Mistakes were made; Design Patterns for Error Feedback in Virtual Reality Training Simulations in a no-code authoring tool

Achilleas Filippidis

Thesis submitted in partial fulfillment of the requirements for the

Masters' of Science degree in Computer Science and Engineering

University of Crete
School of Sciences and Engineering
Computer Science Department
Voutes University Campus, 700 13 Heraklion, Crete, Greece

Thesis Advisor: Professor *George Papagiannakis*

This work has been performed at the University of Crete, School of Sciences and Engineering, Computer Science Department.

The work has been supported by the Foundation for Research and Technology - Hellas (FORTH), Institute of Computer Science (ICS).

University of Crete
Computer Science Department

Mistakes were made; Design Patterns for Error Feedback in Virtual Reality Training Simulations in a no-code authoring tool

Thesis submitted by
Achilleas Filippidis
in partial fulfillment of the requirements for the
Masters' of Science degree in Computer Science

THESIS APPROVAL

Author: ______________________________
Achilleas Filippidis

Committee approvals: ______________________________
George Papagiannakis
Professor, Thesis Supervisor

____
Constantine Stefanidis
Professor, Committee Member

____
Dimitris Grammenos
Principal Researcher, Committee Member

Departmental approval: ______________________________
Polyvios Pratikakis
Professor, Director of Graduate Studies

Heraklion, July 2024

Mistakes were made; Design Patterns for Error Feedback in Virtual Reality Training Simulations in a no-code authoring tool

Abstract

According to the latest neuroscientific advances, four human mechanisms are most prominent in influencing the brain's learning ability: attention, active engagement, error feedback, and consolidation. In this context, error feedback is a crucial mechanism that impacts efficiently the learning process. Today, although error feedback is utilized in many Virtual Reality (VR) training simulations, its implementation does not follow any standard workflow. VR training simulations are customized on an ad hoc basis according to specific requirements. This lack of a standardized framework or methodology does not allow the consistent exploitation of error feedback across diverse VR training simulations.

In this thesis, we present error feedback algorithms specially formulated as VR software design patterns. Aiming to provide accurate, intuitive, and immersive feedback to the user, the proposed five VR software design patterns are based on at least one of the three in-VR human perceptions: haptic, visual, and auditory. Additionally, this thesis provides a methodology for capturing important sensory user data at the time of the incorrect interaction, such as eye-gaze fixation, user position within the virtual world, and the particular interactable object. In this respect, the author of the VR training simulation is empowered with a novel, no-code UI to set up, configure, and visualize the aforementioned user data. Lastly, we provide a novel error-feedback mind-map visualization UI that summarizes all user errors during training, the user's fixation points on interactable objects, the user's position when triggering an error, and all interaction data related to each interactive object.

Mistakes were made: Μοτίβα σχεδιασμού για ανατροφοδότηση σφαλμάτων σε εκπαιδευτικές προσομοιώσεις εικονικής πραγματικότητας με την χρήση συγγραφικού εργαλείου χωρίς την χρήση κώδικα

ΠΕΡΙΛΗΨΗ

Σύμφωνα με τις τελευταίες εξελίξεις στην νευροεπιστήμη, οι τέσσερις μηχανισμοί που επιδρούν σημαντικά στην ικανότητα μάθησης του εγκεφάλου είναι η προσοχή, η ενεργή συμμετοχή, η ανατροφοδότηση των σφαλμάτων και η εμπέδωση. Κατά συνέπεια, η ανατροφοδότηση των σφαλμάτων στους εκπαιδευόμενους είναι ένας κρίσιμος μηχανισμός που ενισχύει την αποτελεσματικότητα στη μάθηση. Σε εκπαιδευτικές προσομοιώσεις εικονικής πραγματικότητας η ανατροφοδότηση σφαλμάτων συνηθως υλοποιείται κατά περίπτωση και χωρίς προκαθορισμένους κανόνες. Ως εκ τούτου, κάθε υλοποίηση είναι προσαρμοσμένη στις εξειδικευμένες απαιτήσεις μιας συγκεκριμένης προσομοίωσης, χωρίς να υπάρχει ένα τυποποιημένο πλαίσιο ή μία μεθοδολογία, γεγονός που καθιστά την αποτελεσματικότητα και τη συνέπεια της ανατροφοδότησης σφάλματος να διαφοροποιείται ευρέως μεταξύ των διαφόρων προγραμμάτων εκπαίδευσης εικονικής πραγματικότητας.

Στα πλαίσια της παρούσας διατριβής αναπτύχθηκαν αλγόριθμοι ανατροφοδότησης σφαλμάτων σε μορφή σχεδιαστικών προτύπων λογισμικου εικονικής πραγματικότητας. Κάθε ένα από τα προτεινόμενα πέντε σχεδιαστικά πρότυπα λογισμικού βασίζονται σε τουλάχιστον μία από τις τρεις ενεργές ανθρώπινες αισθήσεις εντός του εικονικού κόσμου, την απτική, την οπτική και την ακουστική, με στόχο την απόδοση διαισθητικής και εμβυθιστικής ανατροφοδότησης σφαλμάτων με ακρίβεια. Επιπλέον, η διατριβή παρουσιάζει μια μεθοδολογία για την καταγραφή σημαντικών αισθητηριακών δεδομένων από τον χρήστη κατά την εσφαλμένη του αλληλεπίδραση, όπως η εστίαση του βλέμματος, η θέση του χρήστη στον εικονικό κόσμο, και το αντικείμενο με το οποίο αλληλεπίδρούσε την στιγμή του λανθασμένου χειρισμού.

Μέσω μιας καινοτόμου διεπαφής χρήστη, προσαρμοσμένη σε ένα σύστημα ανάπτυξης προσομοιώσεων εικονικής πραγματικότητας χωρίς κώδικα, ο δημιουργός του εικονικού σεναρίου εκπαίδευσης έχει τη δυνατότητα να ρυθμίζει και να οπτικοποιεί τα προαναφερθέντα δεδομένα του χρήστη. Σε αυτό το πλαίσιο, η παρούσα διατριβή παρέχει μια καινοτόμο μέθοδο οπτικοποίησης της ανατροφοδότησης των σφαλμάτων, βασισμένη σε νοητικούς χάρτες, στους οποίους συνοψίζονται όλοι οι εσφαλμένοι χειρισμοί του χρήστη κατά τη διάρκεια της εκτέλεσης της εκπαιδευτικής προσομοίωσης, τα σημεία εστίασης του βλέμματος του, τη θέση του κατά την πρόκληση ενός λανθασμένου χειρισμού, καθώς και τα σχετικά δεδομένα αλληλεπίδρασης με κάθε διαδραστικό αντικείμενο.

Acknowledgements

I am grateful to my supervisor, Professor George Papagiannakis, for his support throughout my master's degree and his continual encouragement to push myself to achieve better results.

Also, I extend my gratitude to my colleagues, John Petropoulos, Dimitris Aggelis, Nikos Marmaras, and Alexandra Plexousaki, for their unwavering support, encouragement, and collaborative brainstorming sessions. Their ideas and motivation have played a significant role in driving me to excel in my master's program.

Moreover, I would like to thank my family and my girlfriend, Dimitra Petrou, for believing in me all these years of my master's degree. Their constant support and encouragement were instrumental as I balanced my academic pursuits with my professional responsibilities, striving to excel in both realms.

I am deeply grateful to my friend, co-worker, and co-supervisor, Paul Zika, whose constant support, encouragement, and assistance were influential throughout this journey. His guidance, motivation, and invaluable help played a pivotal role in completing this master's thesis.

Lastly, I would like to extend my gratitude to my colleague, Dr. Antonis Protopsaltis, for his significant assistance and support.

There is no tomorrow...

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The human brain's capacity for learning is a multifaceted and intricate phenomenon that has captivated researchers across disciplines for centuries. At the core of this cognitive process lies the brain's remarkable ability to assimilate, process, and consolidate information gleaned from sensory inputs and experiential interactions with the environment. Central to the brain's learning machinery are several interrelated cognitive mechanisms that collectively underpin the acquisition and retention of knowledge. Dahaene *et al* 2020 [16] in his book "How we learn" states that there are four essential mechanisms, or "pillars," massively modulate our ability to learn. These are attention, active engagement, error feedback and consolidation.

VR has emerged as a transformative technology in the realm of education and training, offering immersive experiences that can enhance learning outcomes through realistic simulations. Many simulations have been developed over the past few years across various industries. Training simulations are available for activities such as piloting airplanes, enhancing soft skills, and learning medical skills [5]. The utilization of VR in such training environments leverages the brain's natural learning processes by providing engaging and interactive contexts in which learners can practice skills and make decisions. VR simulations are starting to add analytics [58] and other data collection tools [11] that will enhance the precision and scope of user behavior analysis, enabling more sophisticated assessments of learning outcomes and interaction patterns. This integration of advanced analytics facilitates a deeper understanding of user engagement and efficacy within the virtual environment, potentially leading to significant improvements in the design and implementation of VR-based educational and training programs. Most of the time, the aforementioned data are visualized at the end, during the debriefing session. But what happens when the user makes an error? How do we inform the user about an incorrect action they performed?

Right now, the implementations about error feedback to the user are ad hoc. The absence of a standardized approach is problematic, as it can result in inconsistent feedback and inadequately assist users in comprehending errors and identifying corrective actions. This is something we are going to address in this thesis.

1.1 Scope and Objectives

The main goal of this thesis is to tackle the ad hoc implementation of error feedback in VR training scenarios. Specifically, we will investigate the design patterns and strategies employed to provide informative and effective feedback to users in response to errors encountered during VR-based training experiences. In particular, we will develop five error feedback mechanisms (Chapter 3) in that leverage three out of the eight human perceptual modalities. Furthermore, we will design algorithms capable of collecting and visualizing eye-fixation, interaction, and error data (Chapter 4). The visualization will occur both in real-time during the user's engagement with the simulation and during the debriefing session at the conclusion of the training simulation. Finally, we aim to expedite the development process and enable individuals without coding experience to incorporate these error feedback patterns using a no-code authoring tool.

1.2 Achievements

This master thesis contributes to the field of Virtual Reality training by addressing the crucial aspect of error feedback implementation. The thesis introduces a set of error feedback algorithms encapsulated within VR software design patterns. These algorithms leverage three simulated in-VR perceptions - haptic, visual, and auditory - to deliver precise, intuitive, and immersive feedback to users engaged in VR training scenarios. Additionally, we propose a data collection implementation that captures crucial user metrics such as eye-gaze fixation, user's position and orientation, and hand interaction.
Moreover, with a user-friendly interface, we visualize the aforementioned data, both real-time, while the user is training, and offline, on the debriefing. Furthermore, we utilize a mind-map-based theory to consolidate the aforementioned training data, errors, eye-gaze fixation points, user's position and orientation, and interaction data. As a pilot training simulation, we developed a PC building simulations of 25 action-steps that the user has to perform. On each action, we have integrated our 5 error feedback patterns.
The error feedback patterns were linked to actions through the no-code VR authoring tool offered by the MAGES SDK, which was enhanced in this thesis to include the ability to add, delete, and configure these patterns within the authoring tool. We conducted 2 evaluation processes. A heuristic evaluation with 5 User Experience (UX) experts that evaluated the system having as a use-case a PC Building training simulation. A cognitive walkthrough evaluation with 5 experts with Unity game engine that evaluated the no-code authoring tool for the error feedback patterns.
In summary, the thesis advances the training in VR by offering standardized error feedback algorithms, streamlined setup processes, data collection techniques, and sophisticated visualization tools for error feedback analysis.

1.3 Overview and Dissertation

In the following chapters, we will analyze key functionalities such as error patterns, the data collection process and visualization, the no-code authoring tool, and the world-map implementation. In Chapter 2 we will present related work that impacted this thesis. In particular, we studied about VR training scenarios, human perception, error feedback in VR, data analysis in games, visualization tools like mind-maps, human learning processes, authoring tools, analytics in VR, and our related publications. In Chapter 3, we presented the five error feedback patterns. In Chapter 4, we will delve into data collection and visualization regarding the error feedback and interactable objects for each action step that a user must perform in the training simulation. In Chapter 5, we will discuss mind maps and how we implemented this concept to meet our needs. In Chapter 6 we showcased the no-code authoring tool for the 5 error feedback patterns. In Chapter 7 we talked about the evaluation process. Lastly, in Chapter 8, we discuss potential future work that we believe would enhance the system's functionalities.

Chapter 2

State of the art

In this chapter, we will delve into previous work related to virtual reality simulations for training scenarios, exploring how feedback can be delivered through virtual reality headsets. Also, we will describe previous projects and publications in this chapter regarding the error feedback aspect of training simulations in virtual reality environments.

2.1 The influence of Virtual Reality on training scenarios

In recent times, a growing demand for VR training simulations has underscored the importance of immersive learning experiences. More and more industries have jumped into the virtual world to create immersive, fun and replicable experience enhancing both the soft skills, such as leadership and resilience and the hard skills, such as how to perform a correct incision as a doctor or how to remove a tire as a car mechanic.

Aviation was one of the first fields that have started using simulations and serious games as a way to train new pilots. Aviators were using the Full-Flight Simulators (FFS) (Figure 2.1) in order to train within look-alike airplane cockpit. The FFS also includes motion systems and high-quality graphics to provide a lifelike experience.

Figure 2.1: Full-flight Simulator model CAE 7000XR.

Nowadays, many simulations leverage VR technology to enhance user immersion, making training more engaging and cost-efficient. Trinon Hélène *et al.* [57] analyzes immersive technologies with a focus in flight simulation. Below, we will mention some of the simulations that uses virtual reality technologies for training aviators.

Microsoft Flight Simulator[1] is one of the oldest ones. This simulator started as a 3D serious game and it now expanded by using VR headsets. It is a groundbreaking and immersive flight simulation software that sets new standards for realism (Figure 2.2) and detail. Developed by Asobo Studio and published by Microsoft, this simulator offers an unparalleled experience for aviation enthusiasts and casual gamers. It was also used by some aviation companies and flight training organizations as a supplementary tool for pilot training and proficiency. While it may not replace full-motion simulators used in professional pilot training programs, the simulator's realistic flight physics, accurate aircraft models, and detailed global scenery make it a valuable training aid.

Figure 2.2: Microsoft Flight Simulator: Diffent Cockpits of airplanes.

X-Plane[2] is a realistic flight simulator meticulously built by pilots, for everyone. X-Plane was initially launched by Laminar Research in 1995. Their focus lies on the robust physics engine that makes the simulation unparalleled experience for those seeking the thrill and challenges of the real world. X-Plane offers a comprehensive and dynamic platform for virtual aviation exploration. Also, there is a professional version of the simulation for those that want a top-notch immersion of flying an aircraft.Lastly, a version is available (Figure 2.3) that, with some modifications, can achieve sufficient performance to operate in VR [57].

Going to our last example, **Flyinside** [21] is one of the few flight simulators that was originally built to run in VR headsets. Created by Dan Church, FlyInside integrates seamlessly with popular flight simulation platforms such as Microsoft Flight Simulator[1], X-Plane[2], and Prepar3D[3]. They support a lot of ways to interact withing the application's User Interface (UI) such as Leap Motion, Oculus Touch, Vive controllers and the infamous mouse cursor. They also provide an open-source SDK for the developers to create their customs add-ons. The graphics are not as high as the ones of the above applications.

[1]https://www.flightsimulator.com/
[2]https://www.x-plane.com/
[3]https://www.prepar3d.com/

Figure 2.3: X-Plane simulator with VR.

Figure 2.4: Microsoft Flight Simulator: Different Cockpits of airplanes.

Serious games in VR are one of the key-features that improve the learning of the user in various soft and hard skills.They have emerged as a transformative tool in the training process, revolutionizing traditional learning methodologies [10]. By combining the immersive and interactive nature of VR with educational content, serious games create dynamic and engaging environments that enhance user learning experiences. These VR simulations enable users to apply theoretical knowledge in practical scenarios, fostering a deeper understanding of complex concepts. The impact of serious games on training is profound, as they provide a risk-free space for users to make decisions and learn from their outcomes[45]. This experiential learning approach not only accelerates skill acquisition but also promotes retention and transferability of knowledge to real-world situations.

Below you can find some examples of serious games in VR in deep relation with the research field. We will annotate serious games that are used for therapies, training scenarios that improve the cognitive skill of the patient, and serious games in order to train employees. Many solutions have been developed for leadership skills, healthcare, insurance, hospitality, logistics and financial services.

Talespin[4] is an immersive learning platform that provides enterprises with the power of immersive learning. It enables employees to learn critical workplace skills through immersive and engaging interactions with AI-enabled characters. Talespin's platform is trusted by global brands across industries and is scalable and personalized. The platform is deployed in hours, making it a quick and effective solution for employee training. The platform also provides a no-code content creation tool that allows enterprises to create and customize immersive learning content to meet their specific needs [53].

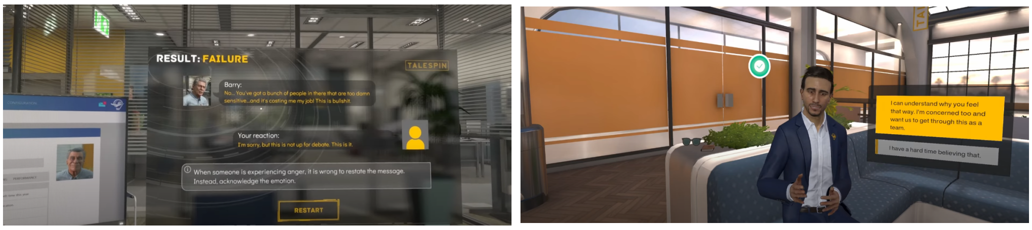


Figure 2.5: Talespin VR training simulation.

They offer simulations across various categories, including leadership, insurance, health, hospitality, logistics, and financial services. Within the leadership category, there is an app called **Effective Performance Feedback** (Figure 2.6). In this simulation, users take on the role of senior managers in a fictitious IT company. Their task is to provide feedback to an employee who has experienced a notable decline in the quality of their work, despite typically performing well. Users must engage in a conversation with the employee about the performance decline and suggest concrete steps the employee can take to improve. This module targets Leaders and Managers to practices their soft skills.In this simulation some of the skills that are practiced are conflict resolution, constructive feedback, verbal communication, resolving issues and probing questions.

PIXO VR[5] is a technology company that create VR training simulations to make work safer, more enriching, and career advancing for employees.They have created a lot of training scenarios that upskill the users' soft and hard skills. In particular, their scenarios annotates the workplace safety, skilled labor, employee wellness and public safety. Their training simulations vary from manufacturing to construction to career development.

One of their many scenarios is for training hard skills is **Hazard Recogntion Training**. The user can choose between two hazardous warehouse environments,

[4]`https://www.talespin.com/`
[5]`https://pixovr.com/`

Figure 2.6: Effective Performance Feedback simulation

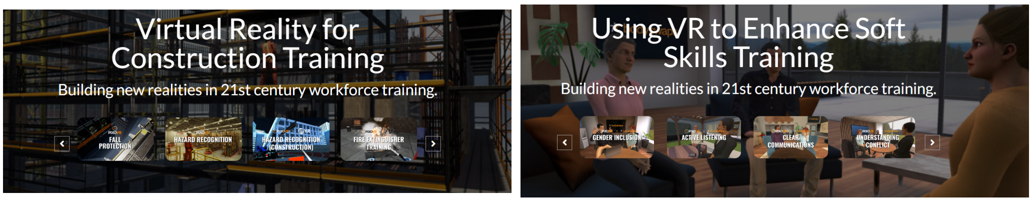


Figure 2.7: PIXO VR Construction and Soft skills available training applications.

a storage warehouse and one storing chemicals. Once the user chooses the environment, then he will must inspect and apply the safety equipment. After doing so, he has to enter the warehouse. Users encounter distinct risks that are randomly selected from millions of potential work environments.The user's selections must all be identified in accordance with their instruction. Employees have access to Safety Data Sheets (SDSs) for evaluation. After completing their evaluation, the staff member departs from the warehouse. The employee may then go back inside the warehouse and retrieve the information to evaluate mistakes, with objects coloured red, yellow, or green to show which dangers they properly recognised. The user's knowledge about protection (checking the personal protective equipment), identification (hazard recognition) and correction (recommending corrective actions for identified hazards) will be challenged throughout the whole simulation.

Another scenario they provide is to train soft skills is **Interview Simulation** that was made with the collaboration of bodyswaps company [7]. In this application, the user participates in any one of four fully immersive learning simulations of a job interview, where they may polish their pitch, pick up interview techniques, and get practice responding to numerous interview questions. The simulation offers pointers and advice on the best and worst practices for each question addressed during the job interview. After the interview is over, candidates can replay the

Figure 2.8: PIXO VR: Hazard Recognition Training

interview and switch places with the interviewer to assess how they performed. In order to help with interview preparation, non-VR exercises are also given to trainees. After mastering these strategies for reducing anxiety, trainees usually go on to virtual interview preparation. Learners can develop their abilities on their own, with constant practice, tailored feedback, and complete psychological safety. The user's will test their skills on career goals, about their character, behavioral matters and on creative questions

Figure 2.9: PIXO VR x Bodyswaps: Interview Simulation.

All the above share a commonality in providing users with feedback on both their positive actions and mistakes. We will discuss further about analytics and feedback in Video Games, both VR and non-VR.

