Politecnico di Torino

Politecnico di Torino

Cinema And Media Engineering

2024/2025

Graduation Session July 2025

Designing for Spatial Computing

Exploring Natural and Traditional Interfaces in a VR Moodboarding Tool

Supervisors:

Andrea Bottino

Francesco Strada

Candidate:

Pietro Uras

Abstract

This study explores the use of multimodal interaction techniques in spatial computing through a VR moodboarding application powered by AI-generated imagery. The system requires an HMD and is based entirely on hand tracking, enabling hands-free interaction without the use of controllers. Three interaction modalities are examined: a traditional interface using virtual buttons, a gesture-based setup leveraging hand poses for core actions such as image generation, voice input, and GUI navigation, and a hybrid mode combining both approaches. The evaluation includes measures of task completion time, accuracy, and user engagement, along with user preferences across different interaction contexts. The work also reflects on user experience design in immersive environments and outlines a structured process for prototyping interactive systems in VR, from concept exploration to implementation.

Acknowledgements

I would like to express my heartfelt gratitude to my family and friends for their unwavering support throughout this journey.

I am also deeply thankful to the supervisors of this thesis for their valuable guidance and for helping me develop and structure this project.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Goal

The evolution of spatial computing and immersive technologies is reshaping the landscape of digital interaction. Among these technologies, **Virtual Reality (VR)** stands out for its capacity to offer users highly immersive, three-dimensional environments where interaction is no longer confined to traditional screens or input devices. In this context, the exploration of **multimodal interaction techniques** — that is, combining different input modalities such as voice, gestures, and traditional controllers — is becoming essential to enhance user experience and efficiency in virtual environments.

This thesis investigates the design and evaluation of a **moodboarding system** within a spatial computing framework, leveraging **AI-generated images** to support creative workflows. The application runs on a virtual reality headset, specifically the **Meta Quest 2**, and operates entirely through **hand tracking**, without the use of physical controllers. The objective is to test and compare three distinct interaction modalities within this immersive context: a **traditional interface** based on virtual buttons, a **gesture-driven interface** relying on custom-designed hand gestures for key operations, and a **hybrid interface** that combines both approaches. The moodboarding task serves as the core activity to evaluate the usability, efficiency, and user preference associated with each interaction model.

The study focuses on identifying the advantages and limitations of each interaction mode in terms of **task performance**, **user engagement**, and **interaction preference**. By evaluating user behavior within the hybrid setup, this research also seeks to uncover patterns in input selection, aiming to define which interaction strategies are most naturally adopted by users for specific tasks. The ultimate goal is to provide design insights that can inform the development of intuitive and effective multimodal interfaces for creative tools in spatial computing.

1.2 Context and Motivation

The increasing adoption of **spatial computing technologies** is transforming how users engage with digital content. In **VR**, users are not merely spectators; they can physically interact with virtual elements, fostering a stronger sense of presence and agency. Creative tasks, such as **moodboarding** — the process of assembling visual materials to convey an aesthetic or conceptual direction — are particularly suited to benefit from these immersive and interactive environments.

Moodboarding is a well-established technique in fields like **design**, **advertising**, **filmmaking**, and **game development**. Traditionally, it relies on two-dimensional tools where images are collected, arranged, and annotated on static boards. However, this process remains largely bound to flat interfaces and mouse-based interactions, which can limit the user's spatial understanding and creative fluidity.

The advent of **AI-generated content** introduces new possibilities for enhancing creative workflows by automating and diversifying the generation of visual assets. Integrating these capabilities into a **VR environment** enables users to interact with **AI systems** in more natural, embodied ways — for example, by using gestures to generate new images or employing speech commands to refine prompts. Such interaction methods can potentially reduce cognitive load, speed up the creative process, and increase user satisfaction.

Despite the growing interest in **multimodal interfaces**, there is still limited research exploring how users combine or switch between different input modalities in creative VR applications. Understanding these behaviors is essential to design systems that feel intuitive, efficient, and enjoyable. Specifically, this thesis aims to answer the following questions:

- Do users prefer traditional virtual button-based interfaces or gesture-based interaction paradigms in virtual reality environments?
- In a hybrid interface, do users naturally prefer to use gestures or virtual buttons for specific types of tasks (e.g., image creation, GUI navigation, text dictation)?
- What role can hand gestures play in shaping the interaction design of multimodal creative applications in spatial computing?

Through this work, the thesis contributes to the ongoing exploration of **natural and intuitive interfaces** within spatial computing, providing practical insights specifically for the design of immersive VR applications based on hand tracking technologies.

The implementation developed for this research will be made available on GitHub at: `https://github.com/PietroUras/VRMoodboarding`

Chapter 2

State of the Art

This chapter provides a comprehensive overview of the current state of immersive technologies, natural user interfaces, and spatial computing, laying the groundwork for understanding the technical and conceptual landscape relevant to this thesis. It begins by defining and distinguishing key terms such as XR, VR, AR, and MR, and presents an overview of head-mounted displays (HMDs) as essential hardware enablers. The chapter then explores the principles of natural user interfaces and multimodal interaction, focusing on input modalities like speech, gaze, and gestures. It proceeds to examine spatial computing with an emphasis on productivity and creative tools. The discussion then narrows to moodboarding practices, tracing their evolution from traditional methods to VR-enhanced workflows. Finally, the chapter briefly introduces AI-powered image generation and AI-driven workflows.

2.1 Immersive Technologies Overview

2.1.1 Differences between XR, VR, MR and AR

The landscape of immersive technologies is characterized by a spectrum of experiences that blend the real and virtual worlds to varying degrees [1]. To accurately discuss these technologies, it is crucial to differentiate between:

- **Extended Reality (XR)**: An umbrella term encompassing all real-and-virtual combined environments and human-machine interactions generated by computer technology and wearables. It represents a continuum from the entirely real to the entirely virtual, including VR, AR, and MR [2].
- **Virtual Reality (VR)**: Immerses users in a fully synthetic digital environment, effectively replacing their perception of the real world. Typically experienced through Head-Mounted Displays (HMDs), which block out external visual and auditory stimuli, VR offers high immersion and presence [3]. Interactions rely on

spatial input/output devices, timely sensory feedback, and spatial information, often involving embodied cognition [4].

- **Augmented Reality (AR)**: Overlays digital information onto the real world, enhancing rather than replacing the user's perception. Commonly accessed via smartphones or transparent AR glasses, AR blends real and virtual content through real-time interaction and accurate 3D registration [5]. AR interfaces often adapt to user needs without the computational intensity of full VR [6].
- **Mixed Reality (MR)**: Enables deeper integration and interaction between real and virtual elements than AR. MR allows virtual objects to respond to the physical environment through spatial mapping and advanced sensors [2]. Devices like Microsoft HoloLens exemplify MR, where digital content is convincingly anchored within the real world.

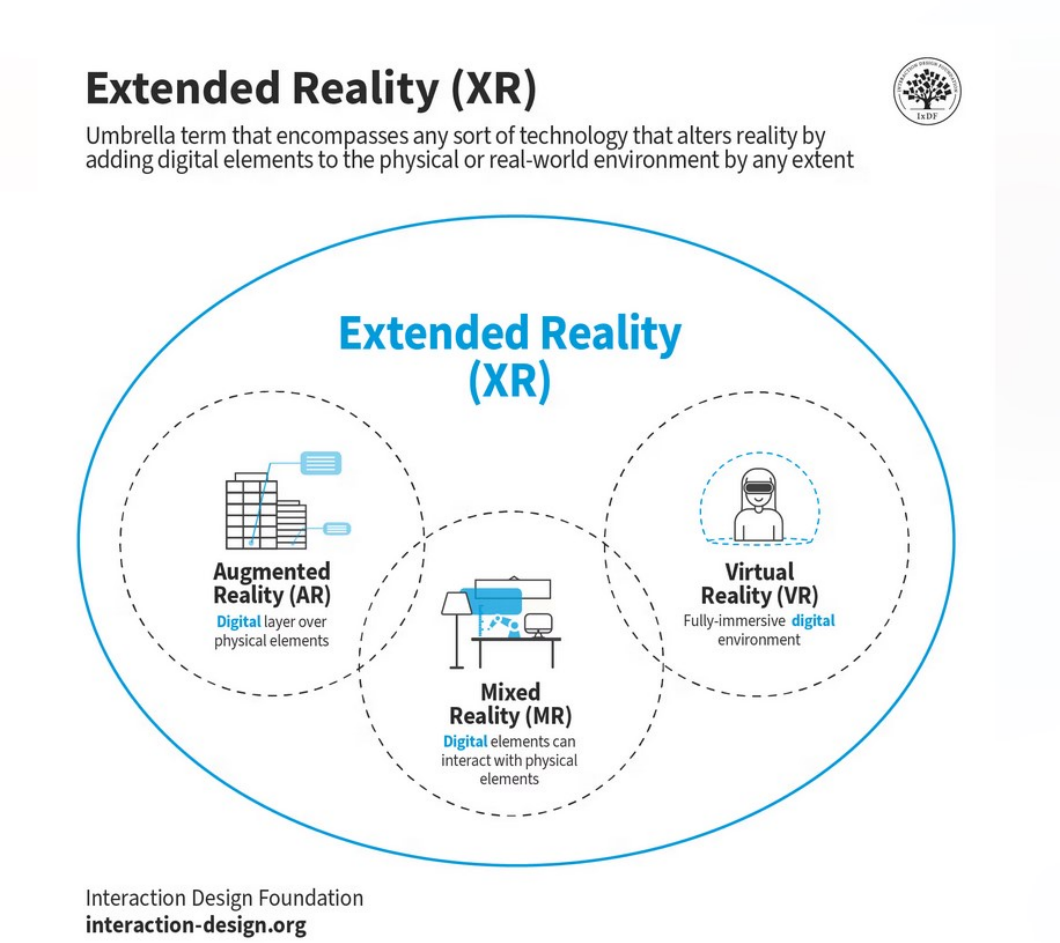


Figure 2.1: Extended Reality Definition [1].

2.1.2 Head-Mounted Displays

Head-Mounted Displays (HMDs) are the primary hardware devices that enable immersive experiences in **VR**, **AR**, and **MR**. These devices are worn on the head, positioning optical systems directly in front of the user's eyes to deliver digital content.

Early HMDs, such as Ivan Sutherland's **"The Sword of Damocles"** in 1968, laid the foundational concepts for immersive viewing [3].

Figure 2.2: The Sword of Damocles

Modern **VR HMDs**, such as the **Meta Quest series**, **HTC Vive**, and **Valve Index**, typically consist of a screen (or two, one for each eye) and lenses that provide a wide field of view and create the illusion of depth. They often include integrated audio and various sensors (e.g., accelerometers, gyroscopes, magnetometers) for head tracking, allowing the virtual environment to respond dynamically to the user's head movements.

A prominent example of a standalone **VR HMD** is the **Meta Quest 2**. Released by Meta (formerly Oculus), it offers a relatively affordable and accessible entry point into virtual reality. It features an LCD screen with a resolution of **1832 × 1920 pixels per eye** and supports refresh rates up to **120 Hz**. Its inside-out tracking system, powered by four onboard cameras, eliminates the need for external base stations, streamlining setup and use. The device supports both standalone operation—running applications natively—and PC VR via **Oculus Link** or **Air Link**. While the Quest 2 is no longer the most advanced headset in Meta's lineup, it includes all the core functionalities necessary for this study, such as spatial tracking, gesture recognition, and voice input capabilities. More recent models, such as the **Quest 3**, build upon these foundations with improved hardware and passthrough features.

Another example is the **Apple Vision Pro**, which represents a significant advancement in mixed reality. Positioned as a **"spatial computer"**, it seamlessly blends digital content with the physical world, emphasizing intuitive interaction through natural input like eyes, hands, and voice. It features an ultra-high-resolution micro-OLED display system, delivering **23 million pixels across two displays**, and is powered by Apple's **M2** and **R1 chips** for low-latency passthrough video and spatial audio. Its design integrates advanced sensors for precise tracking and environmental understanding, aiming to provide a highly immersive and comfortable computing experience.

AR and **MR HMDs**, such as the **Magic Leap One** and **Microsoft HoloLens**, are designed to be transparent, allowing users to see the real world while digital content is projected onto their field of view. These devices incorporate advanced cameras and depth sensors to map the physical environment, enabling precise placement and interaction with virtual objects within real space. The technological advancements in **HMDs**, including higher resolution displays, wider fields of view, improved tracking accuracy, and reduced form factors, are continuously pushing the boundaries of immersive experiences and enabling more natural interactions.

Figure 2.3: Meta Quest 3 and Apple Vision Pro

2.2 Natural User Interfaces

Natural User Interfaces (NUIs) are interaction paradigms that enable users to engage with digital systems through innate human behaviors, such as gestures, speech, and gaze, rather than traditional input devices like keyboards and mice. By leveraging human-centric communication methods, NUIs aim to create intuitive and accessible interactions. A subset of NUIs, **perceptual user interfaces**, further enhance this paradigm by relying on the full spectrum of human perceptual,

motor, and cognitive abilities [7]. These interfaces are particularly well-suited for multimodal **extended reality (XR)** and **three-dimensional user interfaces (3D UIs)**, as they harness spatial memory, position, and orientation [4].

