POLITECNICO DI TORINO

College of Computer Engineering, Cinema and Mechatronics

Master's Degree
in Cinema and Media Engineering

Politecnico
di Torino

Design and Development of Interactive 3D Virtual Reality Environments for Substance Use Disorder Monitoring and Treatment

Tutors

Prof. Andrea Bottino
Prof. Francesco Strada

Candidate

Antonino Tiziano Castagnella

A Morena, i miei traguardi saranno sempre i tuoi.

Abstract

Substance Use Disorder (SUD) represents a global health challenge. Digital therapeutics, particularly those based on Virtual Reality (VR), offer a promising paradigm for increasing the efficacy and accessibility of treatment by providing immersive, controlled environments. This thesis documents the design and implementation of a standalone VR application for Meta Quest, developed in Unity, intended to support SUD therapy. The core of the system is a virtual therapist guided by an advanced conversational agent (Large Language Model, LLM) hosted on a high-performance computing (HPC) infrastructure at the Politecnico di Torino. The therapeutic flow is structured into sequential modules. The patient, guided by the virtual therapist, completes a pre-assessment analysis via an in-VR tablet interface to record levels of craving, pain, and mood. Subsequently, the patient selects from several available therapeutic techniques (e.g., guided breathing, visualization, mindfulness). The system then loads a dedicated therapeutic scene where the patient performs the exercise, concluding with a post-assessment analysis. This project not only demonstrates the technical feasibility of integrating complex, HPC-hosted LLMs with a standalone VR client but also provides a robust architecture for structured data collection. This enables the tracking of patient progress across treatment cycles and paves the way for future personalized, data-driven therapeutic tools.

Table of Contents

Chapter 1

Introduction

1.1 Problem Statement: The Challenge of SUD

Substance Use Disorders (SUD) represent one of the major concerns globally. SUD is defined by DSM-5 as a chronic, relapsing brain disease characterized by compulsive drug seeking and use, despite harmful consequences and body impairments. It includes a wide range of substances, such as alcohol, cannabinoids, cocaine stimulants, tobacco, hallucinogens, opioids (OUD). As defined by DSM-IV [2], both substance abuse and substance dependence refer to "a maladaptive pattern of substance use, leading to clinically significant impairment or distress." The DSM-5 has eliminated the distinct abuse and dependence disorders for several reasons [3]: (1) the distinction provided little guidance for treatment;(2) the distinction created “diagnostic orphans” (individuals who endorsed two dependence symptoms and no abuse symptoms and therefore did not meet any diagnostic criteria); (3) the hierarchical structure did not follow the anticipated relationship between abuse and dependence (that abuse was largely a less severe prodrome4 of dependence); and (4) the separation caused the abuse diagnosis to suffer from significant reliability problems. The DSM-5 combines the abuse and dependence criteria under the new rubric substance use disorder, which requires 2 out of 11 criteria in a 12-month period for diagnosis.[4]

1.2 Thesis Objectives and Scope

The primary objective of this thesis is to address the previously identified gaps in Substance Use Disorder (SUD) treatment—specifically accessibility, stigma, and the "generalization gap"—through the design, implementation, and technical evaluation of a novel Virtual Reality (VR) therapeutic system.

This research was developed in direct collaboration with the University of Maryland, specifically commissioned by the Colloca Lab.

This project is not intended to invent new therapeutic methods. Instead, it aims to create a more effective, engaging, and accessible delivery mechanism for existing, evidence-based practices. As outlined by established medical literature, the standard of care for SUD diagnosis and treatment relies on psychosocial interventions such as Cognitive Behavioral Therapy (CBT) and Mindfulness-Based Relapse Prevention (MBRP) to help patients manage cravings and emotional distress [5]. Our system is designed to translate these proven techniques into an immersive, interactive, and self-guided VR format.

To achieve this primary objective, the following specific technical goals were established for this project:

- **To develop, first of all, a credible and effective environment** in which to carry out the first part of therapy together with the virtual agent.
- **To introduce a visually credible virtual agent** capable of conversing with the user(patient).
- **To implement a state-driven conversational agent** to act as a virtual therapist, capable of guiding a user through a structured, multi-step therapeutic flow.
- **To architect a robust, decoupled client-server system** capable of running a complex Large Language Model (LLM) while maintaining real-time interaction with a lightweight, standalone VR client (Meta Quest).
- **To develop a series of distinct therapeutic modules** including free breathing, guided breathing, visualization, and mindfulness exercises, that can be loaded dynamically based on the agent's state.
- **To develop a series of environments** associated to the different types of treatments.
- **To create an intuitive, VR-native user interface** for handling user input, such as pre- and post-assessment questionnaires.
- **To implement a secure and structured data collection pipeline** that captures session data, user responses, and selected therapies for future analysis.

1.2.1 Limitations

The scope of this thesis explicitly excludes: the conducting of any medical diagnosis of SUD/OUD, and any claim of replacing a human therapist. This system is designed as a supplementary tool to augment existing treatments, not to supplant them.

1.3 Thesis Structure

The remainder of this thesis is organized as follows:

Chapter 2 – Background and Literature Review provides a review of the current state of research on SUD, established psychosocial interventions (CBT, Mindfulness), and the emerging role of Virtual Reality and LLMs in healthcare.

Chapter 3 – System Architecture and Design details the high-level technical blueprint of the project, justifying the choice of a decoupled client-server architecture, the FSM (Finite State Machine) model, and the selection of key technologies like LangGraph, URP and Microsoft Azure Speech SDK.

Chapter 4 – Implementation: The Server-Side Agent offers a deep dive into the Python backend. This includes the implementation of the FastAPI server, the automatic network discovery (UDP Broadcast), the logic of the LangGraph agent, and the database storage.

Chapter 5 – Implementation: The Client-Side VR Application documents the development of the Unity client for Meta Quest. This covers the clinic scene implementation, a quick review of the main script functions inside it, the UI components of the in-VR tablet, and the integration of the specific therapeutic environment scenes.

Chapter 6 – Usability Tests and Results presents the data collected from the usability tests, evaluating all the environments and the virtual agent.

Chapter 7 – Conclusion and Future Work summarizes the achievements of the project against the initial objectives, discusses the system's current limitations, and proposes directions for future research and development.

Chapter 2

Background and Literature Review

This literature review aims to explore the diverse applications of Virtual Reality (VR) in the treatment of substance dependence, categorizing the existing research into distinct therapeutic approaches. By examining the efficacy and mechanisms of these VR-based interventions, this review seeks to provide a comprehensive overview of the current state of the field and identify future directions for research and clinical practice.

2.1 Current Therapeutic Approaches for SUD

The standard of care for Substance Use Disorder (SUD) is a combinatorial approach, addressing the complex interplay of physiological dependence, psychological triggers, and social factors. As outlined by established medical literature, diagnosis and treatment rely on a foundation of psychosocial interventions, often supported by pharmacological assistance [5].
Pharmacological interventions, such as Medication-Assisted Treatment (MAT) for Opioid Use Disorder (OUD), are frequently a first-line defense to manage acute withdrawal symptoms and reduce physiological cravings. However, long-term recovery is contingent on the patient's ability to manage the psychological component of addiction.
To this end, psychosocial interventions are critical. The most prominent include:

- **Cognitive Behavioral Therapy (CBT):** This approach focuses on identifying and modifying dysfunctional thought patterns and behaviors. Patients learn to recognize their environmental and emotional triggers, challenge the automatic thoughts that lead to use, and develop effective coping strategies to prevent relapse.
- **Mindfulness-Based Relapse Prevention (MBRP):** This modality teaches patients to observe their cravings and negative emotions with a non-judgmental awareness, viewing them as transient mental events rather than urgent commands to act. This "de-centering" allows them to tolerate discomfort without engaging in compulsive behavior.
- **Relaxation and Breathing Techniques:** Often used in conjunction with mindfulness, these techniques are rapid-response tools to manage acute moments of anxiety or craving by directly activating the parasympathetic nervous system.

2.2 Virtual Reality in Psychotherapy

While these traditional methods are effective, they face significant barriers, including social stigma, high cost, and a lack of accessibility. Furthermore, they suffer from a critical "general-

ization gap," where skills learned in the sterile environment of a clinic are difficult for patients to apply in the high-risk, trigger-filled real world.
Virtual Reality (VR) emerges as a powerful technological solution to this gap. It provides an immersive, multisensory, and ecologically valid platform to simulate high-risk scenarios within a safe and controlled therapeutic space. This review categorizes the existing research on VR for SUD into several distinct therapeutic approaches.

2.2.1 Cue Exposure Therapy (VRET) in VR

This approach uses VR to expose individuals to drug-related cues (e.g., virtual environments simulating bars, parties, or drug paraphernalia) in a controlled manner, aiming to reduce cravings and desensitize the individual to triggers.

VRET for Alcohol

Study 1 A 2023 study protocol by Lütt et al. [6] outlined a design to measure craving induction using fMRI.

- **Environments:**
    1. *Neutral Environment (Baseline):* A "non-room" (black room with a bright grid) for spatial orientation and control.
    2. *High-Risk Scenario 1 (Living Room):* Patient sits on a couch with their preferred alcoholic drink on a table, simulating a solitary domestic trigger.
    3. *High-Risk Scenario 2 (Bar):* Patient is in a social bar setting where a bartender serves them their preferred drink.
    4. *Acclimatization Environment:* A neutral white room to acclimate the patient to the VR experience.
- **Aims and Results:** As this is a study protocol, final results are pending. The aims are to: (1) Induce craving, (2) Compare reactions between scenarios, (3) Measure the duration of craving, (4) Assess the sense of presence, and (5) Monitor side effects.

Study 2 A 2023 randomized controlled trial (RCT) protocol by Deng et al. [7] proposes a multi-faceted VR intervention.

- **Environments:**
    1. *Relaxation Scene:* Four serene natural landscapes for relaxation.
    2. *High-Risk Scene:* Simulations of real-life drinking situations (street barbecue stands, restaurants, bars, home settings) enhanced with olfactory (scent) stimulation.
    3. *Aversive Scene:* VR videos depicting the harmful health and social consequences of alcohol consumption.
- **Aims and Results:** The ongoing trial aims to evaluate: (1) Reduction in alcohol craving, (2) Reduction in AUD severity, (3) Improvement in comorbid depression/anxiety, (4) Normalization of Event-Related Potentials (ERP), and (5) Comparison between interventions.

Study 3 A 2021 study by Hernández-Serrano O et al. [8] investigated the specific predictors of cue-elicited craving.

- **Environments:** A neutral room (white room with water) and several alcohol-related environments (Bar, Restaurant, Pub, At-Home). The environments included interactive alcohol cues (beer, wine, spirits) and olfactory stimuli.
- **Aims and Results:** The study found several key predictors and correlations:
  - *Predictors of Craving:* Both AUD Severity and the Perceived Realism (PR) of the virtual beverages were significant predictors. Patients who found the drinks more realistic reported higher cravings.
  - *Predictors of Realism:* The PR of beverages was also a key predictor of the overall PR of the environment.
  - *Mediation Effect:* The study found that the PR of beverages mediated the relationship between AUD severity and craving.
  - *Correlations:* Crucially, there was no significant correlation between the PR of the environment and craving, suggesting the realism of the specific cue-object (the drink) is more important than the general background.

Study 4 This 2024 study protocol [9] details an RCT focusing on common daily triggers.

- **Environments:** (1) Heading for the metro (with alcohol ads), (2) Going to the supermarket (alcohol aisle), (3) Being alone at home (bottle on table), (4) Going to a party.
- **Aims and Results:** The primary outcome is the cumulative number of standard drinks consumed at an 8-month follow-up. Secondary outcomes include changes in craving, anxiety, depression, and self-efficacy.

Study 5 (Ryan JJ et al., 2010) A key 2010 study [10] compared reactivity in binge drinkers versus non-binge drinkers.

- **Environments:** Neutral underwater scenes and four alcohol-cue rooms (Kitchen, Bar, Argument, Party) featuring visual, auditory, and olfactory cues.
- **Aims and Results:** The study compared 15 binge drinkers and 8 non-binge drinkers.
  - *Craving for Alcohol:* Binge drinkers reported significantly higher cravings in the Kitchen and Party rooms.
  - *Thinking About Drinking:* Binge drinkers reported more thoughts about drinking in the Bar and Party rooms.
  - *Attention:* Both groups paid similar attention to cues, but attention was higher for all participants in the cue rooms vs. neutral rooms.

VRET for Nicotine

Study 1 An early pilot study [11] tested VR-CET on 16 male adolescent smokers over six 20-minute sessions.

- **Environments:** A virtual bar setting with visual (bottles, cigarettes, ads, smoking avatar), auditory (chatter, music), and interactive (avatar offering cigarettes) cues.
- **Aims and Results:**
  - *Craving Reduction:* While self-reported craving showed a decline, it was not statistically significant. However, a key behavioral measure, the Morning Smoking Count, was significantly reduced from 2.93 to 1.29 cigarettes/day.
  - *Cue Reactivity:* Social pressure from the avatar elicited stronger cravings than passive objects, highlighting the importance of social simulation.

Study 2 This study [12] tested the impact of olfactory cues.

- **Environments:** (1) Neutral room (nature, floral scent), (2) Paraphernalia room (visual smoking cues), (3) Party room (social smoking cues). Half the participants received olfactory cues (smoke, coffee).
- **Aims and Results:** The smoking-related rooms significantly increased attention to visual cues and thoughts about smoking. However, the addition of olfactory cues did not significantly enhance attention or craving, suggesting a "suggestibility effect" as 30% of the no-scent group also reported smelling scents.

Study 3 This study [13] investigated contextual cues.

- **Environments:** Neutral environments (underwater, art gallery) and two convenience store environments (one with explicit cigarette cues, one without).
- **Aims and Results:** Smokers reported significantly higher craving in the store `without` cues compared to the neutral environment. Craving was highest in the store `with` cues. This suggests that the context (the convenience store) is a powerful trigger by itself, even before explicit cues are present.

Study 4 This study [14] tested the difference between passive exposure and active interaction.

- **Environments:** A virtual pub with three conditions: (1) Virtual Cigarette (simulating smoking via microphone), (2) Virtual Darts (neutral distraction), (3) Passive Pub exposure.
- **Aims and Results:** Only the Virtual Cigarette condition produced a significant increase in both self-reported craving and heart rate. This indicates that simulating the behavior (a proximal cue) is more potent at triggering cravings than passive exposure to the context (a distal cue).

Study 5 This study [15] combined VRET with memory reconsolidation theory.

- **Environments:** A neutral mountain landscape, a smoking-related scenario (smoking zone at a college), and an office for acclimatization.
- **Aims and Results:** All groups showed increased craving on Day 1. On Day 3, the group that performed VR extinction training 15 minutes after a smoking memory retrieval task showed no significant craving increase, indicating successful inhibition of memory reconsolidation. The other control groups (no retrieval, immediate extinction) still showed significant craving.

VRET for Opioids, Cannabis, and Other Drugs

While VRET for alcohol and nicotine is well-documented, its application for **Opioid Use Disorder (OUD)** is a significant gap in the literature. Most VR research for opioids focuses on relaxation or pain management, not direct cue exposure.

For **Cannabis Use Disorder**, however, VRET has been validated as a powerful assessment tool. A key study by Bordnick et al. [16] exposed participants to:

- **Environments:** (1) Neutral art gallery rooms, (2) A Paraphernalia Room (bongs, joints, cannabis scents), (3) A Social Interaction Party Room (avatars smoking, offering joints).
- **Aims and Results:** Craving levels, attention to cues, and thoughts about smoking were all significantly higher in the cannabis environments compared to the neutral rooms, with large effect sizes (average d=1.07) demonstrating VR's validity for eliciting cannabis craving.

For other **Drugs (Cocaine)**, a 2024 study protocol by Lehoux et al. [17] details an interactive, multisensory VRCET system.

- **Environments:** Scenarios are tailored to the participant's primary method of use (snorting, smoking, or injecting) and include visual (peers using), auditory (preparation sounds), and olfactory/tactile cues.
- **Aims and Results:** The protocol aims to test VRCET combined with Memory-Focused Cognitive Therapy (MFCT) against traditional picture-based CET, hypothesizing that the higher immersion of VR will lead to greater craving extinction and lower relapse rates.

A 2021 systematic review [18] on VRET for AUD confirmed that VR successfully elicits craving (especially bar and party scenes) and that craving correlates with the ecological validity of the virtual environment. However, the review noted that while VRET shows promise, its long-term superiority over standard CBT for sustained abstinence remains mixed and requires further research.

2.2.2 Relaxation and Stress Management

This approach uses VR to create calming, immersive environments that help individuals manage stress and anxiety, which are common triggers for substance use. While this review found no specific studies for alcohol or nicotine, its application for OUD is a critical emerging field.

Study 1 (Faraj MM et al., 2021) This study [19] tested a VR meditative intervention on OUD patients undergoing methadone maintenance.

- **Environments:** A four-stage narrative: (1) Introduction to child avatars, (2) Teaching of the "Breath Brake®" technique, (3) Confrontation with a "beast" representing addiction in a metaphorical brain, (4) Waterfall Meditation where the beast is washed away.
- **Aims and Results:** The intervention was well-tolerated and showed significant behavioral and neurobiological outcomes:
  - *Behavioral:* Significant reductions in self-reported pain, opioid craving, anxiety, and depression after each session.
  - *Neurobiological:* fMRI data showed reduced activation in the left postcentral gyrus (a key pain region) after the 12-week intervention and changes in brain connectivity.

Study 2 (McGirt et al., 2023) This study [20] utilized "Vx Therapy," an in-home VR toolkit paired with remote CBT for patients with chronic pain (a major OUD risk factor).

- **Environments:** The toolkit included four categories of modules: (1) Education (3D anatomy), (2) Meditation (guided mindfulness in nature), (3) Distraction (puzzles, games), (4) Escape/Entertainment (virtual travel).
- **Aims and Results:** The study found significant acute and long-term benefits.
  - *Pain Reduction:* Pain was reduced by 33% immediately after use, and the duration of relief increased from 2.8 hours to 4.5 hours by the end of the 14-week program.
  - *Mood:* Anxiety scores decreased by 46%.
  - *Long-Term Efficacy:* No habituation was observed; the effects increased over time.

2.2.3 Cognitive-Behavioral Therapy (CBT) in VR

This modality integrates VRET with active skills training, placing patients in realistic scenarios where they can practice CBT strategies.

CBT for Alcohol

Study 1 The CRAVR trial protocol [21] is designed to test VR-CBT against standard CBT for AUD.

- **Environments:** The system uses 30 high-resolution 360° videos of five real-world locations: Pub, Bar/Party, Restaurant, Home, and Supermarket.
- **Method:** Each location features 6 progressively intense scenarios, with interactive avatars in later grades offering drinks or confronting the patient, allowing for graded exposure and skills practice.

Study 2 This study [22] used VR for cognitive rehabilitation in AUD patients.

- **Environments:** The "Systemic Lisbon Battery (SLB)," a virtual city environment (minimarket, pharmacy, art gallery, home) where participants performed activities of daily living (ADLs) like managing a shopping list and budget.
- **Aims and Results:** The pilot RCT found that the VR group showed significant cognitive improvements compared to the control group, particularly in:
  - *Attention:* Correct responses on the Toulouse Pierón test increased significantly ($p<.001$).
  - *Cognitive Flexibility:* Reduced errors on the Wisconsin Card Sorting Test (WCST) ($p=.001$).
  - *Memory (Rey Complex Figure:)* Experimental group showed improved recall (21.54 $\rightarrow$ 37.80, $p=.002$), though no significant group $\times$ time interaction.

CBT for Drugs

Study 1 An earlier pilot study [23] of the same SLB system confirmed its feasibility.

- **Environments:** A virtual city with locations like a mini-market, pharmacy, and art gallery for task-based training, populated by NPCs.
- **Task-Based Training:** participants completed goal-oriented tasks (e.g. purchasing items from a list, organizing objects) with increased difficulty in order to challenge cognitive skills.
- **Aims and Results:** The study confirmed significant improvements in attention and cognitive flexibility, and reported high retention (only 14% dropout), supporting VR's usability in this clinical population.

2.2.4 Motivational Enhancement and Goal Setting

This approach uses VR's narrative and emotive power to enhance a patient's motivation for change, either by visualizing positive outcomes or confronting negative consequences.

Study 1 This study [24] used VR to train medical students, aiming to reduce the stigma toward OUD that often prevents patients from seeking care.

- **Environments:** A 12-episode, 360-degree video narrative following "Destiny," a pregnant woman with OUD, through clinical, home, and community settings.
- **Aims and Results:** The intervention was highly feasible (100% retention) and successful in its primary goal, showing a significant decrease in stigma scores ($p < 0.001$) and a significant improvement in empathy ($p = 0.023$) among participants.

Study 2 This study [25] applied Aversive Counterconditioning (VR-CP) to METH users.

- **Environments:**
  - A 360° video simulating social METH use. Key scenes included Social Interactions with auditory cues, drug paraphernalia, group settings where people consume METH together.
  - Six custom VR videos pairing METH-use scenes with highly aversive scenarios (police arrest, severe hallucinations, infections, contracting sexually transmitted diseases, premature aging/tooth loss, sudden death).
- **Aims and Results:** The VR-CP group showed a significant decrease in both METH-craving and METH-liking ($p < 0.001$) compared to the waiting list group. They also showed a significant reduction in physiological stress (HRV) during cue exposure.

2.2.5 Social Skills Training and Relapse Prevention

This modality specifically targets social pressure, a primary driver of relapse. VR is used to simulate social interactions where patients can practice refusal.

Study 1 This study [26] co-designed a social skills training environment with clinical experts.

- **Environments:** An immersive, messy "social-housing apartment" with alcohol cues and a Persuasive Embodied Conversational Agent (ECA).
- **Method:** The ECA (a virtual peer) exerts peer pressure using emotional appeals ("You used to be more sociable"). The user practices refusal via a dialog interface that provides traffic-light feedback on the risk level of their chosen response.
- **Aims and Results:** Experts rated the IVR as realistic and immersive. A key finding was that the ECA's nonverbal animations (e.g., drinking gestures) were critical for realism, but the text-to-speech voice reduced persuasiveness.

Study 2 This study [27] adapted the Approach-Avoidance Task (AAT) to VR.

- **Environments:** A simple table-top VR setting where participants used a controller to physically "PUSH" (avoid) alcoholic images and "PULL" (approach) non-alcoholic images.
- **Method:** Grasping Conditions. Three variants were tested:
  - Never Grasp: No lever press; only arm movements.
  - Always Grasp: Press lever for both PUSH and PULL.
  - Grasp When PULLing: Press lever only when pulling stimuli.

  Participants sorted stimuli into slots (near for PULL, far for PUSH) while reaction times (RTs) were measured.
- **Aims and Results:** The VR-AAT outperformed the standard PC-joystick version at detecting approach bias in AUD patients. The version without grasping was most effective for detection, while the "Grasp When PULLing" version was the most feasible for users.

Study 3 This protocol [28] describes a multi-site RCT to further validate the VR-based AATP (Approach-Avoidance Training Program).

- **Environments:** An immersive VR bar with high-poly 3D drinks and a virtual bartender, where participants PUSH or PULL the drinks.
- **Aims and Results:** The study hypothesizes that the embodied, realistic nature of the VR-based AATP will prove superior to both PC-based training and treatment-as-usual (TAU) in reducing long-term alcohol consumption, cravings and impulsivity.

Study 4 This study [29] combined cue exposure with aversive conditioning in a three-stage protocol.

- **Environments:** (1) Relaxation (calm landscapes), (2) High-Risk (virtual bar with alcoholic beverages and social drinking environments), (3) Aversive (visuals of people vomiting after drinking paired with the real-world taste of sour kefir, a vomit-like fermented milk that creates disgust).

- **Aims and Results:** After 10 VRT sessions, PET/CT imaging revealed that patients had reduced metabolism in the right lentiform nucleus (reward processing) and right temporal lobe (emotional reactivity). This neurobiological change suggests VRT can help normalize hyperactivity in the brain's reward circuits. Self-reported cravings (VAS) significantly decreased reward sensitivity and emotional reactivity to alcohol cues. Changes in brain metabolism did not directly correlate with craving reduction, hinting at complex mechanisms.

2.2.6 Gamification and Engagement in VR Therapy

This final category applies game design principles to increase user engagement and adherence to treatment.

Gamification for Opioids

Study 1 A pilot study [30] tested a mindfulness-based VR program, "3D Therapy Thrive" (3DTT), with OUD inpatients.

- **Environments:** In this pilot study, the VR intervention used was 3D Therapy Thrive (3DTT), a psychologically informed VR program delivered via the Meta Quest 2.0 headset. The VR environment included five modules designed to address psychological factors, mindfulness, creativity, problem-solving, and self-celebration. The experience featured interactive 3D art, first-person challenges, and guided activities with a calming voice, all aimed at improving mood, reducing tension, and alleviating cravings.
- **Aims and Results:** The intervention was highly feasible (94% satisfaction) and led to significant reductions in depression and tension (after only 1-2 sessions), and opioid cravings. The VR group was also 41% more likely to complete their treatment protocol than the treatment-as-usual (TAU) group.

Gamification for Drugs

Study 1 A "Social-Volitional VR" (SVVR) program [31] was developed to train volunteers in drug abuse prevention for at-risk youth.