2.2 The human perception

Perception [18] is the process by which the human brain derives organised experience from sensory stimuli. That perception, or experience, is a combined result of the procedure and the stimulus. Relationships between different stimulus kinds (such as light and sound) and the corresponding perception produce allow for the development of theories of perception based on conclusions drawn about the characteristics of the perceptual process. The correctness of perceptual theories can only be indirectly verified because the perceptual process itself is not publicly available or immediately visible (except from the perceiver, whose percepts are supplied directly in experience). In other words, acceptable actual evidence is contrasted with theory-derived predictions, often through experimental study. In this section we will not focus on the philosophical aspect of human perception, such as what we can feel, what we can perceive and the existence of another world other than the physical one. The aim of the section is to analyze the input that a human can receive.

One component of a stimulus or what is observed following a stimulus is known as the stimulus modality, sometimes referred to as the sensory modality. These sensory receptors include [2, 55] (2.10):

1. **Vision**: The ability to interpret light patterns reaching the eyes, enabling the perception of shapes, colors, and spatial relationships.
2. **Hearing**: The capacity to perceive sound waves and interpret them as distinct sounds and tones.
3. **Tactile**: The sense of touch, including pressure, temperature, and pain, which is detected by receptors in the skin.
4. **Gustatory**: The ability to detect and distinguish various tastes such as sweet, salty, sour, and bitter through taste buds on the tongue
5. **Olfactory**: The sense of smell, which involves the detection and interpretation of chemical odors through receptors in the nose.
6. **Vestibular**: The vestibular system plays a role in helping the body maintain balance and awareness of its spatial orientation.
7. **Interoception**: The collection of senses that provide information to the organism about the internal state of the body.
8. **Proprioception**: Very similar to the vestibular system, proprioception is how we determine the position of our whole body in space. In contrast,

proprioception refers to how we interpret the relationship and energy between each individual body part.

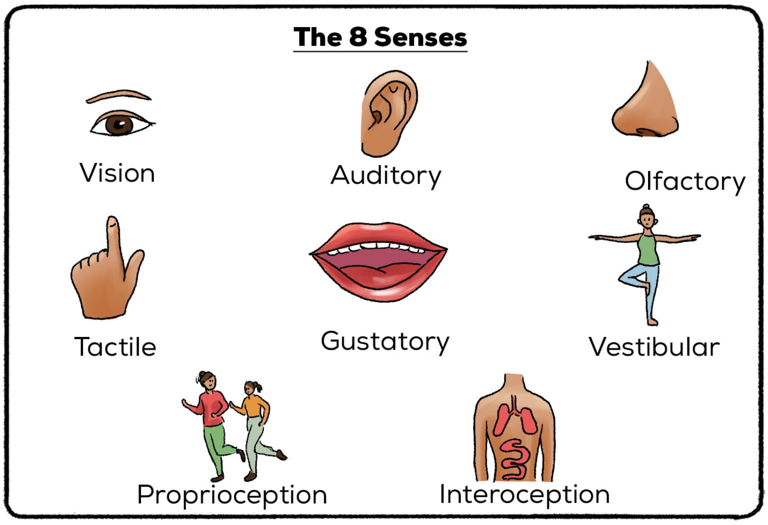


Figure 2.10: The 8 type of sensory receptors[55].

To represent and comprehend the information or environment that is offered, perception is the organisation, identification, and interpretation of sensory data [49].

Faieza Abdul Aziz *et al 2009* [1] discussed about the different haptic feedback that we can have in a VR application to enhance user performance in manufacturing industry. In this research, they categorize the different types of haptic data. For example, haptic is related to the sense of touch, Kinesthetic is the feeling of motions, relating to sensation originating in muscles, tendons and joints, Cutaneous that includes that sensation of pressure, temperature and pain.

For our research, we will focus on 3 of the 8 senses. The **Visual** sense, the **Hearing** sense and the **Touch** sense. VR head mounted displays (HMD) that are commonly used for training or entertaining reasons have lenses, speakers and provide haptic feedback. Extra hardware can be added to increase the human senses of the real world to the virtual world [31].

An example of a multisensory HMD that allows the user to feel the temperature of the virtual world and smell different scents is the VR Game Season Traveller [47]. Season traveller allows the user to feel wind, temperature, and smell on the Samsung-VR HMD (Figure 2.11). By incorporating olfactory and haptic stimuli, including thermal and wind sensations, we expand upon conventional audio-visual VR technologies to enhance sensory immersion in interactive experiences. Through the use of subjective assessments of presence and elicited physiological responses, we assessed the effects of various modalities on the virtual experience.

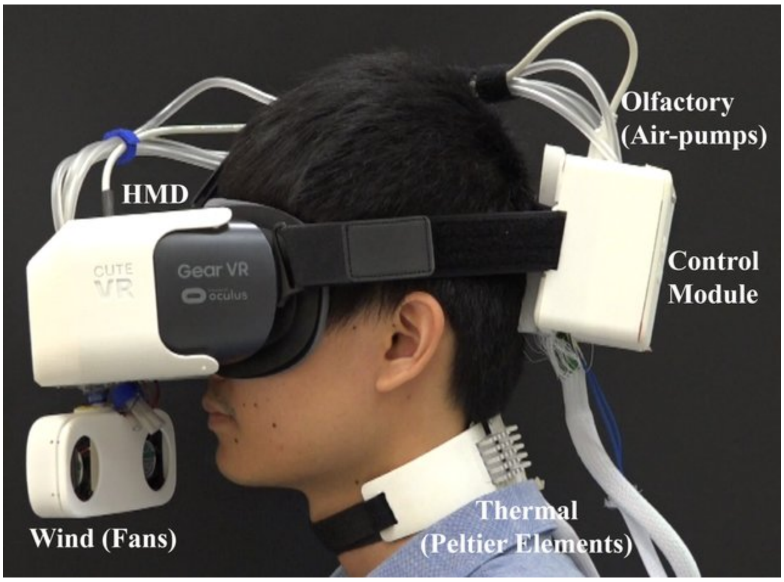


Figure 2.11: The Head Mount Display needed for Season Traveller Game [47].

We won't dive any further with extra sensors equiped onto the headset. This work focuses only on 3 sensory receptors, Visual, Auditory and Tactile. That is because the main goal of this thesis is that the error feedback patterns can work on any consumer-standard VR HMD, like Meta Quest 2, Pico 4 and VIVE XR Elite.

2.3 Error Feedback in VR Games and Training scenarios

In the current era, VR psychomotor training scenarios improves the learning curve of the user for certain tasks. Using gamification techniques [23] and correct targeted feedback the user can enhance his skills. A lot of papers and clinical trials [5, 4, 28, 26, 48] have proven that performing a task first in VR and then in real life can be done faster and better.
But how can we inform the users about an action they need to take, a mistake they did or congratulate them for an answer they gave? Before answering these questions, we first need to acknowledge all the methods that a VR headset can interact with the user. The outputs of a VR HMD are mainly three (Figure 2.15):

1. Visual (through lenses)
2. Auditory (through speakers)
3. Kinesthetic (through controllers)

The above outputs are standarized in the industry of VR headsets. VR games and training scenarios give feedback to the user depending on a particular movement they have to do (eg. Insert a key to the chest's hole) or inform them about their progress (eg. User is heading to a wrong way or Inserting the wrong key to the chest's hole) [54]. The feedback is being given to the user while playing the game, informing for his progress (eg. In a racing game: How many kilometres needs to finish the lap), explaining for wrong actions (eg. In Medical Training: Performing a wrong incision to a 3D human) or teaching the correct gesture of a certain action (eg. In Medical Training: Using the scalpel to create a incision onto a 3D human for an anatomy class)

Figure 2.12: Visual feedback informing the user that he performed a wrong incision.

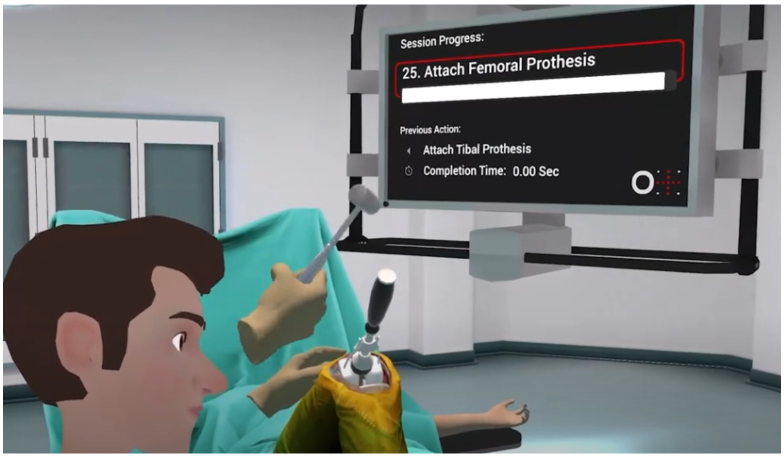


Figure 2.13: Visual feedback informing the user about his progress.

In Figure 2.14 we can see the user being in a medical training scenario, made by ORamaVR [43], using the drill tool for a total knee arthoplasty surgical operation. The feedback is received by the user through all outputs we categorized above:

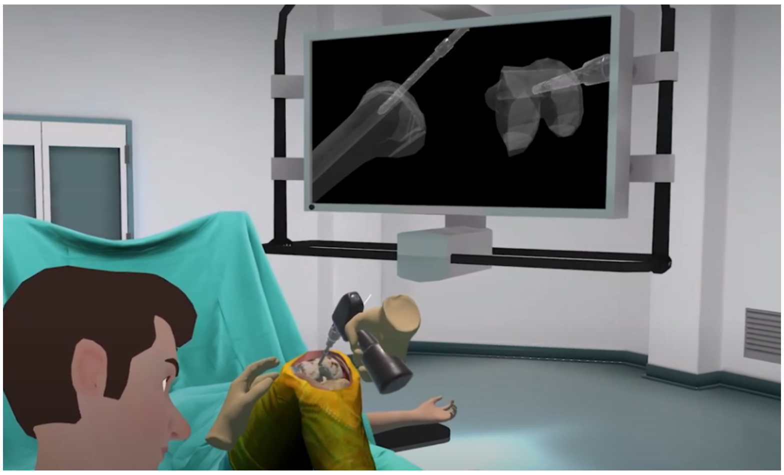


Figure 2.14: Visual, Auditory and Kinesthetic feedback informing the user about his progress using the drill.

visual, auditory and kinesthetic feedback. **Visual** feedback is provided by displaying on the HMD monitor giving the user continuous knowledge of how deep the drill has been inserted into the virtual patient's knee (Figure 2.12, 2.13). **Auditory** feedback is provided by playing the drill sound, the user understands that he has activated the drill. Last but not least **Kinesthetic** feedback is provided by vibrating the user's controllers when the user is using the drill.

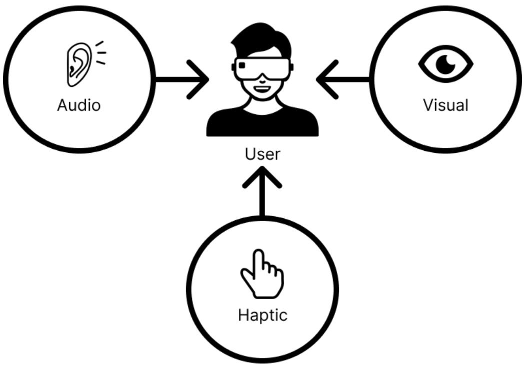


Figure 2.15: The three ways that the user can receive information from VR,

Another example is from the company called Innerspace [6] that create VR training scenarios for pharmaceutical companies to efficiently train and evaluate cleanroom personnel, mitigating errors and minimizing risks. In their simulation they provide accurate feedback on human errors. From their videos, it's evident that their primary method of error feedback is visual. When a user makes a mistake, they provide real-time information through a UI that explains the error. Additionally,

[6] `https://www.innerspace.eu/`

they highlight the VR view in red, prompting the user to quickly recognize and address the issue. An error feedback visualization can be seen in Figure 2.16 from one of their training simulations.

Figure 2.16: Error feedback in the Innerspace Simulator includes visual cues, such as alerts indicating when the user's hand movement is too rapid.

Gibbs *et al.* [24] conducted a study comparing the effects of haptic and visual feedback on presence in virtual reality. Their work examines the contributions of haptic and visual feedback to participants' sense of presence within a virtual environment. Additionally, they conducted tests in which participants experienced a virtual ball bouncing on a virtual stick resting across their avatar's hands. The study found that the sense of presence was heightened when participants could both see and feel the ball's actions, indicating that the combination of visual and haptic feedback contributed significantly to the sense of presence. Furthermore, there was a notable indication that haptic feedback alone elicited a greater sense of presence compared to visual feedback alone.

Thus, studying the research papers, journals but also from industry applications and games in VR, we can see that Virtual, Audio and Haptic are the 3 pylons of output from a VR device. Moreover, from the Section 2.2, we wrote that those 3 are going to concern us in this thesis. Ofcourse,the outputs can be increased by adding third-party equipment such as warbles and extra sensors as said in Section 2.2.

2.4 Data Analysis in Games and Training Scenarios

Analytics in Games and Training scenarios is a field that has gained significant attention in recent years. Utilizing data analytics in video games and training scenarios can enhance user experiences by aiding in the comprehension of both successful and unsuccessful actions. This can lead to more fun, engaging and competitive games and training scenarios that better meet the needs of users. Although video games are mainly for entertainment, the main goal of providing analytics to the user is to improve his skills.

For several years, video games have been widely recognized as a primary source of entertainment. Individuals spanning various ages and generations engage in video games, whether in singleplayer or multiplayer modes, using desktop PCs, laptops, consoles, or mobile phones. This widespread recognition has ignited a passion for competition, thus opening doors for data analysis and analytics visualization.

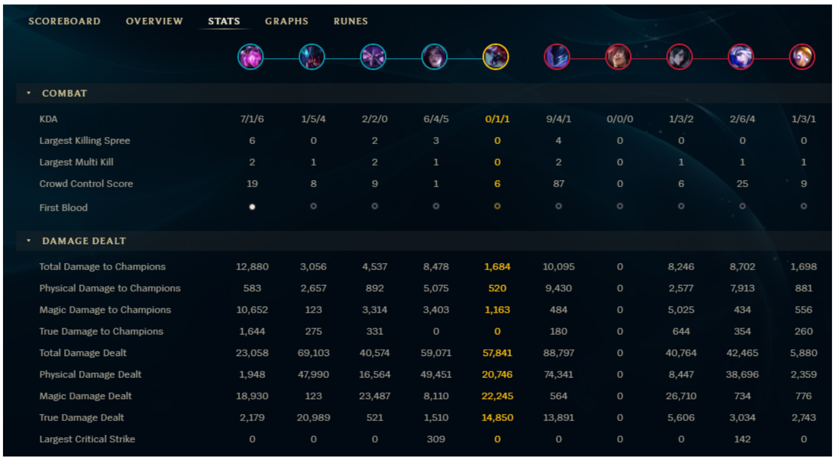
Figure 2.17: Statistics for Counter Strike Global Offensive from an external tool, csstats.gg [13].

First and foremost, analytics in video games provides insights into player behavior, preferences and performance from data collected while playing the game. After that, these data can be visualized in a fun, easy to review and playful manner, allowing game designers and players to explore and understand the game's dynamic in a more engaging way [37]. Analytics systems, through the visualization of extensive player populations and their interactions with the game, offer a comprehensive insight into player behavior. They provide specific details on the utilization of game elements by players and facilitate comparisons between different players or groups. This not only improves the game design process but also enables players to delve into their own gameplay and performance, resulting in a more immersive and enjoyable gaming experience [37].

A player can check his analytics either from the game itself (Figure 2.18) or by using an external tool (Figure 2.19) if the game does not provide these kind of information or the information is incomplete. Counter Strike: Global Offensive[7] is another example of a video game that an external tool is needed in order to

[7] https://store.steampowered.com/app/730/CounterStrike_2/

have extra analytics and information about the players gameplay. An example an external tool can be seen in Figure 2.17.



SCOREBOARD OVERVIEW STATS GRAPHS RUNES

COMBAT										
KDA	7/1/6	1/5/4	2/2/0	6/4/5	0/1/1	9/4/1	0/0/0	1/3/2	2/6/4	1/3/1
Largest Killing Spree	6	0	2	3	0	4	0	0	0	0
Largest Multi Kill	2	1	2	1	0	2	0	1	1	1
Crowd Control Score	19	8	9	1	6	87	0	6	25	9
First Blood	●	○	○	○	○	○	○	○	○	○
DAMAGE DEALT										
Total Damage to Champions	12,880	3,056	4,537	8,478	1,684	10,095	0	8,246	8,702	1,698
Physical Damage to Champions	583	2,657	892	5,075	520	9,430	0	2,577	7,913	881
Magic Damage to Champions	10,652	123	3,314	3,403	1,163	484	0	5,025	434	556
True Damage to Champions	1,644	275	331	0	0	180	0	644	354	260
Total Damage Dealt	23,058	69,103	40,574	59,071	57,841	88,797	0	40,764	42,465	5,880
Physical Damage Dealt	1,948	47,990	16,564	49,451	20,746	74,341	0	8,447	38,696	2,359
Magic Damage Dealt	18,930	123	23,487	8,110	22,245	564	0	26,710	734	776
True Damage Dealt	2,179	20,989	521	1,510	14,850	13,891	0	5,606	3,034	2,743
Largest Critical Strike	0	0	0	309	0	0	0	0	142	0

Figure 2.18: Statistics from a game session in League of Legends[8]

League of Legends [8] is a team-based multiplayer online battle arena (MOBA) game where two teams of five champions face off to destroy the other's base. There are over 160 champions each one specialized in a different role based on their abilities and play styles. League of Legends is the game that helped into creating a competition scene in the electronic sports (e-sports). The prize pool of the world championship exceeds $2 million. Having such a huge prize pool will push the player to improve his skills which made apparent the need for precise, efficient and loads of analytics began. League of legends application provides a lot of statistics from different categories such as combat, damage dealth, vision, income and damage taken and healed for each player, 10 in total, of a game session (Figure 2.18). There is an overview tab where the player can see the valuable statistics of objectives (killing the dragon or barron), kills and gold earned. Also, there is a minimap where you can see where each death took place (Figure 2.20).

A study was made about eSports analytics in League of Legends [36]. The paper discusses the use of data science and analytics for MOBA games in order to develop new metrics for individual performance. It introduces advanced statistics and analytics to measure player performance and its impact on win probability, highlighting the importance of factors such as favorable fights, deaths, vision, time management, and survivability. The study also calculates in-game win probability using a machine learning model and analyzes the contribution of individual performance to the overall team's win probability. Additionally, it introduces advanced metrics, automates player improvement analysis, and provides role-specific isolated correlations of bundles of stats with win probability [36]. Therefore, these

[8] https://www.leagueoflegends.com/

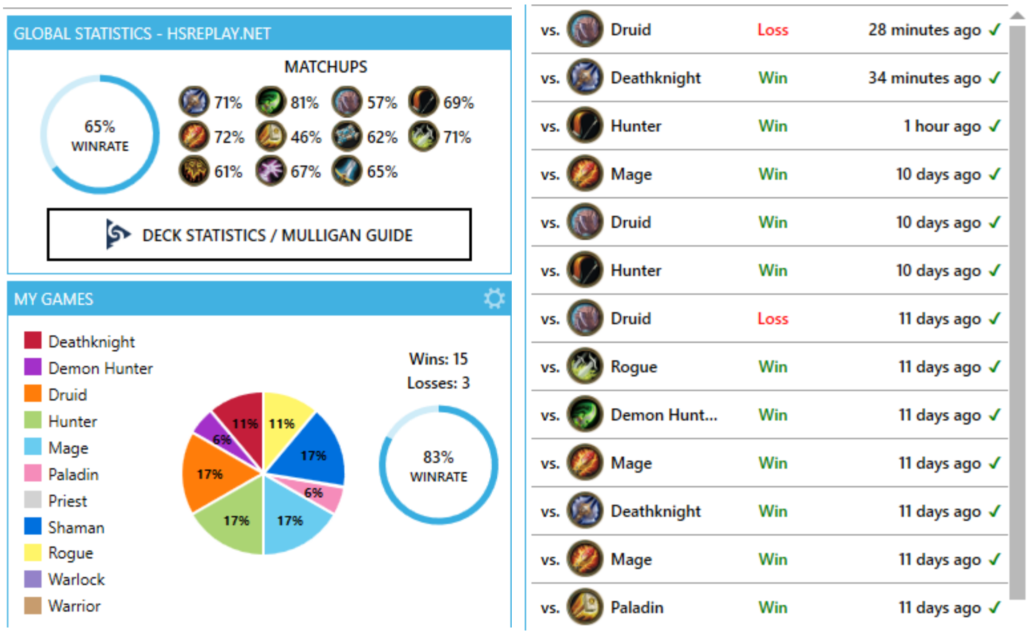


Figure 2.19: Hearthstone Deck Tracker [27] an external tool that shows data for hearthstone decks, winrates and also displays old duels to revisit.

advanced statistics and metrics provide a more accurate assessment of how individual performance impacts team success in League of Legends.

On the other hand, analytics on virtual training scenarios are focused on enhancing the learning outcomes of the user. Analytics there are used to assess trainee performance and identify areas for improvement.Enhancing users' skills based on the analytics and feedback they receive is pivotal for their present and future roles, ensuring more efficient and safer execution of tasks.We will provide some examples of analytics and feedback from various VR training scenarios.

Covid-19 Strikes Back (CVRSB)[58] (Figure 2.21) is a VR training application is dedicated to providing medical personnel with a quicker and more efficient learning experience, specifically targeting the proper techniques for pharyngeal swab procedures and the correct donning and doffing of Personal Protective Equipment (PPE). The application is free for download in SideQuest[9] and it is available for Meta Quest 2 and 3 HMDS.