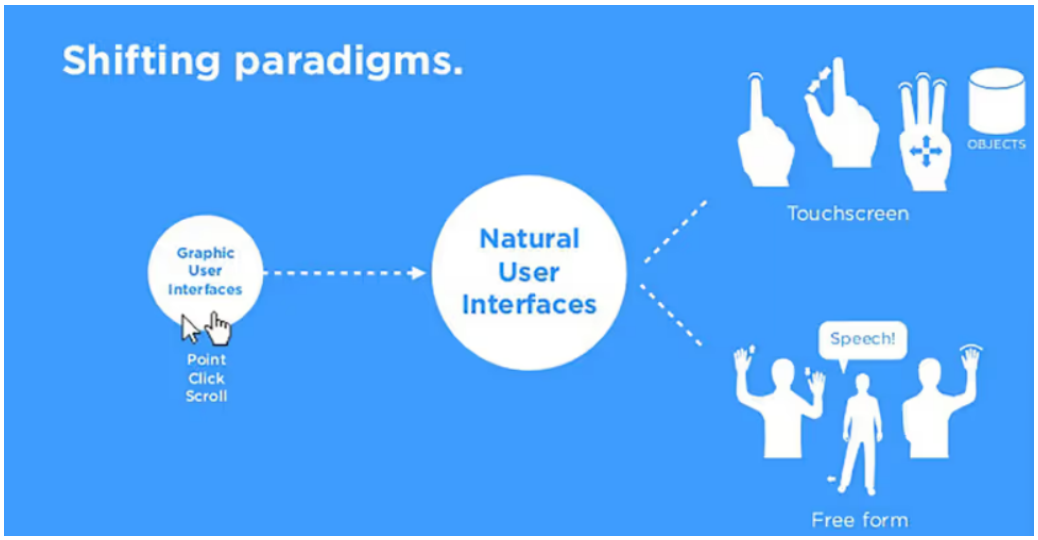


Figure 2.4: Natural User Interface Paradigm

2.3 Multimodality

In **Human-Computer Interaction (HCI)**, multimodality refers to the use of multiple input modalities—such as speech, gesture, gaze, and touch—enabling users to interact with digital systems in a more natural and expressive way [8]. This approach enhances usability and robustness by allowing users to choose the most convenient modality for each task or to combine them for greater clarity. For example, a user might point at an object while saying "move this," allowing the system to interpret both spatial and semantic cues. The integration of modalities not only improves task performance but also increases user satisfaction, particularly in complex or dynamic environments [9, 8].

The concept of multimodal interaction dates back to pioneering systems like Bolt's **"Put-That-There"** (1980) [9], which allowed users to issue voice commands while pointing to objects on a screen. This early work demonstrated the intuitive synergy between modalities, a finding echoed in later **"Wizard of Oz"** studies where users favored combining speech and gesture for object manipulation in virtual environments. Oviatt's research further validated that users naturally combine modalities, especially during manipulation tasks, reinforcing the idea that multimodal systems align closely with innate human behaviors [6].

This interaction style has given rise to distinct paradigms. One is **redundancy**, where different modalities can serve the same function, such as a hand posture performing the same action as a spoken command. Redundancy increases system robustness by providing fallback options and reducing error rates [6]. Another is **sequential mode interpretation**, where input is structured across time—such as using gesture for spatial positioning followed by voice for confirmation. While this sequencing can introduce switching costs, in **extended reality (XR)** environments it often feels fluid, as modalities typically rely on separate cognitive and physical channels [6].

Despite its benefits, multimodal interaction presents significant challenges. Effective fusion of input streams is complex: while machines typically process modalities independently and merge outputs late, human communication is integrative and context-sensitive. Systems must interpret temporally aligned or overlapping gestures, gaze, and speech as unified acts, resolving ambiguity and partial input through context-aware fusion strategies [6].

Recent developments in **head-mounted displays (HMDs)** have driven the advancement of immersive multimodal interfaces. Devices like the **Meta Quest Pro** and **Apple Vision Pro** support real-time tracking of hands, gaze, and voice, enabling users to interact with large spatial canvases through natural input. Gestures allow for precise spatial manipulation, while voice input offers rapid access to system functions.

To better understand the unique affordances and challenges of each modality, the following sections examine **speech**, **gaze**, and **gesture input** individually.

2.3.1 Speech

Speech serves as a powerful modality in **Natural User Interfaces (NUIs)**, allowing users to interact with digital systems using spoken language. Enabled by advancements in **Automatic Speech Recognition (ASR)** and **Natural Language Understanding (NLU)**, speech interfaces have evolved from simple command-and-control mechanisms to systems capable of handling complex dialogue and contextual interpretation [10]. In **XR environments**, speech offers distinct advantages due to its hands-free and eyes-free nature, making it ideal for tasks that require simultaneous physical activity or when visual attention must remain focused elsewhere [4]. For instance, in professional settings such as industrial maintenance, speech enables users to issue commands while manipulating tools or inspecting machinery.

In addition to accessibility, speech is efficient and expressive: users can convey abstract concepts through natural language, reference known objects or actions, and make selections from large option sets more easily than through manual input. These characteristics make it particularly useful for interacting with virtual agents

or characters, where naturalistic dialogue enhances immersion [6]. However, speech input is not without its challenges. Unlike visual modalities, speech is inherently slower when conveying large volumes of information, and **XR environments** often lack optimized visual rendering for text output. While spoken output can be experienced privately through headphones, input and output are vulnerable to noisy environments, despite the partial mitigation offered by noise-cancellation technologies.

Another key issue is recognition error. Given the probabilistic nature of speech recognition systems, users must receive transparent feedback about how their input was interpreted. Effective feedback and repair mechanisms are essential to maintain usability and prevent frustration [6]. Additionally, speech is transient and lacks spatial characteristics, which limits its use for continuous or position-sensitive inputs. Yet, this same quality makes it well-suited for discrete, high-level commands such as tool selection or navigation triggers, where quick verbal shortcuts are advantageous [11].

An unconventional yet intriguing input method is the **Exhalation Interface**, which captures blowing actions through built-in microphones for subtle, hands-free control. Though limited in bandwidth, this method has proven engaging and expressive in gaming and artistic applications, often enhancing users' sense of presence [12].

2.3.2 Gaze

Gaze input leverages users' eye movements to infer focus, attention, and intent. In **XR systems**, gaze tracking provides a low-effort, high-precision way to interact with digital content. It supports both passive uses—like attention analytics or foveated rendering—and active input for selecting objects or navigating menus. When used explicitly, gaze interaction typically involves mechanisms like dwell time, blinks, or multimodal confirmation to avoid unintended activations, a problem known as the **"Midas touch"** [12, 13]. To distinguish intentional commands from casual observation, systems often implement gaze selection strategies such as dwell-select, blink-based triggers, or combined input with gestures or voice [13].

Advanced interaction techniques such as **gaze gestures**—patterns of saccadic movement—and smooth pursuit tracking expand the expressive capacity of gaze input beyond simple pointing. Gaze tracking also plays a pivotal role in user research, from studying attention patterns in shopping simulations to assessing cognitive load and intention in mixed-reality interfaces [14]. Commercial solutions like **Tobii** and **Varjo** integrate gaze tracking into **HMDs** for research and development, supporting metrics-driven UI refinement [12].

In multimodal contexts, gaze is frequently combined with other modalities for mutual disambiguation. Techniques like **SenseShapes** utilize volumes (e.g., cones

from eyes or spheres around hands) to interpret spatial deixis such as "this" or "there" [15]. These regions, analyzed frame-by-frame for object intersection, enable more reliable reference resolution in 3D environments and inform the multimodal integrator when resolving ambiguous commands. Gaze also complements head-tracking techniques, particularly **head gaze raycasting**, where the system projects a ray from the headset to assist in pointing or selection tasks [11].

2.3.3 Gestures and Hand Movements

Gesture-based interaction capitalizes on users' body movements—especially of the hand and fingers—to manipulate digital objects in spatial computing environments. Thanks to innovations in **hand tracking** (e.g., Leap Motion, Meta Quest), gestures have become a staple in immersive systems. Gestures can be categorized as **deictic** (pointing), **manipulative** (grasping or rotating), **symbolic** (representing abstract actions like "OK" or "cancel"), or **navigational** (swiping or teleporting). Their appeal lies in their immediacy, spatiality, and intuitive mapping to real-world actions, which foster user engagement and embodiment [11, 9].

Nonetheless, camera-based gesture tracking presents limitations. Lighting conditions, occlusion, and user variability challenge the reliability of current systems. Recognition errors may stem from difficulties in localizing hands, segmenting gestures, or selecting robust classification features [6]. Despite this, gestures remain valuable in symbolic command execution and continuous manipulations, especially when paired with other modalities. Gesture recognition also depends on movement parameters such as angular velocity and distance; if hands move too fast or outside the detection zone, commands may be missed [11].

Experiments have shown user preferences for one-handed interactions when tasks are simple and for bimanual interaction in complex or prolonged operations, as the latter reduces fatigue [11]. For instance, a user might point with one hand and confirm with the other using a pinch gesture. Gesture-speech multimodality further lowers cognitive load by distributing task demands across verbal and visuo-spatial systems [16].

Legacy gestures, such as **pinch-to-zoom** or **tap-to-select** from touchscreens, continue to influence user expectations in **XR**, offering familiar metaphors and improving discoverability [16].

Beyond hands, the **HeadGesture** system explores head movement as a modality for interaction when hands are occupied. Through machine learning classification of nods, shakes, or tilts, head gestures can support functions such as scrolling, zooming, or navigation, although they tend to be slower and are less suited for continuous tasks [17].

To optimize gesture-based interactions, models such as **Fitts' Law** provide a predictive framework for target acquisition. Originally devised for 1D tasks, **Fitts'**

Law has been adapted to 2D and 3D interfaces to estimate movement times based on target distance (A) and width (W), supporting empirical evaluation of pointing techniques in immersive systems [18].

2.3.4 Evaluation of Multimodal Interaction in Complex Tasks

In the context of evaluating multimodal interaction, case studies involving large virtual interfaces for tasks like photo management have been conducted [11]. Such studies categorize interaction tasks into **Navigation** (operating view to a target location), **Selection** (interface system controls, e.g., enabling filters or selecting photos), and **Manipulation** (moving photos spatially) [11]. Evaluations comparing multimodal combinations (e.g., **Head+Voice**, **Head+Gesture**, **Gesture+Voice**, **Gesture Only**) reveal insights into efficiency and user preference.

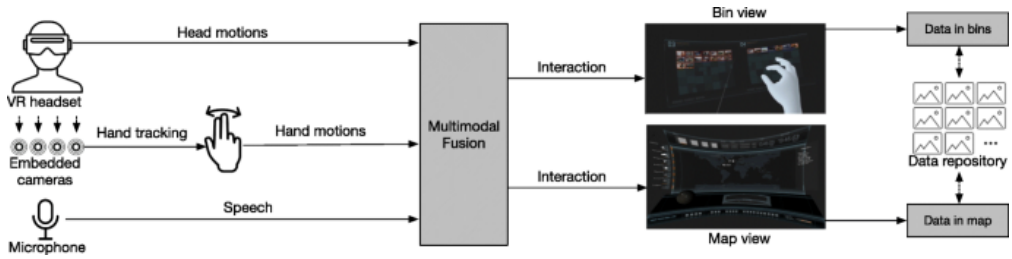


Figure 2.5: Complex Multimodal Fusion System Example [11].

For instance, the combination of **Head+Voice** has shown to be significantly more pragmatically efficient and results in shorter task completion times, especially for manipulation tasks [11]. While quantitative differences in task completion time might not always be statistically significant across all combinations, user preferences observed in qualitative feedback are crucial. For example, users often prefer **head gaze** over **hand gestures** for pointing actions (e.g., Confirm, Zoom-In/Out) and **voice commands** for actions like "Select All," "Move to a Bin," or "Delete," whereas the **"Fist" gesture** might be favored for "Stop Zooming" [11].

These findings underscore the importance of offering multiple interaction modalities for users to choose from, allowing adaptation to task demands and individual preferences, as user prior **VR experience** does not necessarily correlate with adaptation to multimodal interaction [11]. However, **voice recognition** can have a more noticeable latency than **gesture recognition**, making it less suitable for tasks requiring immediate responses, like navigation or stopping a continuous action [11].

2.4 Spatial Computing

Spatial computing is a paradigm that allows digital systems to understand and interact with the physical world in a sophisticated manner, enabling them to process and manipulate real-world objects and spaces as if they were digital assets. This goes beyond traditional computing by integrating 3D data and environmental context into applications, fostering deeper interactions between humans, machines, and the environment. It is the foundation for many **AR** and **MR** applications, and increasingly, **VR** environments that incorporate real-world mapping or persistent virtual spaces.

2.4.1 Productivity Tools

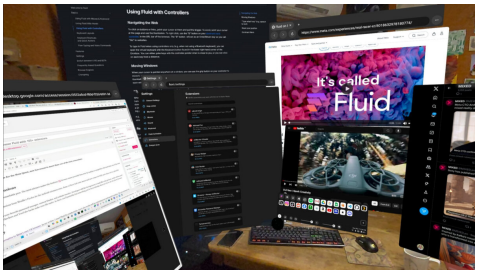
Spatial computing extends the traditional desktop metaphor into three-dimensional space, offering new possibilities for productivity tools. These applications aim to enhance efficiency and collaboration by providing immersive workspaces that leverage the benefits of 3D environments. Examples include:

- **Virtual Desktop Solutions**: Applications like **Virtual Desktop** or **Immersed** allow users to bring their traditional 2D computer screens into a **VR** environment, often providing multiple virtual monitors in an expansive 3D space. This can increase screen real estate, reduce distractions, and create a more focused work environment [19].
- **Multi-Window Environments**: Similar to virtual desktops, these tools provide persistent digital workspaces where users can arrange multiple application windows or panels in 3D space, irrespective of physical screen limitations. This facilitates multitasking and information organization in a spatially intuitive manner.
- **Collaborative Workspaces**: Spatial computing enables shared virtual environments where geographically dispersed teams can collaborate on projects, review documents, and hold meetings as if they were in the same physical room. These tools often integrate features like shared whiteboards, 3D model viewing, and real-time communication. VR can support demands for interaction, collaboration, and knowledge sharing while addressing individual workstyle needs, minimizing distraction, and decreasing crowding. As an adaptable workspace, VR can offer spaces and interior features designed for different work activities, acknowledge cultural diversity, and improve productivity and wellbeing [19].