- **Environments:**
  - Ice-breaking and introductions – Building rapport with students.
  - Tarot card activity – Encouraging self-reflection and problem identification.
  - Handling disengagement – Managing situations where students ignore volunteers.
  - Peer influence scenarios – Using a game-based platform (Kahoot) to explore communication strategies.
  - Social media interaction – Teaching volunteers to engage with students via platforms like Facebook/Instagram to monitor social influences.
- **Aims and Results:** The SVVR group showed significant improvements in problem-solving skills ($p = 0.03$) and self-efficacy ($p < 0.001$) compared to controls. SVVR participants showed greater confidence in assisting students ($p < 0.001$). Activities like

role-playing and mentoring boosted their belief in handling drug-related challenges. No significant improvement ($p = 0.28$) in teamwork, possibly due to limited collaborative tasks in VR. Participants expressed high engagement with SVVR, citing novelty and praticality.

2.3 Conversational AI and Large Language Models (LLMs) in Healthcare

While the environments and interventions reviewed provide the "stage" for therapy, the "actors"—the virtual humans—have traditionally been a limiting factor. Most studies employ pre-scripted 360° videos, simple Embodied Conversational Agents (ECAs) with basic dialog trees [29], or avatars that offer no real-time social interaction. This creates a "brittle" experience that can quickly break immersion and fails to adapt to the patient's unique needs.

The recent advent of powerful Large Language Models (LLMs) presents an opportunity to create a virtual therapist that is truly dynamic, empathetic, and responsive. LLMs are deep learning models trained on massive datasets, capable of understanding context, generating nuanced human-like text, and performing complex reasoning.

In healthcare, conversational AI platforms are already being used to deliver CBT and mindfulness interventions via text, offering significant advantages in accessibility and anonymity. However, these text-based models lack the embodiment and presence that is central to the therapeutic alliance. The integration of an LLM as the "brain" of an animated, 3D virtual therapist within an immersive VR environment represents a significant leap forward, combining the accessibility of AI with the proven power of VR presence.

2.4 Gaps in Current Research and Project Contribution

This literature review highlights several gaps that the current project aims to address.

1. **A Gap in Modality Integration:** The majority of studies focus on a single therapeutic modality (e.g., VRET or Relaxation). There is a lack of integrated systems where a patient can be assessed and then seamlessly guided through a multi-step flow that combines pre-assessment, a choice of intervention, and post-assessment, all within one continuous session.

2. **A Gap in OUD Intervention:** While research on alcohol and nicotine is prevalent, the literature on direct VRET and skills-based VR for Opioid Use Disorder is sparse. Most OUD studies focus on pain management (a comorbidity) or provider-side empathy training [27], not direct craving intervention for the patient.

3. **A Gap in Agent Intelligence:** As noted, many interactive studies rely on pre-scripted videos or simple ECAs with basic dialog trees [29]. This severely limits the system's ability to respond to a user's unique statements, questions, or concerns, preventing the formation of a true therapeutic alliance.

This thesis project contributes to the field by directly addressing these gaps. We propose and document the technical implementation of an integrated system that does not just present a single VR experience, but manages an entire therapeutic loop.

The primary contribution is the novel architecture that combines a high-fidelity, standalone VR client with a state-of-the-art, LLM-driven conversational agent. This agent is not on the device but is hosted on a local server, which in turn queries a remote HPC for LLM processing. This hybrid, three-tier model allows for complex AI reasoning while maintaining the portability of a standalone VR headset. This system moves beyond simple cue exposure to simulate the entire therapeutic alliance, guiding the user from check-in and assessment through to skills practice and conclusion, all managed by a dynamic, intelligent, and responsive virtual therapist.

Chapter 3

System Architecture and Design

This chapter presents the high-level technical blueprint of the proposed system. The architecture was designed to solve a specific engineering challenge: integrating the advanced cognitive capabilities of a Large Language Model (LLM), which requires significant computational power, with the immersive but hardware-constrained environment of a standalone Virtual Reality headset.

3.1 Overview of the Decoupled Client-Server Architecture

To address the computational dichotomy between rendering and reasoning, a decoupled, three-tier architecture was adopted. Unlike tethered VR experiences where a powerful PC handles both rendering and logic, this project targets a Standalone use case to maximize portability and accessibility for patients.

The system is composed of three distinct entities:

1. **The Client:** The Meta Quest headset running the Unity application. It handles input capture (voice/hands) and audiovisual rendering.

2. **The Local Server:** A Python application running on a local machine. It orchestrates the session flow, manages databases, and acts as the bridge between the user and the AI.

3. **The LLM Service:** The core conversational capabilities of the system are powered by Google's Gemma 3 Large Language Model (specifically the 27-billion parameter version, gemma3:27b). The model was deployed and served using the Ollama inference framework, hosted on the High-Performance Computing (HPC) infrastructure of the Politecnico di Torino. This setup allowed for local, low-latency inference without relying on external cloud APIs.

This separation ensures that the VR client remains lightweight and responsive (maintaining high frame rates), while the heavy cognitive lifting is offloaded to external resources.

3.2 Server-Side Architecture

The server-side application is the decision-making core of the system. It does not merely relay messages; it maintains the state of the therapy, tracks patient progress, and determines the flow of the conversation.

3.2.1 Backend Technology Stack: Python, FastAPI, and HPC

The backend is implemented in Python, chosen for its dominance in the AI ecosystem. The web server framework is FastAPI, selected for its asynchronous capabilities (`async/await`), which allow the server to handle multiple concurrent operations—such as polling from the VR headset and waiting for LLM inference—without blocking.

The integration with the HPC (High-Performance Computing) infrastructure is handled via API calls. The Local Server constructs the prompt (including system instructions and conversation history) and sends it to the HPC. The HPC runs the inference using the Llama 3 model and returns the generated text. This setup ensures data privacy and control, as the LLM is hosted on university infrastructure rather than a public commercial cloud.

3.2.2 The Conversational Agent Core: LangGraph

To manage the complexity of a therapeutic session, simple "chain-of-thought" prompting was insufficient. The agent is built using LangGraph, a library that models the conversation as a graph of nodes and edges.

LangGraph allows for cyclic logic. Unlike a linear pipeline, the agent can loop back to previous states (e.g., repeating a question if the user is unclear) or branch into different paths based on user decisions. The graph's state is persisted after every step, allowing the server to "remember" the context even if the connection is temporarily lost.

3.2.3 Finite State Machine (FSM) Design

The therapeutic flow is modeled as a Finite State Machine (FSM). The session is divided into sequential "Modules," each representing a distinct phase of the clinical protocol. The agent transitions between these modules based on specific completion criteria.

Module 1: Check-in

This is the initialization phase. The virtual therapist welcomes the patient and establishes rapport. The primary goal is to assess the patient's current state ("How are you feeling today?"). The system remains in this state until the user provides a valid response that indicates readiness to proceed.

Module 2: Educational

In this phase, the therapist provides psychoeducation. The agent explains the purpose of the session and introduces the concept of craving management. This module is typically linear, consisting of an explanation followed by a confirmation from the user.

Module 3: Pre-Assessment

This module triggers the first interactive UI element. The therapist instructs the user to look at the tablet. The system transitions to a data-collection mode where the user must fill out the questionnaires and rate their baseline parameters (Craving, Pain, Mood). The conversation pauses until the data is submitted to the server.

Module 4: Therapeutic Intervention

This is the core of the session. Based on the user's preference, the system branches into one of the specific therapeutic protocols. The server dynamically loads the appropriate scene on the client.

The available therapies and their associated Virtual Environments (VEs) are:

- **Free Breathing Therapy:** A self-paced relaxation exercise.
  - *Scenarios:* **Modern Loft**, **Seaside**.
- **Guided Breathing Therapy:** A structured exercise where the user synchronizes breath with a visual guide.
  - *Scenarios:* **Hill**.
- **Visualization Therapy:** A guided imagery session where the therapist describes a calming narrative to distract from craving.
  - *Scenarios:* **Library**, **Seaside**.
- **Mindfulness Therapy:** A body-scan meditation focused on grounding and present-moment awareness.
  - *Scenarios:* **Hill**.

Module 5: Post-Assessment

Upon returning from the virtual environment, the user enters the re-evaluation phase. The therapist debriefs the experience and asks the user to answer to the same questions about craving, pain and mood, and then fill out the post-treatment assessment on the tablet. This allows for the immediate measurement of the intervention's efficacy (e.g., delta in craving levels).

Module End: Cyclical Loop

The session does not necessarily terminate after Module 5. The logic includes a routing node that asks the user if they wish to try another technique. If affirmative, the FSM loops back to the choice menu (Module 4); otherwise, it proceeds to the final goodbye and session termination.

3.3 Client-Side Architecture

The client application is developed in **Unity**. The target platform is the **Meta Quest 3**, operating in Standalone mode. This choice dictated several critical architectural decisions regarding rendering and performance optimization.

3.3.1 Platform and Engine: Meta Quest 3 and Unity

Choosing a Standalone platform means the application runs entirely on the headset's internal mobile chipset, without being tethered to a PC. This maximizes user comfort and allows the therapy to be administered in any quiet room. However, it also imposes strict limits on thermal output and battery usage, requiring highly optimized code and assets.

3.3.2 Choosing the Render Pipeline: URP vs. HDRP

One of the most pivotal technical decisions was the selection of the Render Pipeline. Unity offers the High Definition Render Pipeline (HDRP) for photorealistic graphics and the Universal Render Pipeline (URP) for optimized performance.

While HDRP offers superior lighting effects, it is computationally prohibitive for mobile VR hardware. HDRP relies on deferred rendering and heavy post-processing stacks that the Quest's mobile GPU cannot sustain at the required 72 or 90 frames per second (FPS). A drop below this frame rate in VR immediately causes motion sickness (cybersickness), which is unacceptable in a therapeutic context.

Therefore, URP was selected. URP utilizes a forward renderer that is highly optimized for mobile architectures. It allows for the use of custom shaders (such as the water shader in the Seaside scenario) and baked lighting solutions to achieve high visual fidelity without compromising performance. This ensures a smooth, nauseogenic-free experience for the patient.

3.3.3 Real-time Communication: Microsoft Azure Speech SDK

To maintain the illusion of a natural conversation, latency must be minimized. The client integrates the Microsoft Azure Speech SDK for two bidirectional streams:

1. **Speech-to-Text (STT):** Continuous recognition runs on the client, converting the user's spoken audio into text locally before sending it to the Python server. This reduces bandwidth usage compared to streaming raw audio.

2. **Text-to-Speech (TTS):** The server sends back text response, which the client sends to Azure to synthesize into a neural voice. The resulting audio stream is analyzed in real-time to drive the lip-sync animation.

3.4 Conversation Overview

Having introduced both the server and the client sides, below is the complete data flow of a conversational turn—from the user's voice input in VR to the generation of a therapeutic response — is illustrated in Figure 1.

Legend

- Headset side (Meta Quest 3)
- Local server (Engine)
- Large Language Model (HPC)

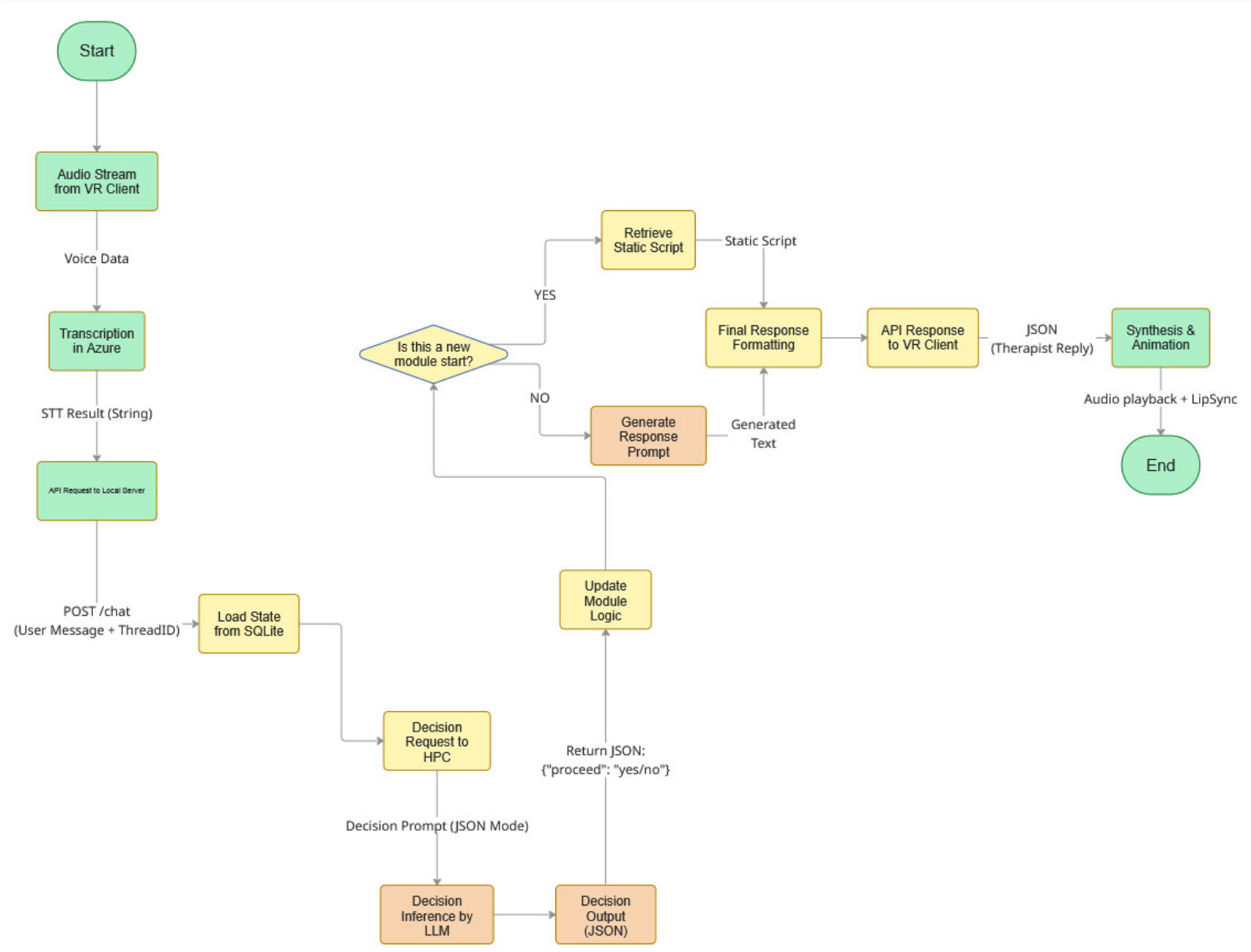

Figure 1: All the steps during the interaction between the user and the virtual human

3.5 Database and State Management

Effective therapy requires memory. The system implements a dual-layer data management strategy to handle both long-term clinical records and short-term conversational context.

3.5.1 Long-Term Patient Database (SQLite)

The `SqlitePatientManager` handles persistent storage. It uses a local SQLite database to store the "Patient Profile." This includes the user's unique ID, their progress history, and the cumulative `TreatmentLog` (a JSON structure containing the results of every assessment and therapy session). This allows the system to resume a patient's therapy across multiple days or sessions.

3.5.2 Short-Term Session Memory (LangGraph Checkpointer)

For the conversational agent to be coherent, it must remember what was said five minutes ago. This is handled by the LangGraph Checkpointer, also backed by SQLite. It stores the serialized state of the FSM (Finite State Machine), including the chat history and the current active module. This allows the server to be stateless between HTTP requests; every time a request comes in, the agent "rehydrates" its memory from the checkpoint.

3.5.3 Cross-Scene State Passing (`CrossSceneData`)

A specific challenge in Unity is that loading a new scene (e.g., switching from the Clinic to the Forest) destroys all objects in the previous scene. To preserve the session context during these transitions, a static class named `CrossSceneData` acts as a bridge.

When the `TherapyManager` initiates a transition, it saves the `ThreadID`, `CurrentModule`, and `TreatmentLog` into this static class. When the new scene loads, the local controller immediately reads from `CrossSceneData` to initialize itself. This ensures that the virtual therapist "remembers" the user and the current task even as the entire virtual world is replaced around them.

Chapter 4

Implementation: The Server-Side Agent

The core intelligence of the proposed system resides in the server-side application. Designed as a decoupled entity from the visualization client, the server acts as the central orchestration unit, managing the dialogue flow, state persistence, patient data, and the integration with the Large Language Model (LLM). This chapter details the technical implementation of the backend "Brain," developed in Python.

4.1 The FastAPI Application Server

The backend is built upon FastAPI, a modern, high-performance web framework for building APIs with Python 3.7+ based on standard Python type hints. FastAPI was selected for its native support of asynchronous programming (`async/await`), which is critical for handling multiple concurrent requests (e.g., network discovery, database writes, and LLM inference) without blocking the main execution thread. The server runs on a local machine, exposing its endpoints over the local area network (LAN).

4.1.1 Endpoint Design

The API exposes a RESTful interface designed to handle specific phases of the therapeutic session. Data validation is enforced via Pydantic models, ensuring that data structures like classes, dictionaries or API inputs exchanged between Unity and Python strictly adhere to the expected schema.

- `POST /chat`: This is the primary endpoint for the conversational loop. It receives a JSON payload containing the `thread_id` (session identifier) and the user's `message`. Upon receipt, the server instantiates or retrieves the LangGraph agent associated with that ID, processes the input through the Finite State Machine (FSM), and returns a JSON response containing the therapist's reply and the updated `current_module`. This endpoint is stateless in terms of connection (HTTP) but stateful in logic, relying on the checkpoint database.
- `POST /create_patient`: Used to initialize a new user profile. It accepts a `patient_name`, generates a progressive unique identifier (e.g., `user_01`), and initializes the database record with an empty state. Crucially, this endpoint also initializes the LangGraph checkpoint, "priming" the agent's memory so it is ready to receive the first message.

- `POST /create_test_session:` Designed for usability testing, this endpoint allows the creation of a patient that starts immediately from a specific therapy module rather than the beginning. It accepts a `base_thread_id` (existing user), a `scene_name`, and a `start_module`. It generates a unique test ID (e.g., `user_01_TestScene`), creates the patient record, and forces the LangGraph checkpoint to the specified `start_module`, enabling direct testing of specific therapeutic scenarios. In the usability tests documented in this thesis, this endpoint has been used for the therapy scenarios (seaside, modern loft, hill and library), with the module "therapy_usability", that was a short version of the "therapy_mindfulness".

- `POST /log_treatment:` This endpoint is dedicated to data collection. It is called by the client at the conclusion of a specific therapeutic exercise. It receives a structured object containing pre-assessment scores (craving, pain, mood), the therapy selected, and post-assessment scores. The server appends this data to the patient's historical log without overwriting previous entries.

- `POST /submit_pre_assessment:` Specifically saves the results of the pre-treatment surveys, filled by the patient in the clinic scenario. It receives the `thread_id` and a list of `SurveyResult` objects. The data is stored in the `pre_treatment_assessment_json` column of the patient's database record. The endpoint /submit_post_assessment consists of the same functions, applied to the post-treatment surveys.

- `POST /reset_starters:` A utility endpoint used to reset the conversation state flags. It accepts a `thread_id` and clears the `module_starters_sent` dictionary in the agent's checkpoint. This ensures that if a patient re-enters the application on a certain module (selected by the real clinician), the virtual therapist will re-deliver the necessary introductory scripts for each module.

- `POST /set_active_session:` Part of the PC-to-VR session initiation protocol. Called by the desktop "Intro" app (MenuManager), run by the it receives the `thread_id`, `patient_name`, and `current_module` of the selected patient. The server stores this data in a volatile global variable (`ACTIVE_SESSION_DATA`), effectively "staging" the session for the VR headset to pick up.

- `GET /get_active_session:` The counterpart to the previous endpoint. This is polled repeatedly by the VR headset (‘SessionPoller‘) on startup. If a session has been staged by the PC (from the "Intro" app), this endpoint returns the session data and clears the global variable. It also triggers the `STOP_BROADCAST_EVENT`, signalling the Network Discovery Service to cease broadcasting, as the connection has been established.

  item `GET /get_threads:` A debugging and management endpoint that returns a list of all active conversation threads currently stored in the LangGraph checkpoint database.

- `GET /delete_thread:` Allows for the deletion of a specific conversation thread from the checkpoint database, useful for resetting testing data or managing storage.

- `GET /get_patients:` Retrieves the full list of registered patients from the SQLite database, including their `thread_id`, name, current module, and last update timestamp. This is used to populate the patient selection list in the desktop "Intro" app, after clicking on the button "Load Session".

- `GET /surveys`: Returns the static configuration data for all questionnaires (questions, types, options) defined in `SURVEY_DATA`. This allows the Unity client to dynamically build the UI for the surveys without hardcoding questions in the application.
- `GET /therapies`: Returns a list of available main therapeutic categories (e.g., Breathing, Mindfulness, Visualization). For each therapy, it provides an ID, title, description, and a dynamically generated `imageUrl`. The server constructs the image URL using its own current IP address (e.g., `(http://192.168.1.)X:9999/...`), ensuring the VR client can download the assets regardless of the network configuration.
- `GET /breathing_therapies`: Similar to the therapies endpoint, but returns the specific sub-options for the Breathing category (Guided vs. Free Breathing), with title, description and dynamically generated image URLs.

4.2 Network Discovery Service

In a decoupled local network architecture, the IP address of the server is subject to change (DHCP) or variation between different testing environments (e.g., laboratory vs. clinic). The PC hpsting the server could even change. Hardcoding the server's IP address into the VR client build is impractical, as it would require to rebuild the application for every network change. To solve this, an automatic Network Discovery Service was implemented.

4.2.1 Implementation of UDP Broadcast for Server Discovery

The solution leverages the User Datagram Protocol (UDP) to broadcast the server's presence to the local network.

On startup, the Python server spawns a background daemon thread. This thread identifies the machine's active network interface (filtering out loopback addresses) and begins broadcasting the string `THERAPY_SERVER_HERE`) to the broadcast address (e.g., `192.168.1.255`) on a specific port (`9998`) at regular intervals.

Simultaneously, the VR client starts a listening thread on port `9998`. When it intercepts the server's broadcast message, it extracts the sender's IP address. This IP is then used to construct the base URL (e.g., `http://192.168.1.105:9999`) for all subsequent HTTP API calls.

To ensure robustness, the Python implementation includes thread-safe socket handling: the socket is created, used, and closed within the worker thread to prevent resource conflicts, and the broadcast loop automatically terminates once a client successfully connects and requests a session.

4.3 Session Initiation Protocol (PC-to-VR)

A significant challenge in standalone VR development is the disjointed nature of the hardware: the researcher operates a PC, while the patient wears the headset. To bridge this gap, a custom "Session Initiation Protocol" was implemented using a polling mechanism.

This architecture uses two of the endpoints mentioned above: `/set_active_session` and `/get_active_session`.

1. The researcher uses a desktop interface (the "Intro" app) to select a patient and sends the configuration to `/set_active_session` by clicking the "Confirm" button, whether it is a new session or a loaded one. The server stores this configuration in a volatile global variable.

2. The VR headset, upon boot, enters a "Waiting Room" scene where it continuously polls `/get_active_session` every 3 seconds.

3. Once the server returns a valid configuration (non-null, meaning that the set_active_session has been successfully set by launching a session from the Intro app), the VR client saves the `thread_id` locally and loads the clinical environment.

This approach eliminates the need for manual data entry inside the VR headset, streamlining the user experience.

4.4 The Conversational Agent

The cognitive core of the virtual therapist is built using `LangGraph` and `LangChain`.

4.4.1 LangGraph

LangGraph is a library designed to build stateful, multi-actor applications with Large Language Models (LLMs). Unlike simple linear chains, LangGraph allows the definition of a state graph (specifically, a Fine State Machine) , which is essential for modeling a therapy session that can loop, repeat, or branch based on user input.This also allows the LLM to loop through steps until a goal is met.
Key Components:

- The State: A shared data structure (schema) that tracks the conversation. It persists across every step of the graph. As the agent works, it updates this State (e.g., adding messages, storing tool outputs).

- Nodes: These are Python functions or LangChain chains. Each node performs work (e.g., "Call the LLM," "Search Google") and updates the State.

- Edges: These define the path between nodes.

- Normal Edges: Go from A to B always.

- Conditional Edges: The "brain." The system looks at the State and decides where to go next (e.g., "If the tool returned an error, go to 'Retry'; otherwise, go to 'End'").

4.4.2 LangChain

LangChain is an orchestration framework designed to simplify building applications with Large Language Models (LLMs). Its primary goal is composability. It allows to chain together different components (prompts, models, data retrievers) to create a sequence of events.
Key Components:

- Prompts: Templates that structure the input for the LLM

- Models (LLMs): The engine (e.g., GPT-4, Claude) that processes the text.
- Retrievers (RAG): Tools that fetch external data (like PDFs or databases) to give the LLM context.
- Output Parsers: Tools that clean up the AI's messy response into structured data (like JSON).
- LCEL (LangChain Expression Language): A syntax used to pipe these components together easily, like this: Prompt | Model | OutputParser.