Upon finishing the simulation and all the steps needed for completion, the user is presented an analytics window UI. There, the user will be able to scroll and see your errors. In CVRSB, there are 3 types of errors, Warning, Error and Critical Error. Also, the user will be able to check his total score and the duration it took to finish an action. Lastly, if the user clicks on an action, he will be able to see

[9] https://sidequestvr.com/

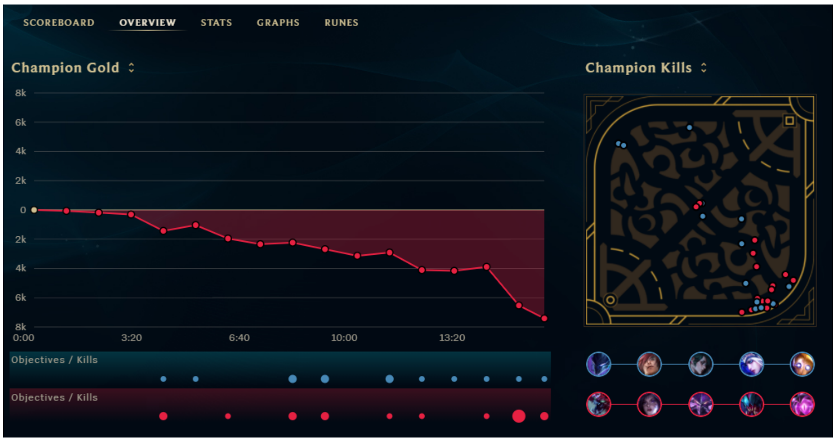


Figure 2.20: League of Legends overview tab of a game session.

the errors with comments on his correct and incorrect actions (Figure 2.22).

XR Labs[10] is a company that creates metaverse training simulations.One of their simulations is about the assemble process of a Hyundai car. From their video[11] we can see that the user receives visual feedback, for example when the user turns the screw a pop-up tick icon appears informing him about finishing the task. When the user finishes the simulation he will be presented a UI score about how well he did in safety, part assembly, connector assembly et cetera. Lastly, XR Labs do not provide further details regarding their analytics and more detailed feedback.

Pixo VR[12] was mentioned in section 2.1 for two of their training simulation. In their website they claim to provide insights of the users data from their session[13]. PIXO VR automatically collect data and then the admin can see the content, time spent doing training, scores from training assessments, and a host of other data points. Thus, every simulation they have created can capture users data and then visualize their playthrough into a website.

PianoVision[14] is a Mixed Reality (MR) application for Meta Quest 3. It allows the user to learn how to play piano using their own physical piano as an MR application or MIDI keyboard or by using a virtual piano as a VR application (Figure 2.23).

The app responds real-time with the user's performance in order to help the user practice more efficiently. The feature that this app is worth mentioning in this section is the performance feedback and statistics for every piece the user plays

[10] https://xrlabs.co/
[11] https://www.youtube.com/watch?v=to3JGNjyBVI
[12] https://pixovr.com/
[13] https://pixovr.com/the-pixo-platform/
[14] https://www.pianovision.com

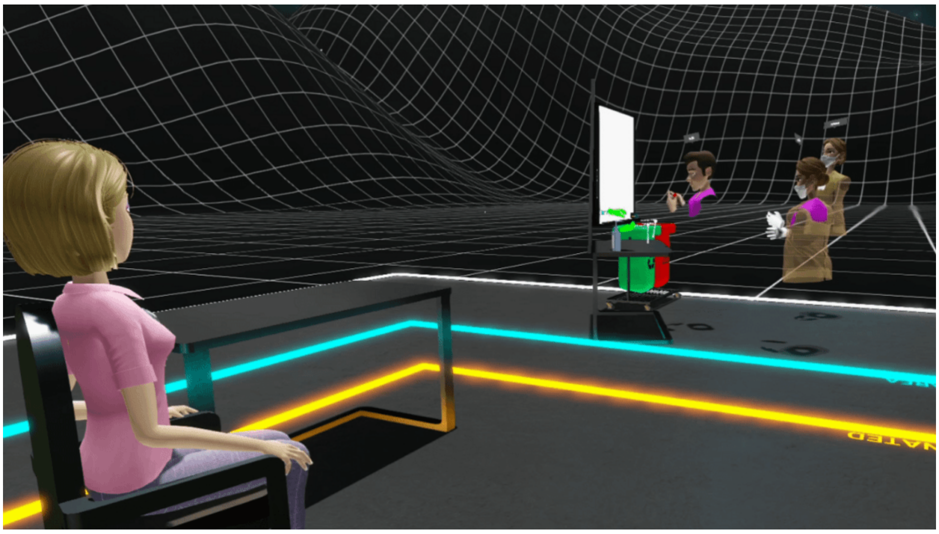


Figure 2.21: "Covid-19 Strikes Back" in game photograph.

(Figure 2.24). The app records Critical performance metrics such as timing, accuracy handedness and more. The user can see the progress after each play in the entirety of their piano journey. The aforementioned data helps users understand what they need to work on. The app is available for purchase at the Meta store in an affordable price.

VirtualSpeech[15] is a VR application that enhances the soft-skills. The key-feature in this application is the integration of AI functionality. Leveraging VR to simulate audiences of different sizes and distractions. It provides an efficient means to rehearse public speaking, and fostering the development of skills and confidence for real-world scenarios.The simulated audience uses the power of Chat-GPT[16]. The AI also reviews the user's speech and actions and gives feedback to them (figure 2.25).

2.5 Mind Maps

Mind mapping was first developed by T. Buzan as a note-taking technique aimed at saving time while maintaining the reader's interest as much as possible.[9]. A mind map is composed of lines, symbols, text, color, and graphic elements (Figure 2.26). This visual tool helps individuals organize, arrange, study, review, and memorize information effectively. By reflecting a natural and related way of thinking, a mind map enables users to construct their own knowledge systems.

Mind mapping serves as a visualization tool employed in instruction, which learners

[15] https://virtualspeech.com/
[16] https://chat.openai.com/

Figure 2.22: "Covid-19 Strikes Back" in game photograph.

can utilize to brainstorm ideas, take notes, structure their thoughts, and formulate concepts [17][51]. It can be used to generalize main ideas into a topic, help individuals remember information, and allow them to plan answers to exam questions [51].

In this thesis, we will use the theory of the mind-map and implement in a UI called "World-Map". In that particular UI we will visualize all the actions-steps that user took while training. Moreover, by mapping out actions, interactions, and eye-fixation data, it creates a detailed representation of the user's journey through the simulation. With the theory of the Mind Map, we aid in correcting mistakes but also empowering users to achieve better outcomes through informed and reflective practice.

Shi *et al.* [51] examine the impact of mind-map based instructions on students' cognitive outcomes. Overall, meta-analyses of the pertinent literature indicate that mind mapping-based instruction significantly enhances students' cognitive learning outcomes compared to traditional instruction. Furthermore, mind mapping-based instruction has demonstrated greater effectiveness when implemented in STEM subjects.

Eppler *et al.* [17] compare various types of visualization formats, including mind

[17] `https://www.nngroup.com/articles/cognitive-mind-concept/`

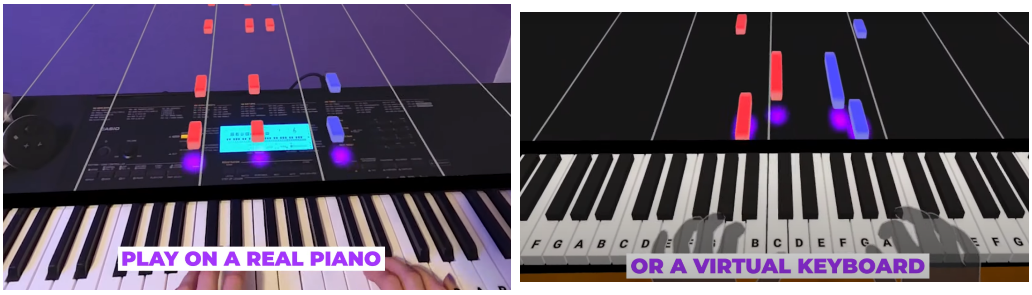


Figure 2.23: PianoVision application. The left image is using AR and the right image is using VR.

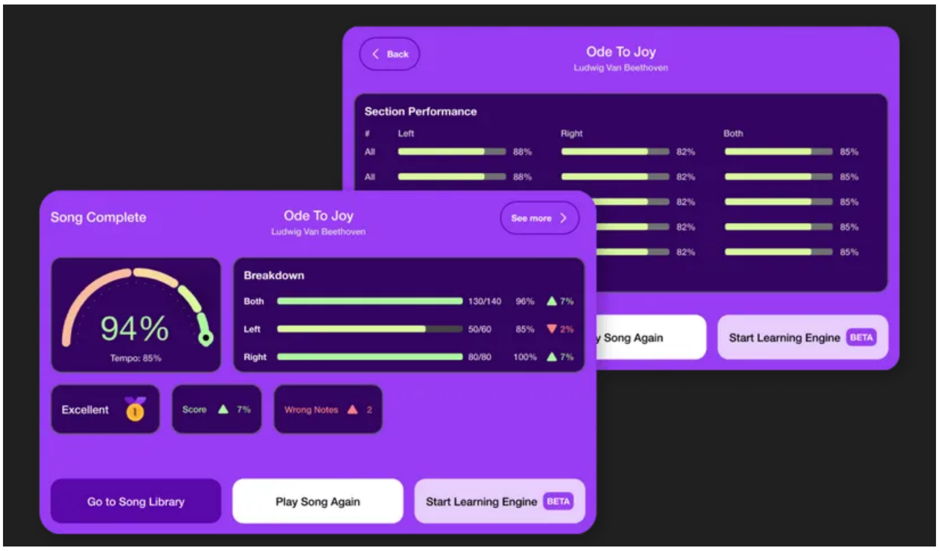


Figure 2.24: Expanding the Time scoring factor.

maps, conceptual diagrams, Novak's concept mapping, and visual metaphors. Subsequently, he identified the advantages, disadvantages and the unique features of each visualization format. He claims that a potential future research direction is to develop mixed-mode visualizations that combine the strengths of the four methods including:

1. Straightforward rules of concept maps.
2. Clarity of conceptual diagrams.
3. Simplicity of mind maps
4. Memorable richness of visual metaphors.

The results indicate that these different visualization formats can be used in complementary ways to enhance motivation, attention, understanding, and recall.

Figure 2.25: VirtualSpeech AI feedback analysis[15].

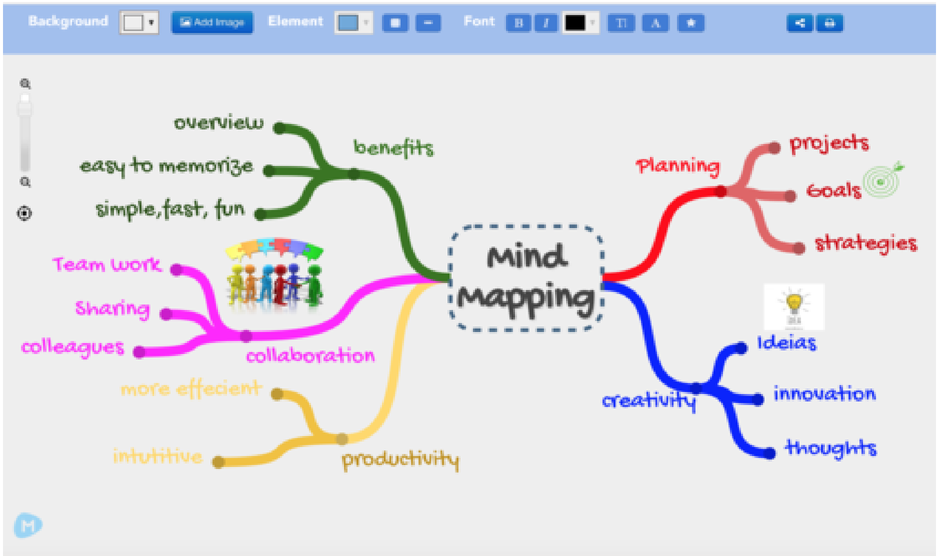


Figure 2.26: An example of a mind-map.

V D'Antoni *et al.* [14] explores if the mind map learning strategy facilitates information retrieval and critical thinking in medical students. He claims that mind map may help medical students organize, integrate, and retain information. Recent research suggests that utilizing mind mapping as a note-taking strategy enhances critical thinking. He concludes that the mind mapping does not increase short-term recall of domain-based information nor critical thinking compared to standard note-taking. Also, medical students using mind maps can successfully retrieve information in the short term and are not disadvantaged compared to students using standard note-taking methods.

Swestyani *et al.* [52] investigate the application of mind mapping as a tool to measure students' logical thinking abilities. The authors emphasize its importance in education, highlighting that mind mapping enables students to broaden their knowledge and comprehension, enhances self-awareness, aids in problem understanding, and provides effective and efficient solutions to everyday challenges.

2.6 Authoring tools feedback and Analytics VR

An authoring tool is a system that refers to a software program equipped with preprogrammed elements designed for developing interactive multimedia software. It can be characterized as software that empowers users to create multimedia applications, facilitating the manipulation of multimedia objects in the process. An example outside of the field of computer science is Canva [18]. Canva is an authoring tool that lets you design and create assets easily. You can create images for social media, videos and gifs to poster, websites and booklets.

For our case, in this section we will study and analyze different authoring tools that collects data and feedbacks the user when he is performing correct or incorrect actions. We will start by checking the VR authoring tool from MAGES SDK [43] focusing on the analytics of each action in the simulation. In this authoring tool, the developer can configure the data that will be collected from the player and create any error cases he wants. The developer can create time related errors, placement of an object, error colliders, stay error colliders, hit perform colliders, question, velocity and their custom errors. The scoring factor component that tracks data from objects or from the student's actions. All the above that were mentioned are different scoring factors.There is also a final score that is calculated depending on the weight of each of the scoring factors[59] (Figure 2.27).

Figure 2.27: Authoring tool from MAGES SDK [43] about Analytics.

Each error type may be configured accordingly. An example can be seen in the Figure 2.27. The Time error for example allows us to choose the factor of importance of the particular action for its time attribute. We must also set the desired time for completion. You can see this in the Figure 2.28

[18] https://www.canva.com/

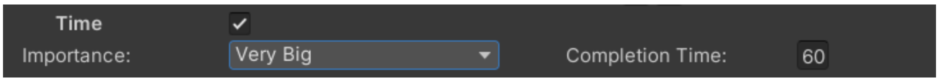


Figure 2.28: Expanding the Time scoring factor.

One can configure the other scoring factors in the same way in order to have the desired outcome. The process is straightforward, developers can create their aspired scoring factors with a lot of precision. Also, in MAGES SDK one can create his own scoring factor without using the provided authoring tool.

Another SDK that collects data and informs the user about mistakes is Cognitive3D. Cognitive3D [11] is a 3D analytics platform designed to measure human behavior within VR and AR simulations, transforming these observations into practical and actionable insights. The SDK include many features that collect data and then visualize them into a portal site. The SDK collected raw data of the users movement inside the VR and then allows to replay in VR all actions. Additionally, these tools include objectives, which serve as a mechanism for rapidly assessing participant behavior in 3D simulations by analyzing both sequential and non-sequential steps indicative of successful outcomes. The steps can be configured using the authoring tool they have created. An example of the authoring tool for the Steps can be seen in the Figure 2.30. Lastly, if the hardware allows it, one can collect biometric data and eye tracking data.

Cognitive3D [11] provides the Objectives feature which is a series of Steps that must occur in a Session. This capability enables a rapid comprehension of the actions undertaken by participants, shedding light on whether they have adhered to a specific process. Creating an objective becomes possible after recording one or more sessions. The generation of new steps necessitates the inclusion of event names and dynamic objects recorded during the session. It is crucial to plan the required steps in advance, allowing for the incorporation of custom events and properties in the content for measurement purposes. An example can be seen in Figure 2.29. The user can assess their performance by determining the number of objectives they have successfully completed. An objective is considered complete only when all its associated steps have been fulfilled.

As you can see in the Figure 2.30, we have created a step and set a type to it. Then, we choose the question set we want to be checked at. After that, we can set the correct answers on each question. The setup is pretty easy and straightforward. In order for the user to finish the steps, he will need to answer correct to each question. If that has not happened, then we can review this at the portal of Cognitive3D, same website that we set the steps.

FundamnetalVR [22] is a healthcare technology company dedicated to expediting human capability through precise simulation. Their mission is to enhance prehuman competence in surgery and other clinical settings, ultimately contributing

Figure 2.29: The Objectives of a session from Cognitive3D[11]. In this example the user has completed 3 objectives. The images was taken from the documentation age of Cognitive3D[11].

to improved patient outcomes. They have created a SDK for easy-budiling of educational modules. For our research, we will focus on the analytics part. In their feature list[19] they offer online results dashboard, real-time results in the VR application and a scoring system (Figure 2.31). Although we can see a beautiful representation for analytics, both in the dashboard and in the VR they do not provide any information about how they apply these scores and how they collect the data.

2.7 How we learn

In this section we will spark the flame about how people learn. This thesis has a focus on creating efficient and correct patterns for giving error feedback to the user. Thus, we need to investigate how do the human brain stores information? How we learn from our mistakes? What causes our brain to drain and save all these data we receive when we are doing something wrong?

Dahaene *et al 2020* [16] claims that the human brain is an extraordinary machine. Its ability to process information and adapt to circumstances by reprogramming itself is unparalleled, and it remains the best source of inspiration for recent developments in artificial intelligence. The book challenges traditional approaches to studying, suggesting that forgetting is a natural part of the learning process and that spacing out learning over time can be more effective than cramming. There are four essential mechanisms, or "pillars," massively modulate our ability to learn [16]:

1. The initial factor is **attention**: A collection of neural circuits that choose,

[19] https://fundamental-core.com/wp-content/uploads/2023/05/Fundamental-Core-Feature-List-PDF-format-June-2023.pdf

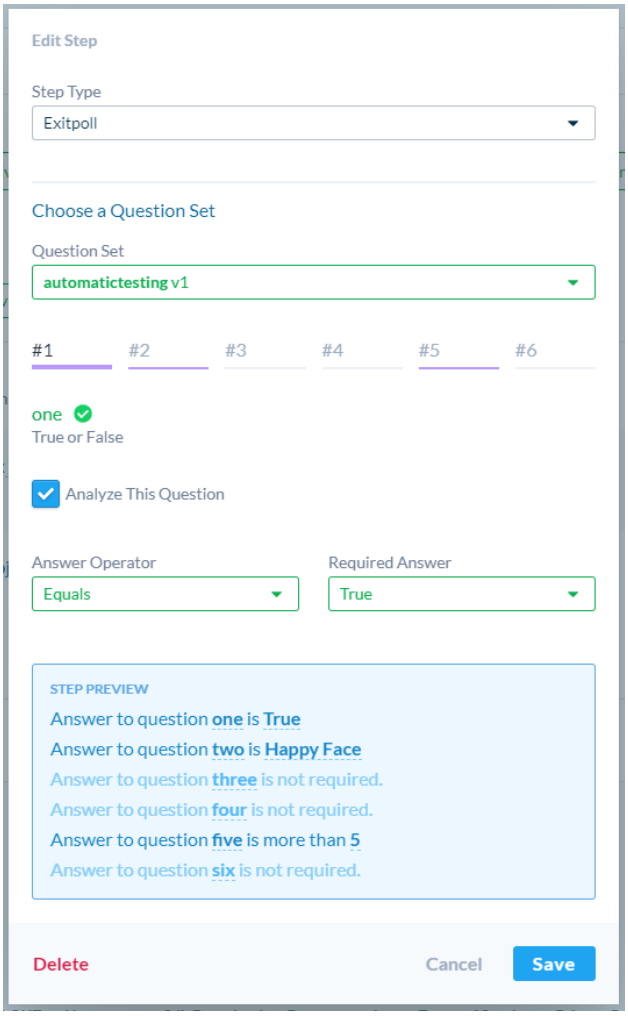


Figure 2.30: Authoring tool from Cognitive3D [11] that creates a series of "Steps" that must occur in a VR training session.

magnify, and transmit signals perceived as relevant, significantly amplifying their impact in our memory.

2. The second pillar is **active engagement**. A passive organism gains minimal knowledge, as learning necessitates the active generation of hypotheses driven by motivation and curiosity.

3. The third essential element, counterbalancing active engagement, is **error feedback**. Whenever surprise arises due to a deviation from our expectations, error signals disseminate throughout our brain. These signals work to correct our mental models, discard inappropriate hypotheses, and solidify the most accurate ones.