The potential benefits of these tools include improved focus, enhanced collaboration, and the ability to manage complex information more effectively by leveraging

spatial memory. Immersive VR environments can significantly impact perceived vitality, stress, and mood, unlike simple changes in environmental features like wall colors or room temperature [19].

(a) Virtual Desktop



(b) Meta Multi-Window System

(c) Meta Workroom

Figure 2.6: Productivity XR tools

2.4.2 Creative Tools

Spatial computing profoundly impacts creative workflows by enabling artists, designers, and engineers to create and manipulate digital content directly within three-dimensional space. Unlike traditional 2D design software, spatial creative tools offer intuitive, embodied interaction methods that can accelerate the ideation and prototyping phases. These tools provide a sense of immersion and direct manipulation that can unlock new creative possibilities. Several applications are available within **VR** ecosystems, such as the **Meta Quest Store**.[1] A common characteristic among many of these established tools is their predominant reliance on handheld controllers rather than freehand gestures or direct hand tracking.

[1]Available at: `https://www.meta.com/it-it/experiences/section/643818773459550/`

Overview of Existing Controller-Based Applications

- **Tilt Brush**: Developed by Google, **Tilt Brush** allows users to create expressive 3D paintings in **VR**. Its intuitive interface and brush variety make it ideal for artistic exploration, but it lacks precision tools for modeling or design. Interaction is entirely controller-based.

- **Quill**: A **VR** tool for 3D illustration and animation, ideal for creating expressive storyboards with detailed, painterly strokes. It focuses on artistic content rather than technical design and relies on controllers for input.

- **Gravity Sketch**: A versatile 3D modeling application used in automotive, industrial, and product design. It supports complex shape creation, CAD-compatible exports, and team collaboration. Controllers are essential for detailed modeling.

- **ShapesXR**: A collaborative **VR** tool focused on **UX/UI** design and **XR** prototyping. It allows rapid interface creation, supports real-time teamwork, and integrates with Unity for extended workflows. While not suited for detailed modeling, it streamlines immersive design through controller-based interaction.

(a) Tilt Brush

(b) Quill

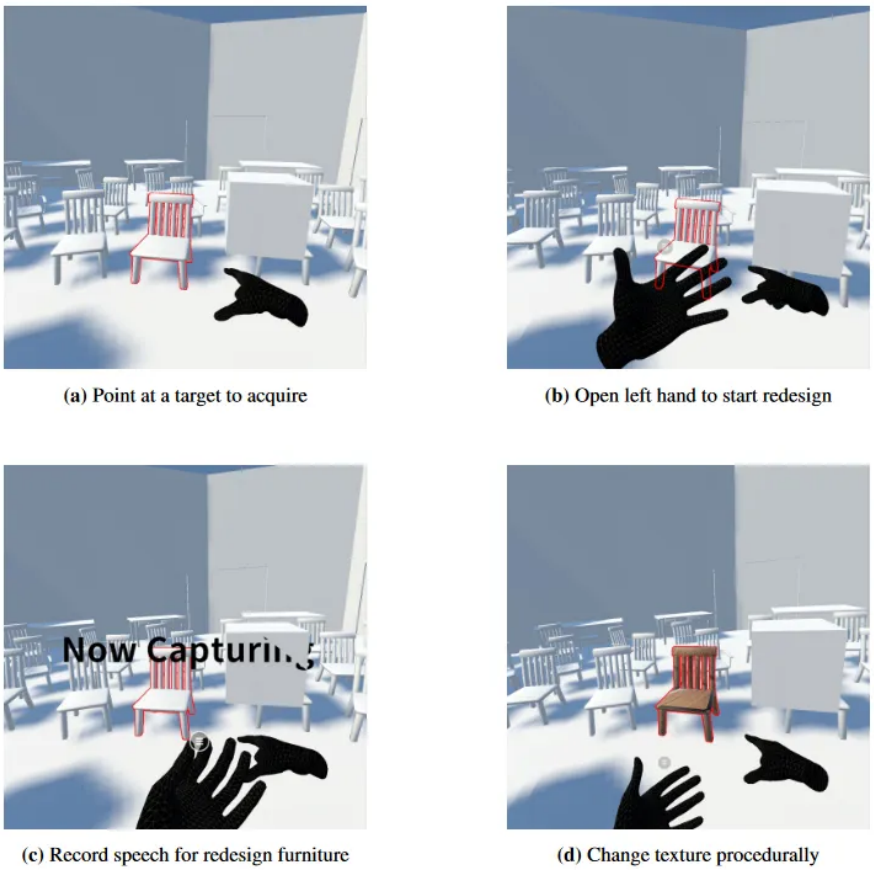
(c) Gravity Sketch

(d) ShapesXR

Figure 2.7: XR creative tools and environments

AI-Enhanced Creative Tools with Multimodal Interaction

Beyond controller-based systems, a new frontier in creative spatial computing involves integrating generative **AI** with natural user interfaces. For instance, recent research explores intuitively redesigning room interiors using gesture, speech, and generative AI [20]. This approach allows users to customize the appearance of a room by capturing desired images through **voice** and **hand gestures**, and then applying **AI-generated textures** in a **VR** environment [20]. Unlike previous **AR**-based interior design methods that mostly focused on furniture relocation and relied on pre-prepared textures, this method enables dynamic texture application based on user communication [20].



(a) Point at a target to acquire

(b) Open left hand to start redesign

(c) Record speech for redesign furniture

(d) Change texture procedurally

Figure 2.8: AI-Enhanced Creative Tools with Multimodal Interaction [20].

2.5 Moodboarding Scenario

Moodboarding stands as a cornerstone in various creative and design disciplines, serving as a **critical tool for visual exploration and conceptual communication**. At its core, a **moodboard** is a compilation of diverse visual elements—images, colors, textures, and material samples—meticulously assembled to articulate emotions, feelings, or a specific "mood" derived from an initial design brief [21]. These inherently visual collections transcend linguistic barriers, effectively conveying abstract concepts [21].

Moodboards not only serve to **visually synthesize** key elements of a project, but also act as a medium to express a **designer's vision**—understood as the imaginative ability to envision future scenarios and alternative lifestyles through visual thinking [22]. This makes visualization a fundamental step in shaping innovative concepts that go beyond mere form or function, embodying values and long-term perspectives.

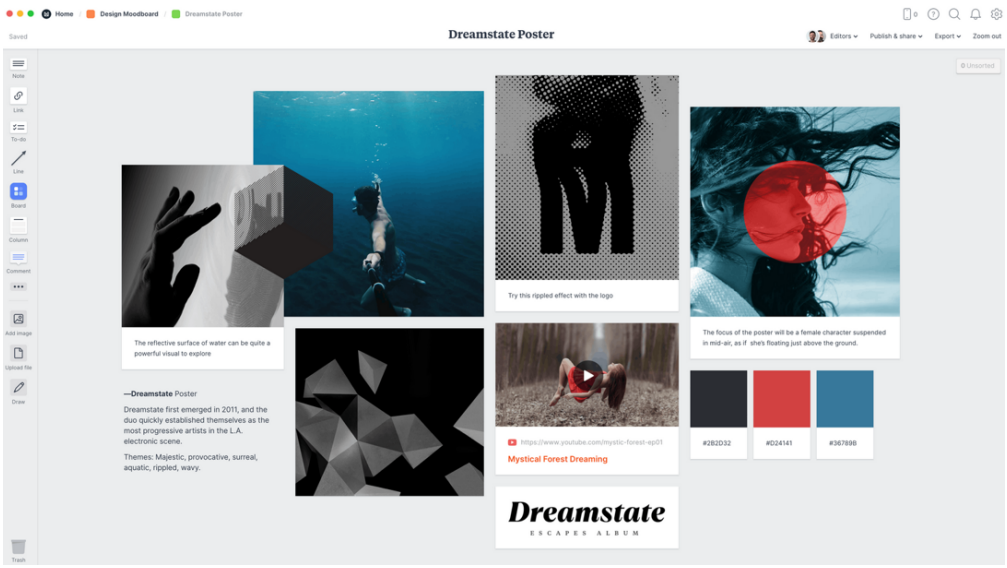


Figure 2.9: Moodboard Example Realized on Milanote.

2.5.1 Design Thinking Foundations

Within the broader context of **design thinking**, moodboards play a multifaceted role in both **problem identification and resolution**. They are dynamic instruments for creative problem interpretation and for facilitating the development and resolution of design problems [21]. Creating moodboards often involves reassembling diverse images, fostering an exploratory, experimental phase where designers

connect with the brief and visualize their perceptions [21]. This process enables problem recognition, envisioning future scenarios, and exploring ephemeral design aspects through **colors**, **textures**, **shapes**, and **images**, infused with personal sensibility [21].

In other words moodboards serve several critical roles in the design process [23]:

- **Framing Role:** Defining task boundaries, encompassing both problem setting and solving.
- **Transmission Role:** Conveying mindset or vision, aligning stakeholders.
- **Research Role:** Visually researching conflicting or paradoxical ideas.
- **Abstracting Role:** Juxtaposing concrete and abstract imagery based on project needs.
- **Directing Role:** Establishing a trajectory for future design efforts.

Furthermore, moodboards are pivotal in the **communication** phase, commonly used by designers to explore, discuss, and convey ideas with clients early in a project [24].

2.5.2 Creation Process Overview

The creation of moodboards is an essential step in the design process, serving as an idea-generating tool that can illustrate **abstract concepts** [21]. Designers often source images from **magazines** or **visual media** to communicate insights about the target audience, product, or company [25]. This activity, though potentially quick (one to two weeks), provides crucial direction for subsequent, more time-consuming development stages [26].

What appears as a simple **"collage"** holds deeper meaning, functioning as a method to build **aesthetic discourse**, preserve **brand identity**, or transform existing concepts through new qualities via **assonance** or **contrast** [26]. Organizing visual materials through **graphic mind maps** or **"walls of evidence"** (highlighting **colors**, **shapes**, **outlines**, **grids**, or **composite styles**) further enhances their utility, assigning weight and value to topics, aesthetics, and specific visual languages [26].

2.5.3 Image Research Tools

Efficient image research is crucial for any moodboarding process. Beyond standard **Google Images** or **social media searches**, specialized platforms and practices include:

- **Magazines, Books, and Catalogs:** Traditional print media remain valuable sources for curated visual content, often providing tactile inspiration.
- **Pinterest:** A well-established visual discovery engine that functions inherently as a moodboard-like platform, allowing users to collect and organize inspirations.
- **Stock Image Libraries:** Platforms like **Unsplash** (`https://unsplash.com/`), **Pexels** (`https://www.pexels.com/`), **Moose Photos** (`https://icons8.com/photos`), and **Pixabay** (`https://pixabay.com/`) offer vast collections of high-quality, often free, images. Commercial libraries such as **Shutterstock** (`https://www.shutterstock.com/`) and **Depositphotos** (`https://depositphotos.com/`) provide even more extensive, professionally curated content, typically via paid subscriptions.

2.5.4 Moodboarding Tools

Historically, moodboarding involved physical collages. However, digital tools have significantly evolved the process, offering increased flexibility, organization, and shareability. These tools range from simple image editing software to dedicated platforms facilitating collaborative visual curation.

Digital moodboarding tools broadly categorize by interface design and spatial metaphor: **image research tools**, **open 2D/3D templates**, **grid-based systems**, **aesthetic-focused templates**, **mobile applications**, and **general-purpose design software**.

Open Template Systems (2D & 3D)

These platforms offer expansive, flexible canvases for free element arrangement.

- **Open 2D Templates:** Tools like **Milanote** (`https://milanote.com/product/moodboarding`) and **Miro** (`https://miro.com/it/moodboard/`) provide infinite two-dimensional whiteboards. Users can freely place images, text, links, and other media, making them highly versatile for initial ideation and exploration. Their strength lies in spatial freedom, allowing for organic arrangements.
- **Open 3D Templates:** A newer development, these tools extend the freeform canvas into a three-dimensional viewport. **Mattoboard** (`https://mattoboard.com/`) allows drag-and-drop elements into a 3D space, intuitive for disciplines dealing with 3D objects. **Planner 5D** (`https://planner5d.com/pro/moodboards`) offers similar capabilities, tailored for interior designers to visualize spatial arrangements. These 3D environments provide a more immersive feel and potential for complex spatial organization.

Grid-Based Systems

In contrast to open templates, **grid-based systems** offer a structured approach with fixed grids and flexible compartments for content. While more rigid, this structure aids organization and prevents visual clutter, making them less dispersive. Examples include templates from **Figma** (`https://www.figma.com/templates/moodboard-maker/`) and **StudioBinder** (`https://www.studiobinder.com/templates/mood-boards/`), favored for clean presentation and collaborative structured workflows.

Aesthetic-Focused Templates

These tools prioritize visual refinement with pre-designed layouts and stylistic options, focusing on graphic design aesthetics. Platforms like **Adobe Express** (`https://www.adobe.com/express/create/mood-board`) and **Canva** (`https://www.canva.com/it_it/creare/moodboard/`) offer user-friendly interfaces and rich template libraries, enabling quick creation of polished moodboards for presentation or social media.

Mobile Applications

Mobile devices have spurred dedicated moodboarding apps, bringing creativity on-the-go. **Shuffles by Pinterest** (`https://www.shffls.com/it`) (2022) enables visual collages from Pinterest pins, blending decoupage with social scrapbooking, often shared on platforms like TikTok. **Morpholio Board** (`https://apps.apple.com/it/app/morpholio-board-moodboard/id761867957`) is a powerful iPad app offering features reminiscent of desktop design software like Photoshop, tailored for mobile designers.