Below is an example of the chain used for the process of decision making:

```
self.decision_chains = {
module_id: ChatPromptTemplate.from_messages([
("system", prompt_text),
MessagesPlaceholder(variable_name="messages")
]).partial(format_instructions=decision_parser.get_format_instructions()) |
self.decision_llm | decision_parser
for module_id, prompt_text in decision_prompts.items()
}
```

In this extract from the chat agent code, the chain is divided in three parts:

- `Prompt construction`, that consists of the prompt_text for that module and a spot for the chat history between the human and the therapist (messages). It format instructions are also injected in the prompt.
- `Decision`, where the formatted prompt is sent to the Large Language Model.
- `Parsing`, where the raw text response from the LLM is passed to the parser, which converts it into a structured Python object

4.4.3 The ChatAgentStm Class Structure

The agent is encapsulated in the `ChatAgentStm` class. This class manages the initialization of the LLM (via an interface to the HPC infrastructure), the connection to the checkpoint database (for memory persistence), and the compilation of the state graph.

The graph state is defined by a Pydantic `BaseModel` named `ChatStateStm`, which holds:

- `messages`: The list of conversation history (Human and AI messages).
- `current_module`: A string tracking the current phase of therapy (e.g., "1", "3b", "therapy_breathing"). The string is set to 1 as default.
- `module_decision`: The output of the decision-making node ("yes", "no", "restart"). This variable will be handled by the LLM and it is set to "no" as default value.
- `summary`: That is a short summary of the messages (between the the virtual agent and the human) that exceed the window size of messages, that represents the memory of the LLM. Due to excessive latency and slow time responses, this feature has been temporarily disabled.
- `module_starters_sent`: A dictionary of flags to track which introductory scripts have already been recited by the therapist.

4.4.4 State Nodes

The graph logic is split into specialized nodes to separate reasoning from execution:

1. `_manage_messages_node`: This node checks if the length of the record of all messages exceeds the window size. If it does, it keeps the last messages that fit into the window size and deletes the older ones. A previous version of this node, that is now annotated in the code, used to make a summary of the messages that were exceeding the window size.

2. `_decide_transition_node`: This node invokes the LLM with a specific prompt based on the current module. It asks the LLM to analyze the user's last message and determine if the conditions to advance to the next stage are met (e.g., "Has the user finished the breathing exercise?"). It outputs a structured "yes/no" decision.

   The structure of the decision prompts is as shown in the following example, taken from `DECISION_PROMPT_1`:

   > *You are a conversation flow controller. Your task is to decide if the conversation should proceed to the next module.*
   >
   > *The current module is '1' (Check-in). The core question is: "How have things been going with your treatment and daily life?"*
   >
   > *Your goal is to determine if the user has provided ANY kind of answer to this question, even if it's vague, polite, or short. You need to interpret their general sentiment.*
   >
   > *Proceed to the next module ("yes") if the user's last message:*
   >
   > - *Directly answers the question (e.g., "I'm doing well," "It's been hard," "Not so well").*
   > - *Gives a vague but conclusive-sounding response (e.g., "Well, thank you," "Okay," "I'm fine").*
   > - *Indicates they are ready to move on.*
   >
   > *Do NOT proceed ("no") if the user's last message:*
   >
   > - *Asks a clarifying question (e.g., "What do you mean by treatment?").*
   > - *Changes the subject completely.*
   > - *Expresses confusion.*
   >
   > *Based on the last user message, have they provided a response that allows the conversation to move forward?*
   >
   > *Format the Output: You MUST format your final output as a single JSON object.*
   >
   > *The JSON object must have a single key: "proceed".*
   >
   > *The value for "proceed" MUST be the literal string "yes" or the literal string "no".*

3. `_update_module_node`: This node contains the deterministic logic of the Finite State Machine (FSM). It does not call the LLM. Instead, it takes the "yes/no" decision and the `current_module` to calculate the `next_module` using a predefined transition map.

   This is the module sequence used by this node:

"1": "2", "2": "2b", "2b": "2c", "2c": "3",
"3":"3b", "3b":"3c", "3c":"3d", "3d":"4",
"4":"routing",
"therapy_free_breathing":"4b",
"therapy_guided_breathing":"4b", "therapy_visualization":"4b",
"therapy_mindfulness":"therapy_mindfulness_b",
"therapy_mindfulness_b":"4b",
"therapy_gamification":"4b",
"therapy_usability":"therapy_usability_b",
"therapy_usability_b":"4b",
"4b":"4c", "4c":"4d","4d":"routing2",
"5":"end"

It also handles a routing logic, which is branching to different therapy scenes based on keyword detection:

if "guided breathing" in last_message:
next_module = "therapy_guided_breathing"
elif "free breathing" in last_message:
next_module = "therapy_free_breathing"
elif "visualization technique" in last_message:
next_module = "therapy_visualization"
elif "mindfulness technique" in last_message:
next_module = "therapy_mindfulness"
elif "gamification technique" in last_message:
next_module = "therapy_gamification"
else: next_module = "therapy_fallback"

A second routing logic is used to understand if the module 5 has to be done or avoided (that is the post-treatment assessment, that has to be filled only when the patient finishes the first treatment of the session).

Another important feature in this function is the logic that decides what to do after module `"end"`. If the string `state.module_decision` contains `"restart"`, the current module will go back to module 3b in order to start another therapy treatment; if `state.module_decision` contains `"stop"`, the session will end. The mechanism behind this is fully explained in 4.4.6.

4. `_get_response_for_module:` This function provides the actual response to the Unity client. First, it checks if the starter response for a certain module has been already delivered. If not, the starter module for that module is chosen as answer from the therapist. If it has already been used for that module, the answer will be asked to the LLM, giving the context of the messages and a response prompt with all the guidelines for that module.

4.4.5 Prompt Engineering: Starter vs. Response Prompts

To maintain clinical validity while allowing for empathetic interaction, a hybrid prompting strategy was employed.

Starter Prompts are static, pre-written scripts used at the beginning of each new module. They ensure that critical clinical instructions (e.g., "Please rate your craving from 0 to 10") are

delivered verbatim and accurately. The system uses the `module_starters_sent` flag to ensure these are spoken only once per module. The following example taken by module 1:

STARTER_PROMPT_1 = """Hi, <break time="500ms"/> my name is Salsa and I will assist you during these sessions to help in your recovery. <break time="1500ms"/> I appreciate you coming in today. Recovery (from substance dependence) can be challenging, and I want to check in on how you're feeling. How have things been going with your treatment and daily life?"""

Response Prompts are dynamic system instructions sent to the LLM during the conversation loop. They define the therapist's persona (empathetic, professional) and the specific goal of the current module (e.g., "Encourage the user to focus on their breath"). This allows the agent to generate context-aware responses to user questions or resistance. The following example taken by module 1:

RESPONSE_PROMPT_1 = CHAT_AGENT_SYS_TEMPLATE +
""" Current Task: Check-in

- *If this is the first message, greet the user and say: "Hi, my name is Salsa and I will assist you during these sessions to help in your recovery. I appreciate you coming in today. Recovery (from substance dependence) can be challenging, and I want to check in on how you're feeling. How have things been going with your treatment and daily life?"*
- *The core question is: "How have things been going with your treatment and daily life?"*
- *If the user asks a question, answer it concisely and if the user didn't answer the core question, repeat your core question at the end of your sentence*
- *If the user has just answered, give a brief, validating acknowledgment like "Thank you for sharing."*

"""

Where CHAT_AGENT_SYS_TEMPLATE is a base prompt where the context, all the communication guidelines, roles and limitation are written for the LLM. It is always given to the LLM with the response prompt.

4.4.6 The EndRoutingDecision Parser

At the end of a therapy cycle, the user can choose to terminate the session or start a new exercise. This represents a complex branching decision. To handle this reliably, the system utilizes LangChain's PydanticOutputParser.

The LLM is instructed to output its decision not as free text, but as a structured JSON object conforming to the `EndRoutingDecision` schema (containing fields like `next_step: "restart" | "stop"`). This ensures that the agent's reasoning can be deterministically parsed by the Python code to trigger the correct state transition (looping back to module 3b or

exiting to the 'end' state), eliminating ambiguity. The routing logic is invoked in the decide_transition_node and the result is stored in the string state.module_decision, which is used in the update_module_node in the "routing2" part. Here is the prompt given to the LLM in order to make the decision:

> *DECISION_PROMPT_END = """ You are a flow controller. The user has just been asked if they want to try another therapy technique or end the session. Your task is to determine the user's intent from their last message.*
>
> *- If the user expresses a desire to continue, try another technique, or go back (e.g., "yes, let's do another one", "go back", "I want to try mindfulness"), you must choose 'restart'. - If the user expresses a desire to stop, end the session, or says no (e.g., "no thanks", "I'm done", "let's end the session", "no, i don't want another session"), you must choose 'stop'.*
>
> *Based on the last message, what is the user's intent?*
>
> *Format the Output: You MUST format your final output as a single JSON object with the key 'next_step'. """*

4.5 Data Management

Persistence is a key requirement for longitudinal therapy. The system utilizes SQLite for its reliability, serverless architecture, and ease of deployment.

4.5.1 The SqlitePatientManager Class

All database interactions are abstracted through the `SqlitePatientManager` class. All the functions in this class are invoked by the endpoints in rest_api_server.py through the chat agent (chat_agent_stm.py). This class handles the connection to the `patient_data.sqlite` file and exposes methods for creating patients, getting the list of all patients, updating their state, and logging treatments. It also saves the results of pre and post treatment assessments into the database. It ensures that database transactions are atomic and thread-safe, which is crucial when the async server handles multiple requests.

4.5.2 Generating Progressive User IDs (user_01, user_02...)

To facilitate data analysis and usability testing, a readable ID system was preferred over random UUIDs. The the `SqlitePatientManager` class implements a logic to scan the database for existing IDs matching the pattern `user_XX`. It extracts the numeric suffix, finds the maximum value, increments it, and formats the new ID (e.g., `user_07`). This also allows researchers to easily track and order participant data during the testing phase.

4.5.3 Appending TreatmentLog Data

A critical feature of the the `SqlitePatientManager` class is the ability to store multiple therapeutic attempts within a single patient record. The `treatments_log_json` column in the database stores a JSON array of treatment objects.

When a therapy session concludes, the `add_treatment_log` function performs a "Read-Modify-Write" operation: it retrieves the existing JSON array, deserializes it into a Python

list, appends the new dictionary containing the session data (timestamp, pre-scores, chosen therapy, post-scores), serializes it back to JSON, and updates the record. This preserves the complete history of the patient's interactions within the single relational database row. Here is an example of the treatments_log_json cell for a patient who has done multiple treatments inside one session:


```
{
"datetime": "09/11/2025 - 21:51",
"pre_craving": "4.",
"pre_pain": "3.",
"pre_mood": "4.",
"therapy_chosen": "Guided Breathing Technique",
"post_craving": "1.",
"post_pain": "4.",
"post_mood": "1."
},
{
"datetime": "09/11/2025 - 22:23",
"pre_craving": "5.",
"pre_pain": "6.",
"pre_mood": "8.",
"therapy_chosen": "Mindfulness Technique",
"post_craving": "5.",
"post_pain": "1.",
"post_mood": "4."
}
```

Chapter 5

Implementation: The Client-Side VR Application

5.1 The virtual agent (therapist)

The virtual agent was created using Character Creator 4, a comprehensive 3D character creation software developed by Reallusion. It is designed to enable artists, game developers, and animators to generate, customize, and render realistic or stylized characters that are fully rigged and animation-ready.

The starting point for this character was the Neutral_Female (CC3+) avatar provided by Character Creator. The morphs were then adjusted in the Morphs panel, adding the visual features of other avatars (such as CC3+_Katherine and CC3+_Jody). Other settings were set, like height, chest, back and makeup. All clothes have been selected from the standard Cloth pack in the "Item" section, in the Content panel. (Fig. 2)

Most of the animations were selected from the Animation pack in Character Creator. Only two - Stand-To-Sit and Sitting Idle (used in the Clinic Scene) - were imported into CC4 from Mixamo.

In order to make the lipsync on Unity work correctly, the right viseme setting had to be chosen in the Facial Profile Editor panel. Visemes represents the blendshapes that are then used by Unity in order to perform facial movements, such as human phonemes. The standard visemes in CC4 are the `8+7 Phoneme Pair`, but for the lipsync we chose (OVR Lipsync), we needed a viseme pair that could directly perform the phonemes (AH, EH, OO, etc.). This is the reason why the viseme setting has been switched to `1:1 Direct` (Fig. 4)

Figure 2: Character Creator 4 - Therapist

Figure 3: Viseme Settings

5.1.1 Optimization

First, I enabled runtime statistics for polygon counts and other metrics (Edit > Preference > Display > Check "Info").
These are the steps that were followed in order to optimize the therapist on Character Creator:

- Hair (4 pieces in my case): Each piece was first converted to an Accessory. Then, I applied Polygon Reduction - Object from the Modify/Attribute tab, setting it to 60%.
- Clothes: I applied Polygon Reduction - Object to each piece here as well, using variable percentages.
- Body Parts: Referring to CC_Base_Tongue, CC_Base_Body, etc., I unchecked `Smooth Mesh` for all parts (except the Eyes). This drastically reduces the triangles count.

5.1.2 Export

The following part refers to the panel that appears when clicking on:
File -> Export -> FBX -> Clothed Character.
Below are the main settings of the export panel:

- Target set to Unity with FBX Options: Mesh and Motion.
- `Delete Hidden Faces` was checked in order to avoid rendering parts of the body that were covered by clothes.
- `Use Smooth Mesh` was not enabled.
- In the Advanced Settings, it is crucial to select `Mouth Open as Morph` and `Convert Skinned Expressions to Morphs`, in order to make the mouth open correctly in Unity.

5.1.3 Unity Import

The CCiC Importer URP 2.1.0 plugin for Unity was used to build all materials and animations for the virtual agent on unity. These are the settings used in the plugin:

- High Quality Materials
- Medium Texture Size
- High Texture Quality
- Parallex Eyes
- Two Pass Hair
- None of the checkboxes in the section Features were checked
- Bake Default Shaders
- Bake Separate Prefab

5.2 LipSync study and plugin integrations

As far as lipsync for the therapist is concerned, three main plugins have been tested and compared. These are:

- Salsa
- uLipSync
- Oculus

The aim of this study was to best recreate the human visemes for the standard phonemes. The following is a chart of the phonemes (4), followed by a picture of the visemes corresponding to the phonemes (5).

Viseme class	phonemes	example
0	none	na
1	p, b, m	put, bed, mill
2	f, v	far, voice
3	T,D	think, that
4	t, d	tip, doll
5	k, g	call, gas
6	tS, dZ, S	chair, join, she
7	s, z	sir, zeal
8	n, l	lot, not
9	r	red
10	A:	car
11	e	bed
12	I	tip
13	Q	top
14	U	book

Figure 4: Phonemes

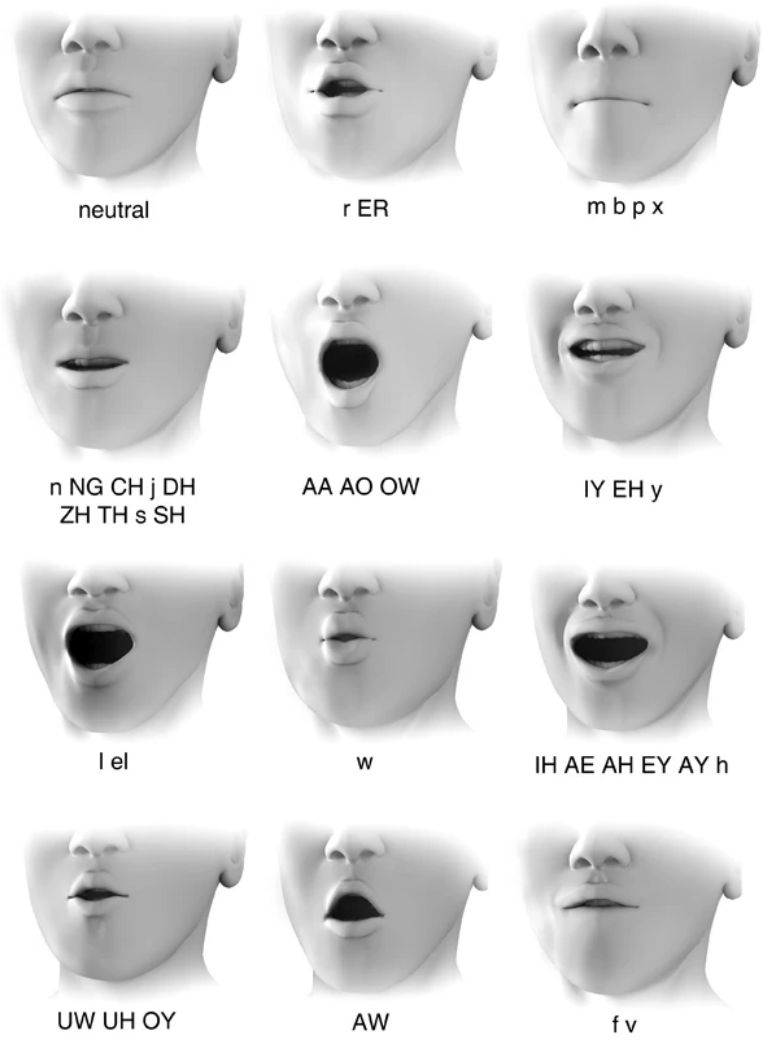

Figure 5: Visemes

Salsa The Salsa plugin provides a wide series of components that automatically set up with the virtual agent's parts of the body and bones and that are able to control not only lipsync, but also head, eyes and face emotes. Talking about lipsync, Salsa was tested with two different Character Creator facial configurations (explained in 5.1):

- `8+7 Phoneme Pair Visemes configuration` : here, a combination of many blendshapes (e.g. V_Open, V_Tight_O, V_Dental_Lip, etc.) was used to recreate one single viseme (e.g. AA). In the comparison between the different plugins, we will relate to this approach as **Salsa**.
- `1:1 Direct configuration` : here, one single blendshape was used to recreate one viseme. We will relate to this approach as **Salsa (profilo 1.1)**.

uLipSync This plugin, using the 1:1 Direct configuration, is built specifically for lipsync and let the user specify the avatar's blendshape related to every viseme. This can be done in its main component in the inspector, called `U Lip Sync Blend Shape`. Furthermore, another component let the plugin adjust itself based on the voice pronoucing the phonemes. That is

why uLipSync was tested with the Azure Text-To-Speech voice that I was using at that moment (we will relate to this as **uLipsync (Emma)**) and also with my voice (we will relate to this as **uLipsync (Tizi)**).

Oculus This plugin, using the 1:1 Direct configuration, let the user specify the number of blendhsape related to every viseme. The numbers go from 0 to the length of the blendshapes list of the virtual agent, where 0 is the first blendshape, that is `None`. We will relate to this approach as **Oculus**.

A comparison was made between all different plugins, making them run separately and then all together (6). In the end, Oculus seemed to be the most likely and it was the chosen one.

The chosen gesture to speak with the therapist is to tap thumb and middle finger together once to start voice recording. During the recording, the hand will turn red and display a small microphone icon. To stop speaking, tap thumb and middle finger together again.

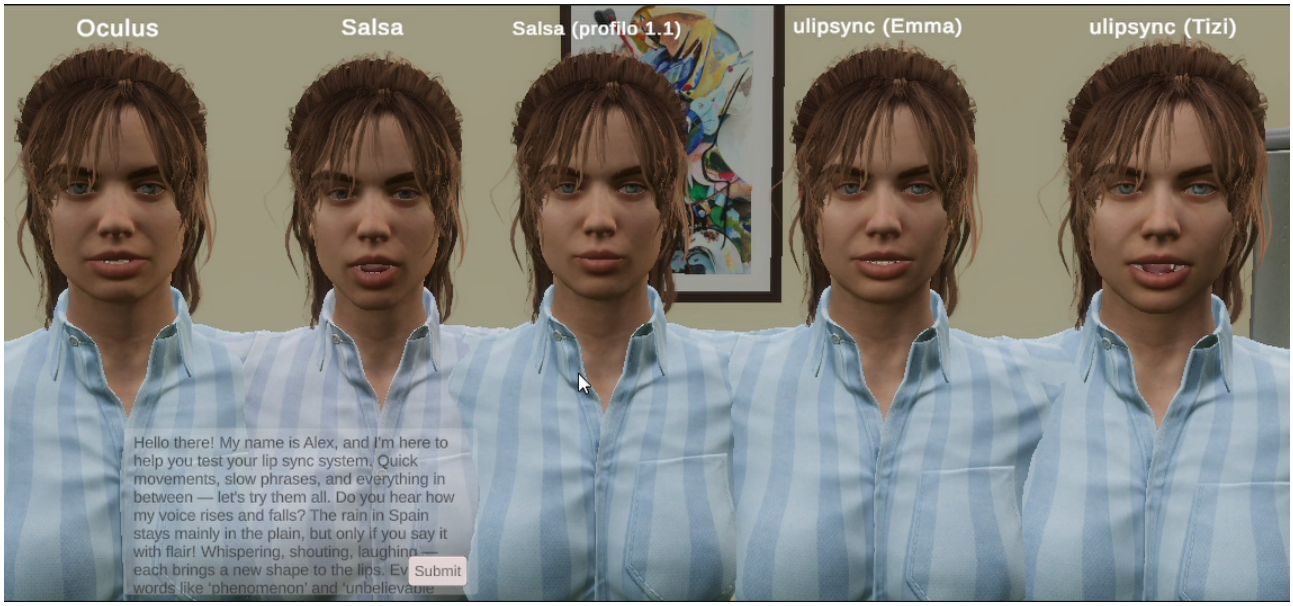

Figure 6: Comparison between lipsync plugins

5.3 The Meta SDK

To achieve a high level of immersion and natural interaction within the virtual environment, the client-side application leverages the Meta Quest Integration SDK (formerly Oculus Integration SDK). Specifically, the implementation adopts the "Building Blocks" workflow. This architectural paradigm provides modular, pre-configured components that ensure best practices in VR development are met while significantly reducing the complexity of setting up tracking and physics interactions.

This chapter details the implementation of the three core pillars of the user experience: the player's presence (Camera Rig), the user interface interaction (Poke System), and object manipulation (Grab System).

5.3.1 The Player Rig: OVRCameraRig and Tracking

The foundation of the VR experience is the `OVRCameraRig`, which serves as the user's eyes and hands within the virtual world. Using the Building Blocks system, the standard Unity camera was replaced with the Meta **Camera Rig** block.

This prefab acts as a wrapper for the tracking data provided by the headset's sensors. Its hierarchy is designed to map physical movements to virtual space:

- `TrackingSpace`**:** The root object that aligns the virtual coordinate system with the user's physical play area.
- `CenterEyeAnchor`**:** Represents the midpoint between the user's eyes. This object carries the `Camera` component and the URP rendering data. This component is also the target for the "stale depth buffer" fix, where the `Opaque Texture` is dynamically toggled to prevent rendering artifacts in the water shader.
- `HandAnchors` **(Left/Right):** These transforms track the position and rotation of the controllers or, in the case of Hand Tracking, the user's actual hands.

To enable interaction, the Camera Rig was augmented with the Interaction OVR Camera Rig building block. This extension injects the necessary `Interactors` (ray casters, poke interactors, and grab interactors) into the hand hierarchy, allowing the system to detect and respond to interactive elements in the scene.

5.3.2 UI: The Poke Interaction

A critical requirement of the application was to make the in-VR tablet feel like a tangible device rather than a floating menu. Traditional VR interfaces often use "Ray Interactors" (laser pointers), which can break immersion by simulating a remote control. Instead, this project implements the Poke Interaction system, allowing users to physically press buttons on the virtual tablet with their index finger.

This system requires a precise configuration of components on both the "actor" (the hand) and the "target" (the UI button).

5.3.3 The Hand: Poke Interactor

The `PokeInteractor` component is attached to the user's virtual hand index finger. It continuously casts a short-range ray to detect proximity to valid surfaces. When the finger penetrates the surface plane of a UI element, the SDK calculates the depth of the press and triggers the corresponding events.

5.3.4 The Tablet: Poke Interactable and Surfaces

Standard Unity UI components (like `Button`) are not natively aware of 3D physical contact. To bridge this gap, the tablet's SurveyCanvas and TherapyCanvas contained a gameobject called `ISDK_PokeCanvasInteraction` with the following SDK components:

1. `PokeInteractable`**:** the core SDK component that handles the logic of the interaction. It listens for the `PokeInteractor` entering the collider and calculates the "Select" state based on the penetration depth.

2. `PointableCanvas:` this component makes the Canvas (with all his buttons) pointable and selectable from the Meta SDK Hands.

3. `Surface child:` the `ISDK_PokeCanvasInteraction` has a `Surface` gameobject as a child. This defines the geometric plane against which the "poke" is measured, ensuring that the user cannot push their finger infinitely through the tablet without resistance feedback being registered visually.

5.3.5 Object Manipulation: The Grab System

To further enhance the sense of embodiment, the tablet is not static; the user can pick it up, hold it, and move it for a better reading angle. This is achieved using the SDK's Grabbable building block.

The interaction is asymmetrical:

- **The Actor:** The user's hands are equipped with `HandGrabInteractors`. These components use heuristic algorithms to detect when the user's fingers curl around an object, initiating a grab attempt.

- **The Object:**
  The tablet GameObject is equipped with the following components:

  - `BoxColliders`, to specify the parts where the user can pick up and grab the tablet. item `Grabbable`, that really makes the tablet grabbable. Needs the ISDK_HandGrabInteraction as parameter.
  - `InteractableUnityEventWrapper`, that acts as a translation layer. It subscribes to the low-level SDK events (such as `WhenGrabbed`) and invokes standard Unity events.