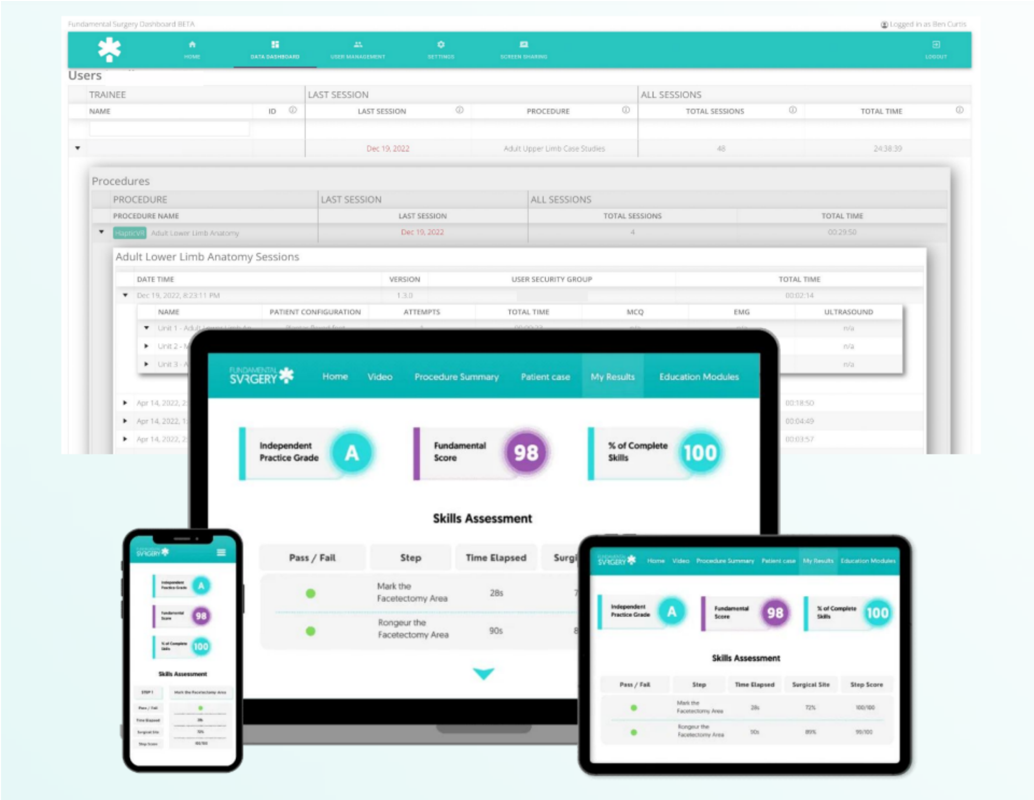


Figure 2.31: The Online Results Dashboard of Fundamental Core. The aggregated results from all completed sessions can be accessed through the Fundamental Surgery Dashboard. Figure taken from https://fundamental-core.com/.

4. Ultimately, the fourth pillar is **consolidation**. Over time, our brain compiles acquired knowledge and transfers it into long-term memory, thereby liberating neural resources for ongoing learning.
   (a) Repetition plays an essential role in this consolidation process.
   (b) Even sleep, far from being a period of inactivity, is a privileged moment during which the brain revisits its past states, at a faster pace, and recodes the knowledge acquired during the day.

To learn, one must encounter new information, pay attention to it, coordinate it with known information store in memory, and apply it [35]. For example, if one wants to learn how to fix a running toilet, he might search for a how-to video, watch it to see if it addresses his needs, and then use the instructions to make the repair. The term "active learning" is often used to describe an interactive process, such as doing a hands-on experiment to learn a concept rather than reading about it [35]. But "passive learning" (reading a text, listening to a lecture, watching a movie) is still learning, and can be effective.¨In this research, the focus will be on the third mechanism of the ability to learn, the error feedback.

Furthermore, Dahlin *et al.* [15] claims that the mechanisms of opportunity, motivation, and ability facilitate learning from failure for individuals, groups, and organizations, effectively bridging gaps across various levels of analysis. Learning from failure comes from understanding mistakes from personal and others' experiences. Motivation to learn is often hindered by punitive leadership and organizational practices. While innate attitudes influence this ability, it can be improved through deliberate analysis and adopting successful practices.

Moreover, Lee *et al.* [33] in his work studied how individuals learn from their own failures, an important microfoundational process that influences organizational learning and firm performance. He claims that the relationship between an individual's accumulated failures and learning will form an inverted-U shape—driven by opposing forces between an individual's opportunity and motivation to learn as failures accumulate—and that this relationship will be moderated by the individual's perceived ability to learn. Lee's studies show that, on average, individuals continued to learn from their own failures, albeit at progressively slower rates, until they reached significantly high levels of accumulated failures. He observed heterogeneity in the degree to which individuals learned from their failures, influenced by varying levels of perceived ability to learn. Specifically, surgeons with elite education, certified expertise, and specialization in patient care exhibited greater persistence in learning from their own failures compared to their counterparts. The theory posits that these individuals possess a higher perceived ability to learn, which leads to stronger motivation to learn and consequently reduces their vulnerability to negative emotions and attribution biases associated with repeated failures.

2.8 Our publications related to this work

In this section we present a short overview of our previous publications related to this work

- **MAGES 4.0: Accelerating the World's Transition to VR Training and Democratizing the Authoring of the Medical Metaverse[59]:** In this work, we presented MAGES 4.0, a novel SDK designed to accelerate the creation of collaborative medical training applications in virtual and augmented reality. Developers can rapidly prototype high-fidelity and high-complexity medical simulations by harnessing the power of the low-code metaverse authoring platform provided by the SDK. The platform introduces several innovations: 1) 5G edge-cloud remote rendering and physics dissection layer, 2) Realistic real-time simulation of organic tissues as soft bodies within 10 milliseconds, 3) A highly realistic cutting and tearing algorithm in under 10ms, 4) Neural network assessment for user profiling, 5) A VR recorder to record, replay, or debrief the training simulation from any perspective.

- **Enhancing Crew Resource Management in VR Medical Training: Tools for analyzing advanced, collaborative group training in the metaverse[30]:**

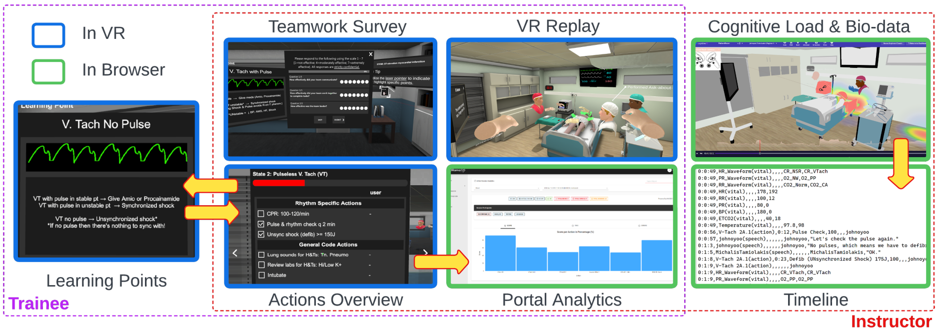
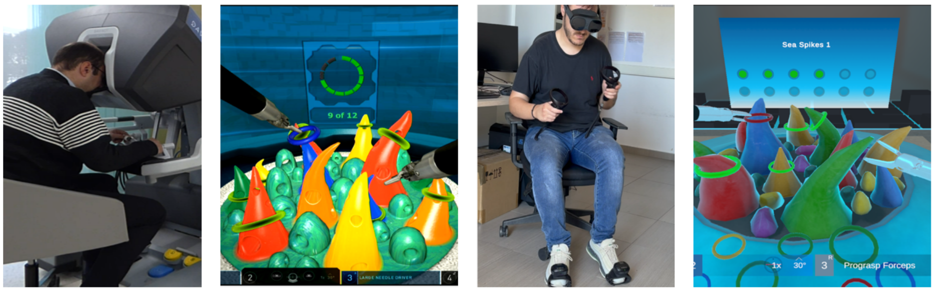


Figure 2.32: Our system offers novel comprehensive analysis for collaborative, group-based VR medical training sessions. This figure outlines the post-session analysis pipeline that (a) [purple rectangle]: Provides trainees with detailed analysis on performance through Actions Overview and Teamwork Survey interfaces. The Learning Points UI offers knowledge relevant to actions performed during the session. VR Replay and Portal Analytics allow learners to examine their session in depth. (b) [red rectangle]: Provides instructors with the same data trainees receive, as well as biometric data to analyze cognitive metrics such as cognitive load and visual attention/gaze.

In this work, we explore the concept of Crew Resource Management (CRM) that has gain traction in medical training (Figure 2.32). It improves the teamwork, communication and decision-making skills that improves the safety and reduces medical errors. Thus, we created a VR tool that aims to enhance the CRM in medical training scenarios. We captured a lot quantitative data that surpasses conventional non-VR evaluation/feedback techniques, such as instructors taking written notes based on their observations. Additionally, capabilities such as recording and replaying VR sessions, combined with timestamped communication log transcripts, enable comprehensive qualitative post-analysis.

- **VR Isle Academy: A VR Digital Twin Approach for Robotic Surgical Skill Development[20]:**

In this paper, we present a cost-effective method for training a Surgical Robotic System (SRS) through a portable, device-agnostic, ultra-realistic simulation that incorporates hand and feet tracking support (Figure 2.33).



(a) (b) (c) (d)

Figure 2.33: A real Surgical Robotic System (SRS) in comparison with a digital twin SRS, VR-Academy. In images (a) and (b), a modern SRS is depicted, showing a user operating from the surgeon's console. In contrast, images (c) and (d) showcase VR-Academy, where the user controls a simulated SRS digital-twin using an inside-out VR HMD, controllers and feet trackers.

> The system offers real-time and offline error assessment, facilitating the monitoring and evaluation of user performance. More over, it is evaluated by multiple untrained testers, demonstrating a significant reduction in error metrics with an increase in training sessions. These results suggest that it effectively enhances the robot-controlling skills of users in an intuitive and immersive manner, while also shortening the learning curve and maintaining minimal costs.

The hypothesis posed in this thesis, "How can we develop accurate error feedback patterns?" is derived from the aforementioned related works. Each of these studies has played a crucial role in shaping the hypothesis and subsequently guiding the research and development process. Our related work made us understand that a no-code authoring solution allows non-developers to participate in shaping and creating a training simulation while giving to expert users a powerful tool to quickly setup and configure a simulation. Moreover, our related work made us understand that we have to distinguish the analytics of a training simulation with the error feedback the user receives. So, patterns had to be implemented for the efficient feedback of the users.

Chapter 3

The Error Feedback Software Design Patterns

3.1 Summary

In this section, we will introduce the error feedback patterns we have developed. We will begin with a brief introduction to gamification patterns and then proceed to showcase the functionality of each pattern.

3.2 What is Gamification?

Gamification is the process of applying game design elements and principles to non-game contexts to engage and motivate people to achieve certain goals [3]. It involves incorporating elements such as points, badges, levels, challenges, leaderboards, and rewards into activities that are typically not considered games. By introducing these elements, gamification aims to make tasks more enjoyable, increase participation, foster competition, and encourage desired behaviors. It's commonly used in various fields, including education, marketing, employee training, health and wellness, and customer engagement, to enhance user experience and drive desired outcomes.

Moreover, in this thesis, great inspiration was taken from the Lee *et.al* [32] in order to create, easy to use, efficient and appealing gamification patterns for error feedback.

Another great example for the gamification patterns was from Costa *et. al* [12] where they created a gamification pyramid, categorizing the different gamification patterns.

Our solution is deeply rooted in the gamification process, which involves applying game design elements in serious games such as VR training for medical procedures or instructional VR scenarios like learning how to change a tire. Numerous studies have demonstrated that we can learn effectively from playing video games [34]. In that respect we will extract the most effective elements from video games and integrate them into training simulations.

3.3 The Software Design Patterns

In Chapter 2 we discussed previous work and many examples of error feedback in different VR simulations. We utilize these as reference points, gathering the most effective elements from each example found in research papers and within the research field. Furthermore, we integrated the three primary methods of providing feedback available in commercial VR headsets like the Quest 3 or the VIVE XR Elite. This approach enables us to create Error Feedback patterns that utilize either all three methods or a combination of them.

Initially, we devised three categories of feedback, each corresponding to one of the available output channels of a VR Head-Mounted Display (HMD). These categories are Haptic, Audio, and Visual. Each one of them implements a functionality to give feedback to the user with its unique way.

1. **Visual Feedback**: The visual feedback is achieved with the incorporation of a notification pop-up message that informs the user of any errors with a text notification. Additionally, it encompasses the alteration of material color to indicate changes or errors. It is effective for highlighting important elements within the VR environment, guiding user attention, and providing context for interactions. It's particularly useful for tasks requiring precise spatial manipulation, navigation, or identification of visual patterns.

2. **Visual Feedback**: The audio feedback is achieved by a played sound to alert the user in case of an error. Moreover, the auditory cues can serve as alerts, notifications, or confirmations, aiding users in understanding their interactions and the state of the virtual world. They are beneficial in scenarios where users may be visually overloaded or where spatial awareness is crucial but cannot be adequately conveyed visually.

3. **Haptic feedback**: The haptic feedback encompasses a manipulation of the vibration of the VR controllers to alert the user when they are holding something incorrectly, for example. Also, the haptic feedback can be utilized to simulate textures, object properties (e.g., hardness, softness), or interactions (e.g., collisions, vibrations). It's particularly valuable for enhancing presence and realism in VR experiences, providing users with a sense of touch and physicality that complements visual and auditory stimuli [46].

So, for the aforementioned classes (visual, audio, haptic), we integrated them into the following five patterns to achieve effective and efficient error feedback.

3.3.1 Time Pattern

Time can be crucial in a training simulation for several reasons. First and foremost, time management skills are often a key component of the training objectives. Moreover, time-bound simulations can also help simulate realistic scenarios where certain actions or events unfold over a specific time frame. This can be particularly relevant in fields such as healthcare, aviation, or military training, where timing is critical for successful outcomes. Thus, one of our error feedback patterns is Time.

In the **Time** pattern, we coordinate a synchronized visual and auditory feedback, strategically implemented when users exceed predetermined time constraints. Upon surpassing the specified time limit, users are promptly alerted through a combination of visual cues and auditory prompts, serving as immediate indicators of their temporal deviation. The visual feedback consists of an error message displayed as a UI, accompanied by a sound notification (Figure 3.1).

Figure 3.1: Time error occured resulting to a error visual feedback

3.3.2 Velocity Pattern

Certain objects require delicate handling, such as ensuring they do not collide with other objects or being cautious about the speed at which they are moved. In this error feedback scenario, we alert the user when moving an object at a velocity higher than deemed safe. This can be applicable to various objects like chemicals, needles, or other potentially hazardous items.

In the **Velocity** error, we orchestrated a multi-modal response to signal users when they exceed predefined speed thresholds while interacting with virtual objects. Upon surpassing the designated speed limit, users are promptly alerted through a combination of visual cues and haptic feedback. A notification message dynamically appears informing users of the velocity-related violation, while simultaneously, the controller being operated by the user undergoes tactile vibrations or shakes. By employing haptic feedback, we instantly communicate to the user that the error they received is related to the object they are interacting with (Figure 3.2).

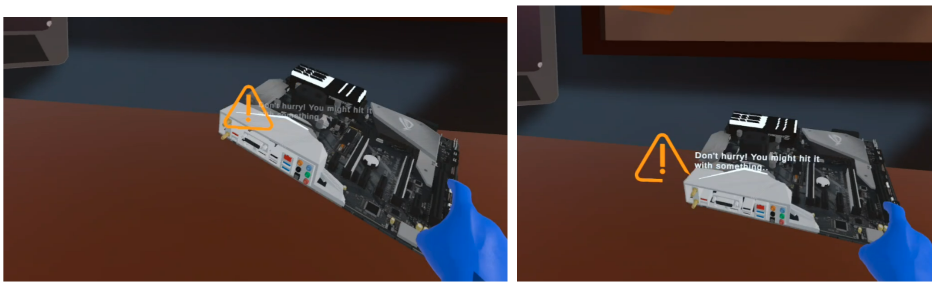


Figure 3.2: A collision resulted in both visual and haptic feedback. Multiple collision patterns can be present in an action, each triggering a response based on different layers or tags. In the left image, the collision error is triggered when the tool touches the glass. In the right image, the error is triggered when the glass is thrown to the floor, causing it to touch the floor.

3.3.3 Incorrect Response Pattern

One effective learning method involves question and answers. When the user gives an incorrect answer we have to inform him that his answer was incorrect. This approach utilizes the principles of active recall and engagement, encouraging learners to retrieve information from their long term memory banks in response to queries.

With the **Incorrect Response** pattern, the user receives visual feedback directly from the Question and Answer object. Additionally, sound feedback is provided to the user. By synchronizing visual and auditory feedback, this multimodal approach ensures that users receive comprehensive feedback throughout their session of a training scenario. By synchronizing visual feedback cues with auditory prompts, this multimodal instructional approach engenders a holistic learning environment wherein users receive nuanced feedback throughout the duration of their training regimen (Figure 3.3).

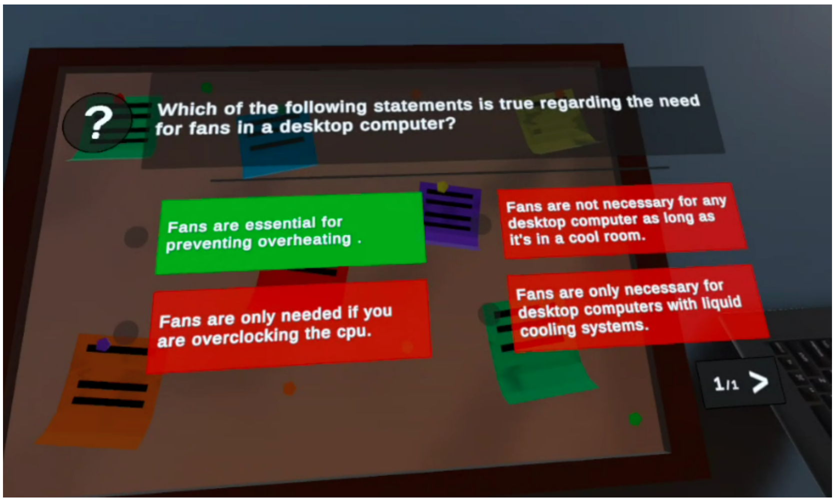


Figure 3.3: Incorrect answer occurred resulting to a error visual and auditory feedback.

3.3.4 Collision Pattern

In numerous contexts, characterized by exacting standards and procedural rigors, such as experimental protocols and industrial manufacturing procedures, the imperative of precision assumes paramount importance.Within these domains, meticulous attention and caution are essential to mitigate the risk of inadvertent collisions between objects for operational efficiency and safety. By exercising due diligence and adhering to established best practices, individuals can prevent collisions between objects.

In the **Collision** pattern, the user receives visual and haptic feedback when they collide the interactable object with other objects configured to trigger the error. This integration of visual and touch feedback exemplifies a user-centered design approach, aimed at increasing user control and reducing the likelihood of incorrect interactions. By imbuing users with real-time sensory feedback indicative of collision risks, this pattern serves as a prophylactic measure against operational inefficiencies, material wastage, or inadvertent damage to equipment. The visual feedback consists of a notification user interface (UI), where the user is provided with an explanation of their mistake, accompanied by a change in the color of the interactable to a distinct hue (such as red or yellow). The haptic feedback will inform the user, using the vibrations that the controllers have. Specifically, it will trigger vibrations in the controller that the user is interacting with (Figure 3.4).

Figure 3.4: Collision error occurred because the user touched the alcohol bottle with the syringe.

3.3.5 Angle Pattern

Inserting objects at specific angles can be critical in various contexts, spanning from manufacturing and assembly processes to surgical procedures and scientific experiments. Precise insertion angles are often vital for ensuring proper alignment, functionality, and safety.

In manufacturing and assembly, inserting components at specific angles may be necessary to guarantee seamless fitting, minimizing gaps, and ensuring structural integrity. This is particularly crucial in industries such as automotive, aerospace, and electronics, where precision is of utmost importance.

In the **Angle** pattern, the user is alerted when a certain amount of time has elapsed (e.g. more than 5 seconds) while attempting to insert an object at an incorrect angle. The user is provided with a visual error notification, indicating the incorrect manner in which they are attempting to insert an object. Additionally, a sound effect is played to further emphasize the error (Figure 3.5).

Figure 3.5: Incorrect positioning of the RAM at an improper angle results in triggering the Angle Error, prompting the system to provide visual feedback to the user.

Chapter 4

Data Collection

4.1 Summary

This section's primary objective is to assist the player in connecting the errors made with the time and place they occurred. We collect various data, including the user's position, fixation points, hand interactions, triggered errors, and the actions in which errors were performed. All of this data is intended to help the user comprehend where and when an error occurred, aiding in their understanding of their mistakes and facilitating the debriefing process. Lastly, all these data will be visualized after the user finishes the scenario, in the Mind Map UI (Chapter 5), providing the user a qualitative post-analysis process with every information needed.

4.2 Data Classes

For the implementation, we created two static classes: the *ErrorDataPool* which is responsible for saving and handling data for the error feedback types and the *FixationDataPool* for the fixation points on the grabbable gameobjects.

The *ErrorDataPool* is a static class that has a static list called *errorsData* and two static members called *AppendError* and *ActionExists*. The *errorsData* member is a list of value-type *ErrorFeedbackStruct* that holds information about the type of error that were made in an action, the action name and the user's position in which an error was triggered. The *AppendError* method appends the data in the static list when the action is initialized and when an error is triggered. The *ActionExists* serves as a utility function checking if an error type has already been triggered in the current action. The *ErrorFeedbackStruct* functions as a variable type, linking actions with the corresponding error types that were triggered. In particular, it stores data about the type of error that was triggered, the action name where it triggered the aforementioed error and the user's position and rotation. The UML for the *ErrorDataPool* and the *ErrorFeedbackStruct* can be seen in the Figure 4.2.