General-Purpose Design Software

Professional designers often utilize powerful, multi-purpose software adaptable for moodboarding. Adobe products like **Photoshop**, **Illustrator**, and **InDesign** provide advanced tools for image manipulation, layering, and precise layout, enabling highly customized and sophisticated moodboards. Even **PowerPoint** can serve as a simple canvas. These tools offer unparalleled flexibility for experienced users, including features like color overlays and blending modes for visual harmony.

2.5.5 Case Study: Funky System

Early studies from 2007 and 2008 explored how digital technologies—and what were then considered immersive setups, such as large interactive displays—could

make the moodboarding process more engaging and expressive. An illustrative example of such efforts is the **Funky System** [25, 24]. Described in papers like *"Funky-Design-Spaces: Interactive Environments for Creativity Inspired by Observing Designers Making Mood Boards"* [25] and *"Funky Wall: Presenting Mood Boards Using Gesture, Speech and Visuals"* [24], this system explored gesture, speech, and visuals to enhance both the creation and presentation of moodboards. The **Funky Wall** project specifically focused on the presentation phase, utilizing a large projection surface and multimodal interaction (gestures, voice commands) to allow designers to contextualize and elaborate on visual narratives for clients. By moving beyond static presentations, the system offered a more dynamic and engaging medium for creative communication.

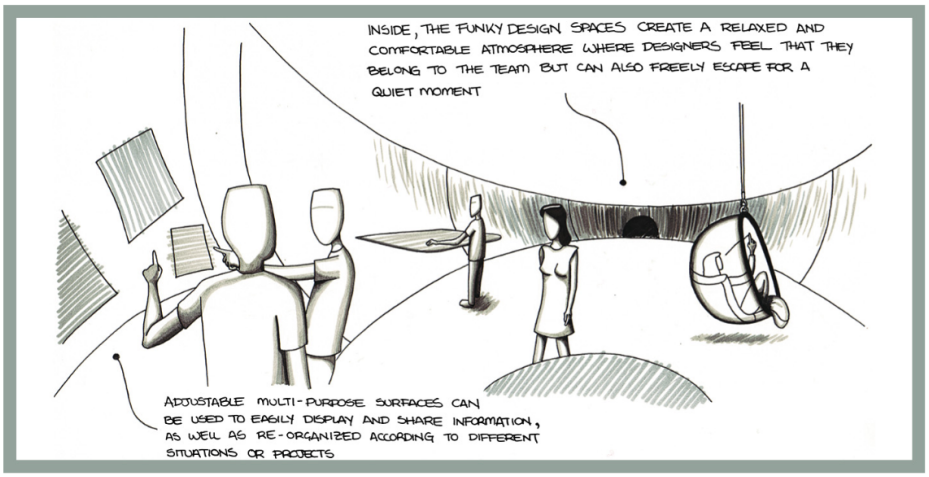


Figure 2.10: Funky Wall [24].

2.5.6 VR Moodboarding

Virtual Reality (VR) offers a uniquely powerful medium for moodboarding, enabling a deeper level of emotional engagement through immersive and multisensory experiences. Research shows that high emotional involvement occurs when individuals are fully immersed in an experience [27], making VR environments valuable not only in entertainment but also in psychology and design research. These immersive settings create a heightened sense of presence, which can significantly enhance the designer's cognitive and emotional connection to their work.

One of the key advantages of using VR for moodboarding is the ability to **spatialize information**. Unlike traditional two-dimensional compositions, VR introduces a third dimension, allowing for the creation of a **three-dimensional trend environment** [28]. This spatialization actively stimulates and engages

designers, encouraging them to interact with and explore the immersive moodboard. By providing the illusion of a potential reality, VR systems offer a powerful foundation for reflection and ideation. Research also suggests that early exposure to immersive environments can significantly support industrial designers in making style-related decisions [28].

2.6 AI-Powered Image Generation

AI-powered image generation represents a significant leap in visual design, particularly within **spatial computing** contexts like **Virtual Reality (VR) moodboarding**. This technology moves beyond traditional image collection, enabling designers to actively generate custom content.

2.6.1 Understanding Diffusion Models

The core of modern AI image generation, particularly for tools like **Stable Diffusion** and **Midjourney**, lies in **diffusion models**. These generative models synthesize images by progressively denoising a random signal until it matches a target visual, translating complex textual descriptions into nuanced visual content through an iterative process. Unlike **Generative Adversarial Networks (GANs)**, diffusion models offer enhanced control over the generation process and often produce higher-quality, more diverse outputs. The process typically involves a sequence of denoising steps, guided by a **text prompt**, where the model refines the image based on the provided input and learned aesthetic patterns. This iterative refinement allows for a nuanced translation of textual descriptions into visual content, addressing complex stylistic and compositional requirements.

2.6.2 Principles of Prompt Engineering

While AI offers immense creative potential, generating high-quality images free from artifacts like the **"uncanny valley"** effect—where outputs appear identifiable yet unnaturally flawed [29]—requires careful **prompt engineering**. This involves crafting effective instructions for AI models by understanding how different linguistic components influence image creation.

Prompt Structure and Order

A highly effective prompt structure often follows: **Subject + Adjectives + Style + Composition + Materials + Lighting + Quality**. Crucially, the **order of elements** within a prompt is vital, as many models process prompts sequentially, giving more weight to information presented earlier. For instance,

specific photographic details like **"moodboard, objects from countryside + cinematic shot + photo taken by ARRI + incredibly detailed + professional lighting, + 50mm + Lightroom gallery"** yield more realistic and usable results. Color-based prompts have also proven effective [29].

The iterative nature of prompt engineering, often involving trial and error, highlights the need for intuitive user interfaces that allow designers to easily manipulate these parameters. Testing various prompt lengths is also important, as conciseness is generally preferred, but optimal length can vary by model and desired complexity.

2.6.3 AI-Powered Image Generation Services Comparison

Understanding current industry practices for AI-driven interfaces provides valuable insights for designing intuitive VR interactions. The landscape of AI image generation services is rapidly evolving, offering various platforms with distinct features and target audiences. These services can be broadly categorized by their accessibility and control levels:

- **Adobe Firefly:** Features a modular, step-by-step GUI with visual prompting and tight integration with the Adobe suite. It emphasizes **"prompt stacking,"** organizing parameters hierarchically (e.g., General Settings → Content & Composition → Style → Effects → Color & Tone → Lighting → Camera Angle). Firefly also provides visual references for composition and style.
- **DALL·E 3 (OpenAI):** Known for its strong understanding of natural language, DALL·E 3 integrates well into conversational interfaces. It offers a more narrative and iterative prompting logic, focusing on resolution, vividness, and quality.
- **Leonardo.ai:** Caters to indie creatives, offering a mix of text prompting and modular settings for texture, lighting, and concept art. It provides a **"flow state"** experience through visual references and pre-defined styles.
- **Stable Diffusion (DreamStudio):** Favored by professionals and enthusiasts, its technical interface offers extensive customization for parameters like CFG Scale, Steps, and Sampler. Its strength lies in high control and flexibility, supporting Image-to-Image generation for reverse engineering and stylistic transfer.
- **Midjourney (v6):** Primarily community-driven via Discord, it excels at generating aesthetically pleasing images with minimal effort. Notably, Midjourney has evolved to accept moodboards as input, influencing the stylistic consistency of subsequent generations, thereby allowing visual input to guide AI.

2.6.4 AI-Driven Moodboarding

The advent of **Artificial Intelligence (AI)**, particularly generative models, introduces a new paradigm for moodboarding. Traditionally, creating moodboards required significant human effort, involving physical drawing, photography, or digital editing [23, 29]. These processes demanded specific graphical and visual skills central to many design fields [29]. However, novel AI tools, especially those leveraging **Generative Adversarial Networks (GANs)** based on deep neural networks, now enable creative visualizations without extensive graphical expertise [29].

Studies exploring major public AI image generators—such as Midjourney—have investigated their generative mechanisms and assessed the design usefulness of their outputs [29]. The integration of AI fundamentally transforms image acquisition and ideation by allowing rapid iteration of visual concepts beyond the scope of traditional image libraries. This accelerates early creative phases and redefines moodboarding from simple collection and arrangement to active content generation. AI can synthesize complex stylistic preferences, translate abstract ideas into concrete visuals, and generate entire moodboards aligned with specific thematic or emotional goals.

Figure 2.11: Example of Moodboard Realized with GAN-based AI [29].

2.6.5 Ethical and Copyright Considerations

While AI-powered image creation offers unprecedented creative opportunities, it simultaneously raises substantial ethical and copyright challenges. Although AI

systems are frequently trained on copyrighted images within the bounds of current legal frameworks, the legitimacy of utilizing AI-generated outputs for commercial purposes remains ambiguous, potentially exposing users to infringement risks. Notable controversies—such as the inadvertent appearance of Getty watermarks in images generated by Midjourney—highlight the complexity of these legal and ethical issues [29].

In addition to copyright and artistic labor concerns, AI-generated content also poses broader ethical questions related to fairness, bias, and privacy, which are critical to consider:

- **Fairness and Bias:** Generative AI models, commonly trained on large-scale datasets harvested from the web, can perpetuate and even amplify existing biases. For instance, Stable Diffusion has exhibited biased outputs under certain conditions. Current research endeavors aim to mitigate these biases through techniques including data pre-processing, incorporating fairness constraints during model training, and applying corrective post-processing to the generated content.
- **Privacy:** The extensive datasets used to train text-to-image algorithms may inadvertently include sensitive or private information. **Membership inference attacks** explore vulnerabilities by attempting to ascertain whether specific data points were part of a model's training set, thereby raising concerns about data privacy and protection.

Chapter 3

Background

3.1 Unity for VR Development

Unity was chosen as the primary development platform for this project because of its strong reputation as a versatile and widely adopted engine for virtual reality (VR) development. One major reason for this choice is Unity's relatively **lightweight architecture** and **lower resource demands**, which are essential for achieving optimal performance on VR devices giving it an edge over more resource-intensive engines like Unreal Engine. Additionally, Unity's **extensive ecosystem**, thorough **documentation, and active community support** made it a clear choice for the project.

3.1.1 OpenXR Integration

OpenXR is an open, royalty-free **standard** developed by the Khronos Group that provides a unified API for high-performance access to virtual reality (VR) and augmented reality (AR) platforms and devices. Its integration into this Unity project is a strategic decision to ensure broad hardware compatibility and future-proof the application.

Key benefits of adopting OpenXR include:

- **Cross-Platform Compatibility**: OpenXR allows developers to write code once and deploy it across a wide range of VR and AR devices, eliminating the need to adapt applications to proprietary APIs from different hardware vendors (e.g., Meta, Valve, Microsoft).
- **Reduced Fragmentation**: By providing a common standard, OpenXR helps to reduce fragmentation within the XR ecosystem, simplifying the development process and allowing for greater focus on application features rather than platform-specific adaptations.

- **Avoidance of Vendor Lock-in**: Leveraging OpenXR minimizes reliance on any single hardware manufacturer's SDK, ensuring the project remains flexible and can easily support new or alternative devices as the market evolves.

- **Optimized Performance**: The standard is designed for high performance, allowing direct access to underlying hardware capabilities, which is essential for maintaining high frame rates and a smooth user experience in VR.

In Unity, OpenXR integration is managed through the XR Plugin Management system, which allows for straightforward configuration and runtime switching between different OpenXR runtimes and device setups.

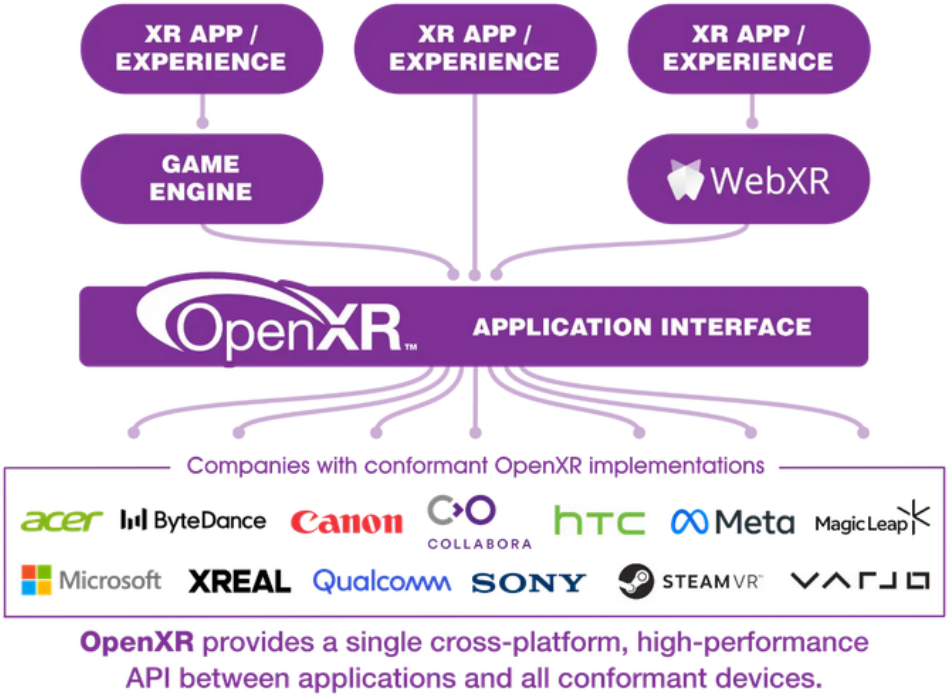


Figure 3.1: Open XR Overview

3.1.2 MRTK3: A Comprehensive Framework for Mixed Reality UI

MRTK3 is an open-source development kit from Microsoft designed to accelerate MR application development in Unity. It's built on top of Unity's XR Interaction Toolkit (XRI) and OpenXR, ensuring a modern and extensible foundation.