  A child block called `ISDK_HandGrabInteraction` has the following components:

  - `Hand Grab Interactable`
  - `Grab Interactable`
  - `Grab Free Transformer`

5.3.6 Physics and Constraints

The `Grabbable` component manages the object's physics (RigidBody) during the interaction. When grabbed, the object's movement is coupled to the hand's movement.

A crucial aspect of the implementation was managing the state of the application based on this interaction. As described in the logic of the `GeminiTTSController`, the system must know when the user is holding the tablet to prevent accidental voice recording (the "pinch" gesture could be confused with holding the tablet).

To achieve this, the `InteractableUnityEventWrapper` was again utilized on the tablet. The `WhenSelect` event (triggered when the grab begins) and the `WhenUnselect` event (triggered when released) are wired directly to the `NotifyTabletGrabbed(bool)` method in the main controller. This ensures that voice input is disabled exactly when the user is physically interacting with the tablet, preventing state conflicts.

5.4 The Clinical Environment: Scene Design and Optimization

The "Clinic Scene" (fig.7 and 8) serves as the central hub of the application. It is the first environment the user enters, designed to simulate a modern, professional, yet welcoming therapist's office. The design philosophy prioritized a sense of calm and safety, essential for reducing the initial anxiety associated with therapy.

Figure 7: Clinic - Office scene

Figure 8: Clinic - Office scene

5.4.1 Environment Composition and Asset Sourcing

The spatial layout was conceptualized based on real-world reference images to ensure architectural plausibility (fig.9 and 10). To achieve a high level of visual fidelity while maintaining development efficiency, a hybrid asset strategy was employed, combining commercial packages, open-source models, and custom modeling.

- **Commercial Assets:** The core furniture, including the cabinets and the main desk, was sourced from the "Apartment Kit" by Brick Project Studio [32]. These assets provided high-quality textures and realistic proportions suitable for a VR setting.
- **Custom Modeling:** To fit the specific dimensions of the virtual room, several structural elements were modeled from scratch by the author. These include the office walls, the window curtains, and the custom shelving units. This ensured the environment matched the precise scale required for the user's tracking space.
- **Imported and Composite Models:** Special attention was given to the seating arrangements. The sofa was imported from Sketchfab [33], selected for its detailed fabric texture which enhances the sense of presence when viewed up close in VR. The therapist's chair was created by compositing different parts of various 3D models found online to create a unique asset that fit the aesthetic of the room.

Figure 9: Reference 1

Figure 10: Reference 2

5.4.2 Mesh Optimization for Standalone VR

Developing for the Meta Quest requires strict adherence to performance budgets, particularly regarding the polygon count (poly-count). High-poly assets, while visually pleasing on PC, can cause frame rate drops on mobile VR chipsets, leading to motion sickness.

To address this, every imported object underwent a rigorous optimization pipeline using Blender. The primary technique used was the **Decimate Modifier**:

1. Models were imported into Blender.
2. The `Modifiers` panel was accessed, selecting `Add Modifier -> Generate -> Decimate`.
3. The "Collapse" ratio was adjusted to reduce the vertex count while preserving the silhouette and essential visual details of the object.

This process significantly reduced the rendering load without a perceptible loss of quality inside the headset.

5.4.3 Lighting Strategy: Baked Illumination

Lighting is crucial for establishing the mood of the therapy session. However, real-time lighting calculations (e.g., dynamic shadows) are computationally expensive. Therefore, a fully Baked Lighting strategy was implemented. This process pre-calculates light bounces and shadows into static textures ("lightmaps") which are overlaid on the geometry, offering high-quality visuals at zero runtime cost.

The lighting setup consists of three main components:

- **Directional Light:** Simulates the sun, providing the primary illumination and casting sharp, defined shadows from the window frames to ground the user in the environment.
- **Spot Lights:** A series of spots were placed on the office ceiling to simulate recessed artificial lighting, highlighting the desk and the seating area where the interaction takes place.
- **Area Lights:** Three Area Lights were positioned at the windows to simulate environmental light scattering (skylight) entering the room, softening the shadows and creating a realistic ambient fill.

5.4.4 Navigation and Pathfinding

To enhance the realism of the virtual therapist, static positioning was avoided. The therapist enters the room and walks to her chair, requiring a dynamic movement system that acknowledges the physical layout of the room.

A NavMesh (Navigation Mesh) was generated for the scene. The NavMesh Surface component analyzed the scene geometry (desk, sofa, walls) and defined the "walkable" areas. This allows the therapist's AI agent to calculate pathfinding in real-time, ensuring she naturally avoids obstacles like the desk corner or the coffee table while moving to her target destination (‘Sitting Idle‘ position).

5.5 The main script: `GeminiTTSController`

The `GeminiTTSController` script is the most critical component of the client-side application. It functions as a monolithic orchestration unit that bridges three distinct domains: the user's physical interactions in VR, the cognitive processing of the remote Python server, and the audiovisual rendering of the virtual therapist via Microsoft Azure services.

This script is responsible for the entire lifecycle of a conversation turn, from capturing the user's voice to animating the therapist's response. The following subsections detail the implementation of its core functions and logic flow.

5.5.1 Initialization and Session State Management

The application's state is not static; it changes depending on whether a session is just beginning or resuming after a therapeutic exercise. The `Start()` method implements a conditional initialization logic based on the persistent `CrossSceneData` class.

- **Returning from Therapy:** If the `CrossSceneData.ThreadId` is populated, the controller recognizes that the user is returning from a specific module (e.g., Guided Breathing). In this case, it retrieves the session ID and the current module state, clears the static data to prevent state pollution, and executes the `therapistController.SetupAsSeated()` method to position the avatar instantly in the counseling chair. It also checks for any `MessageToSendAfterDelay` to trigger a context-aware welcome back message.
- **Fresh Start:** If no cross-scene data is found, the controller treats this as a new session. It loads the patient credentials from `PlayerPrefs`, triggers the `ResetStarterFlagsOnServer` coroutine to ensure the server-side agent resets its conversation scripts, and instructs the therapist to perform the "Walk-In" animation sequence (`SetupForWalkIn`), simulating the start of a real-world appointment.

5.5.2 Input Handling and Safety Gates

User input is managed through the `OnRecordButtonPressed` and `OnRecordButtonReleased` methods, which are bound to the Oculus Interaction SDK's pinch gesture or controller inputs. To ensure a robust user experience, this logic is protected by several boolean "gates":

1. `isProcessing`**:** Prevents the user from speaking while the system is already transcribing audio or waiting for a server response.
2. `terOcc` **(Therapist Occupied):** Ensures the user cannot interrupt the therapist while she is speaking.
3. `isConversationReady`**:** A flag that remains false until the initial "Hello" sequence is complete, preventing premature interaction.
4. `isTabletGrabbed`**:** A crucial usability feature. If the user is physically holding the virtual tablet (detected via the `NotifyTabletGrabbed` callback), voice recording is disabled to prevent conflict between moving the UI and triggering the voice command.

When validation passes, the method provides immediate visual feedback by changing the user's virtual hand color to red via the `handRenderer`, activates the visual microphone indicator, and initiates the Azure `StartContinuousRecognitionAsync` process.

5.5.3 The Core Pipeline: ProcessAndSpeak

The `ProcessAndSpeak` method is the asynchronous heart of the controller. It orchestrates the transformation of raw text into a synchronized audiovisual performance. Its execution flow is as follows:

1. Server Communication

The method first calls the `SendChatMessage` task (wrapping the `PostRequest` coroutine). This sends the user's transcribed text to the remote Python server via a REST API call. The coroutine waits for the JSON response, which contains both the textual reply from the Large Language Model (LLM) and the updated state of the therapy module.

2. SSML Injection and Emote Mapping

Before generating audio, the raw text response is processed by the `InjectSSMLEmoteMarks` helper function. This function scans the text for emotional keywords defined in the `emoteMappings` list. When a match is found, it wraps the word in a specific SSML (Speech Synthesis Markup Language) tag:

```
<mark name='emote:Happy'/> word
```

This markup is essential for the lip-sync system, as it embeds "invisible" events into the audio stream that the engine can detect during playback.

3. Audio Synthesis and Playback

The annotated SSML string is sent to the Azure `SpeechSynthesizer`. Upon successful synthesis, the resulting byte stream is converted into a Unity `AudioClip` using the custom `WavByteArrayToAudioClip` utility, which parses the RIFF/WAV header structure. The clip is then assigned to the `characterAudioSource`. Crucially, the method parses the timing of all SSML bookmarks and sorts them into the `upcomingMarks` queue before playing the audio.

4. Post-Speech Logic

The method awaits the completion of the audio playback using `Task.Delay`. Once finished, it performs a keyword check on the text (e.g., checking for the word "rise") to determine if a special cinematic sequence (`TherapySessionSequence`) should be triggered, such as the therapist fading out to begin a meditation session.

5.5.4 Azure Text-to-Speech Implementation and Audio Conversion

The auditory output of the virtual therapist is generated via the Microsoft Azure Text-to-Speech (TTS) service. The implementation within `GeminiTTSController` goes beyond simple API calls, requiring low-level audio data manipulation to function within the Unity engine.

The synthesis process is handled asynchronously in the `ProcessAndSpeak` task to prevent frame-rate drops in VR. The system utilizes the `SpeechSynthesizer.SpeakSsmlAsync` method, which accepts the XML-based SSML string containing both the text and the emote markers.

Upon successful synthesis, Azure returns the audio data not as a standard audio file, but as a raw byte array (`byte[]`) encoded in PCM format (Riff-24Khz-16Bit-Mono, as configured in `Awake`). Unity, however, requires an `AudioClip` object to play sound. To bridge this gap, the custom utility method `WavByteArrayToAudioClip` was implemented.

This method performs a manual parsing of the WAV byte stream:

1. **Header Parsing:** It iterates through the byte array to locate the "fmt " and "data" sub-chunks of the RIFF header. This is crucial to extract metadata such as the number of channels, sample rate, and bit depth directly from the binary data.

2. **PCM to Float Conversion:** Unity's audio engine processes audio as floating-point numbers ranging from -1.0 to 1.0. The method converts the 16-bit integer PCM data received from Azure into the float array format required by `AudioClip.SetData`.

Only after this conversion is the clip assigned to the `AudioSource`, ensuring that the generated voice is played back with correct timing and pitch.

5.5.5 Visual Feedback and Interaction State

In Virtual Reality, providing immediate visual confirmation of system state is essential to prevent user frustration. The `GeminiTTSController` implements a direct feedback loop linked to the recording state.

The `OnRecordButtonPressed` method, triggered by the pinch gesture, performs two simultaneous visual updates:

- **Haptic/Visual Embodiment:** It accesses the `SkinnedMeshRenderer` of the user's virtual right hand. The material color is instantly changed to a specific `recordingColor` (typically red). This creates a "contextual glove" effect, signaling that the hand is currently acting as an input device for voice.

- **UI Feedback:** It activates a `microphone` GameObject (a 3D icon floating near the user or tablet), providing a secondary diegetic cue that the system is listening.

Conversely, the `OnRecordButtonReleased` method ensures the restoration of the neutral state. It retrieves the originally cached material color (`originalHandColor`) stored during the `Awake` phase and reapplies it to the hand renderer, while simultaneously deactivating the microphone icon. This clear binary state (Red/Normal) eliminates ambiguity regarding when the user's voice is being transmitted to the Azure Speech-to-Text service.

5.5.6 Deterministic Logic and Scene Transitions

While the core dialogue is managed by the remote LLM, critical navigation decisions—specifically the selection of therapeutic environments—are handled by a deterministic logic layer within the client. This "Offline" processing ensures immediate responsiveness and prevents hallucination errors where the AI might misunderstand a navigation command.

The `StopRecognitionAndProcess` coroutine includes specific branching logic controlled by boolean flags (e.g., `isWaitingForFreeBreathingChoice`). When the system is in this state, it bypasses the standard `/chat` endpoint and instead performs a local keyword analysis on the transcribed text (`finalRecognizedText`).

For example, if the user is choosing a setting for Free Breathing:

1. The system checks if the input contains specific keywords defined in the inspector (e.g., "Mountain", "Seaside", "Loft").

2. Upon detecting a match (e.g., "Seaside"), it executes a transition sequence:

    - It populates the `CrossSceneData` static class with the session ID and the chosen therapy name, ensuring persistence across the scene load.
    - It updates the local `TreatmentLog` to record the user's choice for data analysis.
    - It configures the `SceneTransitionTrigger` with the target scene name (e.g., `FB_SeaScene`) and initiates the visual fade-out/fade-in sequence.

If no keyword is recognized, the system falls back to a re-prompting mechanism (`AskForFreeBreathingSceneChoice`), guiding the user back to valid options without server latency.

5.5.7 Lip-Sync and Animation Synchronization

Realistic facial animation is achieved through an event-driven system. The Azure SDK triggers the `MyBookmarkReachedHandler` event whenever the audio playback reaches a timestamp associated with an SSML mark.

This handler invokes `TriggerActionForMark`, which parses the mark name (e.g., "emote:Happy") and commands the **SALSA (Simple Automated Lip Sync Approximation)** engine to trigger the corresponding blend shape on the 3D model. This ensures that facial expressions are perfectly timed with specific words in the sentence, significantly enhancing the perceived empathy of the agent.

5.5.8 Data Collection and Logging

Beyond conversation, the controller acts as a data logger. The `CaptureTreatmentData` method monitors the `currentModule` state. When specific modules conclude (e.g., Module "3b" for Craving Assessment), it captures the user's verbal response and stores it in the `currentTreatmentLog` object.

When the post-assessment phase is complete (Module "4d"), the `SendTreatmentLogToServer` coroutine is activated. This method performs a timezone conversion (UTC to Eastern Standard Time) to ensure data consistency, serializes the `TreatmentLogData` object into JSON, and POSTs it to the `/log_treatment` endpoint, permanently archiving the session results for analysis.

5.6 The In-VR User Interface (The Tablet)

5.6.1 `TabletController`: Animating and Showing the UI

The `TabletController` script manages the physical behavior and content display of the virtual tablet. In a VR environment, UI ergonomics are paramount; a static menu can be difficult to read, while a menu that is always "stuck" to the user's face can be intrusive.

This controller implements a dynamic positioning system that moves the tablet from a passive "idle" state (resting on the sofa) to an active "interaction" state (floating comfortably in front of the user) only when needed.

Spatial Animation

The `MoveTablet` coroutine handles the smooth transition between positions. This is used when the user finishes to fill the pre or post treatment assessments, chooses one of the treatments available or presses the "X" button to stop using the tablet. In all of these situations, the tablet has to move back to its resting position.

- **Targeting:** The script utilizes a defined `closeupTarget` transform to determine the position where tablet is while the user interacts with it.
- **Interpolation:** It employs `Vector3.Lerp` and `Quaternion.Slerp` over a variable `moveDuration` to ensure the movement feels mechanical yet fluid, avoiding the motion sickness potential of instant snapping.

Context-Aware Content Loading

The tablet is not a single-purpose device; it adapts its interface based on the current state of the therapeutic session, as dictated by the `ShowTablet(string threadId, string moduleNumber)` method.

This method acts as a router:

1. **Assessment Mode (Modules 3 and 5):** If the current module corresponds to the Pre-Treatment or Post-Treatment assessment, the controller activates the `surveyCanvas` and delegates logic to the `SurveyManager`. This loads the appropriate questionnaires (IPQ, VRSUQ, etc.) from the server.

2. **Intervention Mode (Module 4):** If the module is the Therapeutic Intervention phase, it hides the surveys and activates the `therapyCanvas`. It then calls the `TherapyManager` to fetch and display the available treatment options (e.g., Guided Breathing cards) for the user to select.

This modular approach allows a single physical object (the tablet) to serve multiple functions throughout the session, maintaining immersion by avoiding the need for popping-up 2D floating windows.

5.6.2 Assessment: `SurveyManager`

The `SurveyManager` is responsible for dynamically building, displaying, and collecting data from the pre and post treatment questionnaires. Unlike traditional vertical forms, the UI was redesigned as a horizontal "swiper" to improve usability and reduce cognitive load in VR. This design presents one question at a time, allowing the user to focus on a single task before swiping to the next.

The manager operates by fetching a JSON configuration from the server (`/surveys`), which defines the structure of each questionnaire. Based on the "type" field of each question (slider, input, checkbox), it instantiates the corresponding prefab into a horizontal layout group.

The `ScrollRectSnap` Component

To achieve the "paginated" feel, a custom component named `ScrollRectSnap` was developed and attached to the Unity `ScrollRect`. This script intercepts the drag events (`OnEndDrag`) and overrides the default inertia. Instead of scrolling freely, it calculates the nearest "page" index based on the current horizontal position and smoothly interpolates (snaps) the content to center that specific question.

This component exposes a public property `CurrentPage` and an event `OnPageChanged`, which are crucial for synchronizing the UI buttons with the scroll position.

Dynamic Button and Swipe Logic (`UpdateNavigationUI`)

The navigation logic is centralized in the `UpdateNavigationUI` method. This function is triggered every time the page changes (via swipe or button click) and is responsible for two key tasks:

1. **Context-Aware Buttons:** It determines which navigation button to display based on the current position.
    - *Intermediate Pages:* Shows a "Next Arrow".
    - *End of Survey:* Shows a "Next Survey" button to load the next questionnaire in the sequence.
    - *End of Assessment:* Shows a "Submit" button to finalize the data collection.
2. **Interaction Gating:** It enforces the validation logic. The "Next" arrow is only interactable if the current question has been answered. Furthermore, it actively disables the horizontal scrolling of the `ScrollRect` if the current page is invalid, effectively locking the user on the question until they provide a response.

Answer Validation (`ISurveyElement` and `IsAnswered`)

To support this validation logic in a polymorphic way, all question prefabs implement a custom interface: `ISurveyElement`. This interface mandates a boolean property `IsAnswered`. Each concrete implementation defines validity differently:

- **Sliders:** Require at least one interaction (value change).
- **Input Fields:** Require a non-empty string.
- **Checkboxes:** Require at least one option to be selected.

The `TitlePageElement` (used for section headers) returns `true` by default. This abstraction allows the `SurveyManager` to validate the current page without knowing the specific type of question being displayed.

5.6.3 Intervention: `TherapyManager`

While the `GeminiTTSController` handles the verbal dialogue, the `TherapyManager` script is responsible for the visual presentation of therapeutic choices. It populates the tablet's "Therapy Menu" dynamically based on data retrieved from the backend, ensuring that the available interventions can be updated on the server without requiring a client-side update.

Asynchronous Data Retrieval

The core of the manager is the `FetchTherapiesFromServer` coroutine. This method executes a GET request to the `/therapies` endpoint (or `/breathing_therapies` for sub-categories).
To handle the JSON response within Unity's utility limitations, the script employs a wrapper strategy: the raw JSON array received from the server is encapsulated in a temporary object (`{"therapies": ...}`) before parsing. This allows `JsonUtility` to correctly deserialize the list of `TherapyOption` objects, each containing an ID, title, description, and a dynamically generated image URL.
Once parsed, the manager instantiates a `therapyCardPrefab` for each option, populating the UI elements (text and images) with the retrieved data. This creates a scrollable, interactive menu for the user.

Handling User Selection and Branching Logic

The `OnChooseButtonPressed` method is the central decision node for the therapeutic flow. It implements complex branching logic depending on the nature of the selected therapy:

- **Category Expansion:** If the user selects a broad category (e.g., "Breathing Techniques"), the manager does not initiate a session but instead triggers a secondary fetch (`ShowBreathingTherapyOptions`) to load the specific sub-options (Guided vs. Free Breathing), updating the UI in real-time.
- **Direct Scene Transition:** For therapies with a single, fixed environment (e.g., "Mindfulness"), the manager immediately prepares the transition. It captures the current session state (Thread ID, Module), updates the `TreatmentLogData` with the chosen therapy name, stores this bundle in the `CrossSceneData` static class, and triggers the `SceneTransitionTrigger` to load the target scene (e.g., `MNDF_HillScene`).

- **Parameter Negotiation:** For therapies that support multiple environments (e.g., "Free Breathing" or "Visualization"), the manager delegates control back to the `GeminiTTSController`. Instead of loading a scene, it instructs the virtual therapist to verbally ask the user for their preference (e.g., "Would you prefer the beach or the loft?"). This initiates a sub-dialogue loop where the user's spoken choice determines the final destination scene.

This architecture allows for a flexible and scalable therapy menu, capable of supporting simple one-click interventions as well as complex, multi-step configuration flows.

5.7 Therapeutic Scenes Implementation

5.7.1 Therapies

As written in 3.2.3, the application offers four distinct therapeutic modules, each targeting a different coping mechanism for Substance Use Disorder. These modules are triggered by the user's selection on the tablet and are guided by specific scripts delivered by the virtual therapist.

Free Breathing Therapy This module offers a self-paced relaxation experience designed to reduce performance anxiety. Unlike structured exercises, the "Free Breathing" approach does not impose a specific rhythm. The user selects a preferred environment (the *Modern Loft* or the *Seaside*) to establish a sense of safety. The virtual therapist instructs the user to simply "observe" their natural breath without attempting to control or alter it. The therapeutic goal is to anchor the user in the present moment ("here and now"), using the breath as a stable focal point to weather the transient intensity of a craving wave, a technique often referred to as "urge surfing."

Guided Breathing Therapy This module implements a structured physiological intervention known as "paced respiration." In the *Hill* environment, a visual aid (a 3D sphere) appears in front of the user. The therapist guides the user to synchronize their breathing with the object: inhaling as the sphere expands and exhaling as it contracts. This forces a slow, rhythmic breathing pattern (typically targeting a resonance frequency of approx. 6 breaths per minute). The mechanism aims to directly stimulate the vagus nerve and activate the parasympathetic nervous system, thereby reducing heart rate and physiological arousal associated with stress and withdrawal.

Mindfulness Therapy This module follows a standard "Body Scan" meditation protocol within the *Hill* environment. The virtual therapist guides the user's attention systematically through the body, starting from the facial muscles (forehead, jaw, eyes) and moving downwards through the neck, shoulders, arms, torso, and legs. The script emphasizes the concept of "non-judgmental awareness," instructing the user to simply notice sensations—such as tension, warmth, or contact with the ground—without trying to "fix" them. This practice is designed to increase interoceptive awareness and help the patient decouple the physical sensation of discomfort from the emotional reaction to it.

Visualization Therapy This intervention utilizes guided imagery to induce a state of deep relaxation and mental escapism. The user chooses a setting that aligns with their personal preference for calmness, such as a cozy *Library* or a *Seaside* beach. The therapist's narrative

actively engages the user's sensory imagination, prompting them to visualize specific details (e.g., the light filtering through a window, the texture of a book, the sound of waves). By occupying the brain's visuospatial and sensory processing centers with safe, vivid imagery, this technique aims to disrupt the cognitive loop of craving and provide a mental respite from immediate stressors.

5.7.2 Modern Loft

The Modern Loft environment (fig.11 and fig.12) was imported from an external project on Sketchfab [34]. Most of the assets have been optimized on Blender (Add Modifier -> Generate -> Decimate) for the VR standalone version. The fireplace was replaced with the UnityTechnology `LargeFlames` and `FireEmbers` and more realistic Logs were added, in order to build a more immersive and realistic experience.
All the lights, either lamps or ceiling lights, were set to Baked mode. A Reflection probe was also included for a consistent illumination and reflections. Adaptive Probe Volumes were used to enlighten the therapist, which was not marked as `Static` and could not receive light from Baked lights.