The *FixationDataPool* is a static class tasked with storing information about the fixation and interaction data of each interactable object spawned by the action prototypes. This class has a static dictionary called *fixationData* with key-type string and value-type *FixationDataStructure*. The *AddFixationData* and *AddInteractionData* members are invoked when we add fixation and interaction data to the aforementioned dictionary. The *FixationDataStructure* functions as a variable type and includes parameters where fixation and interaction data can be stored. It includes fields for storing the name of the interactable, the total number of fixations spawned onto it, the timestamp of the first fixation since the spawn of the interactable, the duration the interactable was held in hand, and which hand the user interacted with the object. The UML for the *FixationDataPool* and the *FixationDataStructure* can be seen in the Figure 4.2.

The visualization of these data are taken care of by World Map UI which we

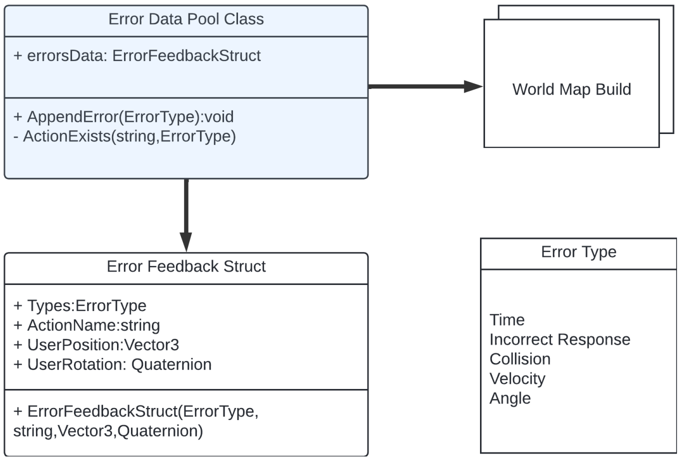


Figure 4.1: The UML of the Error Data Pool

described in Chapter 4.

4.3 Collecting and storing the Data

In Section 4.2, we examined and illustrated the categories of error, fixation, and interaction data. In the following section, we will detail the methods employed for collecting and storing this data.

As discussed in previous chapters, we utilize the MAGES SDK to develop a training simulation. In MAGES, creating a training simulation involves creating action nodes. An action node represents a training step that a user must complete to proceed to the subsequent step. Error patterns described in Chapter 3 can be added on each action node using a no-code solution described in Chapter 6. Therefore, we store data when a user performs an action, makes an error, or interacts with an interactable object. An example can be seen in the Figure 4.3 where the creator of the training scenario adds the "Time" error feedback in a remove action called "Remove the RAM1".

When a user performs an **action**, we store the fixation points on the interactables that the user gazed at, along with the interaction data corresponding to their actions. We obtain the interaction data through an event invocation triggered when a user holds an interactable object. The interaction data we store includes the hand that interacted with the item, and the duration of the interaction. When a user makes an **error**, we record the user's position, the user's rotation, the type of error committed, and the action name. The above data are the visualized at the end of the training scenario, on our mind map. More about this can be found in Chapter 5 Section 5.2.

Furthermore, we record the type of error that occurs and the total number of

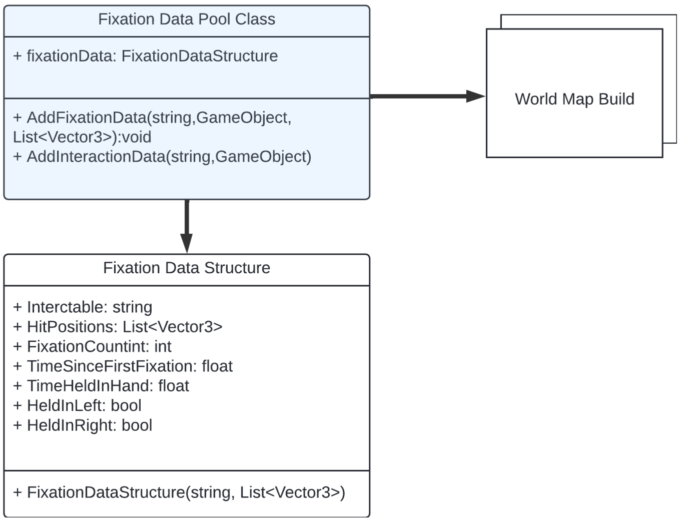


Figure 4.2: The UML of the Fixation Data Pool.

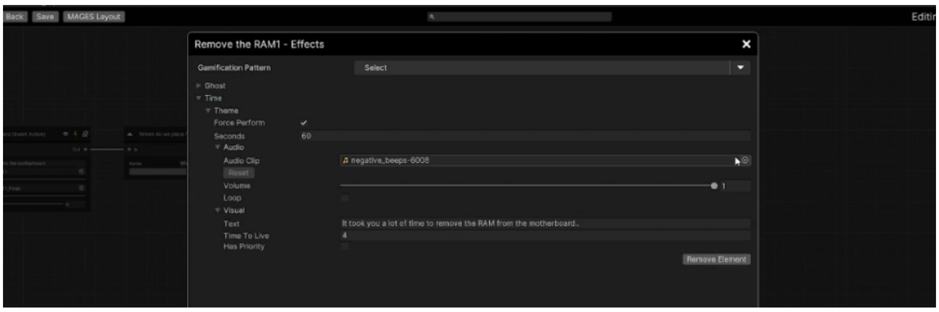


Figure 4.3: Our patterns using MAGES no-code authoring.

Figure 4.4: (Left) Runtime visualization of the frequency of error occurrences. (Right) Wrist UI displaying the total number of errors committed.

errors the user commits during a single play-through. These data are accessible to the user during training within the simulation, facilitating real-time feedback and enabling immediate adjustments of the user's attention and movements. The availability of such detailed information enhances the training experience by allowing users to monitor their performance metrics, understand their errors, and refine their actions accordingly. The visualization of these data is directly accessible to the user. By turning his left hand with his palm facing upwards, the user visualizes a UI with all error patterns along with their frequency. On his right hand, a wrist-watch-like user interface, allows the user to view the total number of errors committed at any time. The above can be see in the Figure 4.4

Chapter 5

The Mind Map

5.1 What is a Mind Map?

A Mind Map is a simple yet powerful tool for brainstorming thoughts in a way that happens or develops naturally, free from the constraints of order and structure. It visually organizes ideas, facilitating analysis and recall. A Mind Map is a diagram used to represent tasks, words, concepts, or items linked to and arranged around a central concept or subject. Utilizing a non-linear graphical layout, it enables users to create an intuitive framework centered around a core idea. By transforming a long list of monotonous information into a colorful, memorable, and highly organized diagram, a Mind Map aligns with the brain's natural way of processing information.

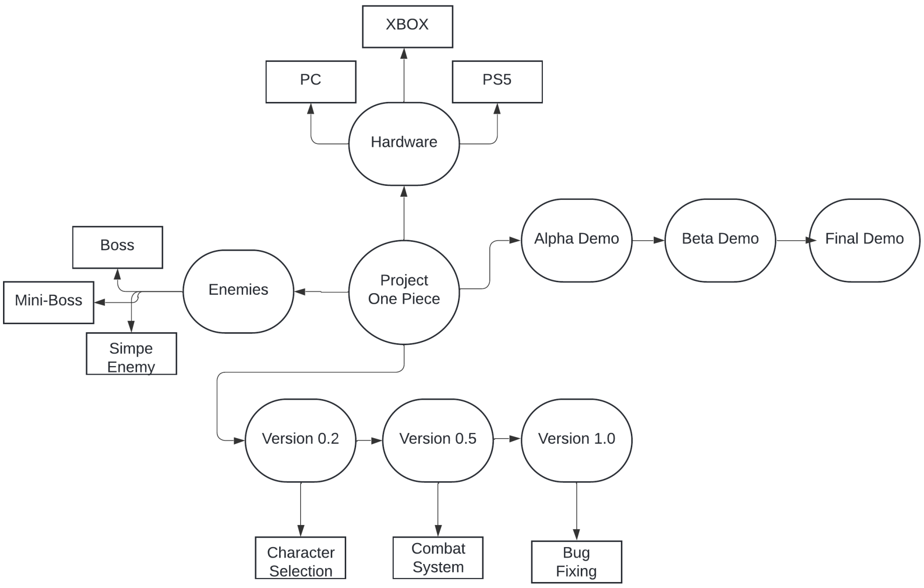


Figure 5.1: An example of a mind map.

Moreover, a Mind Map is an interactive learning method that encourages students to focus more precisely on topics. It transforms learning from a mundane task into an engaging and beneficial activity, thereby enhancing user engagement and comprehension. In Figure 5.1 we can see an example of a mind map for a project named "One Piece".

5.2 Implementation

In our case, we wanted to provide a visualization to the user with all the steps he performed while being inside the virtual training simulation. We chose mind maps for our background theory in order to build an interactive UI that is not boring nor overwhelming. Mind maps help in creating an engaging, visually appealing interface that organizes information intuitively, making it easier for users to understand and interact with complex data.

This highlights the dual benefits of avoiding boredom and preventing users from feeling overwhelmed, emphasizing engagement and intuitive information organization. Thus, we condensed the user's actions, their positions during each action, fixation data points, and interaction data into an interactive UI, having the mind map theory as a base.

Figure 5.2: Our interactive UI, instantiated during the debriefing session of the scenario.

In Figure 5.2, the thesis mind map is illustrated. This UI functions as an administrator panel, allowing the user to interact with buttons, sliders, and dropdown menus to locate the necessary data. These interactive components collectively serve as navigational tools, empowering the user to traverse through the manifold layers of data and informational nodes encapsulated within the thesis framework. By harnessing the intuitive affordances of user-centric design principles, this UI cultivates an environment conducive to seamless information retrieval and manipulation, fostering a symbiotic synergy between human cognition and technological mediation.

At the center of the mind map, all actions performed by the user are displayed. Each action is represented by a UI element that shows the action name and includes interactive buttons. These buttons allow the user to view the position where an error occurred and the interactable object associated with that specific action. Additionally, the user can identify error patterns through designated icons. All the 5 error patterns from Chapter 3 are displayed. If the user has made an error, the icon will be red. Otherwise, the icon will be white. Furthermore, hovering over each of the five error icons will display a text annotation explaining the meaning of each icon. A scroll bar is located at the bottom of the UI, indicating the length of the mind map. It can be used to navigate and scroll through the actions the user has performed.

Figure 5.3: The centric part of the "World Map" UI. We can see that the collision error has been invoked in the "Insert Motherboard" action.

On the right side of the mind map, users will find a few buttons that they can interact with. As illustrated in Figure 5.4, users can find zoom in and zoom out buttons. Pressing these buttons will adjust the zoom level of the central part of the mind map accordingly. Moreover, by pressing the top button the user can spawn or despawn the eye-fixation points. Lastly, with the "Spawn Object" button, the user can spawn the interatable object of the selected action.

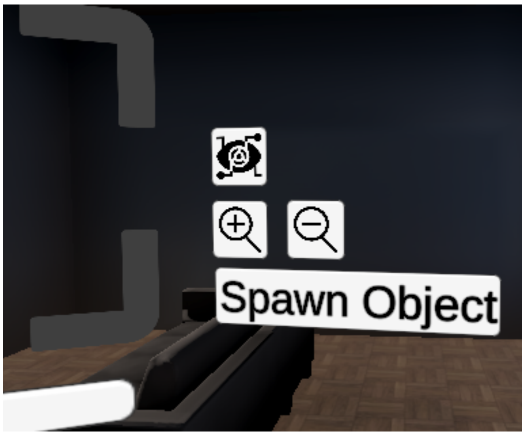


Figure 5.4: On the right side of the mind map UI, buttons are provided for zooming in and out on the actions UI in the central part. Additionally, the user can enable and disable the eye-fixation point spheres. Lastly, the user can spawn the interactable object associated with each action.

In this UI, located on the right side of our mind map, users can observe the interaction data presented in a straightforward and informative manner. At the top of the UI, the name of the interactable item is displayed. Directly below, users can find the name of the action associated with the interactable item. Finally, all relevant data, including fixation and interaction information linked to the specific action, is provided. An example can be seen from the Figure 5.5

Figure 5.5: On the left side, the interaction data UI.

Chapter 6

The Authoring Tool

6.1 Summary

This section will discuss the expansion of a no-code VR training authoring solution, MAGES SDK, in order to apply the discussed error patterns from Chapter 3 and enabling data collection from Chapter 4. We'll outline how we extended the no-code solution and the rationale behind our choice to do so.

6.2 The no-code authoring tool

Our solution builds upon the MAGES SDK[1]. One of the reasons we opted for MAGES as our SDK of choice because it already includes a pre-existing no-code solution, called "Effects", for gamification patterns like ghost, label, and arrow. Leveraging the same no-code solution, we will implement and bolster our error feedback patterns into it and adding data capturing in certain interactable objects of action data.

Moreover, MAGES also includes a functionality called "Steps". In "Steps", the developer adds functions that will be invoked. The invocation can happen when an action is initialized, performed or undoed. In those steps we empower the developer to enable the data collection of the interactable object of the contextual action.

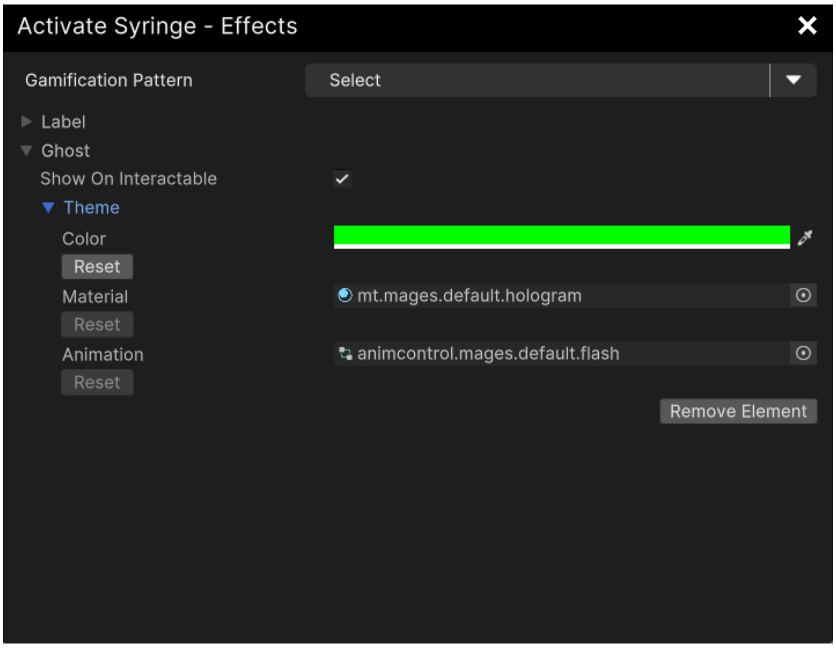


Figure 6.1: The no-code solution to set a gamification pattern in an action

[1] https://assetstore.unity.com/packages/tools/game-toolkits/mages-sdk-create-cutting-edge-medical-xr-simulations-277259

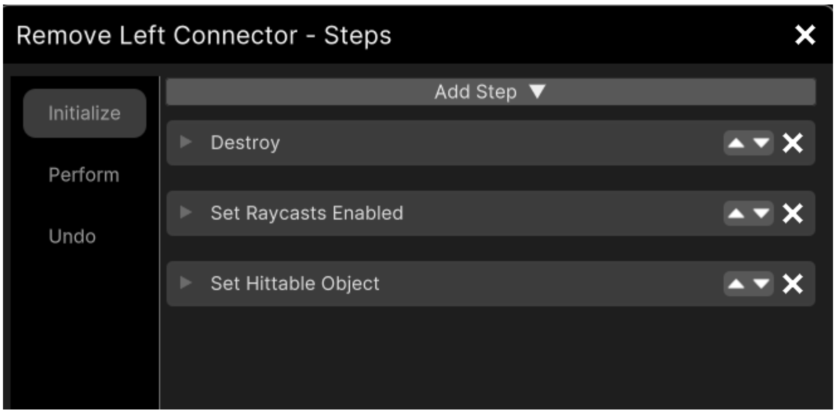


Figure 6.2: The "Steps" functionality where the user can set functions to be invoked when an action is Initialized, Performed or Undoed.

Below, we will show the no-code implementation of our 5 Error Feedback Patterns as well as the Data Collection.

6.2.1 Error Feedback Patterns

The user can set the error feedback patterns using the "Effects". This can be found when configuring an action type in the Scene Graph editor window. Not all error feedback patterns are applicable to every action. Some error feedback patterns may not provide meaningful assistance or help the user understand what they did wrong. Below, in Table 6.1, we outline the error feedback types available for each action.

	Time	Velocity	Incorrect Response	Collision	Angle
Insert Action	Yes	Yes	No	Yes	Yes
Remove Action	Yes	Yes	No	Yes	Yes
Use Action	Yes	Yes	No	Yes	No
Question Action	Yes	No	Yes	No	No
Trajectory Action	Yes	No	Yes	No	No
Activate Action	Yes	Yes	No	Yes	No

Table 6.1: Table indicating the error feedback types corresponding to each action.

An example of how to configure an error feedback pattern can be seen in the Figure 6.3 and Figure 6.4. In the action "Insert RAM" we have set a "Collision" and a "Time" error feedback pattern.

For the "Time" type (Figure 6.3), developers can specify whether the action will be automatically triggered when the time threshold is reached. Additionally, they can set the time threshold, specify the sound to be played along with its volume, define the notification text to be displayed its time to live.

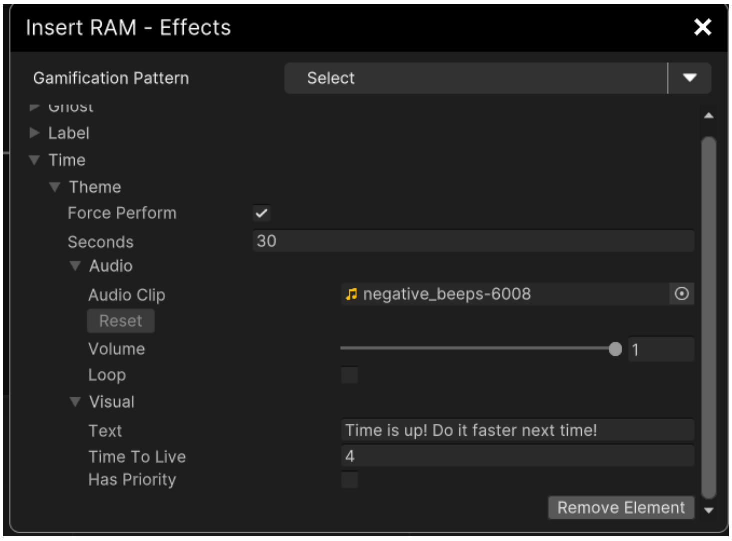


Figure 6.3: The configuration of "Time" error feedback pattern.

For the "Collision" type (Figure 6.4), the developer can specify the notification text to be displayed and its time to live. They can also define the sound to be played when the error is triggered, determine if there will be haptic feedback, and set the strength and duration of the feedback. Lastly, they can configure how the collision error type will be triggered and specify the color the interactable will change to..

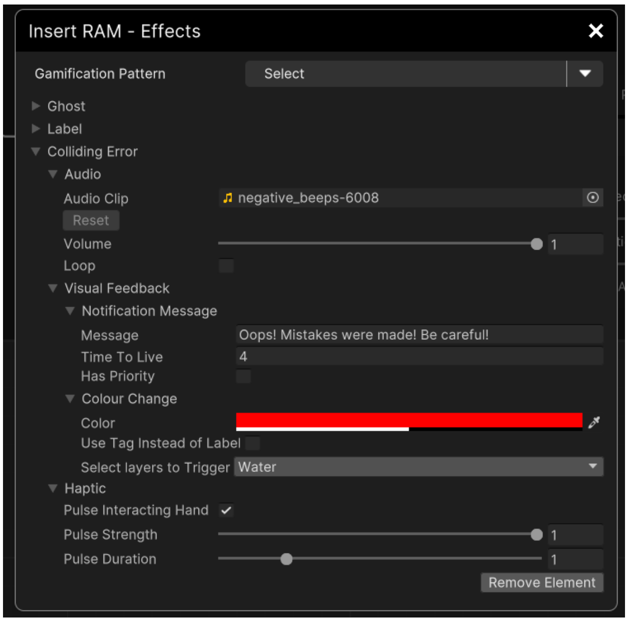


Figure 6.4: The configuration of "Collision" error feedback pattern.

After connecting every error feedback type with each action, the developer can configure an action to have an error feedback type. First the developer needs to open the "Scene Graph Editor" window. After that, he can either create an action type, such as insert action or remove action, or find an already created action. Then, the developer needs to click the eye icon which refers to the effects, as seen in Figure 6.5. In the "Effects" window, developers can configure either gamification patterns, aiding players in identifying specific elements, such as which item to interact with, or error feedback patterns. As a configuration example we can see in Figure 6.6 we can see that the developer set the "Time" error feedback type. From the configuration the player will have 50 seconds to complete the insert action. If 50 seconds passed and the action was not completed, then the action will be skipped, playing a sound and having a notification message informing the user.

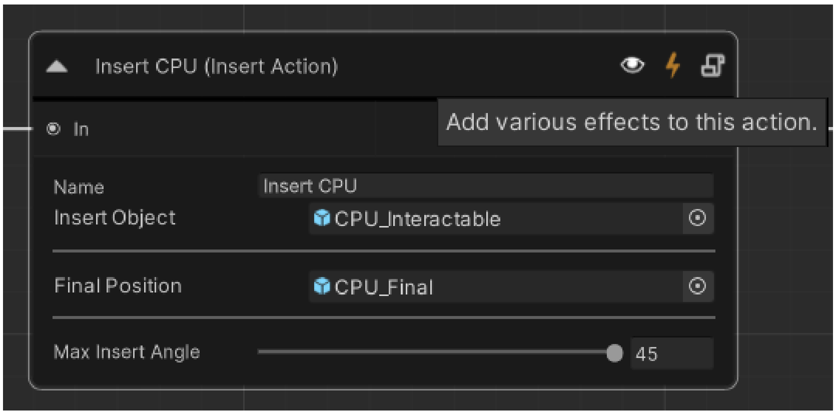


Figure 6.5: The "Effects" button of the Insert CPU action.