Key advantages of MRTK3 include:

- **Advanced Interaction Models**: MRTK3 provides a rich set of interactions tailored for mixed reality, such as direct manipulation, gaze-based interaction, and sophisticated hand tracking. Components like **ObjectManipulator** allow for intuitive 3D object manipulation, offering more specialized functionalities compared to XRI's `XRGrabInteractable`. It also includes features for managing **Bounds Control** and **Constraint Manager** to limit object manipulation (e.g., restricting rotation to a single axis).

- **Comprehensive UI Components**: MRTK3 offers a wide array of ready-to-use UI components, from standard buttons (**Canvas Button**) to specialized elements like **Slates** for displaying content and **Hand Menus** that appear with specific hand gestures. These components are designed to be highly customizable and responsive, leveraging UnityUI's AutoLayout groups for flexible and adaptive layouts across various physical contexts.

- **Input Simulation**: For efficient development and testing without a physical device, MRTK3 provides a robust **XR Simulator**, enabling developers to simulate hand movements and interactions directly within the Unity editor. This includes controls like 'Y' and 'T' for hand visibility and 'Space' and 'Left Shift' for hand selection.

- **Performance Optimization**: MRTK3's approach to UI design emphasizes performance. It recommends using multiple smaller Canvas elements for dynamic parts of the UI rather than a single large Canvas to minimize CPU overhead. The framework also incorporates advanced shader capabilities from the Mixed Reality Graphics Tools package to enable sophisticated visual effects without significant performance impact.

Figure 3.2: MRTK3 Overview

3.1.3 Comparison with Alternative UI Systems

The development of intuitive and robust **user interfaces (UI)** is paramount for creating compelling **mixed reality (MR)** experiences. While exploring various UI solutions for Unity, including **native Unity UI**, **Unity's UI Toolkit**, and **Meta's Interaction SDK (ISDK) UI Set**, **Mixed Reality Toolkit 3 (MRTK3)** emerged as the most suitable framework for this project due to its comprehensive features, cross-platform compatibility, and strong integration with Unity's **XR Interaction Toolkit (XRI)**.

During the research phase, other UI systems were considered:

Unity UI (UGUI)

Unity's built-in **UI system (UGUI)** is widely used and artist-friendly, facilitating rapid prototyping. It operates via the **Unity Event System**, where UI elements like buttons have direct callback events. While effective for traditional 2D UIs, UGUI's performance can degrade with complex hierarchies, as updating a single element might trigger a refresh of the entire Canvas. For VR, Unity UI requires careful optimization, such as dividing UI elements across multiple smaller Canvases to manage refresh rates. However, its native **World Space Canvas** allows for rendering UI elements within a 3D environment.

Unity UI Toolkit

Unity's **UI Toolkit** represents a more modern approach, separating UI design from functional implementation, akin to web development with **HTML**, **CSS**, and themes. It employs a comprehensive **Event System** where scripts register callbacks to visual elements. UI Toolkit also features a flexbox implementation called **Yoga**, enabling responsive layouts. While offering a more robust and scalable solution for complex UIs, its native support for **"World Space UI"** in VR environments was not fully mature at the time of evaluation (early 2025), with community discussions indicating potential bugs and workarounds needed for seamless XR integration.

Meta Interaction SDK (ISDK) UI Set

Meta provides its own **Interaction SDK UI Set** for Unity, primarily targeting Meta's Horizon OS. This library offers a rich set of pre-built UI components and direct access to **Figma design files**, which is appealing for designers. While visually comprehensive and integrated with Meta's ecosystem, its primary focus on Meta's specific hardware and software stack made it less ideal for a solution aiming for broader **OpenXR compatibility** and potential future deployment across various mixed reality headsets beyond Meta's ecosystem.

Conclusion on UI Choice

Ultimately, **MRTK3** was chosen for its comprehensive support for **mixed reality interactions**, robust input simulation, performance-conscious design principles, and its foundation on **OpenXR** and Unity's **XR Interaction Toolkit**, which aligns with the goal of creating a versatile and high-performance VR moodboarding application. The ability to integrate with Unity UI's Canvas system also provided flexibility for various UI paradigms, including the creation of floating, constrained windows for interactive elements.

Additionally, **Unity UI** was utilized to develop a basic desktop interface for testing purposes, allowing users to select profiles and input modalities before entering the VR environment.

3.1.4 Microsoft Azure for Speech Recognition

Microsoft Azure Speech Services provide a scalable and production-ready cloud solution for integrating **voice recognition** into interactive applications. Designed to support **multimodal and intelligent systems**, Azure's speech capabilities are particularly well-suited for immersive environments such as VR.

Azure's **Speech Service** offers robust and flexible features, allowing developers to transcribe user voice input into text for various applications, including **command recognition** and **natural language processing** within immersive environments.

The integration process involves creating a **Speech Service resource** in the Azure portal and then configuring the Unity project to use the **Azure Speech SDK**.

For optimal performance and responsiveness in a real-time VR application, it is vital to handle speech recognition asynchronously. The **Azure Speech SDK** facilitates this by offloading the recognition process to separate threads, preventing the main Unity thread from being blocked. This ensures a smooth user experience, even during potentially time-consuming operations like audio recording, transmission, and transcription. Proper resource management, such as disposing of the speech recognizer after use, is also critical to prevent memory leaks and ensure efficient application performance.

While this study primarily focuses on **speech-to-text**, **Azure AI services** offer additional functionalities like **Text Analytics**, which can analyze sentiment and extract key phrases from transcribed text. This capability, though not implemented in the current prototype, represents a **future integration possibility** or a **recommendation for subsequent development**. Leveraging **Text Analytics** could further enhance a moodboarding tool by allowing the system to derive deeper meaning from user vocal inputs, potentially influencing **AI-generated image suggestions** or refining search queries based on inferred emotional tones or critical concepts.

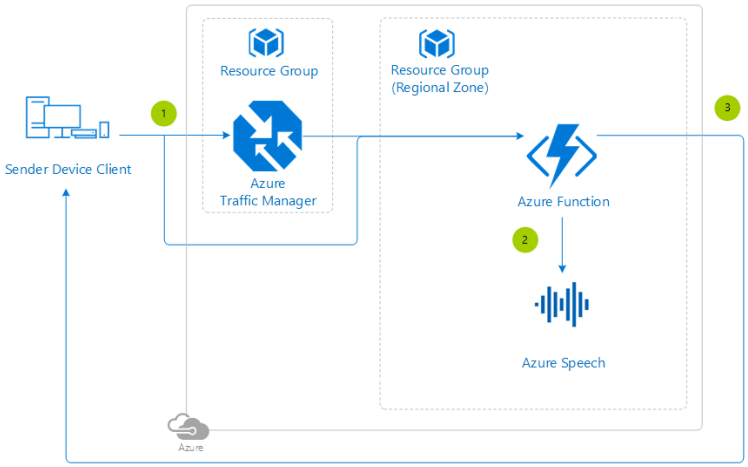


Figure 3.3: Azure Speech To Text Overview

3.2 AI-Based Image Creation

The project utilizes **Diffusion** models for image generation due to their balance of quality, flexibility, and community support. As of mid-2024, **SDXL 1.0** is widely considered a performant option for detailed and coherent outputs. For faster results, particularly during iterative creative workflows in VR, models like **FLUX.1 schnell**—available on **Hugging Face**—are preferred for their optimization and reduced latency.

Figure 3.4: Flux.1 Example

Each of these models supports a range of configurable parameters that can be fine-tuned to influence image fidelity, coherence, and stylistic qualities. [30]

3.2.1 Key Generation Parameters

Several generation parameters significantly influence the final output, offering nuanced control over fidelity, creativity, and visual clarity [30]:

- **CFG Scale (Classifier-Free Guidance):** Determines the strength of adherence to the prompt. Higher values enforce strict conformity, ideal for precise visual targets, while lower values promote creativity.
- **Steps:** Represents the number of inference iterations. More steps generally yield higher quality but increase generation time. A standard range (e.g., 30–50 steps) balances efficiency and detail.
- **Seed:** Initializes the random noise input. Consistent seeds allow reproducibility, critical for refining outputs over iterations.
- **Scheduler:** Controls how noise is removed during inference. It defines the denoising trajectory and influences generation speed, sharpness, and overall visual characteristics.
- **Negative Prompt:** Filters undesired features. For VR-specific use cases, terms like `blurry, small features, excessive texture` are used to improve readability and mitigate aliasing artifacts in **Head-Mounted Displays (HMDs)**.
- **Weighted Tagging:** Some models support weights for prompt components (e.g., `subject:(1.2)`), enabling precise emphasis on specific elements of the prompt.

3.2.2 Scheduler Selection

In the context of moodboard generation, different schedulers were evaluated for their alignment with various creative goals:

- **Euler / DDIM:** Recommended for rapid prototyping and conceptual sketching. These schedulers are fast and provide reasonably sharp results with minimal inference time.
- **DPM-Solver:** Suitable for high-fidelity results with strong structural coherence. It is especially effective when realism or architectural consistency is desired in the generated output.
- **Euler Ancestral:** Ideal for more abstract or stylized compositions, offering slightly more creative variation and artistic noise—valuable in exploratory visual tasks like moodboarding.

3.2.3 API Integration and Implementation

To support dynamic and user-driven image generation, the system integrates Python-based APIs that interface with leading generative AI providers. The generated outputs are subsequently stored locally and loaded as textures within the **Unity** environment.

Hugging Face API

The **Hugging Face Inference API**—notably the `black-forest-labs/FLUX.1-dev` and `stable-diffusion-xl-base-1.0` models—was initially adopted due to its configurability. It supports parameter control for `CFG scale`, `steps`, `sampler`, `seed`, and `negative prompt`, offering granular tuning to suit diverse design requirements. This flexibility proved especially useful in allowing real-time customization based on user preferences in VR.

NScale API

Due to recent restrictions in **Hugging Face's free-tier usage policies**, the system transitioned to **NScale's API** (accessed via the **OpenAI client**) using the `FLUX.1-schnell` model. While it offers high-quality generation and generous free usage, it limits direct access to certain parameters (e.g., seed, steps, guidance scale), instead relying on internally optimized defaults. This abstraction reduces fine-tuning capabilities but offers consistent performance and accessibility during development.

```python
# Set up OpenAI client for NScale
nscale_api_key = Api_Token.NSCALE_API_KEY
nscale_base_url = "https://inference.api.nscale.com/v1"

client = openai.OpenAI(
    api_key=nscale_api_key,
    base_url=nscale_base_url
)

try:
    # Generate image
    response = client.images.generate(
        model="black-forest-labs/FLUX.1-schnell",
        prompt=full_prompt,
        size=f"{width}x{height}",
        n=1,
    )

    # Decode base64 image
    image_base64 = response.data[0].b64_json
    image_data = base64.b64decode(image_base64)

    # Save image
    filename = f"image_{datetime.now():%m_%d_%H_%M_%S}.png"
    file_path = os.path.join(directory_path, filename)
    with open(file_path, "wb") as f:
        f.write(image_data)

    print(file_path)
```

Figure 3.5: NScale API Request

3.2.4 Multimodal Prompt Construction in VR

The generation pipeline is designed around multimodal input channels to reflect the core principles of Natural User Interfaces (NUIs). The system synthesizes voice input, GUI selections, and contextual project information into a cohesive prompt:

- **Vocal Input:** Captures the subject or central concept of the image.
- **Project Brief:** Provides soft environmental influence, giving thematic direction without overshadowing the main subject.
- **GUI Parameters:** Define visual styling—composition, lighting, color palette, mood, camera angle.

Prompt Prioritization and Structuring

To preserve **user agency and creativity**, the vocal input is placed at the beginning of the prompt to assign it the highest conceptual weight. Contextual elements and stylistic parameters follow, acting as modifiers rather than dominant themes. This hierarchy ensures that the generated content reflects the user's intention while benefiting from aesthetic guidance.

Contextual Influence Strategy

Rather than imposing the project brief as a rigid design requirement, it is framed as an **environmental backdrop**—shaping tone and atmosphere without dictating specific content. For example:

```
f"{subject}, in a scene reflecting {project brief} as ambient
context, {lighting}, {mood}."
```

This approach supports creativity while maintaining alignment with the broader project direction.

Mitigating Prompt Ambiguity in Image Generation

We acknowledge that current generative AI models do not always interpret prompts with full semantic precision, especially in creative or abstract contexts. To address this limitation, the system generates multiple images—typically three—for each prompt submission. This practice increases the likelihood of producing a satisfactory result that aligns with the user's intent. Such **redundancy** is common among AI-powered image generation systems and reflects a shared strategy to compensate for the probabilistic nature of these models.

3.3 UX and Interaction Design

User Experience (UX) and **Interaction Design** are critical pillars in the development of immersive Virtual Reality (VR) applications. Unlike traditional 2D interfaces, VR demands a rethinking of fundamental **design principles** to leverage the three-dimensional space, natural input modalities, and the inherent sense of presence. This section delves into key considerations and strategies employed in designing the user experience and interactions for the VR moodboarding tool, drawing insights from established design guidelines [31, 32] for XR applications [33], and the latest research on multimodal interaction and selection techniques [11, 34].

To frame this challenge within a broader historical context, it is useful to consider how previous paradigms have structured interaction design. Just as command-line interfaces defined 1D interaction models and the **WIMP** (Windows, Icons, Menus, Pointer) paradigm enabled the rise of 2D graphical user interfaces, immersive environments now require the development of new spatial metaphors and interaction standards suited to 3D. This project explores also how familiar 2D interface patterns can be meaningfully "extruded" into three-dimensional space—leveraging depth, **embodied interaction**, and spatial reasoning—while still retaining the usability lessons and mental models shaped by decades of traditional UI design.