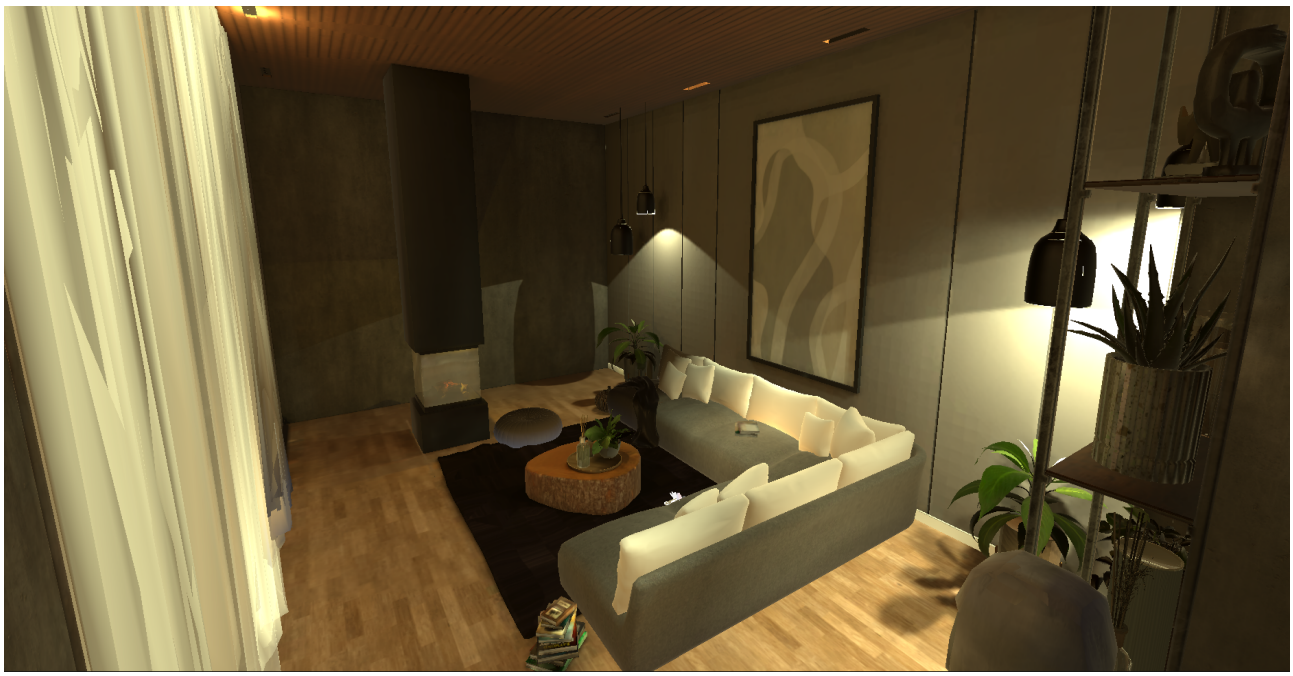

Figure 11: Loft

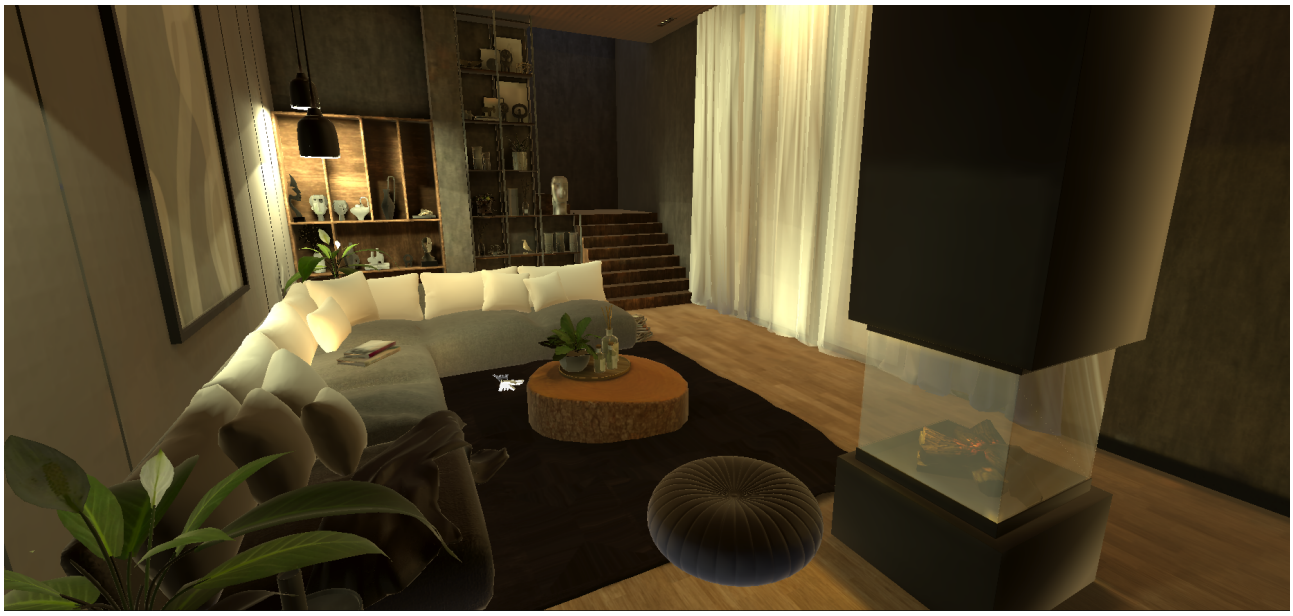

Figure 12: Loft

5.7.3 Seaside

The seaside scenario (fig.13) was built modelling a terrain with a sand ground texture. The ocean is implemented with a shader from [35], and the trees in the back are from the Toby Foliage Engine [36].
As far as lighting is concerned, the directional light used was Baked for all the environment, and then a Realtime directional light was added in order to reproduce the lightind and shadow on the sand for the therapist. In order to only enlighten the therapist with the realtime light, its Culling Mask was set to `Therapist`, and the therapist's layer was set to `Therapist` as well.

Figure 13: Seaside

5.7.4 Library

The library scenario (fig.14) was imported from [37]. As the environment was very big and the amount of batches (numbers of draw calls at every frame, that means the number of different materals and shaders rendered) extremely high, the plugin MeshBaker was used to merge some of the assets and materials (fig.15). Furthermore, the space between the last tables and the door was shortened and the roof material, which was opaque, was replaced with a transparent glass that lets the user see a starry sky. All lights in the chandeliers were set to Baked mode.

Figure 14: Library

Figure 15: Use of MeshBaker in the Library Scene

5.7.5 Hill

The starting ground and rocks in this scenario were taken from [36]. All trees were replaced with other URP tree models [38].
Creating a dense, immersive natural environment, such as a forest or a meadow, presents a significant challenge for mobile VR hardware. Rendering thousands of individual GameObjects for grass blades would drastically increase the draw calls, overwhelming the CPU and causing unacceptable frame rate drops. To address this, the `SineWaveInstanced` script implements a high-performance rendering technique known as **GPU Instancing**.

Procedural Generation and Placement

Upon initialization (`Start`), the script performs a procedural "planting" process. Instead of instantiating GameObjects, it calculates the transformation data for thousands of grass instances (e.g., 50,000) and stores them in a list of 4×4 matrices (`Matrix4x4`).

The placement logic ensures the vegetation conforms naturally to the terrain:

1. A random (x, z) coordinate is selected within the defined `dimensioneArea`.
2. A raycast is fired downward from a set height (`altezzaRaycast`) to detect the terrain surface layer.
3. Upon impact, the hit point determines the position. The rotation is calculated to align the grass perpendicular to the surface normal (`Quaternion.FromToRotation`), ensuring it grows "out" of slopes rather than vertically. A random Y-axis rotation and a random scale factor are applied to avoid visual repetition.

The Rendering Loop

The core optimization lies in the `Update` method. Instead of relying on Unity's standard renderer components, the script utilizes the `Graphics.RenderMeshInstanced` API.
This command instructs the GPU to draw the same mesh (`meshErba`) using the same material (`materialeErba`) multiple times in a single draw call. The CPU only needs to upload the array of transformation matrices once per frame. To further optimize performance for the Quest headset, shadow casting and receiving are explicitly disabled in the `RenderParams`, as dynamic shadows on such small, numerous objects are computationally expensive and visually negligible in this context.
This approach allows for the rendering of vast fields of grass with a minimal performance cost, maintaining the high frame rate required for VR comfort.

Figure 16: Hill

Chapter 6

Usability Test and Results

To validate the system architecture, the quality of the immersive environments, and the interaction with the LLM-driven virtual therapist, a structured usability test was conducted. This chapter details the experimental protocol, the setup of the testing sessions, the demographic characteristics of the participants, and the analysis of the collected data.

6.1 Protocol for Usability Test

The primary objective of this study was twofold: first, to evaluate the user experience (UX) and the credibility of the conversational agent within the main clinical setting; second, to assess the sense of presence and immersion provided by the specific therapeutic environments.

To achieve these goals without inducing user fatigue, the testing experience was architecturally divided into two distinct phases: the **Clinical Phase** and the **Environmental Phase**.

6.1.1 The Two-Phase Structure and the "Usability Hub"

To streamline the testing process and avoid the necessity of repeating the full onboarding procedure for every environment, a dedicated scene named the **"Usability Hub"** was developed specifically for this study.

1. **Phase 1: The Clinic.** The user begins in the standard application flow (Scene: *Intro* $\rightarrow$ *Clinic*). They undergo the full "walk-in" sequence, interact with the virtual therapist, complete the check-in, and perform the pre-assessment via the tablet. This phase concludes at the moment of therapy selection. The aim here is to evaluate the naturalness of the dialogue, the functionality of the UI (the tablet), and the perceived empathy of the agent.

2. **Phase 2: The Environments.** For the second phase, the user is transferred to the **Usability Hub**. This is a debugging and testing environment (fig. 17) that allows for the direct injection of session states. From here, the user can instantly load specific therapeutic scenarios without repeating the conversational preamble. This setup was crucial to allow rapid A/B testing of different environments.

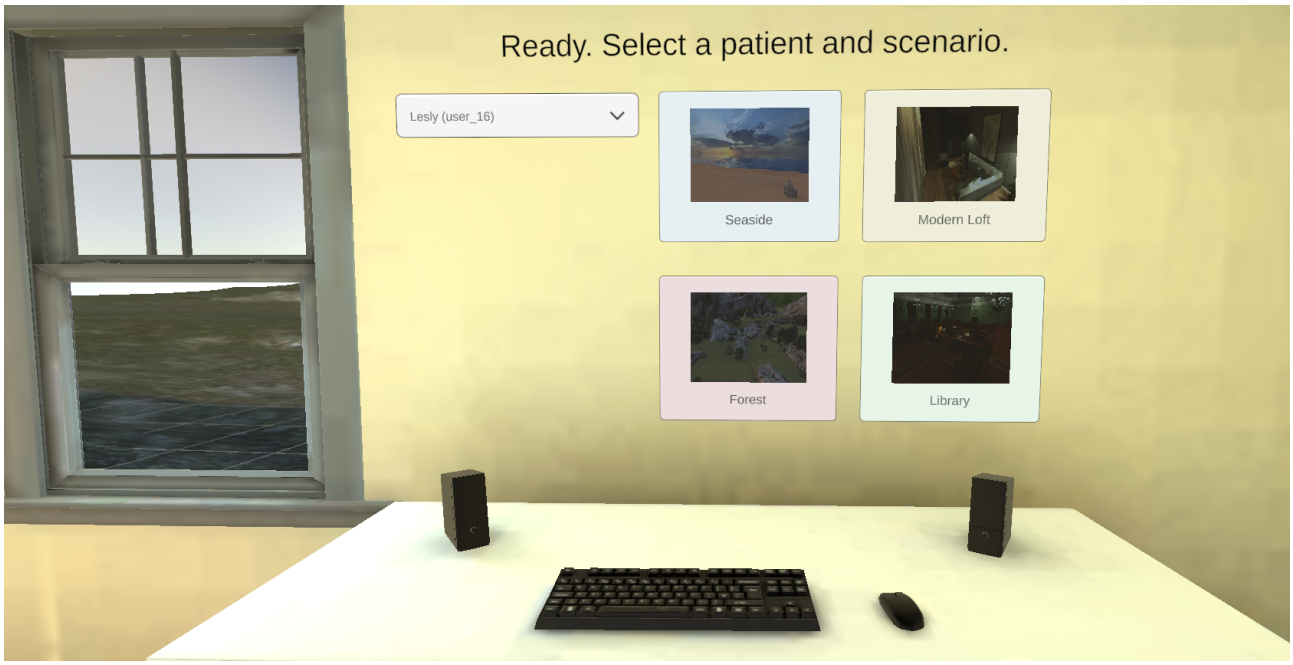

Figure 17: Usability Hub

6.1.2 Experimental Design and Balancing

A total of 20 participants were recruited for the study. To ensure a robust evaluation of the four available environments (*Modern Loft*, *Seaside*, *Library*, *Hill*), a balanced incomplete block design was employed.
Each participant was required to test the Clinic scene (mandatory for all) and subsequently experience exactly two out of the four therapeutic environments. The pairing of environments was distributed according to a counterbalanced matrix to ensure that (fig.18):

- Each user experienced a different combination of scenarios compared to the immediately preceding or following user.
- Each of the four environments was tested exactly 10 times in total across the usability tests.

	A	B	C	D	E
1	User	Loft	Seaside	Library	Hill
2	user 01	x			x
3	user 02		x	x	
4	user 03	x	x		
5	user 04	x		x	
6	user 05		x		x
7	user 06			x	x
8	user 07	x			x
9	user 08		x	x	
10	user 09	x	x		
11	user 10	x		x	
12	user 11		x		x
13	user 12			x	x
14	user 13	x			x
15	user 14		x	x	
16	user 15	x	x		
17	user 16	x		x	
18	user 17		x		x
19	user 18			x	x
20	user 19	x			x
21	user 20		x	x	

Figure 18: Usability test - Environments division

Control Variable: The Mindfulness Therapy To ensure fairness in the judgment of the environments, it was necessary to eliminate confounding variables arising from the different mechanics of the therapies (e.g., the cognitive load of the *Guided Breathing* visual sphere vs. the auditory focus of *Free Breathing*). Therefore, the `Mindfulness Therapy` module was selected as the constant control condition for all environmental tests. This choice was driven by the nature of the mindfulness script, which focuses on "being present" and body awareness, thereby encouraging the user to observe the surrounding environment without the distraction of complex interactive tasks.

6.2 Setup and Assessment Tools

The data collection was managed administering surveys after each session.

6.2.1 Questionnaires: Clinical Phase

Upon completing the interaction in the Clinic (Phase 1), participants were administered a comprehensive battery of standardized questionnaires to evaluate the system's core usability

and the agent's performance:

- **IPQ (Igroup Presence Questionnaire):** To measure the sense of presence, spatial realism, and involvement within the office setting.
- **Believability Questionnaire:** A specific set of questions designed to assess the perceived intelligence, agency, and emotional awareness of the virtual therapist.
- **VRSUQ (Virtual Reality Sickness User Questionnaire):** To monitor any adverse physiological effects (nausea, oculomotor strain, disorientation) caused by the locomotion or rendering.
- **UEQ-S (User Experience Questionnaire - Short):** To evaluate the pragmatic quality (efficiency, perspicuity) and hedonic quality (stimulation, novelty) of the user interface and the overall application.

6.2.2 Questionnaires: Environmental Phase

After experiencing each of the two assigned therapeutic environments (Phase 2), participants were administered a shorter assessment focused strictly on the quality of the Virtual Environment (VE):

- **IPQ (Igroup Presence Questionnaire):** To measure the sense of presence, spatial realism, and involvement within each environment.

6.2.3 Survey items and metrics

Below are the items used for each type of questionnaire and the corresponding metrics used for the data analysis.

IPQ

Likert Scale 1-7. The anchors for the scale are specified for each question
Questions:

1. In the computer generated world I had a sense of "being there". [Not at all - Very much]
2. Somehow I felt that the virtual world surrounded me. [Fully disagree - Fully agree]
3. I felt like I was just perceiving pictures. [Fully disagree - Fully agree]
4. I did not feel present in the virtual space. [Fully disagree - Fully agree]
5. I had a sense of acting in the virtual space, rather than operating something from outside. [Fully disagree - Fully agree]
6. I felt present in the virtual space. [Fully disagree - Fully agree]
7. How aware were you of the real world surrounding you while navigating in the virtual world? (i.e. sounds, room temperature, other people, etc.)? [Extremely aware - Not aware at all]
8. I was not aware of my real environment. [Fully disagree - Fully agree]

9. I still paid attention to the real environment. [Fully disagree - Fully agree]
10. I was completely captivated by the virtual world. [Fully disagree - Fully agree]
11. How real did the virtual world seem to you? (1) [Completely real - Not real at all]
12. How much did your experience in the virtual environment seem consistent with your real world experience? [Not consistent - Very consistent]
13. How real did the virtual world seem to you? (2) [About as real as an imagined world - Indistinguishable from the real world]
14. The virtual world seemed more realistic than the real world. [Fully disagree - Fully agree]

Key metrics:
IPQ_GP_SCORE = IPQ1
IPQ_SP_SCORE = (IPQ2 + (8 - IPQ3) + (8 - IPQ4) + IPQ5 + IPQ6)/5
IPQ_REAL_SCORE = (IPQ7 + IPQ8 + (8 - IPQ9) + IPQ10)/4
IPQ_INV_SCORE = ((8 - IPQ11) + IPQ12 + IPQ13 + IPQ14)/4

Believability

Likert Scale 1-7. In all the items, the anchors for the scale are [Strongly disagree - Strongly agree]
Questions:

1. The visual design of the virtual agent caught my attention.
2. I think the virtual agent's appearance is aesthetically pleasing.
3. I think the virtual agent's visual design is realistic.
4. The virtual agent's behavior drew my attention.
5. I felt the virtual agent's behavior was coherent and natural.
6. I think the virtual agent's behavior was easy to understand.
7. I felt the virtual agent's behavior was appropriate to the context.
8. I felt sometimes the virtual agent behaved inappropriately.
9. I felt that the virtual agent perceived the environment around him/her/them.
10. I felt that the virtual agent reacted to the change in the environment.
11. I felt that the virtual agent was aware of my presence.
12. ~~I felt that the virtual agent was aware of the presence of other virtual agents.~~
13. The virtual agent was unaware of its surroundings.
14. The virtual agent interacted socially with me.
15. I felt that the virtual agent was able to coordinate with me.

16. ~~The virtual agent interacted socially with the other virtual agent(s).~~
17. ~~I felt that the virtual agent was able to coordinate with the other virtual agent(s).~~
18. I felt that the virtual agent was able to make plans.
19. I felt that the virtual agent learned from past experiences.
20. I felt that the virtual agent seemed to have memory.
21. I felt that the virtual agent was capable of having feelings.
22. I felt that the virtual agent expressed emotions.
23. I felt that the virtual agent's expressed emotions were easy to understand.
24. I felt that the virtual agent's expressed emotions were appropriate to the context.
25. I felt that the virtual agent had a personality.
26. I felt that the virtual agent was extraverted and enthusiastic.
27. I felt that the virtual agent was sympathetic and warm.
28. I felt that the virtual agent was dependable and self-disciplined.
29. I felt that the virtual agent was emotionally stable.
30. I felt that the virtual agent was open to new experiences.
31. I felt that the virtual agent seemed to have self-awareness.
32. The virtual agent took actions without inputs from others.
33. The virtual agent seemed to have its own goals.
34. I felt that the virtual agent was believable.
35. I felt that the virtual agent behaved like a real person.
36. I enjoy the interaction with the virtual agent.

Items n. 12, 16 and 17 have been removed (crossed out in red in the above list) from the standard Believability questionnaire because there are not other virtual agents in the Clinic environment.

Key metrics:
Visual Properties = (Q1 + Q2 + Q3)/3
Behaviour = (Q4 + Q5 + Q6 + Q7 + (8 - Q8))/5
Awareness = (Q9 + Q10 + Q11 + (8 - Q13))/4
Social Relationships = (Q14 + Q15)/2
Intelligence = (Q18 + Q19 + Q20)/3
Emotion = (Q21 + Q22 + Q23 + Q24)/4
Personality = (Q25 + Q26 + Q27 + Q28 + Q29 + Q30)/6
Agency = (Q31 + Q32 + Q33)/3

Overall Believability = (Q34 + Q35 + Q36)/3

Total Believability = (Q1 + Q2 + Q3)/3 + ((Q4 + Q5 + Q6 + Q7 - Q8)/5) * 2 + ((Q9 + Q10 + Q11 - Q13)/4) * 2 + (Q14 + Q15)/2 + ((Q18 + Q19 + Q20)/3) * 3 + ((Q21 + Q22 + Q23 + Q24)/4) * 4 + ((Q25 + Q26 + Q27 + Q28 + Q29 + Q30)/6) * 4 + ((Q31 + Q32 + Q33)/3) * 2 + ((Q34 + Q35 + Q36)/3) / 20

VRSUQ

Likert Scale 1-5. In all the items, the anchors for the scale are [Strongly disagree - Strongly agree].
Questions:

1. The system responded well to my manipulations as expected with no delays.
2. I think the virtual reality system provides clear feedback on my manipulations.
3. I kept making errors/mistakes while using the virtual reality system.
4. I could clearly understand the information presented within the virtual environment.
5. I think this system is user-friendly, straightforward to learn, and designed in such a way that most people will find it easy to adapt to.
6. I think it is easy to correct errors made during virtual reality experiences.
7. I enjoyed the virtual reality experience.
8. I felt dizzy, motion sickness, or a headache while experiencing virtual reality.
9. While experiencing virtual reality, I felt mental burdens such as tension, frustration, and time pressure.

Key metric:
VRSUQ score = ((Q1 + Q2 + (6 - Q3) + Q4 + Q5 + Q6 + Q7 + (6 - Q8) + (6 - Q9))/9) – 1) * 100/4

UEQ (Short Version)

Likert Scale 1-7. The anchors of the scale are specified for each item.
The question is the same for all the items and it is: `Rate your experience with the application and the system`.
What changes are the anchors for the scale.
Questions:

1. Obstructive - Supportive
2. Complicated - Easy
3. Inefficient - Efficient
4. Confusing - Clear
5. Boring - Exciting

6. Not interesting - Interesting
7. Conventional - Inventive
8. Usual - Leading edge

Key Metrics:
Data are rescaled to the range -3 to 3.
Pragmatic Quality = (Q1 + Q2 + Q3 + Q4)/4
Hedonic Quality = (Q5 + Q6 + Q7 + Q8)/4
Overall = (Q1 + Q2 + Q3 + Q4 + Q5 + Q6 + Q7 + Q8)/8

In addition, mean, variance, and standard deviation are calculated for each item.

6.3 Results

6.3.1 Clinic

IPQ

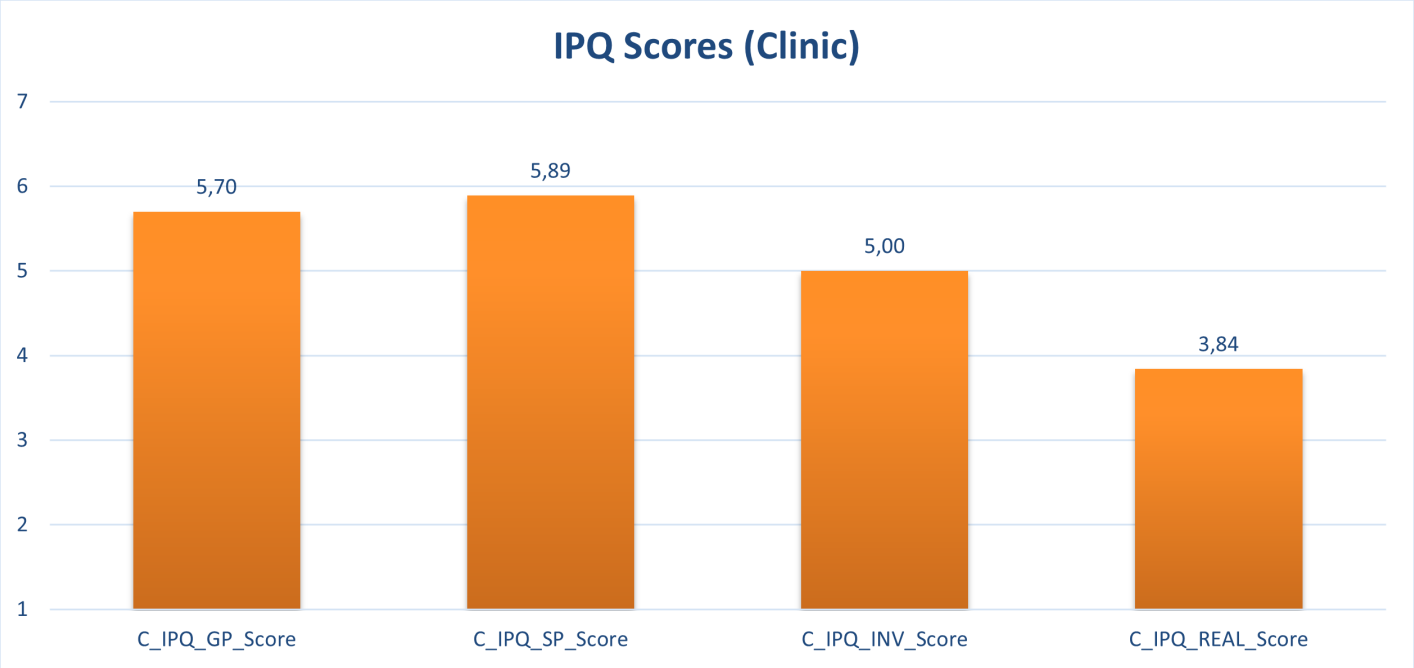

Figure 19: Overall Score, General Presence, Spatial Presence, Involvement, Realism

Item	Mean	Std. Dev.	Variance
CIPQ1	5,70	0,98	0,96
CIPQ2	6,05	0,94	0,89
CIPQ3	1,90	0,72	0,52
CIPQ4	1,95	1,00	1,00
CIPQ5	5,50	0,83	0,68
CIPQ6	5,75	0,64	0,41
CIPQ7	4,70	1,81	3,27
CIPQ8	4,80	1,77	3,12
CIPQ9	3,00	1,65	2,74
CIPQ10	5,50	1,32	1,74
CIPQ11	3,70	1,42	2,01
CIPQ12	4,85	1,04	1,08
CIPQ13	4,25	1,07	1,14
CIPQ14	1,95	0,76	0,58