In the subsections below you can find the configurations you can do while using the no-code solution of the error patterns.

6.2.1.1 Time Error Feedback Configuration

The "Time" error feedback includes both visual and auditory components. The training simulation creator can configure the time threshold and determine whether the action will be performed automatically when the time limit is reached. Moreover, for the visual feedback, configurations can be applied about the notification UI message. The text and duration of the UI are configurable. Additionally, if the creator enables the "Has Priority" property, the UI will be displayed even if another UI is already active. For the auditory feedback, the creator can customize the audio clip that plays when the user makes this error. The volume is adjustable, and the sound can be set to play on a loop if desired. The UI configuration can be seen in Figure 6.3

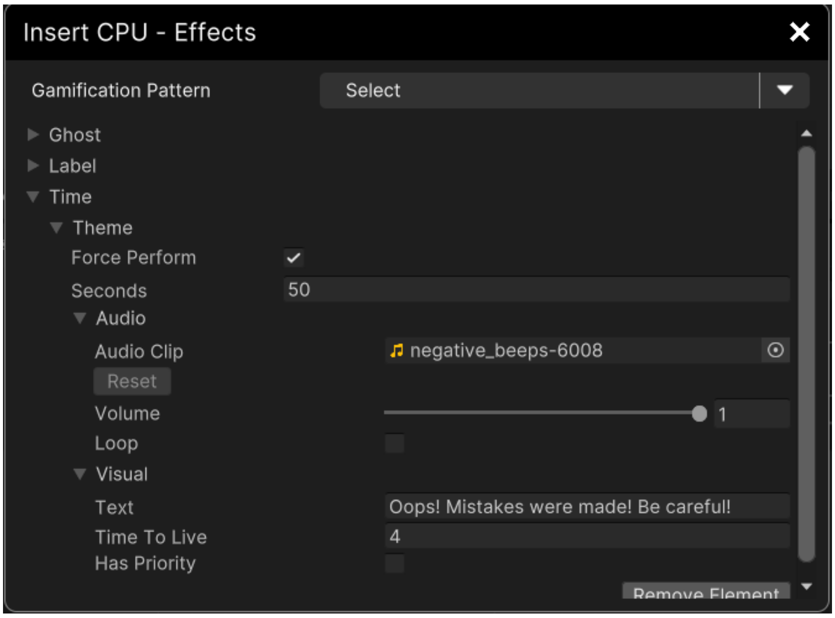


Figure 6.6: The "Effects" button of the Insert CPU action.

6.2.1.2 Angle Error Feedback Configuration

The "Angle" error feedback includes only visual component feedback. The creator of the training simulation can configure the visual feedback for the notification UI message. Both the text and the duration of the UI are customizable. Additionally, if the "Has Priority" property is enabled, the UI will be displayed even if another UI is currently active. The UI configuration can be seen in Figure 6.7

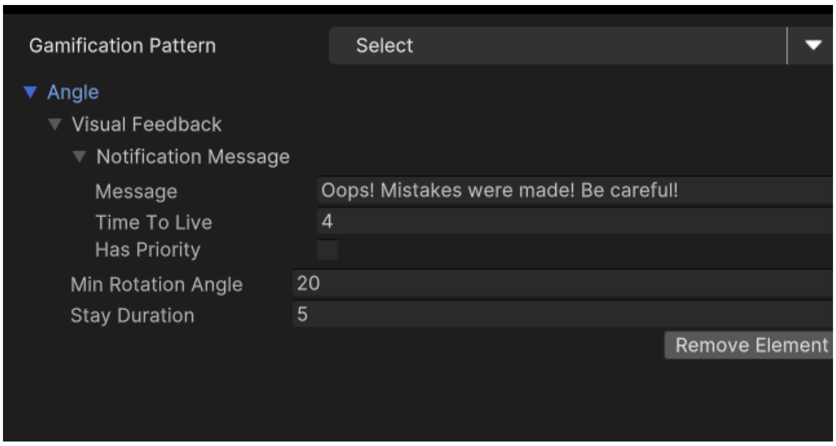


Figure 6.7: The configuration UI for the "Angle" error type.

6.2.1.3 Velocity Error Feedback Configuration

The "Velocity" error pattern includes visual and haptic feedback. For the visual feedback, the notification UI message can be configured. Both the text and the duration of the UI are customizable. Additionally, if the creator enables the "Has Priority" property, the UI will be displayed even if another UI is currently active. Regarding haptic feedback, the creator can customize the shake intensity and duration on the controllers can be configured. This is applicable only when the "Pulse Interacting Hand" option is enabled. The configuration UI can be seen in Figure 6.8

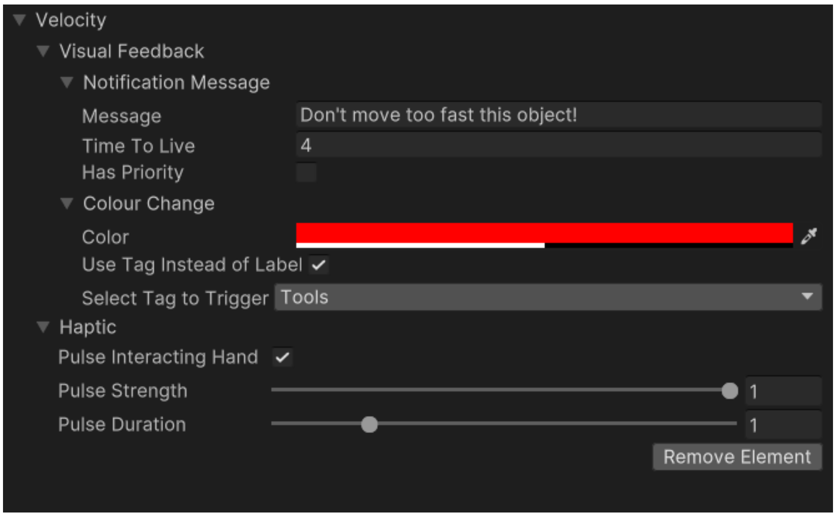


Figure 6.8: The configuration UI of the "Velocity" error feedback.

6.2.1.4 Incorrect Response Error Feedback Configuration

For the "Incorrect Response" feedback, users receive both visual and auditory cues. While visual feedback is provided, it is not configurable by the creator; when a user selects an incorrect answer, the selected answer automatically turns red. When a user selects the correct answer, the selected answer automatically turns green. For the auditory feedback, the creator can customize the audio clip that plays when the error occurs. The volume can be adjusted, and the sound can be set to play on a loop if desired. The configuration UI can be seen in Figure 6.9

6.2.1.5 Collision Error Feedback Configuration

For the "Collision" feedback, user receives visual, auditory and haptic feedback. The notification UI message for the visual feedback can be configured, allowing customization of both the text and the duration. Additionally, if the "Has Priority" property is enabled, the UI will be displayed even if another UI is active. Furthermore, the creator can set the color that the interactable object will change

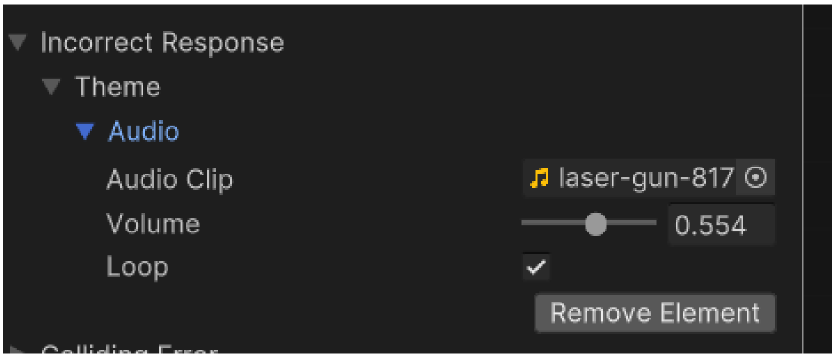


Figure 6.9: The configuration UI of the "Incorrect Answer" error feedback.

to upon colliding with an unwanted object. The unwanted object can be specified by either Tag or Layer. The auditory feedback can be customized by the creator, allowing for the selection of the audio clip that plays when an error occurs. The volume is adjustable, and the sound can be configured to play on a loop if desired. Lastly, for the haptic feedback, the shake intensity and duration on the controllers can be configured. This is applicable only when the "Pulse Interacting Hand" option is enabled. The configuration UI can be seen in Figure 6.4

6.2.2 Data Collection Configuration

Enabling the data capture for the interactable objects in each action takes place in the "Steps" functionalities from the "Scene Graph Editor" window of MAGES SDK.

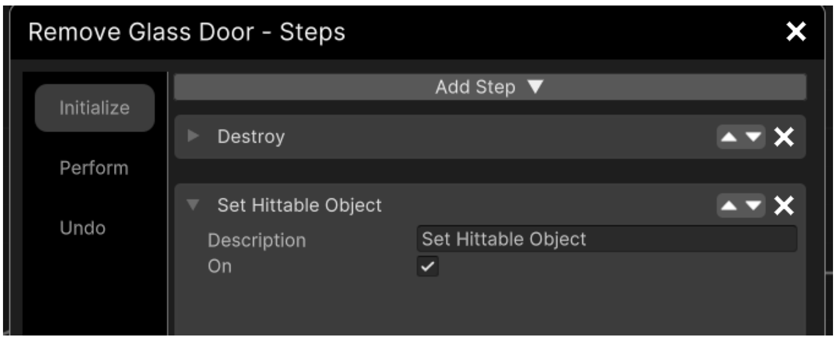


Figure 6.10: The "Step" "Set Hittable Object" that enabled the data collection for the grabbable object in the "Remove Glass Door" action.

To enable data collection for the interactable of the corresponding action, the developer can access the "Steps" from the "Scene Graph Editor" window and create a new action or find an existing one. Then, by pressing the lightning button, the developer will open the "Steps" window. Within this window, functions can be invoked in the Initialize, Perform, or Undo stages of the corresponding action. To enable data collection, the developer needs to add the ”Step” labeled ”Set Hittable Object,” as shown in Figure 6.10.

Chapter 7

Evaluation

In this chapter, we will discuss the three evaluation processes we conducted in cooperation with the Human Computer Interaction (HCI) Laboratory of ICS-FORTH[1]. We assessed the framework of this thesis in terms of design and efficiency, the authoring tool's usability and its effectiveness in developing and improving training simulations, and the error feedback patterns with respect to task load, mental and physical demands, and user frustration. For this purpose, we conducted two evaluation processes:

1. A **Heuristic Evaluation** was conducted to assess the framework of the thesis. The evaluators examined the utility of the World-Map UI. Finally, the evaluators assessed the visualization of interaction and eye-fixation gaze data.
2. A **Expert-based Evaluation** was also conducted featuring various methods and instruments to assess its learnability, user experience, and potential uptake by end users. The evaluation focused on how intuitive the no-code authoring tool is for new users, examining the steps they would naturally take.
3. A **Task load Evaluation** tool to measuring perceived workload for the 5 error feedback types. The evaluation assesses the effectiveness of the 5 error feedback patterns in helping users understand the nature and timing of their errors.

To facilitate the evaluation, five user-experience (UX) experts for the heuristic evaluation and five expert developers for the second part of the evaluation. These experts were selected based on their prior experience in XR technologies and in Unity game engine. Additionally, one individual from the development team served as the facilitator for the process. In the sections below we will analyze the evaluation process as well as showcase the data we collected.

7.1 Framework - Heuristic Evaluation

In this section we will describe the heuristic evaluation process we conducted as well as the results we gathered from 5 User Experience (UX) experts in VR applications. These individuals were considered ideal candidates for our user group, given their expertise and background in the field.

7.1.1 Heuristic Evaluation

Heuristic evaluation is a method of usability analysis where a number of evaluators are presented with an interface design and asked to comment on it [42]. In this method, a small group of evaluators (three to five individuals) is invited to examine an interface and assess its compliance with recognized usability principles (heuristic

[1] `https://www.ics.forth.gr/hci/`

rules) [40]. We used the 10 rules that describe the common properties of easy-to-use interfaces taken by Jakob *et al.* [41]. These rules are:

1. **Visibility of System Status**: The design should always keep users informed about ongoing processes by providing appropriate feedback within a reasonable timeframe.

2. **Match Between the System and the Real World**: The design should communicate in the users' language by employing words, phrases, and concepts that are familiar to them, rather than using internal jargon. It should adhere to real-world conventions, presenting information in a natural and logical sequence.

3. **User Control and Freedom**: Users frequently make mistakes and require a clearly marked "emergency exit" to abandon the unintended action without having to undergo a lengthy process.

4. **Consistency and Standards**: Users should not be left uncertain about whether different words, situations, or actions have the same meaning. Adhere to platform and industry conventions.

5. **Error Prevention**: While good error messages are important, the most effective designs proactively prevent problems from arising. This can be achieved by either eliminating error-prone conditions or by detecting potential errors and providing users with a confirmation option before they proceed.

6. **Recognition Rather than Recall**: Minimize the user's cognitive load by making elements, actions, and options visible. Users should not need to remember information from one part of the interface to another. Information necessary for using the design, such as field labels or menu items, should be visible or easily accessible when needed.

7. **Flexibility and Efficiency of Use**: Hidden shortcuts, accessible to experienced users, can expedite interactions while remaining hidden from novices. This approach allows the design to accommodate both inexperienced and proficient users, offering customization options for frequent actions.

8. **Aesthetic and Minimalist Design**: Interfaces should exclude irrelevant or infrequently needed information. Each additional piece of information in an interface competes with relevant information and reduces its relative visibility.

9. **Help Users Recognize, Diagnose, and Recover from Errors**: Error messages should be written in clear and straightforward language, avoiding the use of error codes. They should accurately identify the problem and provide constructive suggestions for resolving it.

10. **Help and Documentation**: Ideally, the system should be intuitive enough to not require additional explanation. However, documentation may be necessary to assist users in understanding how to accomplish their tasks.

Based on the aforementioned 10 rules, we will gather our data and utilize these rules as guidelines. In the following subsection, we will discuss our evaluation process, specifically focusing on our heuristic evaluation.

7.1.2 The Evaluation Process

In this evaluation process, we devised a compact training scenario comprising 25 sequential actions centered around PC assembly. Participants were tasked with inserting, removing, and utilizing various PC components to achieve correct assembly. Prior to initiating the assessment procedure with every expert, we explained to them the theoretical context of the master's thesis, the tasks ahead of them in the simulation and that they will have to make mistakes in order to give us feedback for the 3 output modalities we use for the error feedback patterns from Chapter 3.

We categorized the feedback we received from the experts using the 10 rules by Jakob *et al.*[41]. In Table 7.1 you can see the feedback we received, from the heuristic evaluation, from the 5 experts. Moreover, in Figure 7.1 you can see the heuristic evaluation results per guideline.

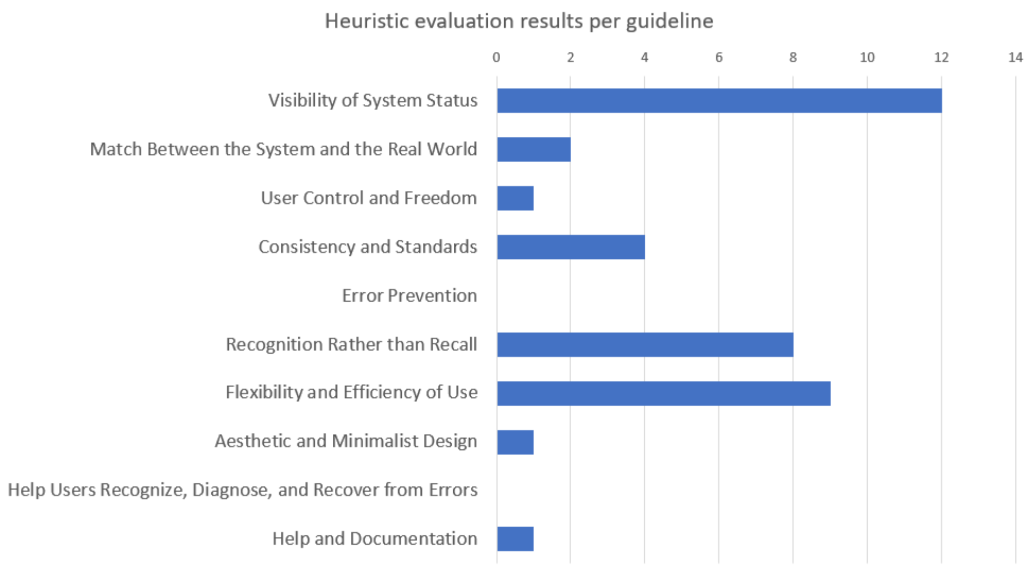


Figure 7.1: The heuristic evaluation results per guideline.

Rule	Results
Visibility of System Status	Many experts provided feedback on the World-Map UI, suggesting various improvements such as adjustments to the color scheme, button placement, addition of a title, and other UI enhancements. Specifically, the absence of a title on the world-map UI makes it challenging for users to comprehend what they are viewing. Additionally, the buttons positioned on the left side of the world-map were not intuitive for users to understand that they influence the World-Map. Furthermore, feedback was provided regarding the UIs of the left and right hand. Specifically, the font size needs adjustment, and the background color should be modified. Additionally, the left-hand UI frequently appeared with even slight rotations of the left hand, making interaction difficult due to visibility issues.
Match Between the System and the Real World	When turning the virtual palm up during an assembly task, the analysis of errors presented may interfere with the instructions or the interactable objects. Consider activating the left palm UI upon a specific action. Moreover, a couple of syntax and style mistakes (eg. Typos) were noticed by the evaluators.
User Control and Freedom	For the Left-palm UI, evaluators commented that it appears too fast.
Consistency and Standards	Many experts commented that other areas of the UI (world map) icons are used for each error type. They should be used consistently throughout the interface, including the error statistics shown on left-palm-up action. Moreover, the experts noted that the text 'Total errors' shown on the wrist employs a different font family than the error details components.
Error Prevention	We did not received any feedback in this category.

Recognition Rather Than Recall	Regarding the error feedback patterns, the evaluators suggested implementing a distinct color code for each type of error feedback would benefit users. Additionally, the evaluators commented that the error feedback does not persist with respect to angles.
Flexibility and Efficiency of Use	There are errors that may have detrimental effects (e.g. breaking something) and errors that have less severe effects (e.g. placing something upside down). The system treats all types of errors in the same manner. Providing different ways of addressing varying levels of errors according to their severity would be beneficial for the user. Although the user may make the same error multiple times, it is only counted once. Therefore, through the dashboards it will not be possible to understand where the user faced difficulties repeatedly. Additionally, evaluators noted that the slider in the World-Map UI is not convenient for navigating through the actions in the mind-map.
Aesthetic and Minimalist Design	The text displayed on the wrist UI of the right-hand appears stretched.
Help Users Recognize, Diagnose, and Recover from Errors	We did not received any feedback in this category.
Help and Documentation	A short tutorial would be useful inside the VR training simulation. How to move around and how to interact with the interactable items.

Table 7.1: Rules and results from the heuristic evaluation.

Furthermore, the classified feedback provided by the evaluators into four severity categories. In total, we identified 38 issues from the evaluators: 4 were categorized as aesthetic, 19 as minor, and 15 as major. You can see the categorization of the evaluators' feedback in Table 7.2. A pie chart illustrating the distribution of these issues can be observed in Figure 7.2.

Severity	Results
Not an error	We did not received any feedback in this category.
Aesthetic	For the right hand, the text "Total errors" displayed on the wrist uses a different font family than the components detailing the errors. Additionally, the "Total errors" text appears stretched. Regarding the palm UI on the left hand, errors could be categorized. Furthermore, the UI appears too quickly.
Minor	Most evaluators provided feedback on the world-map UI. Specifically, they noted that the position of the buttons affecting the world-map was not intuitive. Additionally, the slider used to navigate through the actions of the world-map was found to be very inconvenient. Furthermore, the absence of a title was highlighted as a significant issue. Furthermore, a common piece of feedback was that the left-palm UI should include icons for the error feedback, ensuring consistency with the visualization on the world-map.
Major	For the World-Map UI, many evaluators did not understand its purpose. The zoom-in/zoom-out buttons are dissociated from the area they control and should be placed closer to the world map, such as below it. Additionally, the "eye-fixation" icon is not easily understood and would benefit from a hover text or label for clarification.
Catastrophic	We did not received any feedback in this category.

Table 7.2: The severity categories with the results from the heuristic evaluation.