3.3.1 Play Area and User Posture

Acknowledging the varying "Play Area" scenarios (Seated, Standing, Room-Scale) as outlined in existing research [35]. For a moodboarding tool, a seated or standing experience is most common for prolonged use. This influenced ui and gesture design to be less physically demanding, allowing users to perform actions comfortably while seated at a desk or standing in a fixed position, promoting sustained engagement. This adheres to the guideline to **"Prioritize User's Comfort"** and **"Be Mindful of Physically Draining Interactions"** [33].

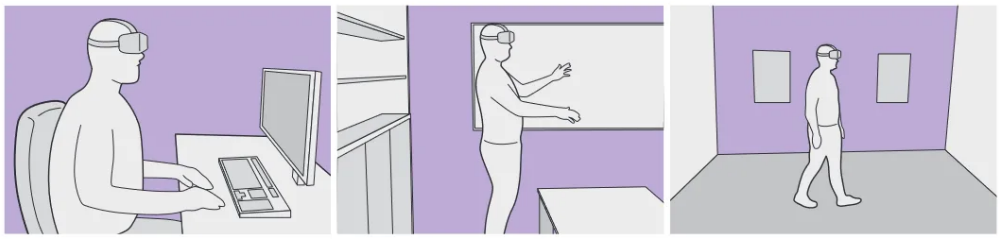


Figure 3.6: Different User Play Area Configurations

3.3.2 Applying Gestalt Principles to 3D UX

Structuring information effectively in a **3D spatial environment** is a fundamental challenge in VR UX design. Unlike flat 2D screens, spatial interfaces offer an expansive canvas, but without thoughtful organization, users can quickly become disoriented or overwhelmed. The design of the VR moodboarding tool integrated spatial organization strategies and drew on insights from existing VR applications to inform layout and interface structure. The overarching principle was to "Use space as an organizational tool" to reduce cognitive load and enhance discoverability [33].

To further support this spatial organization, perceptual principles from cognitive psychology were incorporated to shape how users intuitively interpret and navigate the environment. Gestalt principles, initially formulated for 2D visual perception, are equally, if not more, relevant in the design of 3D user experiences. These principles describe how humans perceive order in visual chaos and group elements, providing a powerful framework for structuring information and interactions in spatial interfaces. The application of these principles is crucial for creating a coherent and intuitive spatial organization, aligning with the guideline to **"Use space as an organizational tool"** to minimize conscious thinking [33].

- **Proximity:** Objects close to each other are perceived as a group [33]. In 3D UX, this means spatially grouping related UI elements or AI-generated images on the moodboard can intuitively convey their relationship, for example, placing images generated from the same prompt near each other. This aligns with the guideline to **"Group Similar Objects to Make Them Easier to Find"** [33].
- **Similarity:** Elements sharing visual characteristics (color, shape, size) are perceived as related. Using consistent styling for interactive elements or images generated with similar prompts helps users understand their commonality.
- **Continuity:** Users tend to follow lines and curves, creating a flow in perception. This can be applied to guide the user's gaze through a spatial layout or direct attention to specific interaction points.
- **Closure:** The human brain tends to perceive incomplete shapes as complete. In 3D, this can mean implying boundaries or relationships even when elements are not explicitly enclosed, allowing for a cleaner aesthetic.
- **Figure-Ground:** The ability to distinguish between foreground (figure) and background (ground) is critical. In VR, careful use of depth, lighting, and visual prominence ensures that interactive elements stand out from the ambient environment or static content. This supports the guideline to **"Keep Visual and Physical Restrictions in Mind When Arranging Content"** for comfortable viewing [33].
- **Common Region:** Elements located within the same bounded area are perceived as grouped. This is particularly useful in 3D to create virtual containers

or panels that clearly delineate functional zones within the immersive space, such as a dedicated area for prompt input or asset management.

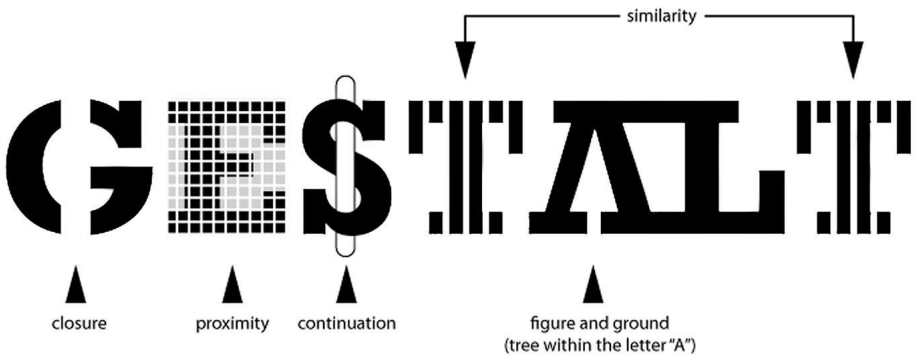


Figure 3.7: Gestalt Principles

Applying these principles to 3D UX helps in organizing the vast spatial canvas, **reducing cognitive load**, and enhancing the discoverability and usability of interactive elements. In this project, Gestalt principles informed both the design of standard 2D UI components and their spatial reinterpretation around the user in VR. They were instrumental in defining the logic behind **prefab construction** for images and boards—ensuring **consistent spacing**, **alignment**, and **grouping behaviors**. This design strategy supports not only functional clarity but also aligns conceptually with moodboarding itself, which is inherently a spatial practice.

3.3.3 Skeuomorphic Interfaces in VR

Skeuomorphism, the design concept of making digital interfaces resemble their **real-world** counterparts, plays a significant role in VR. In a new and often unfamiliar medium like VR, skeuomorphic design can help users quickly understand and interact with virtual objects and functionalities by drawing upon their existing mental **models** from the physical world [33, 36]. This approach is effective because our brains are accustomed to deciphering the **affordances** and **signifiers** of the physical world, where physical laws establish a "mechanic of operation." This forms the basis of our mental models.

Figure 3.8: Job Simulator - Example Of Skeuomorphic Interface

However, VR presents a unique "breaking point": **the absence of mechanical limits** means the physics of the virtual world do not necessarily coincide with those of the real world. While this offers immense creative potential, it can also lead to user disorientation and even motion sickness. For example, if a user expects to grab a virtual frying pan and it clips through the stovetop, or they try to grasp an object and their hand passes through it, this breaks expectations and causes frustration [36]. Therefore, efficient design in VR must be "good communication" [36], where every element acts as an "expression" or statement.

In the context of the VR moodboarding tool, skeuomorphism manifests in ways such as a virtual corkboard, evoking the physical act of arranging ideas. Yet, the design consciously avoids blindly replicating physical limitations. For instance, while a virtual "door" acts as a signifier for navigation, its underlying "mechanics" can allow for immediate teleportation, which is impossible in the real world. The aim is to leverage familiarity to onboard users, bridging the gap between two worlds, while progressively introducing **novel interactions** that capitalize on VR's unique capabilities, moving beyond mere replication towards enhanced functionality. This means the virtual environment is designed to be **intuitive** without being cumbersome, providing tools like a menu on the hand that is accessible on command but never invasive or flow-breaking [33].

3.3.4 Hand Tracking and Object Manipulation

A core design principle adopted in this project is **"Hand Tracking First"** —prioritizing direct, tactile interaction with virtual objects over reliance on hidden or abstract UI elements, which tend to be less discoverable and less intuitive in VR. This approach aligns with findings that users naturally prefer hand gestures for interaction due to their immediacy and familiarity [11].

The application was explicitly designed to be used **without VR controllers**,

relying solely on bare-hand input as detected by the built-in hand tracking systems of modern headsets. While many VR devices ship with physical controllers, this project intentionally excludes them to focus on **natural interaction paradigms**. This constraint reflects the goal of exploring more accessible and embodied interaction strategies that do not depend on handheld hardware.

To support experimentation and evaluation, a simple cube was used as the baseline object for manipulation tasks. As suggested in prior research [34], the cube's symmetrical shape offers clear visual cues for orientation and minimizes the need for object-specific grips, making it ideal for testing rotation, translation, and grasping behaviors.

Within the application, users interact with **multiple moodboards** positioned around them by grabbing dedicated handles, allowing for intuitive spatial arrangement. Images function like virtual **Post-it notes**—they can be detached from boards and repositioned freely within the workspace. To delete an image, users simply throw it to the floor, a metaphorical action grounded in familiar real-world behavior.

3.3.5 Interaction Strategy for UI Components

The VR moodboarding tool incorporated a variety of **UI components**, each designed with a specific interaction strategy to support the multimodal nature of the application. The overall approach aimed to balance the **naturalness** of gestures with the **precision** and reliability of traditional virtual button interfaces, facilitating a hybrid interaction model.

- **General UI Navigation and Content Manipulation (Remote Pinch):** Users interact with menus and moodboard content using a **remote pinch** gesture, which activates a distant pointer for selecting, moving, and scaling elements. This method enables efficient interaction across large virtual areas without requiring physical movement, reducing fatigue and supporting fluid exploration. It mirrors the "point and pinch" paradigm found in apps like Meta's browser and YouTube VR, and aligns with the goal of **prioritizing user comfort** [33].

Figure 3.9: Remote Pinch Interaction.

- **Alerts and Prompting Details (Direct Touch):** Critical interactions—such as confirming actions or accessing AI prompting details—were handled through **direct touch**, allowing users to press virtual elements with their hands. This method provides immediate, tactile feedback, enhancing presence and clarity. It's well-suited for small, focused UI elements and mirrors the interaction style used in the Meta Quest interface for close-range windows, supporting the principle of **"allowing users to feel in control"** [33].

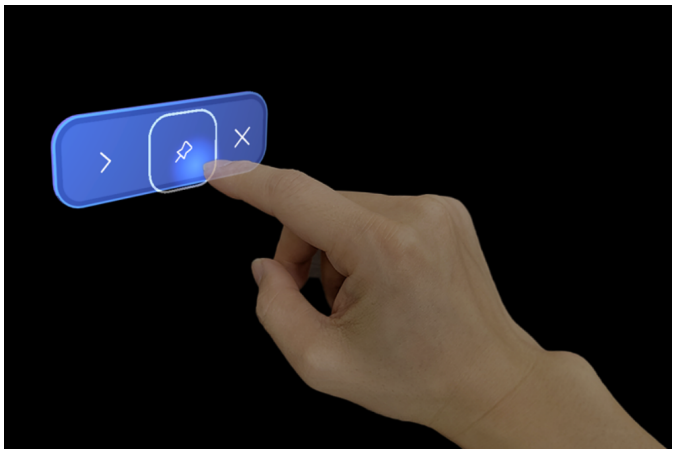
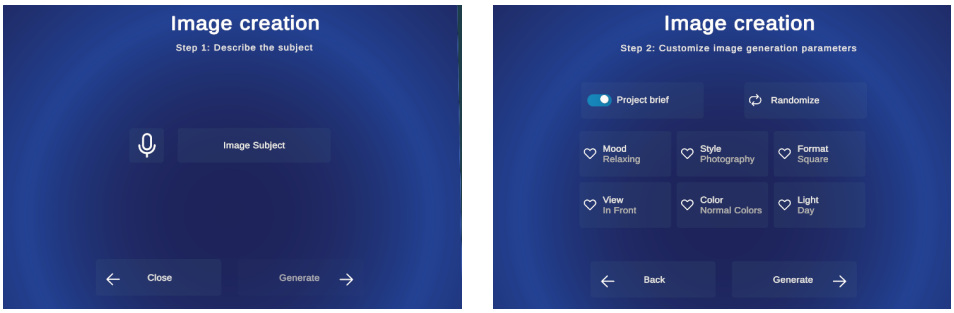


Figure 3.10: Direct Touch Interaction [37]

- **Modular and Contextual UI Panels:** To reduce visual clutter and cognitive load, the interface uses modular and contextual panels that dynamically update existing visual spaces rather than creating new canvases. For example, the primary input canvas adapts to show or hide **voice controls** and **image previews** as needed. Detailed **prompting information** and controls appear only on demand via a dedicated "details" button. This approach keeps relevant information accessible while minimizing distractions, supporting a clean workspace and smooth creative flow [33].



(a) Step 1 (b) Step 2

Figure 3.11: Create Image Component Modular Structure

- **On-Demand Content Panels:** Project and image details (e.g., prompts, metadata) are shown in **on-demand panels** triggered by specific user actions, such as selecting a "details" button or a project tab. These panels can be opened via **remote pinch** or virtual buttons, and automatically close surrounding elements to reduce clutter. This design follows the principles of **progressive disclosure** and **focused interaction**, supporting user concentration and avoiding cognitive overload [33].

Figure 3.12: Image Details On Demand

3.4 Gesture-Driven Interface

A fundamental aspect of this study revolves around the design and evaluation of a **gesture-driven interface**. For every defined gesture, a corresponding virtual button alternative exists, allowing for a comparative analysis within the hybrid interaction model. The design process for these gestures was rooted in a blend of real-world **affordances** and considerations for ease of use and **recognition**, aiming to create intuitive interactions for key processes within the application. Subsequent testing and analysis verified the correctness and efficacy of these design assumptions.

3.4.1 Designing and Evaluating Hand Gestures

Designing effective hand gestures for VR interfaces is essential to creating natural and intuitive interactions. Unlike traditional controller-based input, hand tracking

offers a direct and embodied way for users to engage with virtual environments. However, this advantage also brings new design challenges: gestures must be **learnable**, **memorable**, and **resistant to misinterpretation**—all while remaining physically comfortable and contextually appropriate.

The design process followed a structured and iterative approach:

1. **Research and Inspiration:** Initial exploration drew from natural human communication patterns and real-world actions. The futuristic, gesture-based interface of Tony Stark in *Iron Man* served as a conceptual reference—highlighting how spatial gestures can be expressive and intuitive. This step reinforced the importance of **"Building upon Real World Knowledge"** when crafting interactions, aligning with the principle that virtual systems should leverage users' preexisting mental models [33, 38, 39].

Figure 3.13: Example of a natural user interface depicted in the film Iron Man.