Figure 20: Mean, Standard Deviation and Variance per item

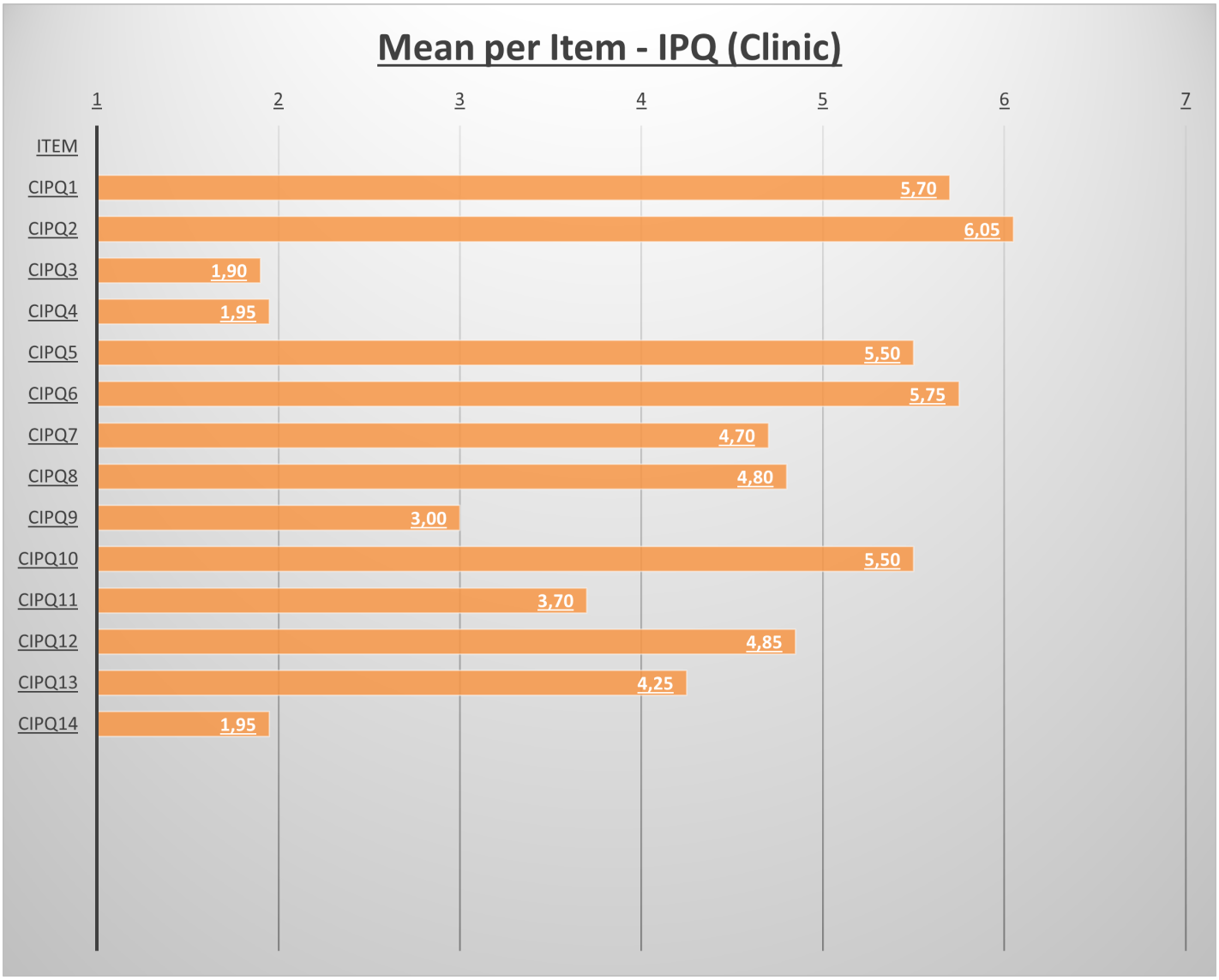

Figure 21: Mean per item

Believability

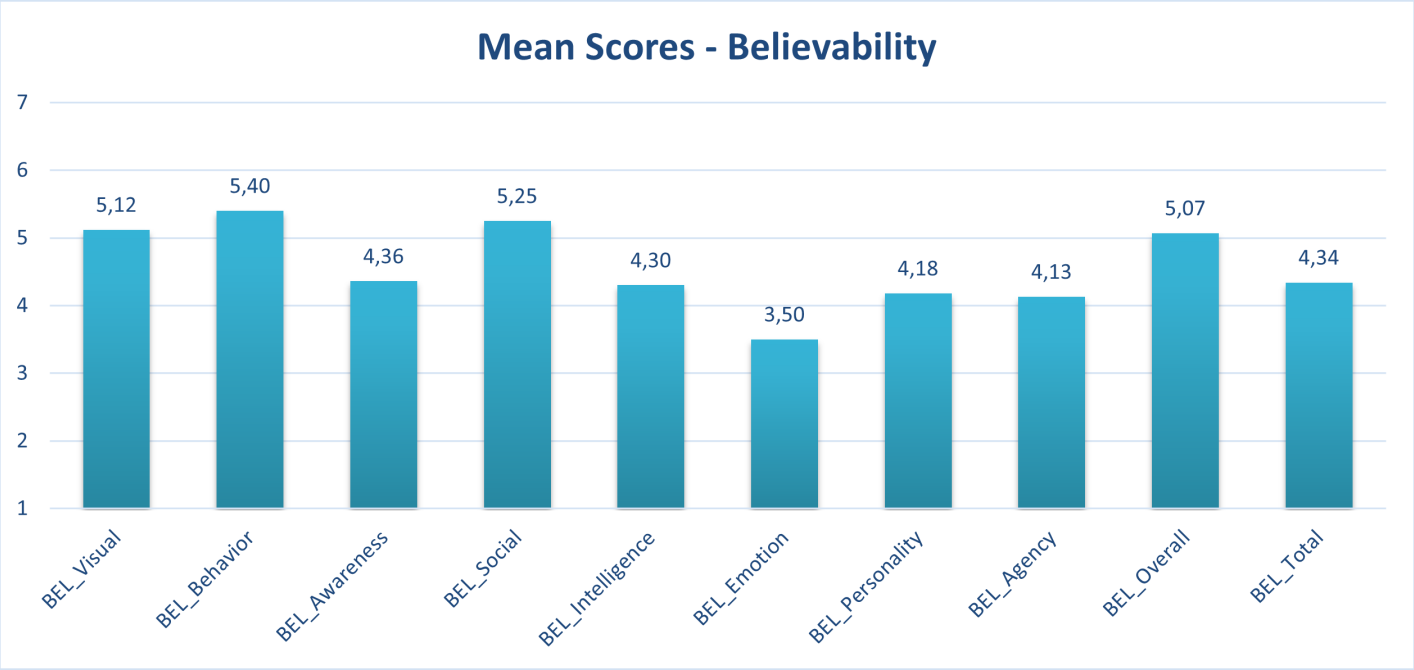

Figure 22: Mean scores of the key metrics for the believability of the viirtual agent

Item	Mean	Std. Dev.	Variance
CB1	5,30	1,26	1,59
CB2	5,45	1,36	1,84
CB3	4,60	1,43	2,04
CB4	4,75	1,68	2,83
CB5	4,45	1,39	1,94
CB6	5,50	1,32	1,74
CB7	6,00	1,08	1,16
CB8	1,70	1,30	1,69
CB9	3,85	2,16	4,66
CB10	3,85	1,98	3,92
CB11	5,50	1,67	2,79
CB12	3,75	2,05	4,20
CB13	5,25	1,37	1,88
CB14	5,25	1,55	2,41
CB15	4,75	1,52	2,30
CB16	4,05	1,57	2,47
CB17	4,10	1,68	2,83
CB18	3,15	1,63	2,66
CB19	3,25	1,59	2,51
CB20	3,30	1,49	2,22
CB21	4,30	1,63	2,64
CB22	3,75	1,68	2,83
CB23	3,15	1,73	2,98
CB24	4,35	1,95	3,82
CB25	5,20	1,47	2,17
CB26	5,10	1,83	3,36
CB27	3,55	1,88	3,52
CB28	4,05	2,21	4,89
CB29	3,65	2,16	4,66
CB30	4,70	1,72	2,96
CB31	5,10	1,62	2,62
CB32	4,55	1,36	1,84
CB33	5,55	1,54	2,37

Figure 23: Mean, Standard Deviation and Variance per item

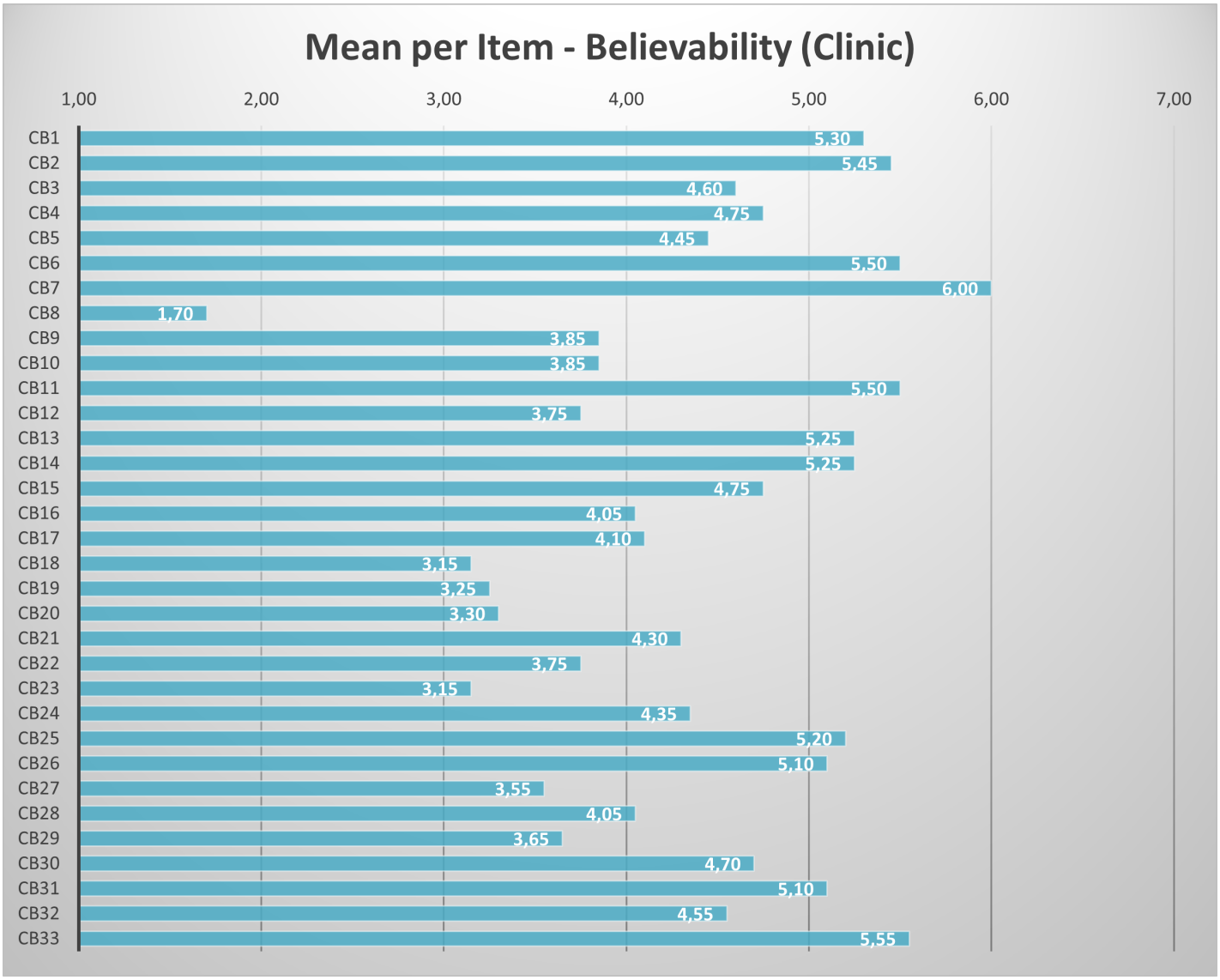

Figure 24: Mean per item

VRSUQ

VRSUQ General Result	
Total Mean Score:	87,64

Figure 25: Total mean score for VRSUQ

Questions	Mean	Std. Dev.	Variance
CVR1	4,35	0,93	0,87
CVR2	4,3	0,92	0,85
CVR3	2,15	1,27	1,61
CVR4	4,75	0,44	0,2
CVR5	4,55	0,6	0,37
CVR6	4,3	0,86	0,75
CVR7	4,75	0,44	0,2
CVR8	1,15	0,49	0,24
CVR9	1,15	0,49	0,24

Figure 26: Mean, Standard Deviation and Variance per item

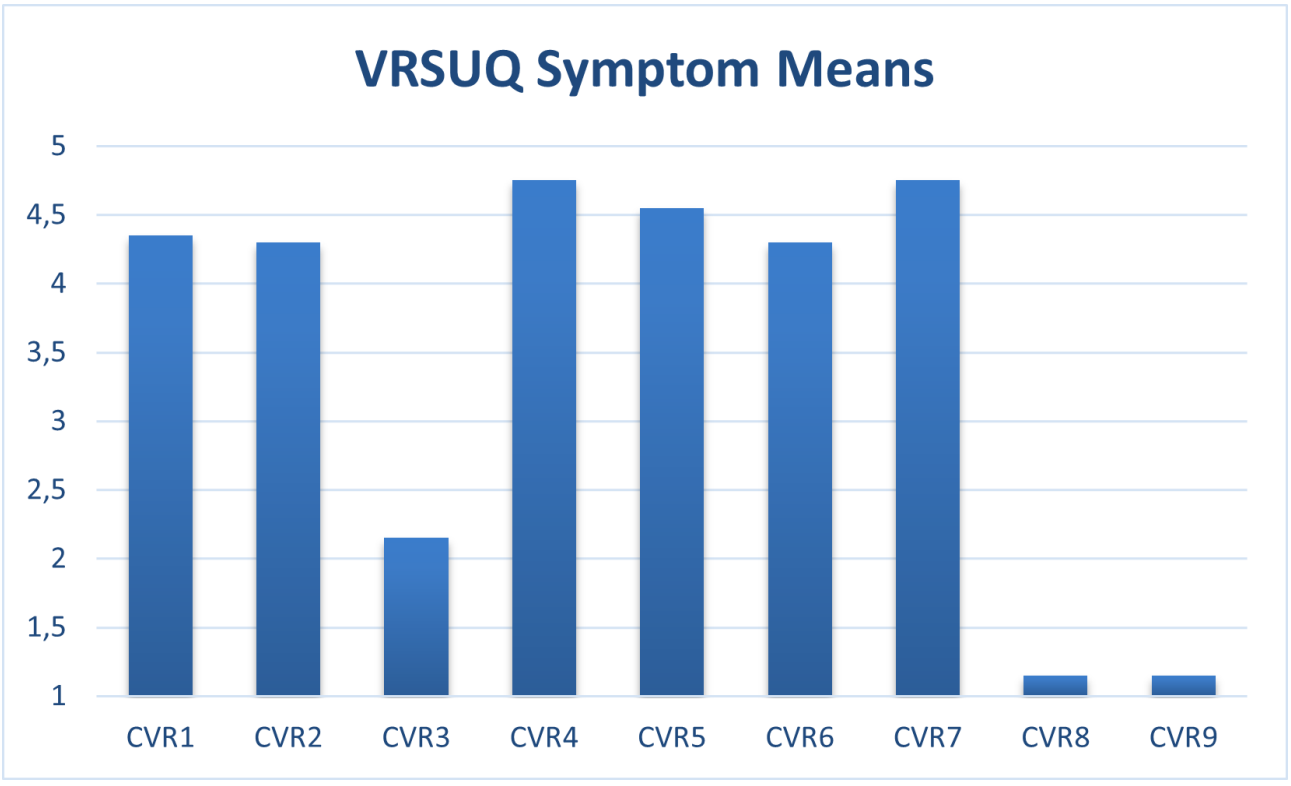

Figure 27: Mean per item

UEQ (Short)

Short UEQ Scales	
Pragmatic Quality	2,038
Hedonic Quality	2,288
Overall	2,163

Figure 28: Mean values

Item	Mean	Variance	Std. Dev.	No.	Negative	Positive	Scale
1	1,9	0,8	0,9	20	obstructive	supportive	Pragmatic Quality
2	2,2	0,6	0,8	20	complicated	easy	Pragmatic Quality
3	1,9	1,2	1,1	20	inefficient	efficient	Pragmatic Quality
4	2,2	1,1	1,1	20	confusing	clear	Pragmatic Quality
5	1,8	1,5	1,2	20	boring	exciting	Hedonic Quality
6	2,5	0,7	0,8	20	not interesting	interesting	Hedonic Quality
7	2,4	0,8	0,9	20	conventional	inventive	Hedonic Quality
8	2,6	0,5	0,7	20	usual	leading edge	Hedonic Quality