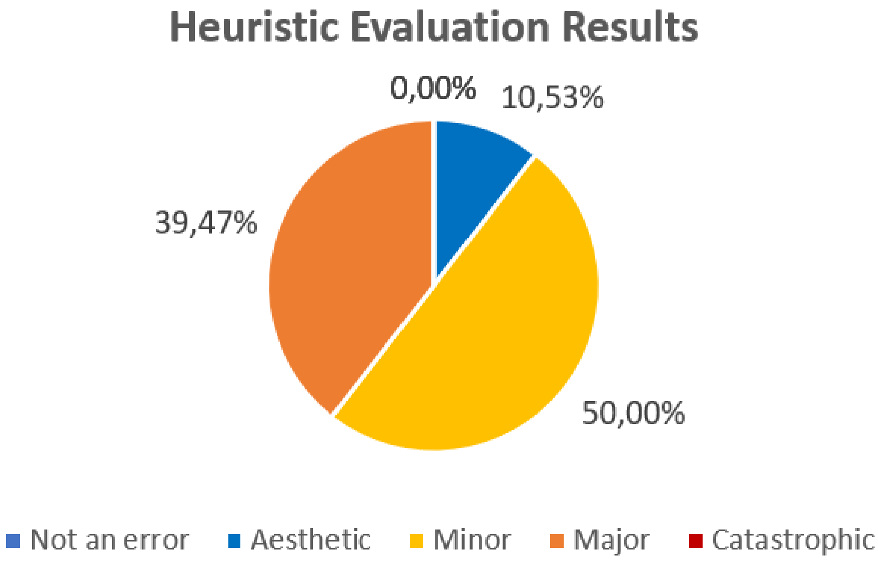


Figure 7.2: The heuristic evaluation results.

7.2 No-code Authoring Tool - Expert-based Evaluation

In this section, we discuss the expert-based evaluation conducted for the no-code authoring tool used to implement our error feedback patterns. Specifically, we involved five experts with proficiency in the Unity game engine and tasked them with completing four steps to add and configure error feedback patterns within MAGES SDK actions.

7.2.1 Cognitive Walthrough

A cognitive walkthrough is a method utilized to assess a system's ease of learning from the viewpoint of a new user[19]. A cognitive walkthrough is conducted in a workshop environment where user tasks to be assessed are pre-defined. One individual assumes the role of facilitator, while all other participants act as evaluators, providing insights into how a typical user might interact with the interface and respond in specific scenarios. During the evaluation of a specific task, the facilitator executes the task and pauses at each new screen or distinct step in the interaction. Moreover, Blackmon *et al.*[6, 29] proposes four key questions for assessors to use during a cognitive walkthrough:

1. Will the user try and achieve the right outcome?

2. Will the user notice that the correct action is available to them?

3. Will the user associate the correct action with the outcome they expect to achieve?
4. If the correct action is performed, will the user see that progress is being made towards their intended outcome?

7.2.2 The Evaluation Process

In this evaluation process, we involved five experts developers proficient in the Unity game engine. We provided an overview of the master's thesis and provided details about the functionalities to be evaluated. Once they had no further questions, we assigned them four tasks to complete. The participants had to evaluate the no-code solution that adds, removes and configures an error feedback pattern to an action of MAGES. Assistance was not offered while performing a step. The 4 tasks can be seen in Table 7.3.

Step	Question
1	Add a "Time" error feedback pattern in the "Remove Left Connector" action. Configure it by making it force perform with the time limit of 5 seconds. Also, it will display "What took you so long to remove a small plug?" for 4 seconds.
2	Find the "Remove Glass Door" action and add a "Colliding Error" error feedback pattern. Configure the "Colliding Error" error feedback pattern and make it use tag, instead of layer. This will trigger the error feedback. The tag will be the Floor.
3	Find the "Remove Glass Door" (same as Step 2) action and add another "Colliding Error" error feedback pattern. Configure the "Colliding Error" error feedback pattern and make it use tag, instead of layer. This will trigger the error feedback. The tag will be the Tools.
4	Find the "Insert Motherboard" action and add "Velocity" error feedback pattern. Configure the "Velocity" error feedback pattern. Play the audio clip "beep" with a volume of 0.85 and display the text "Be careful with the Motherboard" for 5 seconds.

Table 7.3: The steps and questions of our cognitive walkthrough evaluation process.

After completing each task, participants were asked to rate the difficulty of the task on a scale from 1 (very difficult) to 7 (very easy). The results of these ratings are presented in Figure 7.4. After rating the difficulty of the step, participants were asked to answer 7 questions, from 0 (no) to 5 (yes), that resembles the ones from Blackmon *et al.* [6, 29]. The participants could give extra explanatory comments for each question depending on the specific functionality. These questions are:

1. Will the user interface give clues that show that the function is available?
2. Will the user try to achieve the right effect?
3. Will the user associate the right clue with the desired function?
4. Will the user associate the correct action with the desired effect?
5. Will the user get sufficient feedback to understand that the desired function has been chosen?
6. Will the user get sufficient feedback to understand that the desired function has been performed?

The results are depicted in Figure 7.3. Additionally, Table 7.4 presents the evaluators' comments for each corresponding task.

Task	Comments
1	The effects icon is not easily perceivable, making it challenging for users to predict that feedback patterns can be found there, thus hindering their ability to initiate the action. Additionally, an apply button is missing, further complicating user interactions. The system's usability is predominantly geared towards developers familiar with the MAGES SDK. Furthermore, the concept of variable time to live is not easily comprehensible. It is recommended that the label "time" be renamed to "Time Error" to enhance clarity. When a evaluator-developer selects "time," instead of displaying a select option, the system should present the corresponding action. Initially, eveluators-developers may not understand that their selection is presented in the side menu, which can lead to confusion.
2	The evaluators commented that it would be preferable if the tag or layer were positioned outside the "Color Change" category, as the switch to tag or layer was not anticipated to be represented as a checkbox. Similar to a dropdown UI, this switch alters the text on the side but changes it from tag to layer. Additionally, the layer checkbox does not clearly indicate its availability. The icon should be a plus symbol (+), as the user is consistently adds elements. Moreover, the absence of an apply button is a notable omission.
3	No comments from the evaluators.
4	No comments from the evaluators.

Table 7.4: The steps and questions of our congitive walkthrough evaluation process.

Moreover, we performed a Net Promoter Score (NPS) evaluation[39]. NPS is a

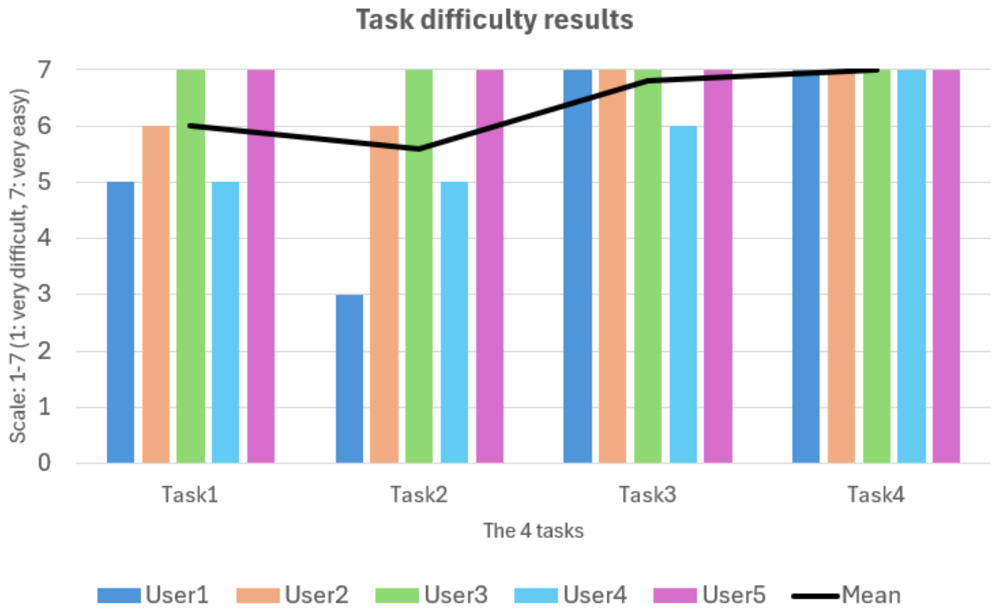


Figure 7.3: The cognitive walkthrough results for each of our 5 users and the mean value.

metric that organizations use to measure customer loyalty towards their brand, product, or service. Our results can be seen in the Table 7.5.

Users	Score
User 1	9
User 2	10
User 3	10
User 4	9
User 5	9

Table 7.5: The NPS results from the evaluation.

Lastly, upon completion of the tasks and questionnaires, we conducted a brief interview with the participants to gather their feedback on aspects they liked and disliked. In particular, participants noted several positive aspects of the system. The evaluators-developers appreciated the ease with which patterns could be added and the capability to incorporate multiple patterns simultaneously. They found the process to be efficient, straightforward, and user-friendly, emphasizing that the configuration options were accessible and intuitive. The system was also praised

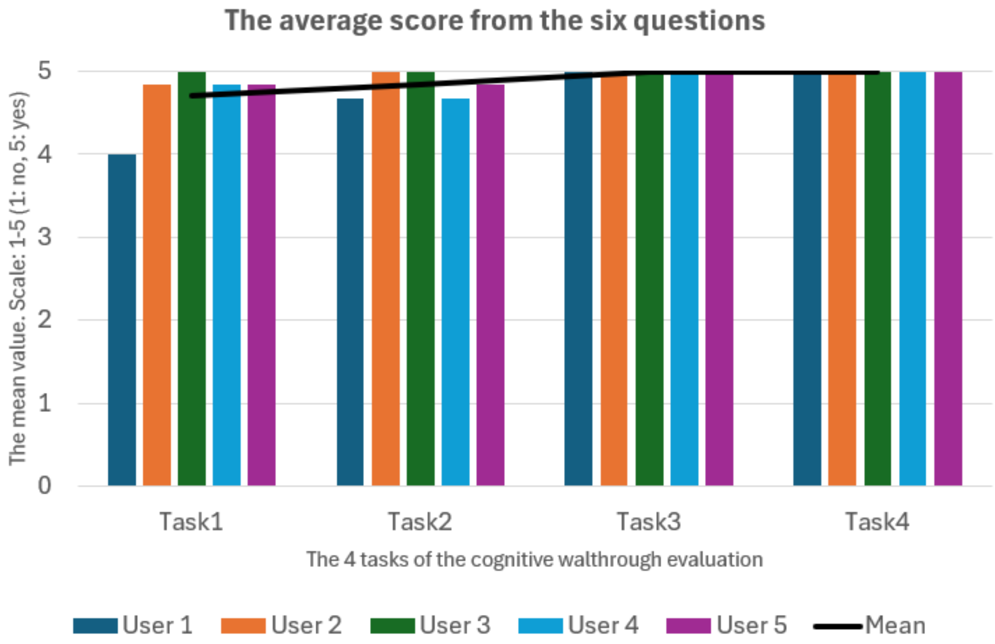


Figure 7.4: The task difficulty results from the expert-based evaluation.

for accelerating the development process. Additional recommendations included the node-based solution, the structure of the UI, the Unity-style theme, and the availability of configuration options across multiple points and parameters. On the other hand, participants provided several suggestions for improvement. They noted the need for clearer labeling of the ”add” icon and expressed concern that naming patterns as ”effects” could be confusing. The absence of an ”apply” button was also mentioned as a drawback. Other feedback included the design of the effects icon, the use of tag or layer in the color change category, and the lack of differentiation between identical error feedback patterns. Additionally, they suggested that button types should be reconsidered, favoring checkboxes over radio buttons or dropdown menus, and that some label wording be revised. Finally, the default name for the beep sound valve was identified as an area for improvement.

7.3 Error Feedback Patterns - Task Load Index Evaluation

In this section, we discuss the impact of error feedback patterns on task load of the user by evaluating the task load index. Specifically, we engaged six VR experts and divided them into two groups. The first group experimented with training scenario A (PC Assembly involving error feedback patterns in each training action). The second group experimented with the training scenario B (PC assembly with no involvement of error feedback patterns).

In this section, we discuss the expert-based evaluation, conducted for the no-code authoring tool used to implement our error feedback patterns. Specifically, we involved five experts with proficiency in the Unity game engine and tasked them with completing four steps to add and configure error feedback patterns within MAGES SDK actions.

7.3.1 NASA Task Load Index

NASA Task Load Index (TLX) is an usability evaluation protocol for measuring perceived workload [25]. Developed by the Human Performance Group at NASA's Ames Research Center, the NASA-TLX measures the mental and physical demands on individuals completing a task. It incorporates a multi-dimensional rating procedure that derives from an overall workload score based on weighted average of ratings on six categories (Figure 7.5):

1. Mental Demand
2. Physical Demand
3. Temporal Demand
4. Performance
5. Effort
6. Frustration

7.3.2 The Evaluation Process

In this evaluation process, we involved 6 experts users in VR. We initialλy provided them with an overview of the master's thesis. The participants had to execute the same VR training scenario as in Section 7.1.2. We adjusted the scenario to increase the difficulty of performing certain training actions, thereby prompting more frequent error feedback. The first group utilized the error feedback patterns, while the second group did not. Our objective was to assess the task load index of the training simulation both with and without the implementation of error feedback patterns.

After completing the entire training simulation, the users were requested to fill the NASA-TLX form (Figure 7.5). Table 7.6 summarizes the TLX index for the group with the error feedback patterns enabled, while Table 7.7 presents the data for the group without error feedback patterns. A comparison of the two data sets reveals significant differences in the 'Frustration' and 'Effort' fields. The second group, that did not receive error feedback during errors, exhibited higher levels of Frustration and Effort related to the training simulation. This is attributed to the

Metric	User 1	User 2	User 3
Mental Effort	15	10	10
Physical Demand	15	10	20
Temporal Demand	0	0	0
Performance	15	10	10
Effort	15	10	15
Frustration	0	0	0
Time	5.17	5.45	5.26

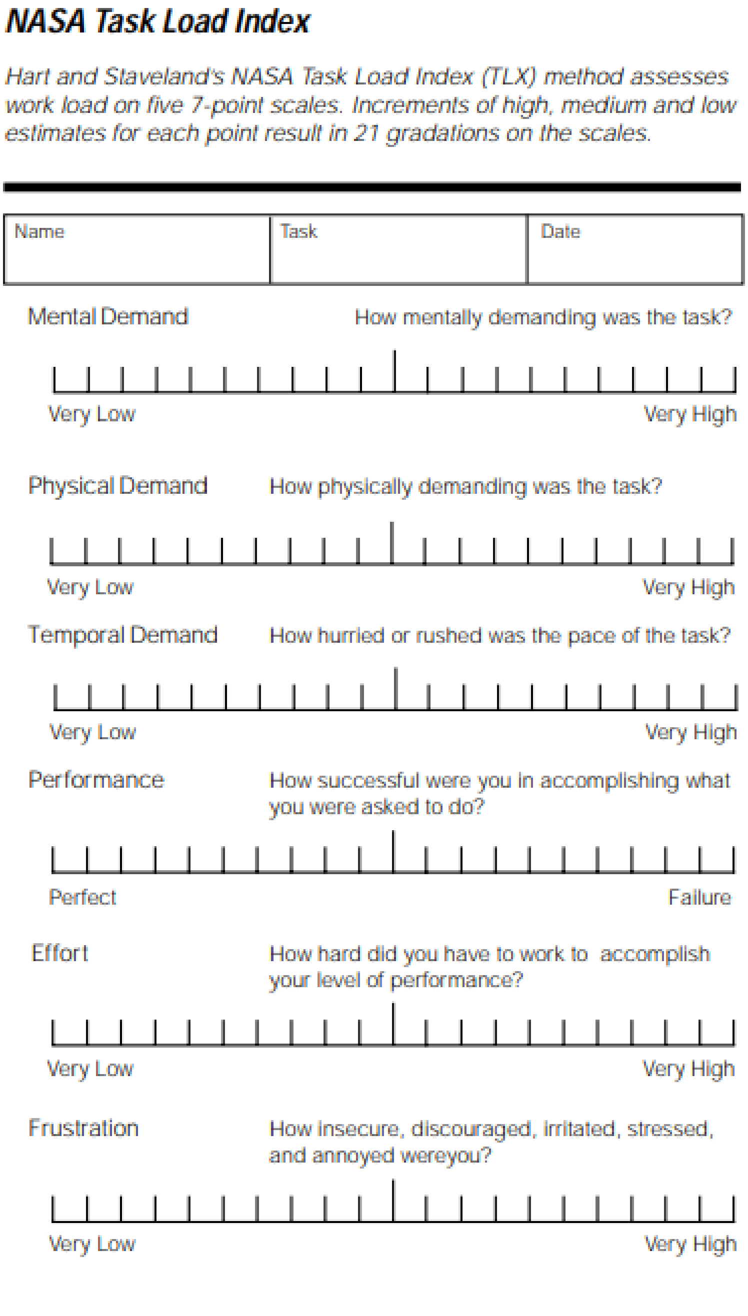
Table 7.6: The NASA TLX index data for the first group (with Error Feedback Patterns enabled during the training simulation)

lack of additional guidance provided to the users on how to insert, remove, or use an interactable object.

Metric	User 4	User 5	User 6
Mental Effort	45	55	45
Physical Demand	15	15	10
Temporal Demand	5	5	5
Performance	20	15	20
Effort	65	65	55
Frustration	70	75	65
Time	6.54	7.22	7.03

Table 7.7: The NASA TLX index data for the second group (without error feedback patterns during the training simulation).

Lastly, we performed a paired T-test analysis from the data of Table 7.6 and 7.7. From the results (Figure 7.6 and Figure 7.7) we see significant differences in "Mental Effort", "Effort", "Frustration" and "Time". Specifically, for "Mental Effort," the t-statistic is -8.315 with a p-value of 0.014. For "Effort," the t-statistic is -10.961 with a p-value of 0.008. The "Frustration" measure has a t-statistic of -24.249 with a p-value of 0.002, and "Time" also has a t-statistic of -24.249 with a p-value of 0.002. Since the p-value is less than 0.05, we reject the null hypothesis, indicating a significant difference in frustration between the two groups.



NASA Task Load Index

Hart and Staveland's NASA Task Load Index (TLX) method assesses work load on five 7-point scales. Increments of high, medium and low estimates for each point result in 21 gradations on the scales.

Name	Task	Date

Mental Demand — How mentally demanding was the task?

Very Low — Very High

Physical Demand — How physically demanding was the task?

Very Low — Very High

Temporal Demand — How hurried or rushed was the pace of the task?

Very Low — Very High

Performance — How successful were you in accomplishing what you were asked to do?

Perfect — Failure

Effort — How hard did you have to work to accomplish your level of performance?

Very Low — Very High

Frustration — How insecure, discouraged, irritated, stressed, and annoyed wereyou?

Very Low — Very High

Figure 7.5: The NASA Task Load Index survey.

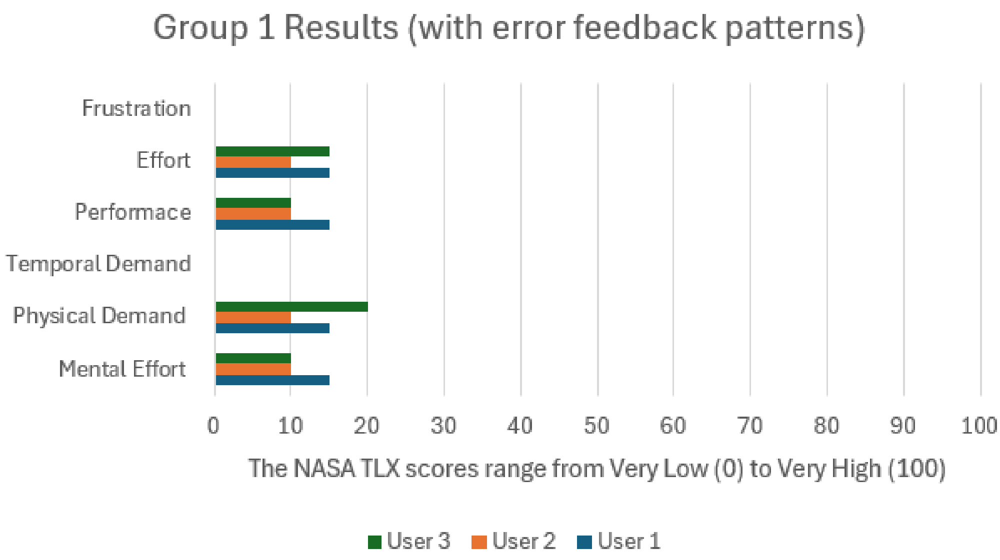


Figure 7.6: Visual representation of the NASA TLX index data for the first group (with error feedback patterns enabled during the training simulation). A lower score signifies a more manageable and less demanding workload.

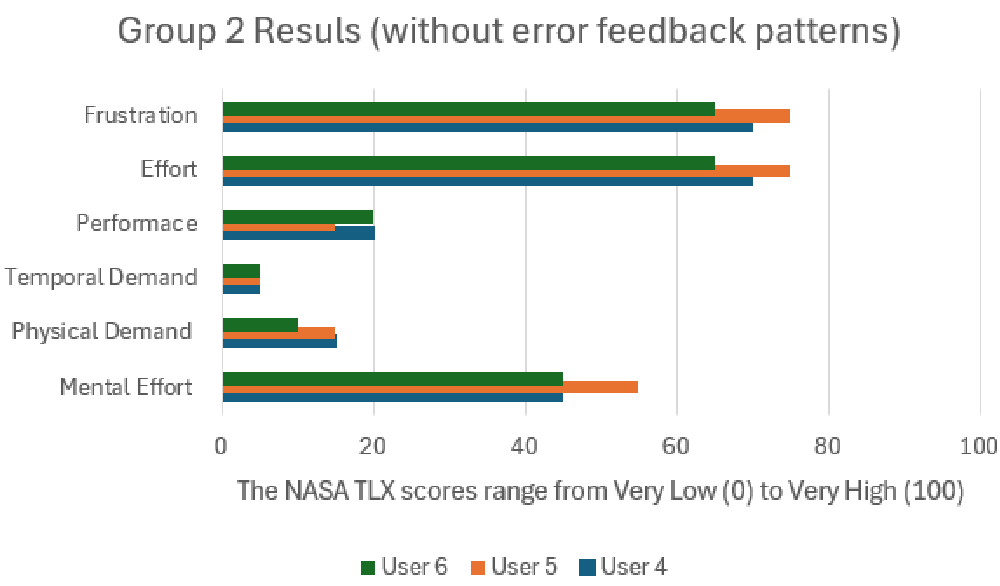


Figure 7.7: Visual representation of the NASA TLX index data for the second group (without error feedback patterns during the training simulation). A higher score signifies a less manageable and more demanding workload.

