1.2 Playing sounds

The internal function **llPlaySound** takes care of sound playing, using the name of the sound file and the volume at which the sound is to be played (values of 0.0 to 1.0). The sound file needs to be placed in the contents of the object containing the relevant script.

6.1.3 Giving images to a player

The internal function **llGiveInventory** can provide a user with an inventory item from the contents of the object containing the script. In the GUESS project, we use this function to give the player posters as 2D textures containing educational material relevant to the scenario.

6.1.4 Talking to NPCs

We use two special wrappers of the internal function **llDialog** to implement player-to-NPC conversations in GUESS. The wrappers are **displayDialog** and **displayDialogWOListen** (Figure 59), and take as input the line of dialogue (in the form of a list containing the line in each language), the names of the buttons that should be displayed for the player to respond, and the key of the NPC that the line is associated with. Both wrappers reset the dialog listener that looks for responses from the player. The first wrapper then creates a new listener that waits for a response from the player, while the second wrapper does not, as it does not require an answer from the player. Then, both wrappers display the dialog box to the player for the conversation to take place. A talking animation is also played. Finally, the first wrapper sets a timer event for the expectation of a response from the player. The state of the scenario is used to track the conversations the player makes with NPCs and their progress.

On the top of the script, we define (hardcoded) all the lines of dialogue (in all languages) that are used in the conversation (Figure 57). They are represented as lists per line, where the elements of the list contain the different languages.

```
list gld_jack_dialog_5_b = ["Panam Alzamar:
-    Convincing consumers to buy clothes of better quality that last longer (slow fashion)
-    Generally steering consumer behavior towards more sustainable options

Your answer:
1. Can you be a bit more specific?", "Panam Alzamar:
•    Convencer os consumidores a comprar roupa de melhor qualidade e mais duradoura (moda lenta)
•    Orientar de forma geral o comportamento do consumidor para opções mais sustentáveis

A tua resposta:
1. Pode ser um pouco mais específico?", "Panam Alzamar:
• Convencer a los consumidores de comprar ropa de mejor calidad que dure más (moda lenta)
• En general, orientar el comportamiento del consumidor hacia opciones más sostenibles

Tu respuesta:
1. ¿Puedes ser un poco más específico?", "Panam Alzamar:
• Πείθοντας τους καταναλωτές να αγοράζουν ρούχα καλύτερης ποιότητας που διαρκούν περισσότερο (αργή μόδα)
• Γενικά, κατεύθυνση της συμπεριφοράς των καταναλωτών προς πιο αειφόρες επιλογές

Η απάντησή σας:
1. Μπορείτε να γίνετε λίγο πιο συγκεκριμένος;", "Panam Alzamar:
• Přesvědčování spotřebitelů, aby si kupovali oblečení lepší kvality, které déle vydrží (slow fashion)
• Obecné směřování chování spotřebitelů k udržitelnějším možnostem

Tvoje odpověď:
1. Můžete být trochu konkrétnější?"];

list gld_jack_dialog_6 = ["Panam Alzamar: Absolutely! Let's start with the definition of circular economy.

Your answer:
1. Continue.", "Panam Alzamar: Com certeza! Vamos começar com a definição de economia circular.
```

Figure 57: Example of dialogue lists.

Player responses are tracked in the **listen** function of each script (Figure 58). Here the state of the scenario is updated depending on player choices.

```
// Initial Choice
if (room == 9)
{
    if (message == "1")
        setRoom(902);
    else
        setRoom(901);
}
else if (room == 923)
{
    if (circularEconomyCorrect == 2)
        setRoom(room+1);
    else
        setRoom(room+2);
}
else if (room == 928)
{
    if (textileCorrect == 2)
        setRoom(room+1);
    else
        setRoom(room+2);
}
```

Figure 58: Example of listening for a player response and updating the scenario state accordingly.

Two important technical limitations to keep in mind are that only one dialog box can be open at any time for a single player and that the contents of the dialog box can be a string of a maximum of 512 bytes. This should inform the writing stage of a scenario, as it hinders how long a piece of conversation is. It is likely that a single piece of conversation will have to be split into smaller parts.

```
displayDialog(list dialogList, list buttons, key npc)
{
    llListenRemove(gListener);
    gListener = llListen(dialogChannel, "", user, "");
    llDialog(user, llList2String(dialogList, Lang - 1), buttons, dialogChannel);
    playAnimation("talk", npc);
    llSetTimerEvent(120.0);
}

displayDialogWOListen(list dialogList, list buttons, key npc)
{
    llListenRemove(gListener);
    llDialog(user, llList2String(dialogList,Lang-1) , buttons, dialogChannel);
    playAnimation("talk", npc);
    llSetTimerEvent(0.0);
}
```

Figure 59: The player-to-NPC dialogue wrappers.

6.1.5 Scenario sensors

Each scenario uses sensor functionality to ensure that only one player is inside each scenario at any time. The internal sensor capabilities are leveraged for this purpose.

6.1.6 Scenario component communication

To allow for the various objects of a scenario to communicate and update each other, we use a simple messaging framework. Each scenario has a unique internal listener that tracks specially formatted messages from other components in that scenario and updates script variables accordingly. Internal functions like **llShout** are used for this.

6.2 Game Mechanics Guidelines

In the early stages of the project, the partners agreed that to fully realise the potential of the 3D VWLE, it was essential to create documentation outlining the capabilities of the GUESS 3D VWLE. This documentation would serve as a guideline for incorporating these features into the scenarios. To achieve this, Upatras, with input from all partners, developed a game mechanics guideline, which has been incorporated into this deliverable in its final form. This guideline is intended to assist anyone looking to use, modify, or create new scenarios that incorporate the game mechanics developed during the GUESS project. It includes information on:

- User interaction with the environment
- Game mechanics that can be integrated into scenarios
- Technical limitations

- Unique features developed specifically for the GUESS project