Figure 29: Statistics per item

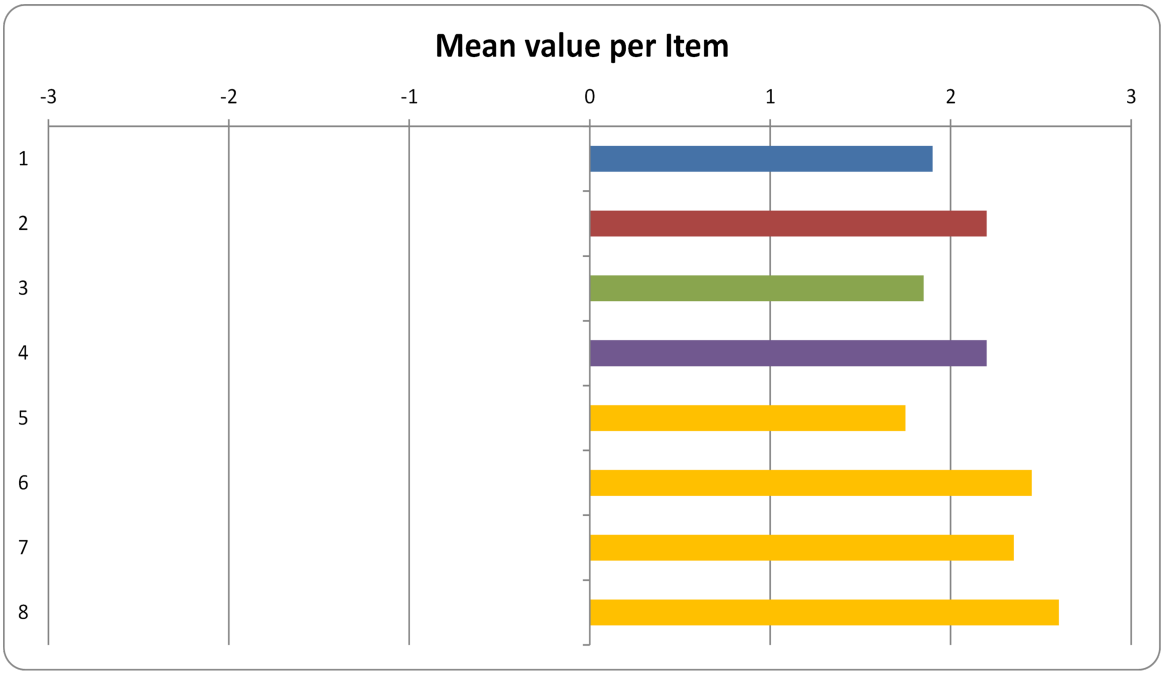

Figure 30: Mean values per item

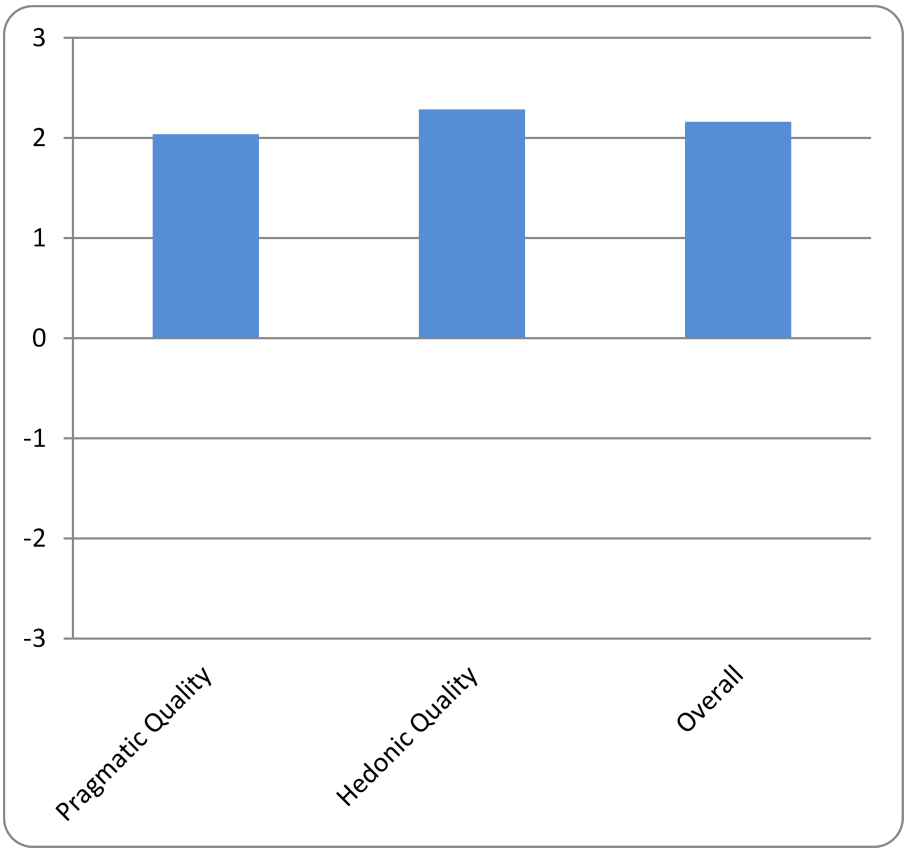

Figure 31: Overall mean values

Correlation between VRSUQ and IPQ results

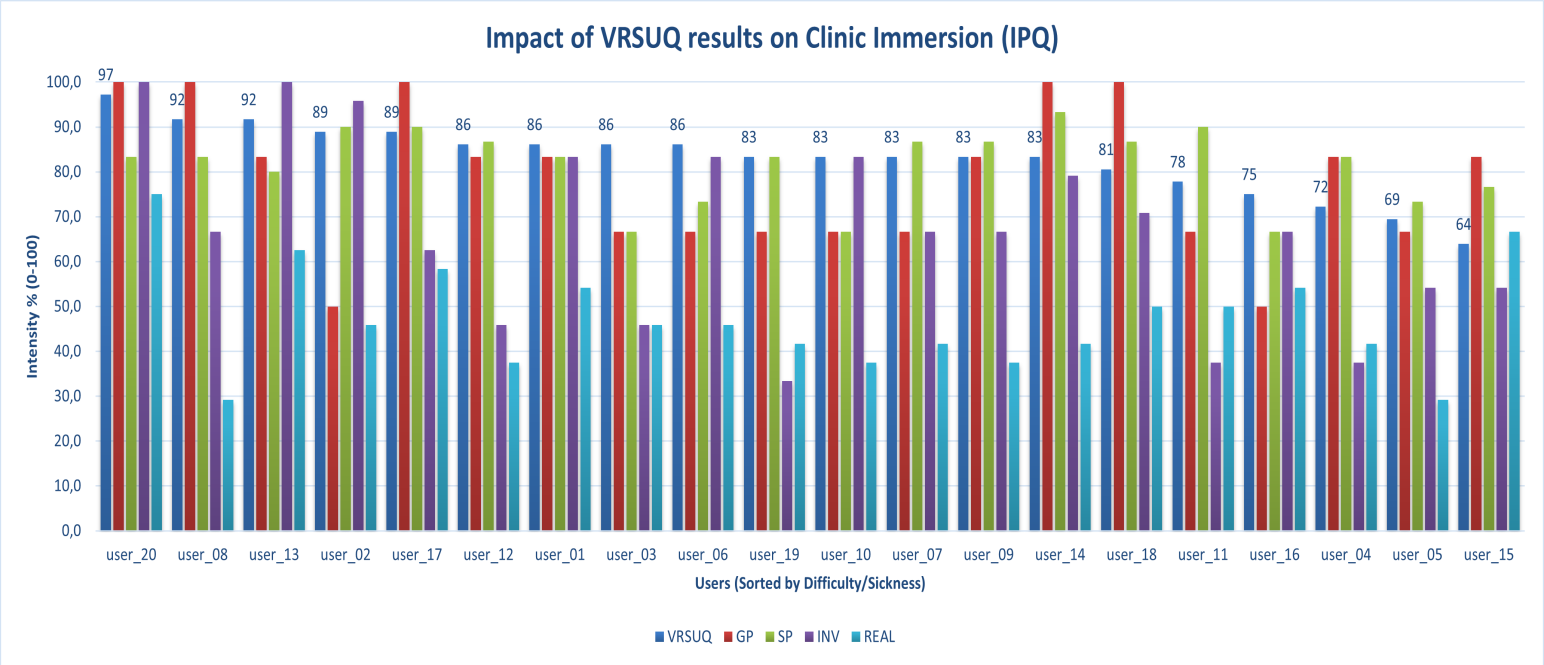

Figure 32: VRSUQ and IPQ results for the Clinic

Impact of prior VR Experience on Clinic Results

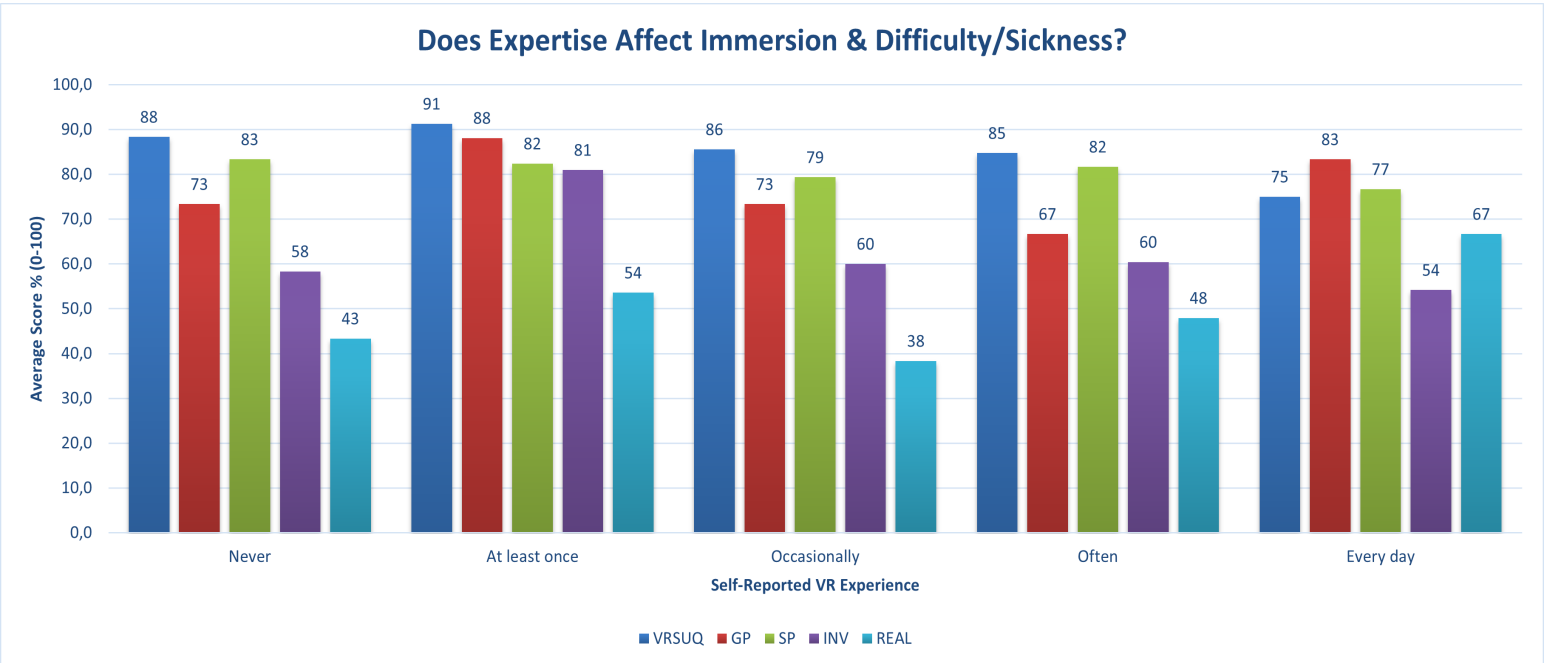

Figure 33: VRSUQ and IPQ results, grouped by previous VR experiences

6.3.2 Modern Loft

IPQ

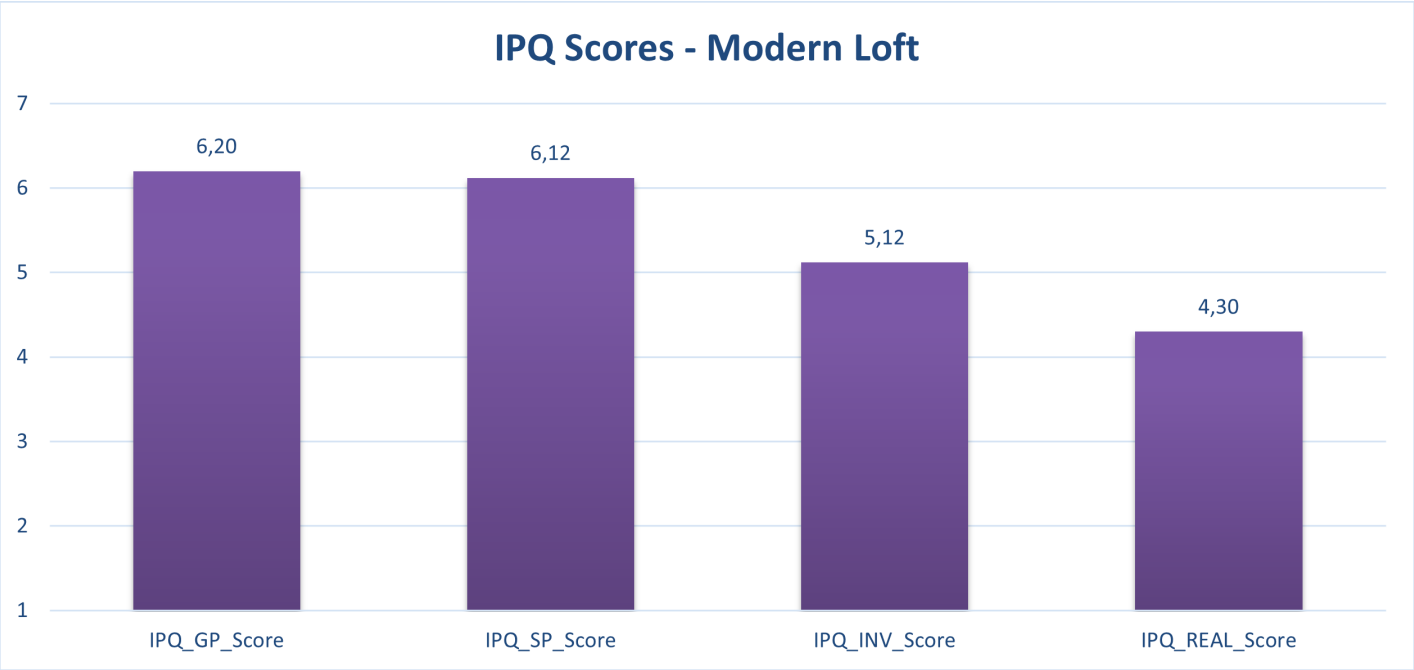

Figure 34: Overall Score, General Presence, Spatial Presence, Involvement, Realism

Item	Mean	Std. Dev.	Variance
IPQ1	6,20	0,79	0,62
IPQ2	6,40	0,70	0,49
IPQ3	1,70	0,48	0,23
IPQ4	2,20	1,48	2,18
IPQ5	6,00	0,82	0,67
IPQ6	6,10	0,74	0,54
IPQ7	5,80	1,23	1,51
IPQ8	4,80	2,25	5,07
IPQ9	3,40	2,22	4,93
IPQ10	5,30	1,57	2,46
IPQ11	3,10	1,52	2,32
IPQ12	5,30	1,06	1,12
IPQ13	4,70	1,42	2,01
IPQ14	2,30	0,82	0,68

Figure 35: Mean, Standard Deviation and Variance per item

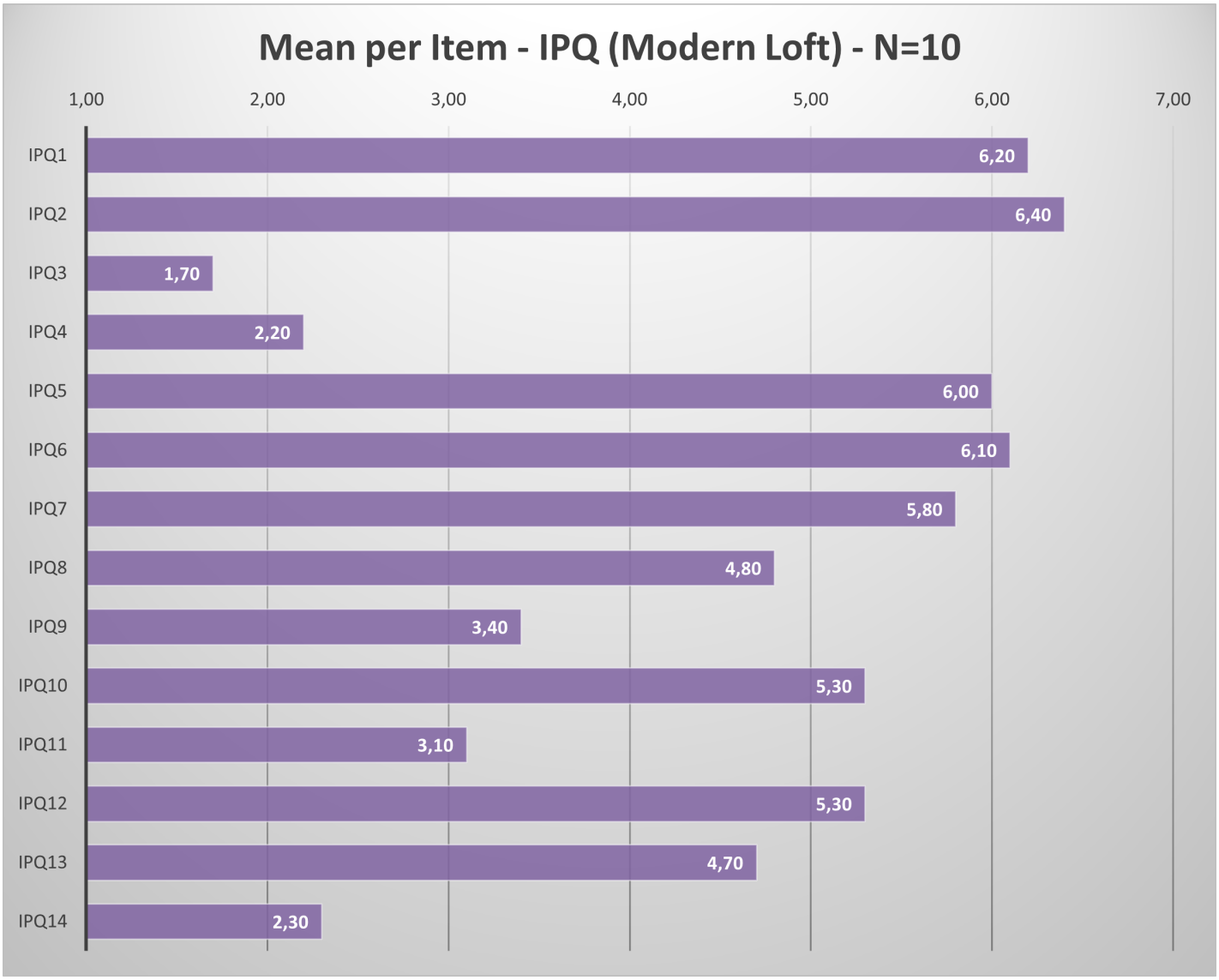

Figure 36: Mean per item

6.3.3 Seaside

IPQ

Figure 37: Overall Score, General Presence, Spatial Presence, Involvement, Realism

Item	Mean	Std. Dev.	Variance
IPQ1	6,20	0,79	0,62
IPQ2	6,40	0,84	0,71
IPQ3	1,40	0,52	0,27
IPQ4	1,80	1,03	1,07
IPQ5	5,80	1,62	2,62
IPQ6	6,00	1,25	1,56
IPQ7	5,20	1,32	1,73
IPQ8	5,00	1,70	2,89
IPQ9	2,60	1,43	2,04
IPQ10	5,80	1,48	2,18
IPQ11	2,80	1,93	3,73
IPQ12	4,80	1,93	3,73
IPQ13	4,60	1,17	1,38
IPQ14	2,10	1,45	2,10

Figure 38: Mean, Standard Deviation and Variance per item

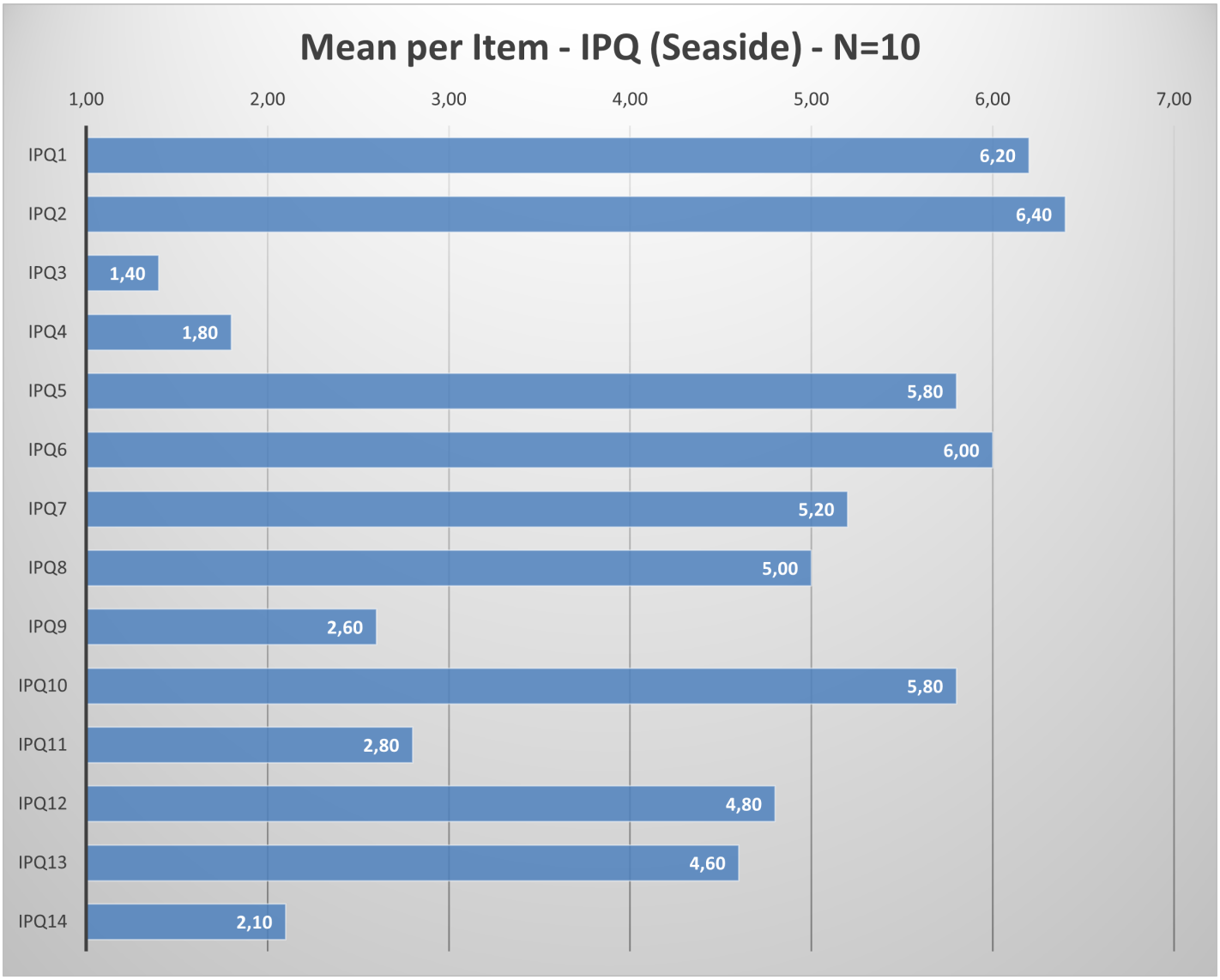

Figure 39: Mean per item

6.3.4 Library

IPQ

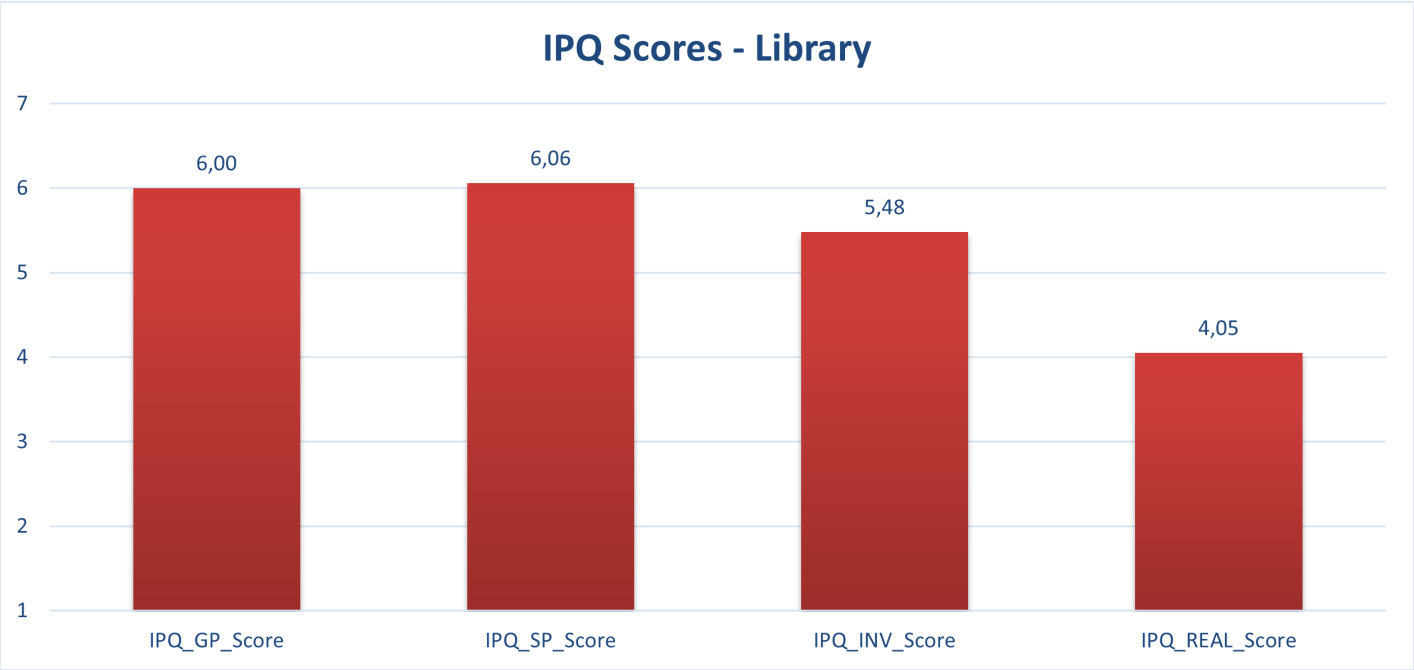

Figure 40: Overall Score, General Presence, Spatial Presence, Involvement, Realism

Item	Mean	Std. Dev.	Variance
IPQ1	6,00	1,33	1,78
IPQ2	6,10	1,37	1,88
IPQ3	1,80	0,92	0,84
IPQ4	1,80	1,03	1,07
IPQ5	5,70	1,49	2,23
IPQ6	6,10	1,10	1,21
IPQ7	5,70	2,00	4,01
IPQ8	5,30	2,11	4,46
IPQ9	2,80	2,35	5,51
IPQ10	5,70	1,64	2,68
IPQ11	3,40	1,90	3,60
IPQ12	5,00	1,56	2,44
IPQ13	4,30	1,49	2,23
IPQ14	2,30	1,77	3,12

Figure 41: Mean, Standard Deviation and Variance per item

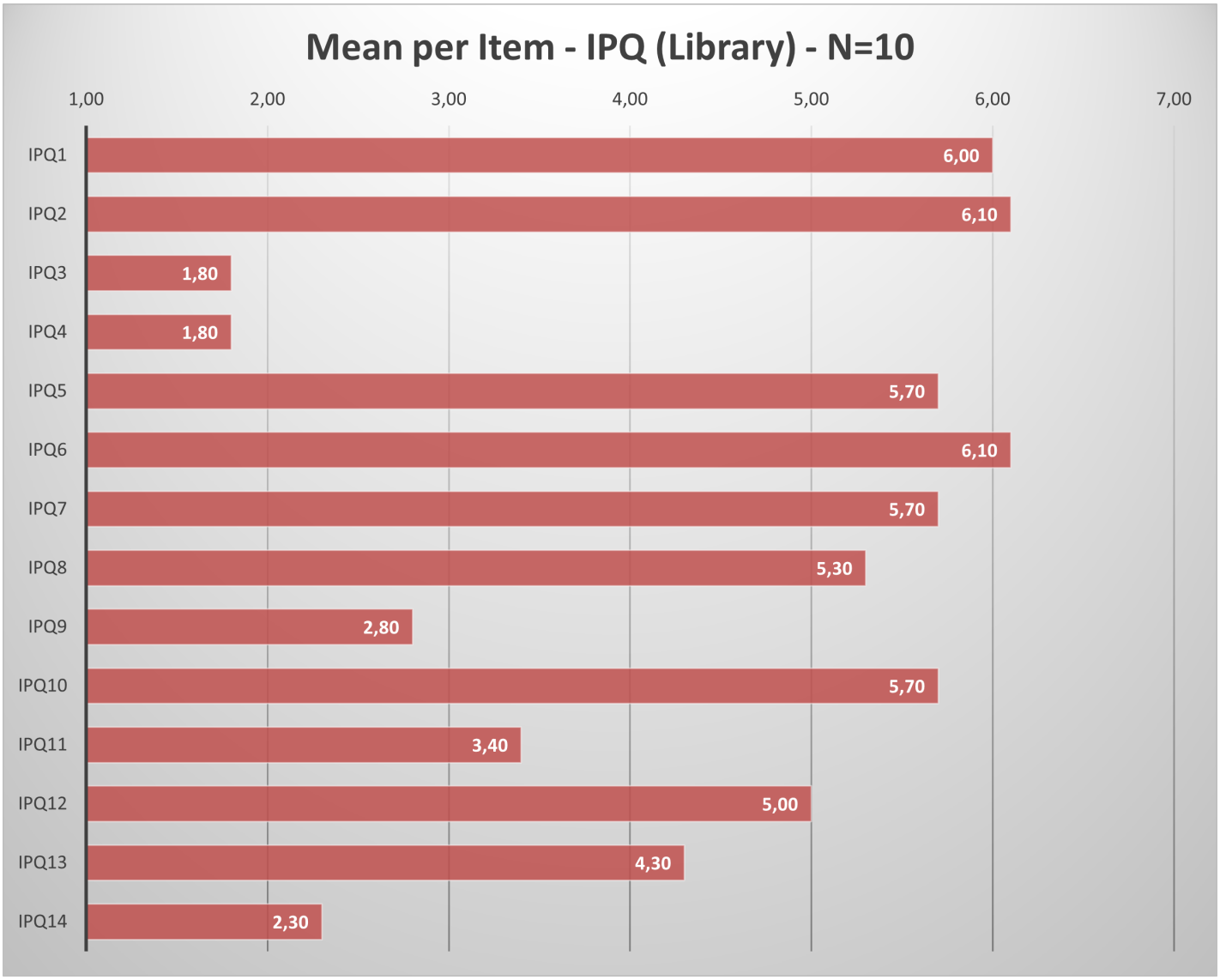

Figure 42: Mean per item

6.3.5 Hill

IPQ

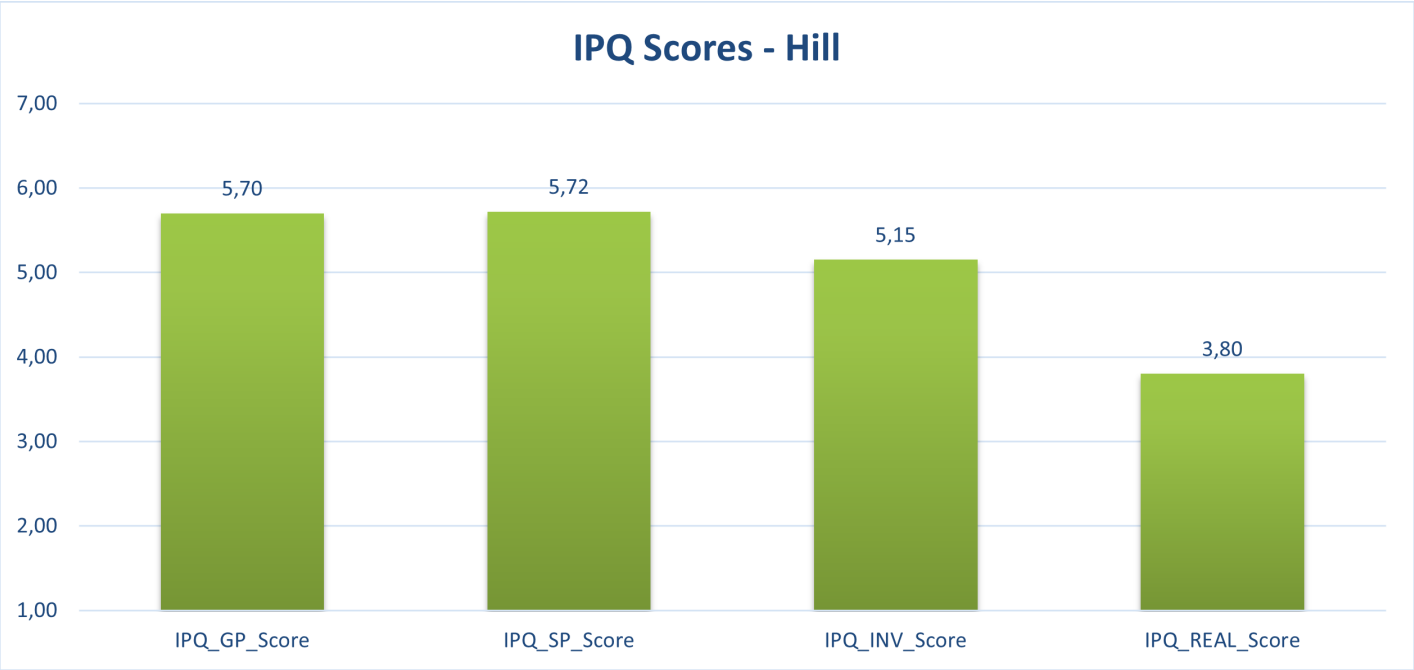

Figure 43: Overall Score, General Presence, Spatial Presence, Involvement, Realism