Chapter 8

Conclusion

8.1 Summary

The main purpose of this Master's thesis was to tackle the ad-hoc implementation of error feedback in VR training scenarios. We created 5 error feedback patterns as mentioned in Chapter 3. In this section, we summarize the achievements and main novelties of the proposed system.
This was possible by utilizing the three output modalities available in commercial HMDs: visual, auditory, and haptic. Each of these five patterns employs at least one of the three modalities to fulfill its intended purpose. Moreover, we incorporated data collection algorithms to enhance the user's experience and provide more detailed insights into their interactions, enabling personalized feedback, improving user engagement, and allowing for continuous optimization of our services based on user behavior and preferences. The data visualizations are rendered in real-time during training, displaying the total number and types of errors made by the user, and highlighting when the scenario concludes, thereby improving the debriefing process. Finally, we apply mind-map theory to illustrate the user's actions along with the corresponding error feedback for each action.

8.2 Comparison of implemented tools

In this section, evaluators-developers we will compare our work on error feedback for VR training scenarios, utilizing a no-code authoring tool, with other related works in both academic research and industry practices.

In the Table 8.1, you can see the comparison between different types of SDK that are all about the training scenarios in VR, either by creating them or by enhancing them.

	MAGES SDK	**Fundamental Core**	**Cognitive3D**	**Our Solution**
Error Feedback Patterns	No	No	No	Yes
No-code Authoring Tool	Yes	No	Yes	Yes
Eye-fixation and Interaction data collection	No	No	Yes	Yes

Table 8.1: Comparison of different types of SDK related with VR training simulations

8.3 Future Work

In this work, Mistake's were made, provided the basis and opened a lot of questions regarding how people learn with error feedback, both in runtime and offline. This thesis presented patterns on how to address the ad hoc nature of these errors. We allowed experienced developers to setup and configure an error feedback pattern. But we do not stop here.

8.3.1 Create more patterns

In this thesis we implemented 5 basic error feedback patterns. The creation of these algorithms was driven by the hypothesis that providing correct error feedback to users could be patterned, thereby addressing the limitations associated with their ad hoc implementation. For future work, we would like to add more patterns. For example, "Boundary" error feedback that triggers when the user moves inside/outside the play area boundary. Another one is "Wrong Interaction" error feedback that triggers when the player attempts to interact with an object that shouldn't.

8.3.2 Headset with eye-gaze integration

Right now the eye-gaze fixation points are taken from a forward vector of the virtual camera. These fixation points are not correct all the time. The users' eyes might not look straight at all times. Thus, we aim to extend our current implementation with an eye-tracking algorithm and test it using a HMD equipped with eye-tracking sensors. Obtaining the true eye-gaze points of the user will allow for accurate data collection of the fixation points, thus, having a more in-depth and correct debriefing session.

8.3.3 Color gradient for the eye-fixation data

Every eye-gaze fixation point is represented as a purple sphere on an interactable object. When many fixation points are to be spawned, the provided detail is not enough to identify in which part of the object with more fixation points. So, we would like to add a color-gradient algorithm [50]. The colormap will be picked from Brewer *et al.* [8] that proposes guidelines for colormaps for different datasets. Utilizing a gradient color scheme may discretize gaze fixation patterns. For instance, by employing a colormap that transitions from blue (representing cold) to red (indicating warmth), we can dynamically adjust the color of individual fixation points based on their significance [56].

8.3.4 Support for multi-sensory devices

The integration of multi-sensory devices with the VR HMD will enhance user experience through the simultaneous engagement of multiple senses. Some examples include haptic feedback systems, olfactory emitters, and auditory enhancements. The synchronous stimulation of visual, auditory, tactile, and olfactory senses facilitates a more profound sense of presence and realism within virtual environments [38]. In particular, adding more sensors to our system will increase the different ways we can provide feedback to the user. That is because these sensors can function as input modality on the training scenario. For instance, by incorporating a haptic VR glove [44], we can measure the strength a user exerts when grabbing an object as an input modality, and provide tactile feedback to specific locations on the VR glove as an output modality.

8.3.5 Improvements from the Evaluation

The evaluation revealed issues of major, minor, and aesthetic severity. We have addressed several items from the major and minor categories. Plans are in place to address all remaining issues in these categories in future development efforts.

Bibliography

[1] Faieza Abdul Aziz and Maryam Mousavi. "A Review of Haptic Feedback In Virtual Reality for Manufacturing Industry." In: *Journal of Mechanical Engineering* 40 (Sept. 2009), p. 68. DOI: 10.3329/jme.v40i1.3476.

[2] Ali Alhawaj and Anna Marzvanyan. "Physiology, Sensory Receptors." In: (Apr. 2019).

[3] Darius Ašeriškis and Robertas Damaševičius. "Gamification Patterns for Gamification Applications." In: *Procedia Computer Science* 39 (2014). The 6th international conference on Intelligent Human Computer Interaction, IHCI 2014, pp. 83–90. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2014.11.013. URL: https://www.sciencedirect.com/science/article/pii/S1877050914014318.

[4] T. Birrenbach, R. Wespi, W. E. Hautz, J. Berger, P. R. Schwab, G. Papagiannakis, A. K. Exadaktylos, and T. C. Sauter. "Development and usability testing of a fully immersive VR simulation for REBOA training." In: *International Journal of Emergency Medicine* 16.1 (Oct. 2023), p. 67. ISSN: 1865-1380. DOI: 10.1186/s12245-023-00545-6. URL: https://doi.org/10.1186/s12245-023-00545-6.

[5] Tanja Birrenbach, Josua Zbinden, George Papagiannakis, Aristomenis K Exadaktylos, Martin Müller, Wolf E Hautz, and Thomas Christian Sauter. "Effectiveness and Utility of Virtual Reality Simulation as an Educational Tool for Safe Performance of COVID-19 Diagnostics: Prospective, Randomized Pilot Trial." In: *JMIR Serious Games* 9.4 (Oct. 2021), e29586. ISSN: 2291-9279. DOI: 10.2196/29586. URL: http://www.ncbi.nlm.nih.gov/pubmed/34623315.

[6] Marilyn Blackmon, Peter Polson, Muneo Kitajima, and Clayton Lewis. "Cognitive walkthrough for the Web." In: vol. 4. Apr. 2002, pp. 463–470. DOI: 10.1145/503376.503459.

[7] *Bodyswaps*. URL: https://bodyswaps.co/.

[8] CA Brewer. *Color use guidelines for mapping and visualization. Visualization in Modern Cartography*. 1994.

[9] T. Buzan and B Buzan. *The Mind Map Book: How to Use the Radiant Thinking to Maximize Your Brain's Untapped Potential.* Penguin Book Ltd., London., 1993.

[10] David Checa and Andres Bustillo. “A review of immersive virtual reality serious games to enhance learning and training.” In: *Multimedia Tools and Applications* 79.9 (Mar. 2020), pp. 5501–5527. ISSN: 1573-7721. DOI: 10.1007/s11042-019-08348-9. URL: https://doi.org/10.1007/s11042-019-08348-9.

[11] *Cognitive3D*. URL: https://cognitive3d.com/.

[12] Carlos J. Costa, Manuela Aparicio, Sofía Aparicio, and João Tiago Aparício. “Gamification usage ecology.” In: *Proceedings of the 35th ACM International Conference on the Design of Communication* (2017). URL: https://api.semanticscholar.org/CorpusID:12772106.

[13] *Counter Strike Stats*. URL: https://csstats.gg/.

[14] Anthony V D’Antoni, Genevieve Pinto Zipp, Valerie G Olson, and Terrence F Cahill. “Does the mind map learning strategy facilitate information retrieval and critical thinking in medical students?” en. In: *BMC Med. Educ.* 10.1 (Dec. 2010).

[15] Kristina B. Dahlin, You-Ta Chuang, and Thomas J. Roulet. “Opportunity, Motivation, and Ability to Learn from Failures and Errors: Review, Synthesis, and Ways to Move Forward.” In: *Academy of Management Annals* 12.1 (2018), pp. 252–277. DOI: 10.5465/annals.2016.0049. eprint: https://doi.org/10.5465/annals.2016.0049. URL: https://doi.org/10.5465/annals.2016.0049.

[16] S. Dehaene. *How We Learn: The New Science of Education and the Brain.* Penguin Books Limited, 2020. ISBN: 9780241366479. URL: https://books.google.gr/books?id=R9C2DwAAQBAJ.

[17] Martin Eppler. “A Comparison between Concept Maps, Mind Maps, Conceptual Diagrams, and Visual Metaphors as Complementary Tools for Knowledge Construction and Sharing.” In: *http://www.alexandria.unisg.ch/Publikationen/54807* 5 (June 2006). DOI: 10.1057/palgrave.ivs.9500131.

[18] William Epstein, Louis Jolyon West, and William N Dember. “perception.” In: *Encyclopedia Britannica.* Oct. 2023.

[19] *Evaluate Interface Learnability with Cognitive Walkthroughs.* URL: https://www.nngroup.com/articles/cognitive-walkthroughs/.

[20] Achilleas Filippidis, Nikoloas Marmaras, Manos Maravgakis, Alexandra Plexousaki, Manos Kamarianakis, and George Papagiannakis. *VR Isle Academy: A VR Digital Twin Approach for Robotic Surgical Skill Development.* 2024. arXiv: 2406.00002 [cs.RO].

[21] *Flyinside*. URL: https://flyinside-fsx.com/.

[22] *FundamentalVR*. URL: https://fundamentalsurgery.com/.

[23] Papagiannakis George. “Gamification and Serious Games.” In: *Encyclopedia of Computer Graphics and Games*. Ed. by Newton Lee. Cham: Springer International Publishing, 2019, pp. 1–4. ISBN: 978-3-319-08234-9. DOI: 10.1007/978-3-319-08234-9_90-2. URL: https://doi.org/10.1007/978-3-319-08234-9_90-2.

[24] Janet K. Gibbs, Marco Gillies, and Xueni Pan. "A comparison of the effects of haptic and visual feedback on presence in virtual reality." In: *International Journal of Human-Computer Studies* 157 (2022), p. 102717. ISSN: 1071-5819. DOI: `https://doi.org/10.1016/j.ijhcs.2021.102717`. URL: `https://www.sciencedirect.com/science/article/pii/S107158192100135X`.

[25] Sandra G. Hart and Lowell E. Staveland. "Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research." In: *Human Mental Workload*. Ed. by Peter A. Hancock and Najmedin Meshkati. Vol. 52. Advances in Psychology. North-Holland, 1988, pp. 139–183. DOI: `https://doi.org/10.1016/S0166-4115(08)62386-9`. URL: `https://www.sciencedirect.com/science/article/pii/S0166411508623869`.

[26] Mary Hassandra, Evangelos Galanis, Antonis Hatzigeorgiadis, Marios Goudas, Christos Mouzakidis, Eleni Maria Karathanasi, Niki Petridou, Magda Tsolaki, Paul Zikas, Giannis Evangelou, George Papagiannakis, George Bellis, Christos Kokkotis, Spyridon Rafail Panagiotopoulos, Giannis Giakas, and Yannis Theodorakis. "*A* virtual reality app for physical and cognitive training of older people with mild cognitive impairment: Mixed methods feasibility study." en. In: *JMIR Serious Games* 9.1 (Mar. 2021), e24170.

[27] *Hearthstone Deck Tracker*. URL: `https://hsreplay.net/downloads/`.

[28] Jessica Hooper, Eleftherios Tsiridis, James E. Feng, Ran Schwarzkopf, Daniel Waren, William J. Long, Lazaros Poultsides, William Macaulay, George Papagiannakis, Eustathios Kenanidis, Eduardo D. Rodriguez, James Slover, Kenneth A. Egol, Donna P. Phillips, Scott Friedlander, and Michael Collins. "Virtual Reality Simulation Facilitates Resident Training in Total Hip Arthroplasty: A Randomized Controlled Trial." In: *The Journal of Arthroplasty* 34.10 (2019), pp. 2278–2283. ISSN: 0883-5403. DOI: `https://doi.org/10.1016/j.arth.2019.04.002`. URL: `https://www.sciencedirect.com/science/article/pii/S0883540319303341`.

[29] *How to Conduct a Cognitive Walkthrough*. URL: `https://www.interaction-design.org/literature/article/how-to-conduct-a-cognitive-walkthrough`.

[30] Mike Kentros, Achilleas Filippidis, John Petropoulos, Michael Cole, Vitaly Popov, Bryan Harmer, Manos Kamarianakis, and George Papagiannakis. *Enhancing Crew Resource Management in VR Medical Training: Tools for analyzing advanced, collaborative group training in the metaverse*. 2024.

[31] Erika Kerruish. "Arranging sensations: smell and taste in augmented and virtual reality." In: *The Senses and Society* 14.1 (2019), pp. 31–45. DOI: `10.1080/17458927.2018.1556952`. eprint: `https://doi.org/10.1080/17458927.2018.1556952`. URL: `https://doi.org/10.1080/17458927.2018.1556952`.

[32] Benjamin Lee, Michael Sedlmair, and Dieter Schmalstieg. "Design Patterns for Situated Visualization in Augmented Reality." In: *IEEE Transactions on Visualization and Computer Graphics* (2023), pp. 1–12. ISSN: 2160-9306.

DOI: 10.1109/tvcg.2023.3327398. URL: http://dx.doi.org/10.1109/TVCG.2023.3327398.

[33] Sunkee Lee and Jisoo Park. "Giving up learning from failures? An examination of learning from one's own failures in the context of heart surgeons." In: *Strategic Management Journal* n/a.n/a (). DOI: https://doi.org/10.1002/smj.3609. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/smj.3609. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/smj.3609.

[34] Debra A Lieberman. "What can we learn from playing interactive games?" In: *Playing video games*. Routledge, 2012, pp. 447–469.

[35] Front Matter. *How People Learn: Brain, Mind, Experience, and School: Expanded Edition*. Expanded Edition. Washington, DC: The National Academies Press, 2000.

[36] Philip Z. Maymin. In: *Journal of Quantitative Analysis in Sports* 17.1 (2021), pp. 11–27. DOI: doi:10.1515/jqas-2019-0096. URL: https://doi.org/10.1515/jqas-2019-0096.

[37] Ben Medler and Brian Magerko. "Analytics of play: Using information visualization and gameplay practices for visualizing video game data." In: *Parsons Journal for Information Mapping* 3.1 (2011), pp. 1–12.

[38] Miguel Melo, Guilherme Gonçalves, Pedro Monteiro, Hugo Coelho, José Vasconcelos-Raposo, and Maximino Bessa. "Do Multisensory Stimuli Benefit the Virtual Reality Experience? A Systematic Review." In: *IEEE Transactions on Visualization and Computer Graphics* PP (July 2020), pp. 1–1. DOI: 10.1109/TVCG.2020.3010088.

[39] *Net Promoter Score*. URL: https://www.techtarget.com/searchcustomerexperience/definition/Net-Promoter-Score-NPS.

[40] Jakob Nielsen. "How to conduct a heuristic evaluation." In: *Nielsen Norman Group* 1.1 (1995), p. 8.

[41] Jakob Nielsen. *Usability heuristics for user interface design*. 10.

[42] Jakob Nielsen and Rolf Molich. "Heuristic evaluation of user interfaces." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '90. Seattle, Washington, USA: Association for Computing Machinery, 1990, pp. 249–256. ISBN: 0201509326. DOI: 10.1145/97243.97281. URL: https://doi.org/10.1145/97243.97281.

[43] *ORamaVR*. URL: https://oramavr.com/.

[44] Jérôme Perret and Emmanuel Vander Poorten. "Touching Virtual Reality: a Review of Haptic Gloves." In: June 2018.

[45] Stéphanie Philippe, Alexis D. Souchet, Petros Lameras, Panagiotis Petridis, Julien Caporal, Gildas Coldeboeuf, and Hadrien Duzan. "Multimodal teaching, learning and training in virtual reality: a review and case study." In: *Virtual Reality Intelligent Hardware* 2.5 (2020). VR/AR research and commercial applications in Singapore, pp. 421–442. ISSN: 2096-5796. DOI: https://doi.org/10.1016/j.vrih.2020.07.008. URL: https://www.sciencedirect.com/science/article/pii/S2096579620300711.

[46] Pushpakumar R, Karun Sanjaya, S. Rathika, Ahmed Alawadi, Khamdamova Makhzuna, S. Venkatesh, and B. Rajalakshmi. "Human-Computer Interaction: Enhancing User Experience in Interactive Systems." In: *E3S Web of Conferences* 399 (July 2023). DOI: 10.1051/e3sconf/202339904037.

[47] Nimesha Ranasinghe, Pravar Jain, Nguyen Thi Ngoc Tram, Koon Chuan Raymond Koh, David Tolley, Shienny Karwita, Lin Lien-Ya, Yan Liangkun, Kala Shamaiah, Chow Eason Wai Tung, Ching Chiuan Yen, and Ellen Yi-Luen Do. "Season Traveller: Multisensory Narration for Enhancing the Virtual Reality Experience." In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18. ¡conf-loc¿, ¡city¿Montreal QC¡/city¿, ¡country¿Canada¡/country¿, ¡/conf-loc¿: Association for Computing Machinery, 2018, pp. 1–13. ISBN: 9781450356206. DOI: 10.1145/3173574.3174151. URL: https://doi.org/10.1145/3173574.3174151.

[48] Sandrine de Ribaupierre, Bill Kapralos, Faizal Haji, Eleni Stroulia, Adam Dubrowski, and Roy Eagleson. "Healthcare Training Enhancement Through Virtual Reality and Serious Games." In: *Virtual, Augmented Reality and Serious Games for Healthcare 1*. Ed. by Minhua Ma, Lakhmi C. Jain, and Paul Anderson. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 9–27. ISBN: 978-3-642-54816-1. DOI: 10.1007/978-3-642-54816-1_2. URL: https://doi.org/10.1007/978-3-642-54816-1_2.

[49] Daniel Schacter. *Psychology*. Worth Publishers., 2011.

[50] Michael Waskom / Seaborn. *Choosing color palettes*. URL: https://seaborn.pydata.org/tutorial/color_palettes.html.

[51] Yinghui Shi, Huiyun Yang, Yi Dou, and Yong Zeng. "Effects of mind mapping-based instruction on student cognitive learning outcomes: a meta-analysis." In: *Asia Pacific Education Review* 24 (Mar. 2022), pp. 1–15. DOI: 10.1007/s12564-022-09746-9.

[52] S Swestyani, Mohammad Masykuri, Baskoro Prayitno, Y Rinanto, and Sri Widoretno. "An analysis of logical thinking using mind mapping." In: *Journal of Physics: Conference Series* 1022 (May 2018), p. 012020. DOI: 10.1088/1742-6596/1022/1/012020.

[53] *Talespin*. URL: https://www.talespin.com/.

[54] Wei-Peng Teo, Makii Muthalib, Sami Yamin, Ashlee Hendy, Kelly Bramstedt, Eleftheria Kotsopoulos, Stephane Perrey, and Hasan Ayaz. "Does a Combination of Virtual Reality, Neuromodulation and Neuroimaging Provide a Comprehensive Platform for Neurorehabilitation? – A Narrative Review of the Literature." In: *Frontiers in Human Neuroscience* 10 (June 2016). DOI: 10.3389/fnhum.2016.00284.

[55] *The 8 Senses: How Your Child Interacts with the Environment*. URL: https://chlss.org/blog/8-senses-parenting-sensory-processing-disorder/.

[56] *Theory on heatmap generation*. URL: https://help.sticky.ai/hc/en-us/articles/8629243856145-Theory-on-heatmap-generation.

[57] H. Trinon. *Immersive technologies for virtual reality - Case study : flight simulator for pilot training.* 2019. URL: https://matheo.uliege.be/handle/2268.2/6443.

[58] Paul Zikas, Manos Kamarianakis, Ioanna Kartsonaki, Nick Lydatakis, Steve Kateros, Mike Kentros, Efstratios Geronikolakis, Giannis Evangelou, Achilles Apostolou, Paolo Alejandro Alejandro Catilo, and George Papagiannakis. "Covid-19 - VR Strikes Back: Innovative Medical VR Training." In: *ACM SIGGRAPH 2021 Immersive Pavilion.* SIGGRAPH '21. Virtual Event, USA: Association for Computing Machinery, 2021. ISBN: 9781450383684. DOI: 10.1145/3450615.3464546. URL: https://doi.org/10.1145/3450615.3464546.

[59] Paul Zikas, Antonis Protopsaltis, Nick Lydatakis, Mike Kentros, Stratos Geronikolakis, Steve Kateros, Manos Kamarianakis, Giannis Evangelou, Achilleas Filippidis, Eleni Grigoriou, Dimitris Angelis, Michail Tamiolakis, Michael Dodis, George Kokiadis, John Petropoulos, Maria Pateraki, and George Papagiannakis. *MAGES 4.0: Accelerating the world's transition to VR training and democratizing the authoring of the medical metaverse.* 2023. arXiv: 2209.08819 [cs.GR].