2. **Definition of Core Operations:** Key functions of the VR moodboarding tool were identified as the primary candidates for gesture control. These included: **generating images and boards**, **initiating voice input**, and **navigating the interface**.

3. **Sketching and Prototyping:** For each core operation, multiple gesture concepts were **brainstormed** and **implemented** in early Unity prototypes. These prototypes allowed for rapid in-VR testing to evaluate gesture comfort, clarity, and the risk of false positives.

4. **User Testing and Iteration: Feedback** was collected from users during initial trials, focusing on how easy the gestures were to remember, perform, and distinguish. Gestures requiring excessive wrist rotation or prolonged awkward postures were revised or discarded.

3.4.2 Gesture Classification and Design

The gestures within the system are primarily classified into two categories: **Trigger Gestures** and **UI Navigation Gestures**. This classification helps structure the interaction model and provides clarity for users regarding the purpose of each hand movement. **Trigger Gestures** are used to initiate key actions, such as confirming selections or generating content, while **UI Navigation Gestures** support movement through menus or panels. This distinction also facilitates a more intuitive learning curve and minimizes gesture ambiguity within the immersive environment.

Trigger Gestures

Trigger gestures are designed to initiate core functionalities, often replacing traditional virtual buttons for enhanced immersion and efficiency.

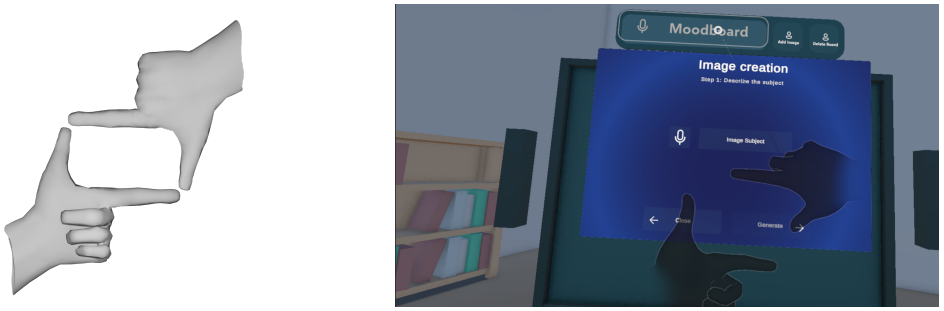
- **Start Mic (Voice Acquisition):** This gesture initiates **voice acquisition**, directly replacing a virtual button adorned with a microphone icon. The design provides a natural and immediate way for users to begin speaking, aligning with the common mental model of **"activating"** a microphone. The gesture consists of bringing two fingers close together in front of the mouth, simulating either holding a physical microphone or the act of drawing attention before speaking—both familiar actions grounded in real-world social interactions.

(a) Start Mic Gesture (b) Voice Acquisition in the application

Figure 3.14: Start Mic Gesture and its application.

- **Frame Gesture (Image/Moodboard Creation):** Inspired by the familiar act of **"snapping a photo"**, this gesture involves forming a rectangle with both hands in front of the face. Its function changes based on gaze context:
  - **Looking at a board:** Creates a new image on that board.
  - **Looking at empty space:** Creates a new moodboard in that location.

  Using gaze as a contextual cue reduces cognitive load, eliminating the need for multiple distinct gestures.



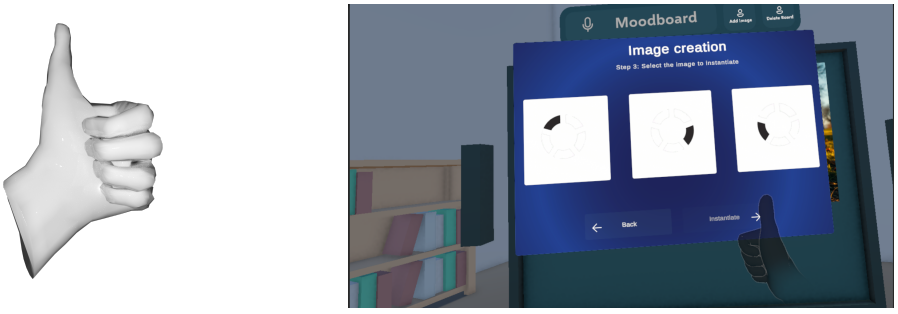
(a) Frame Gesture (b) Frame Gesture in the application

Figure 3.15: Frame Gesture (Image/Moodboard Creation) and its application.

UI Navigation Gestures

UI Navigation gestures facilitate fluid movement and confirmation within the user interface, drawing inspiration from natural human interactions.

- **Thumbs Up (Quick Confirm):** Used for rapid confirmation, this gesture maps to **"proceed"**, **"yes"**, or **"next"**, depending on context. Its universal association with approval makes it intuitive and efficient for frequent confirmation tasks.



(a) Thumbs Up Gesture (b) Thumbs Up in the application

Figure 3.16: Thumbs Up Gesture and its application.

- **Swipe (Directional Navigation/Negation):** Inspired by the natural action of **"flipping through a book,"** this gesture allows for directional navigation and negation based on the hand's movement direction. Moving the hand to the **left** is interpreted as **"proceed"** or **"next"** (akin to turning a page forward). Moving the hand to the **right** signifies **"go back"** or **"negate"** (similar to turning a page backward or dismissing an item). This gesture provides a natural and embodied way to navigate sequential content or confirm/deny options.

(a) Swipe Gesture

(b) Swipe in the application

Figure 3.17: Swipe Gesture and its application.

3.5 Figma-Based Desktop Prototyping for VR Interfaces

To translate the previously defined design principles into a concrete user interface, the initial prototyping phase was conducted using **Figma-based desktop tools**. This approach enabled a low-cost, rapid, and iterative exploration of interface layouts and interaction patterns before transitioning to immersive development. Although inherently two-dimensional, desktop prototyping aligns well with the principle of **"Building Upon Real World Knowledge"**, allowing designers to apply established UI paradigms to VR contexts [33].

The workflow followed a structured progression composed of six key phases:

1. Collection of Visual and Interaction References

The process began with the compilation of visual and interaction references to guide the interface's aesthetic and functional direction. A "meta moodboard" was assembled using imagery gathered from Pinterest (`https://it.pinterest.com/pietro2p/moodboard/`) and supplemented with generative content from Adobe

Firefly. These visual materials served as a foundation for identifying **patterns** aligned with natural and spatial interactions.

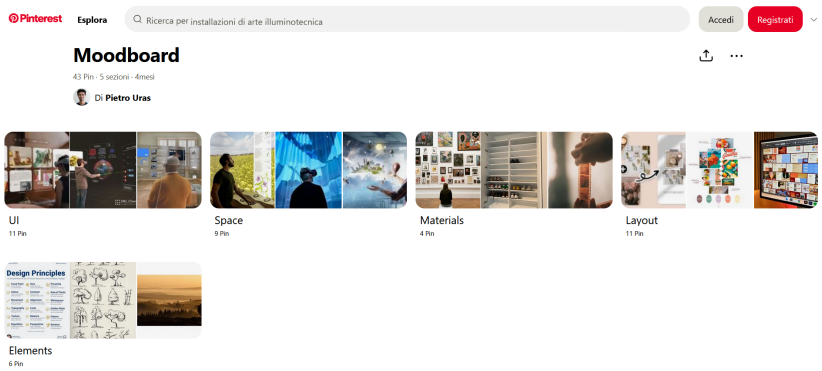


Figure 3.18: Images collected on Pinterest.

Figure 3.19: Meta Moodboard.

2. Comparative Analysis of Existing Tools and Frameworks

A parallel analysis was conducted between **conventional 2D moodboarding tools** (e.g., Milanote) and **immersive design frameworks** such as MRTK3. This informed early design decisions regarding component structure, interaction models, and user needs, helping to bridge traditional workflows with the demands of spatial computing.

3. UI Composition Using the MRTK3 Figma Toolkit

To ensure visual fidelity with the final VR implementation, components from the official **MRTK3 Figma toolkit** [40, 37] were used to create custom elements. The

use of standardized components allowed for consistency across platforms and eased the transition to 3D implementation. Having a ready-made toolkit significantly accelerated the design process, reducing time spent on creating and validating UI elements. This approach also ensured alignment with best practices and improved collaboration between design and development teams.

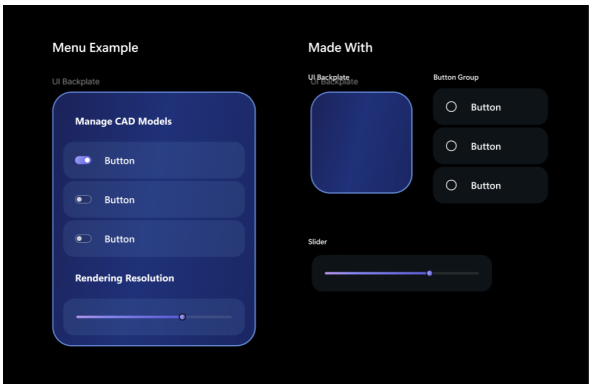


Figure 3.20: Example of MRTK3 Figma Toolkit.

Figure 3.21: Custom components built using the MRTK3 Figma Toolkit.

4. Layout Design with VR-Specific Ergonomic Constraints

Although Figma is a 2D tool, the interface layout was designed to respect VR-specific constraints such as **field of view** (FOV), **depth**, and **ergonomic reach**. The working canvas was set to **3664 × 1920 pixels**, simulating the combined resolution of the Meta Quest 2 headset (*1832 × 1920 pixels per eye*) [41].

Visibility and comfort zones were calculated based on pixel-per-degree metrics derived from VR usability studies [42], and further informed by the Figma community resource *"Guide for Spatial Design of VR"* [43], which was adapted to account for the specific optical properties of the Meta Quest 2.

- **Green Zone** (central clarity, 50° FOV): **1000 × 1000 px**
- **Yellow Zone** (peripheral area, 50°–90°): **1600 × 1600 px**
- **Gray Zone** (outside 90°): areas discouraged for UI placement

Comfort boundaries were also represented visually:

- Horizontal neck rotation (±30°): **±1000 px**
- Vertical head tilt (±20°): **±700 px**

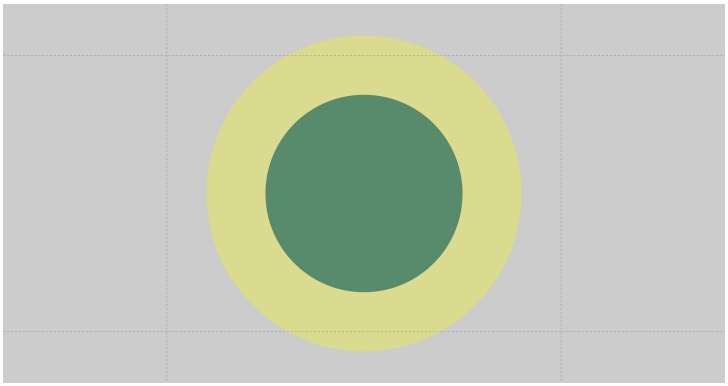


Figure 3.22: Visual comfort zones and ergonomic limits for UI placement.

5. Z-Axis Planning and Interaction Precision

The design strategy carefully considered spatial positioning along the **Z-axis**, with the principle that each component should ideally remain fully within the **green zone**. The distance of a component from the user depends on its content density: simpler and smaller UI elements—such as action buttons—are placed closer to enable **direct manipulation** through touch, while more complex or content-rich

components—like full canvases or parameter panels—are positioned farther away, determining the use of **indirect interaction** methods such as pinch or remote gestures. A further essential condition is that all interactive components, especially those positioned at a distance, must be **sufficiently large and well spaced** to allow easy and accurate selection, minimizing input errors even when accessed from suboptimal angles or depths.

In cases where the interface displayed **two parallel windows**—for example, a dual-panel layout—it was recommended to extend the layout partially into the **yellow zone**, ensuring that at least part of each window remained within the green zone. This spatial compromise allowed users to engage with both panels simultaneously while maintaining an overall comfortable interaction experience.

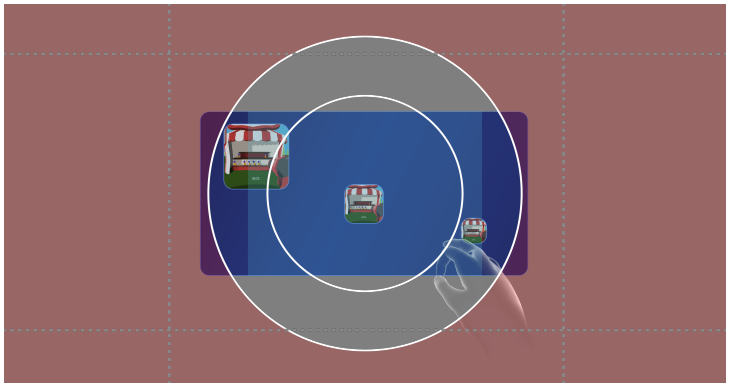


Figure 3.23: Figma mockup of moodboard interface with comfort visibility zones and neck movement limits.

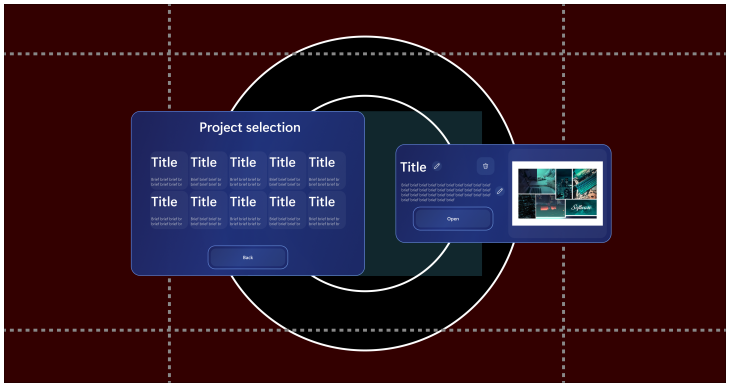


Figure 3.24: Figma mockup of project selection screen with ergonomic placement guides.