Item	Mean	Std. Dev.	Variance
IPQ1	5,70	1,06	1,12
IPQ2	5,90	0,74	0,54
IPQ3	1,90	0,88	0,77
IPQ4	2,90	1,60	2,54
IPQ5	5,50	1,51	2,28
IPQ6	6,00	0,67	0,44
IPQ7	5,10	1,45	2,10
IPQ8	5,40	1,17	1,38
IPQ9	3,40	1,78	3,16
IPQ10	5,50	1,27	1,61
IPQ11	3,80	1,55	2,40
IPQ12	5,30	0,95	0,90
IPQ13	3,60	1,26	1,60
IPQ14	2,10	0,74	0,54

Figure 44: Mean, Standard Deviation and Variance per item

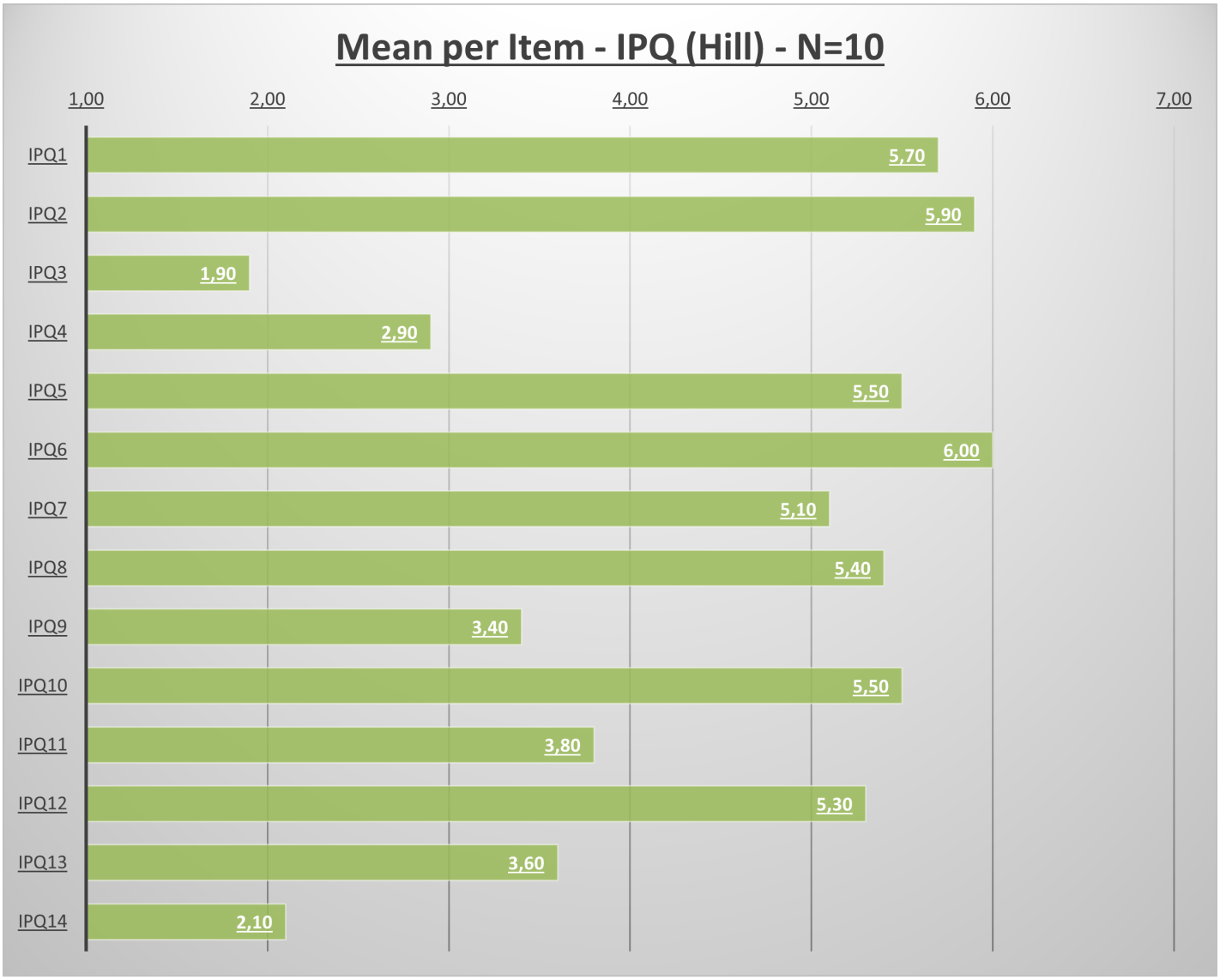

Figure 45: Mean per item

6.3.6 Comparison between environments

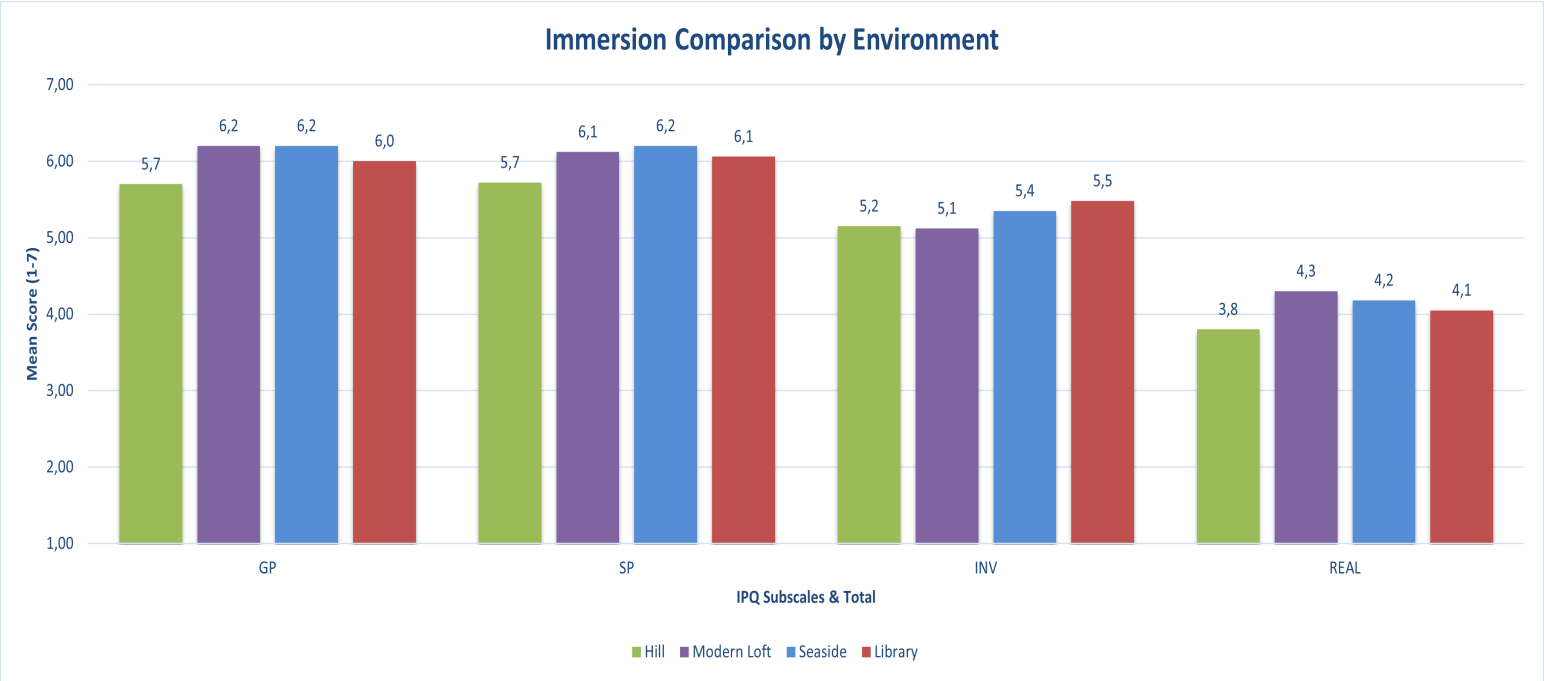

Figure 46: IPQ metrics (General Presence, Spatial Presence, Involvement, Realism) compared for each environment

Chapter 7

Conclusion and Future Work

This thesis has presented the design, development, and evaluation of a novel Virtual Reality system for the treatment of Substance Use Disorder (SUD). By integrating a standalone VR client with a remote, LLM-driven conversational agent, the project successfully demonstrates that high-fidelity immersion and advanced artificial intelligence can be combined to create a scalable, accessible therapeutic tool.

7.1 Summary of Achievements

The primary objective of this work was to bridge the gap between clinical efficacy and technological accessibility. The usability testing conducted with 20 participants has yielded highly encouraging results, particularly regarding the visual fidelity and the sense of presence.

The quantitative data from the IPQ (Igroup Presence Questionnaire) indicates that the therapeutic environments—specifically the *Seaside* and the *Modern Loft*—were effective in generating a strong sense of "being there." Participants reported that the high quality of the baked lighting, the realistic textures, and the spatial audio contributed significantly to their ability to disengage from the real world and focus on the therapeutic task.

Furthermore, the decoupled architecture proved robust. The custom UDP network discovery service allowed for seamless connection in various network conditions without manual configuration, and the latency introduced by the remote HPC inference was deemed acceptable for a paced therapeutic conversation. The system successfully managed the complex state machine of the therapy, validating the technical feasibility of the proposed three-tier architecture.

7.2 Limitations of the Current System

Despite the technical success, the usability study highlighted specific limitations that must be addressed to maximize clinical efficacy.

The most significant limitation concerns the Virtual Therapist. While the character model was perceived as visually realistic (high-poly mesh, realistic skin shaders), the interaction suffered from a lack of non-verbal communication. Participants noted that while the *content* of the therapist's speech was empathetic and context-aware, her *delivery* was static. The current system relies on basic lip-syncing (OVR lipsync), head and eyes movement (SALSA), some facial expressions based on keywords (SALSA) and idle animations, but it lacks the dynamic facial expressions, prosody-driven gestures. This discrepancy creates a slight "uncanny valley"

effect, where the visual realism sets an expectation of emotional intelligence that the animation system does not yet meet.

Additionally, while the URP pipeline delivers good performance, the visual realism is still constrained by the mobile hardware of the Meta Quest, preventing the use of advanced volumetric lighting or real-time global illumination that could further enhance immersion.

7.3 Future Work

The development of this platform is an ongoing process. Based on the feedback and limitations identified, the following areas have been prioritized for future development.

7.3.1 Changes for Compatibility with all VR Headsets

Currently, the application is tightly coupled with the Meta ecosystem, utilizing the `OVRCameraRig` and the `Oculus Interaction SDK` for hand tracking and UI interaction. To ensure the tool can be widely adopted by clinics using different hardware (e.g., HTC Vive, Pico, or Apple Vision Pro), a migration to the OpenXR standard is necessary.

Future iterations will replace the Oculus-specific scripts with Unity's XR Interaction Toolkit (XRI). This abstraction layer will allow the application to follow a "write once, deploy anywhere" paradigm, making the software hardware-agnostic and future-proof.

7.3.2 Therapist's Emotions and Expressiveness

To bridge the emotional gap identified during testing, the virtual agent requires a sophisticated "Non-Verbal Behavior (NVB) Generation System". Future work will focus on implementing a "Sentiment Analysis" module within the Python server.

On the Unity client, instead of simple idle loops, the therapist will trigger specific blend shapes for micro-expressions (e.g., furrowing brows for concern, smiling for encouragement) and body gestures (e.g., leaning forward to show active listening). This will transform the agent from a static speaker into an expressive digital human, crucial for maintaining patient engagement over long-term therapy.

7.3.3 Environmental Fidelity and Flexibility

To further enhance the sense of presence, future development will focus on two environmental aspects:

- **Scenario Expansion:** Creating a broader library of environments (e.g., a Forest at dusk, a Japanese Garden) to cater to diverse user preferences.
- **Decoupling Logic from Scene:** The codebase will be refactored to allow a "Mix-and-Match" approach. Users will be able to select their preferred therapeutic technique (e.g., Guided Breathing) and apply it to *any* available environment.

7.3.4 Tests with Patients

The ultimate validation of this system lies in its clinical efficacy. The usability test confirmed the system is stable and usable by healthy volunteers. The next critical step is to deploy the system in a real-world clinical setting.

A pilot clinical trial is planned in collaboration with the `University of Maryland School of Medicine`. This study will involve actual patients diagnosed with Substance Use Disorder (SUD). The focus will shift from usability metrics (UEQ, IPQ) to clinical outcomes, specifically measuring the reduction in craving (VAS scores) and the retention rate in treatment programs compared to standard care.

Bibliography

[1] American Psychiatric Association (2013) Diagnostic and statistical manual of mental disorders: DSM-5. 5th edn. Washington, D.C.: American Psychiatric Publishing.

[2] American Psychiatric Association: Diagnostic and statistical manual of mental disorders, edn 4. Washington, DC: American Psychiatric Association Press; 1994

[3] Substance Abuse and Mental Health Services Administration: Impact of the DSM-IV to DSM-5 Changes on the National Survey on Drug Use and Health, Rockville (MD); 2016 Jun.

[4] Hasin, D. S., O'Brien, C. P., Auriacombe, M., Borges, G., Bucholz, K., Budney, A., et al. (2013). DSM-5 criteria for substance use disorders: Recommendations and rationale. American Journal of Psychiatry, 170(8), 834–851. doi:10.1176/appi.ajp.2013.12060782

[5] *The Gale Encyclopedia of Medicine*, 6th ed., vol. 8, J. L. Longe, Ed. Gale, 2020.

[6] Lütt A, Tsamitros N, Wolbers T, Rosenthal A, Bröcker AL, Schöneck R, Bermpohl F, Heinz A, Beck A, Gutwinski S. An explorative single-arm clinical study to assess craving in patients with alcohol use disorder using Virtual Reality exposure (CRAVE)-study protocol. BMC Psychiatry. 2023 Nov 14;23(1):839. doi: 10.1186/s12888-023-05346-y. PMID: 37964300; PMCID: PMC10647047.

[7] Deng H, Zhang R, Wang C, Zhang B, Wang J, Wang S, Zhang J, Shari NI, Leong Bin Abdullah MFI. The efficacy of virtual reality exposure therapy for the treatment of alcohol use disorder among adult males: a randomized controlled trial comparing with acceptance and commitment therapy and treatment as usual. Front Psychiatry. 2023 Aug 22;14:1215963. doi: 10.3389/fpsyt.2023.1215963. PMID: 37674551; PMCID: PMC10477784.

[8] Hernández-Serrano O, Ghiţă A, Fernández-Ruiz J, Monràs M, Gual A, Gacto M, Porras-García B, Ferrer-García M, Gutiérrez-Maldonado J. Determinants of Cue-Elicited Alcohol Craving and Perceived Realism in Virtual Reality Environments among Patients with Alcohol Use Disorder. J Clin Med. 2021 May 21;10(11):2241. doi: 10.3390/jcm10112241. PMID: 34064120; PMCID: PMC8196721.

[9] Nègre F, Lemercier-Dugarin M, Gomet R, Pelissolo A, Malbos E, Romo L, Zerdazi EH. Study on the efficiency of virtual reality in the treatment of alcohol use disorder: study protocol for a randomized controlled trial : E-Reva. Trials. 2024 Jun 27;25(1):417. doi: 10.1186/s13063-024-08271-x. PMID: 38937776; PMCID: PMC11212355.

[10] Ryan JJ, Kreiner DS, Chapman MD, Stark-Wroblewski K. Virtual reality cues for binge drinking in college students. Cyberpsychol Behav Soc Netw. 2010 Apr;13(2):159-62. doi: 10.1089/cyber.2009.0211. PMID: 20528271.

[11] Lee J, Lim Y, Graham SJ, Kim G, Wiederhold BK, Wiederhold MD, Kim IY, Kim SI. Nicotine craving and cue exposure therapy by using virtual environments. Cyberpsychol Behav. 2004 Dec;7(6):705-13. doi: 10.1089/cpb.2004.7.705. PMID: 15687806.

[12] Traylor AC, Bordnick PS, Carter BL. Using virtual reality to assess young adult smokers' attention to cues. Cyberpsychol Behav. 2009 Aug;12(4):373-8. doi: 10.1089/cpb.2009.0070. PMID: 19630582; PMCID: PMC4104935.

[13] Paris MM, Carter BL, Traylor AC, Bordnick PS, Day SX, Armsworth MW, Cinciripini PM. Cue reactivity in virtual reality: the role of context. Addict Behav. 2011 Jul;36(7):696-9. doi: 10.1016/j.addbeh.2011.01.029. Epub 2011 Jan 26. PMID: 21349649; PMCID: PMC4104934.

[14] García-Rodríguez O, Weidberg S, Gutiérrez-Maldonado J, Secades-Villa R. Smoking a virtual cigarette increases craving among smokers. Addict Behav. 2013 Oct;38(10):2551-4. doi: 10.1016/j.addbeh.2013.05.007. Epub 2013 May 23. PMID: 23793042.

[15] Thomas Zandonai, Giulia Benvegnù, Francesco Tommasi, Elisa Ferrandi, Elettra Libener, Stefano Ferraro, Bogdan Maris, Cristiano Chiamulera, A virtual reality study on postretrieval extinction of smoking memory reconsolidation in smokers, Journal of Substance Abuse Treatment, Volume 125, 2021, 108317, ISSN 0740-5472, https://doi.org/10.1016/j.jsat.2021.108317.

[16] Bordnick PS, Copp HL, Traylor A, Graap KM, Carter BL, Walton A, Ferrer M. Reactivity to cannabis cues in virtual reality environments. J Psychoactive Drugs. 2009 Jun;41(2):105-12. doi: 10.1080/02791072.2009.10399903. PMID: 19705672; PMCID: PMC4104948.

[17] Lehoux T, Capobianco A, Lacoste J, Rollier S, Mopsus Y, Melgire M, Lecuyer F, Gervilla M, Weiner L. Virtual reality cue-exposure therapy in reducing cocaine craving: the Promoting Innovative COgnitive behavioral therapy for Cocaine use disorder (PICOC) study protocol for a randomized controlled trial. Trials. 2024 Jun 27;25(1):421. doi: 10.1186/s13063-024-08275-7. PMID: 38937824; PMCID: PMC11212420.

[18] Lebiecka, Z., Skoneczny, T., Tyburski, E., Samochowiec, J., Kucharska-Mazur, J. (2021). Is Virtual Reality Cue Exposure a Promising Adjunctive Treatment for Alcohol Use Disorder? Journal of Clinical Medicine, 10(13), 2972. https://doi.org/10.3390/jcm10132972

[19] Faraj MM, Lipanski NM, Morales A, Goldberg E, Bluth MH, Marusak HA, Greenwald MK. A Virtual Reality Meditative Intervention Modulates Pain and the Pain Neuromatrix in Patients with Opioid Use Disorder. Pain Med. 2021 Nov 26;22(11):2739-2753. doi: 10.1093/pm/pnab162. PMID: 33956146; PMCID: PMC11494379.

[20] McGirt MJ, Holland CM, Farber SH, Zuckerman SL, Spertus MS, Theodore N, Pfortmiller D, Stanley G. Remote cognitive behavioral therapy utilizing an in-home virtual reality toolkit (Vx Therapy) reduces pain, anxiety, and depression in patients with chronic cervical and lumbar spondylytic pain: A potential alternative to opioids in multimodal pain management. N Am Spine Soc J. 2023 Oct 19;16:100287. doi: 10.1016/j.xnsj.2023.100287. PMID: 38033880; PMCID: PMC10684389.

[21] Thaysen-Petersen D, Hammerum SK, Vissing AC, Arnfred BT, Nordahl R, Adjorlu A, Nordentoft M, Oestrich IH, Düring SW, Fink-Jensen A. Virtual reality-assisted cognitive behavioural therapy for outpatients with alcohol use disorder (CRAVR): a protocol for a

randomised controlled trial. BMJ Open. 2023 Mar 29;13(3):e068658. doi: 10.1136/bmjopen-2022-068658. PMID: 36990475; PMCID: PMC10069573.

[22] Gamito P, Oliveira J, Matias M, Cunha E, Brito R, Lopes PF, Deus A. Virtual Reality Cognitive Training Among Individuals With Alcohol Use Disorder Undergoing Residential Treatment: Pilot Randomized Controlled Trial. J Med Internet Res. 2021 Jan 29;23(1):e18482. doi: 10.2196/18482. PMID: 33512329; PMCID: PMC7880813.

[23] Man DWK. Virtual reality-based cognitive training for drug abusers: A randomised controlled trial. Neuropsychol Rehabil. 2020 Mar;30(2):315-332. doi: 10.1080/09602011.2018.1468271. Epub 2018 May 8. PMID: 29734923.

[24] Rehl D, Mangapora M, Love M, Love C, Shaw K, McCarthy J, Beverly EA. Feasibility of a cinematic-virtual reality training program about opioid use disorder for osteopathic medical students: a single-arm pre-post study. J Osteopath Med. 2024 Jul 5;124(11):509-516. doi: 10.1515/jom-2023-0188. PMID: 38965036.

[25] Wang YG, Liu MH, Shen ZH. A virtual reality counterconditioning procedure to reduce methamphetamine cue-induced craving. J Psychiatr Res. 2019 Sep;116:88-94. doi: 10.1016/j.jpsychires.2019.06.007. Epub 2019 Jun 14. PMID: 31226580.

[26] Langener S, Kolkmeier J, VanDerNagel J, Klaassen R, van Manen J, Heylen D. Development of an Alcohol Refusal Training in Immersive Virtual Reality for Patients With Mild to Borderline Intellectual Disability and Alcohol Use Disorder: Cocreation With Experts in Addiction Care. JMIR Form Res. 2023 Apr 26;7:e42523. doi: 10.2196/42523. PMID: 37099362; PMCID: PMC10173034.

[27] Ascone L, Wirtz J, Mellentin AI, Kugler D, Bremer T, Schadow F, Hoppe S, Jebens C, Kühn S. Transferring the approach avoidance task into virtual reality: a study in patients with alcohol use disorder versus healthy controls. Virtual Real. 2023;27(3):2711-2722. doi: 10.1007/s10055-023-00835-7. Epub 2023 Aug 3. PMID: 37614715; PMCID: PMC10442255.

[28] Mellentin AI, Nielsen AS, Ascone L, Wirtz J, Samochowiec J, Kucharska-Mazur J, Schadow F, Lebiecka Z, Skoneczny T, Mistarz N, Bremer T, Kühn S. A randomized controlled trial of a virtual reality based, approach-avoidance training program for alcohol use disorder: a study protocol. BMC Psychiatry. 2020 Jun 30;20(1):340. doi: 10.1186/s12888-020-02739-1. PMID: 32605614; PMCID: PMC7324964.

[29] Son JH, Lee SH, Seok JW, Kee BS, Lee HW, Kim HJ, Lee TK, Han DH. Virtual Reality Therapy for the Treatment of Alcohol Dependence: A Preliminary Investigation With Positron Emission Tomography/Computerized Tomography. J Stud Alcohol Drugs. 2015 Jul;76(4):620-7. doi: 10.15288/jsad.2015.76.620. PMID: 26098039.

[30] Greenwald HJ, Berger A, Wilson RLH, Greenwald DJ, Lannon E, Johnson-Smith P, Bergman BG, Wilens TE. A pilot study of virtual reality for inpatients with opioid use disorder. Am J Addict. 2024 Jul;33(4):423-429. doi: 10.1111/ajad.13526. Epub 2024 Mar 2. PMID: 38430207.

[31] Chiang CH, Huang CM, Sheu JJ, Liao JY, Hsu HP, Wang SW, Guo JL. Examining the Effectiveness of 3D Virtual Reality Training on Problem-solving, Self-efficacy, and Teamwork

Among Inexperienced Volunteers Helping With Drug Use Prevention: Randomized Controlled Trial. J Med Internet Res. 2021 Nov 2;23(11):e29862. doi: 10.2196/29862. PMID: 34726606; PMCID: PMC8596241.

[32] "Apartment Kit" by *Brick Project Studio* on Unity Asset Store. Available at: `https://assetstore.unity.com/packages/3d/environments/apartment-kit-124055`

[33] "SOFA DEC12" by *jordmarzeti* on Sketchfab, used under CC BY 4.0 License. Available at: `https://sketchfab.com/3d-models/sofa-dec12-adf542b6a74b40c9ad85dc59c6de0dcc`

[34] "Modern Living Room" by *Visthétique* on Sketchfab, used under CC BY 4.0 License. Available at: `https://sketchfab.com/3d-models/modern-living-room-ec7179648b1e43739104994d64e95673`

[35] "Simple Water Shader URP" by *IgniteCoders* on Unity Asset Store. Available at `https://assetstore.unity.com/packages/2d/textures-materials/water/simple-water-shader-urp-191449`

[36] "The Toby Foliage Engine / Light" by *Tobyfredson* on Unity Asset Store. Available at `https://assetstore.unity.com/packages/vfx/shaders/the-toby-foliage-engine-light-282901`

[37] "Library Reading Room" by *IL.ranch* on Unity Asset Store. Available at `https://assetstore.unity.com/packages/3d/environments/urban/library-reading-room-hdrp-bip-83806`

[38] "URP Tree Models" by *ALIyerEdon* on Unity Asset Store. Available at `https://assetstore.unity.com/packages/3d/vegetation/trees/urp-tree-models-253340`