6. Integration into Unity and In-Headset Evaluation

The final Figma mockups were exported as **PNG assets** and integrated into Unity using MRTK canvases and 3D planes. Visual proportions and spatial relationships were validated through in-headset **testing**, which confirmed the reliability of Figma prototypes for approximating scale, comfort, and usability in VR environments.

This workflow effectively applied the principle of **"Keeping it Simple: Do Not Overwhelm the User"**, focusing on clarity and layout simplicity before introducing full 3D complexity [33]. It also followed the guideline to **"Design Around Hardware Capabilities and Limitations"**, supporting a rapid iteration cycle without the need for continuous VR deployment.

3.6 User Operations and Application Flow

The VR moodboarding tool offers a comprehensive set of **user operations** embedded within a seamless interaction flow. This section details the core actions available to users and describes the typical sequence of interactions from application launch through image creation and management.

Application Structure and Entry Points

At launch, users are prompted to select an existing profile or create a new one. Following profile selection, they choose among three distinct **input modalities**:

- **Traditional:** Interaction is based primarily on virtual button presses.
- **Gesture-Based:** Core functions are executed through hand gestures using hand tracking.
- **Hybrid:** A combination of traditional UI controls and gesture-based inputs.

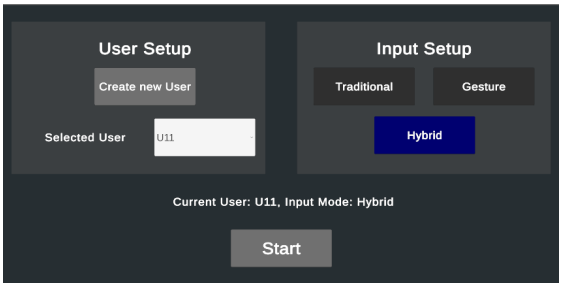


Figure 3.25: Testing setup.

Upon selecting the interaction mode, users enter the virtual environment and are presented with a **central main menu** offering the following options:

- **Gesture Tutorial:** A guided walkthrough introducing all supported gestures. Instructions are presented through on-screen text and synthesized voice narration (via `https://ttsmp3.com/`). Upon completion, users return to the main menu.
- **New Project:** Launches a voice-driven interface for creating a new project. Users dictate a *project title* and *brief*, both subject to character limits. Once confirmed, the project is instantiated and associated with an empty moodboard positioned in front of the user.
- **Load Project:** Opens a panel displaying all saved projects, enabling users to select, open, or permanently delete them. Each user may manage up to **nine active projects** in addition to a default template.
- **Quit:** Closes the application.

Each time the user returns to the initial scene, a dedicated component automatically recenters the main menu window to align with the user's current gaze direction, thereby preventing disorientation and maintaining a consistent spatial reference.

Figure 3.26: Main menu.

Moodboarding Environment and Image Interaction

Each project can contain up to **five moodboards**, spatially arranged around the user within predefined **positional boundaries** to ensure legibility and avoid visual overlap. Moodboards feature the following characteristics:

- Repositionable via lateral **handles**, manipulated through distant pinch gestures to rotate boards around the user.
- Fixed depth, with a visible frame to enhance spatial presence and support a skeuomorphic design metaphor.
- Editable titles positioned above each frame, modifiable via selection or voice command.
- Dedicated buttons for image generation and board deletion.

Figure 3.27: Example of Moodboard Interaction

Each moodboard accommodates up to **ten images**, which support the following interactions:

- **Overlapping** is permitted; the most recently selected image is brought to the foreground.
- **Scaling** is enabled within fixed bounds to preserve readability and ease of selection.

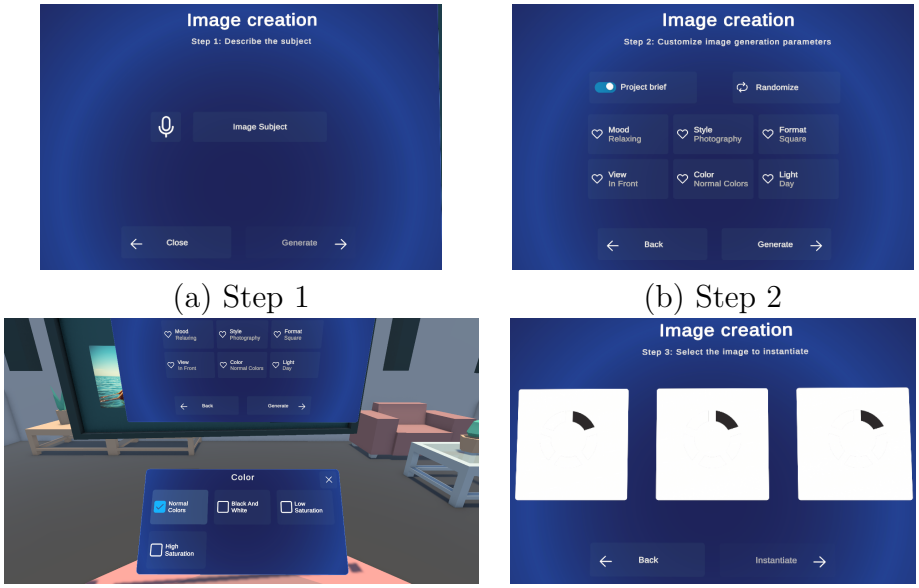
- **Transfer** between boards is accomplished via a pull-toward-face gesture, simulating the action of detaching a post-it note. Dropping the image onto another board completes the transfer; dropping it into empty space reverts it to its original position.
- **Deletion** is performed by releasing the image over a red floor zone, which triggers a confirmation dialog.

When an image is **held in place** for five seconds, a floating *information card* appears next to the moodboard, displaying **metadata** related to the image generation parameters.

Image Generation Workflow

Image creation can be initiated via a dedicated gesture or by selecting the corresponding button on the moodboard. The process consists of three distinct phases:

1. **Prompt Acquisition:** Users dictate a prompt using voice input.
2. **Parameter Selection:** A sequence of floating panels allows users to define visual parameters:
    - Lighting style
    - Artistic style
    - Aspect ratio
    - Color
    - Subject framing
    - Mood
    - Application of the project brief
    - Randomization of selected parameters
3. **Image Generation:** The system produces three image candidates (approx. 15 seconds each). Users may:
    - Instantiate a selected image onto the moodboard.
    - Review and adjust parameter selections to refine inputs and generate new candidates. In this process, a confirmation dialog is presented to verify the user's intent.



(a) Step 1 (b) Step 2

(c) Step 2.1 (d) Step 3

Figure 3.28: Image Creation Process

When a new image is spawned, all existing images momentarily fade to direct visual attention on the new one.

Confermation Dialogs

Confirmation dialogs are displayed for all destructive actions, including deletion of projects, boards, and images. This measure aligns with usability best practices aimed at preventing accidental data loss.

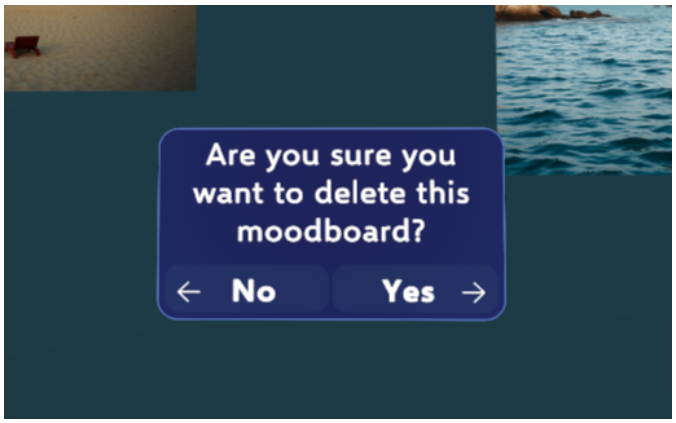


Figure 3.29: Delete Moodboard Dialog.

Hand Menu and Global Controls

A palm-up gesture directed toward the face opens the **hand menu**, a floating interface offering access to global controls:

- View and edit project details (title and brief).
- Return to the main menu, exiting the current project session.
- Create a new moodboard.

This design aims to minimize head movement and to preserve task continuity, aligning with the heuristic of **reducing cognitive load** in immersive systems.

Figure 3.30: Hand menu for global controls.

Spatial Environment and Audio Feedback

Each interaction—including gestures, selections, and dialog activations—is accompanied by distinct **audio cues**, reinforcing system feedback and improving clarity.

While initial iterations utilized HDRI backgrounds, these were replaced due to aliasing issues and reduced immersion. The final implementation features a stylized 3D environment constructed using free assets from `https://kenney.nl/`, which enhances spatial coherence and improves depth perception within the virtual scene.

3.7 Connection Between Moodboard Design Process and User Path

The design process of creating moodboards is inherently intertwined with the user's journey through the app, reflecting a natural progression of creative activities that the tool supports. Each stage of **moodboard creation** [23] corresponds to specific user operations and interaction flows within the application, ensuring that the system facilitates and enhances the designer's creative path.

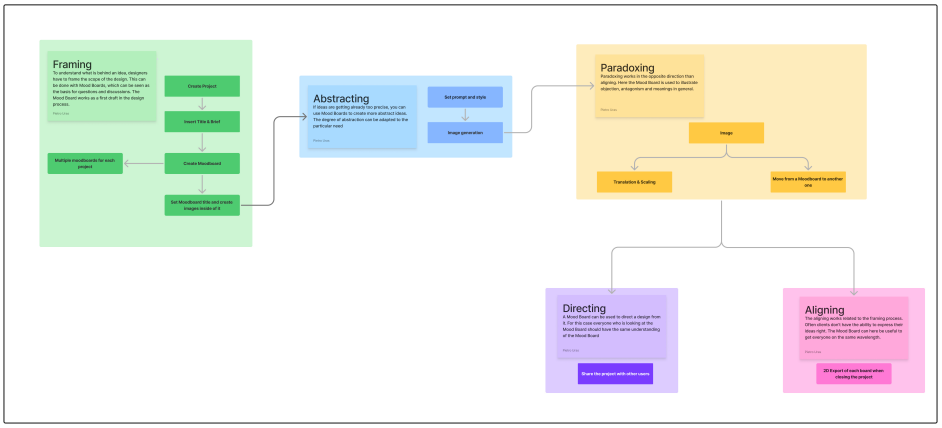


Figure 3.31: Connection between moodboard design process and user path

3.7.1 Establishing Structure and Context

At the start, users define the creative boundaries and organize their content within the **moodboard environment**. This corresponds to the app's initial steps where users create or select projects and moodboards, setting the scope and context for their work. The interface guides them in framing the workspace, allowing them to position and group elements in a meaningful way that reflects their conceptual goals.

3.7.2 Simplifying and Abstracting Ideas

As users gather images and inspirations, the app supports operations that help distill and abstract core themes. This stage aligns with user actions such as moving or layering images to capture overarching moods and concepts rather than overwhelming detail. The flow encourages focusing on essence, enabling users to curate content that best represents their design intent.

3.7.3 Exploring Contrasts and Divergent Paths

Creative exploration often involves embracing contradictions or experimenting with diverse ideas. Within the app, users can juxtapose contrasting images and rearrange elements freely, facilitating parallel exploration of different design directions. The interaction model supports branching and iterative refinement, encouraging users to engage with paradoxes and tensions to discover novel combinations.

3.7.4 Guiding and Focusing the Creative Direction

As the moodboard evolves, users begin to prioritize and emphasize certain elements, steering the overall narrative. This phase transitions from open exploration to focused decision-making within the user path.

3.7.5 Harmonizing and Unifying Elements

Finally, users work toward aligning all components into a coherent and balanced composition. The user path culminates in refining the moodboard to achieve a cohesive design vision, supported by intuitive interactions that encourage integration and consistency.

3.8 Software Architecture Patterns

This section outlines the architectural design patterns employed in the VR moodboarding application. These patterns support modularity, maintainability, and flexibility.

3.8.1 Observer Pattern for Gesture and Interaction Events

Gesture detection and interaction events in the system are managed using the **Observer pattern**, which promotes modularity and scalability. A central `GestureEventManager` serves as a broadcaster, dispatching high-level gesture events—such as the frame gesture, start mic, swipe, or thumbs up—when detected. These events are generated by dedicated gesture detectors including `HandPullDetectorXR`, `SwipeDetector`, and `XRHandGestureHandler`, each responsible for recognizing specific patterns of motion or hand pose.

System components that respond to gestures—such as UI controllers, image spawners, or audio processors—subscribe to these events and react accordingly. This decoupled architecture ensures that gesture recognition and gesture response remain cleanly separated, making it easier to expand or modify the gesture vocabulary without disrupting the broader system.

Gesture Recognition Implementation

Underlying this event system is a robust gesture recognition pipeline powered by **XR Hands**, Unity's official hand tracking subsystem introduced in version 1.5 via the `com.unity.xr.hands` package [44]. XR Hands provides low-level access to comprehensive tracking data, including joint positions, orientations, and per-finger curl metrics, enabling fine-grained gesture detection without the need for third-party SDKs.

Gestures in this system are defined through a combination of:

- **Hand shape** — identified by analyzing the **curl values** of each finger to determine whether they are extended, bent, or in transition.
- **Hand pose** — including the relative orientation and spatial configuration of the joints, essential for distinguishing similar shapes with different meanings (e.g., thumbs up vs. palm up).

To author and fine-tune gestures, the development process leveraged the built-in debugging and visualization tools provided by XR Hands. These tools allowed for real-time hand posing, visualization of joint hierarchies, and inspection of curl metrics. Once identified, gesture parameters were stored in dedicated `ScriptableObjects`, which served as reference templates during runtime matching.

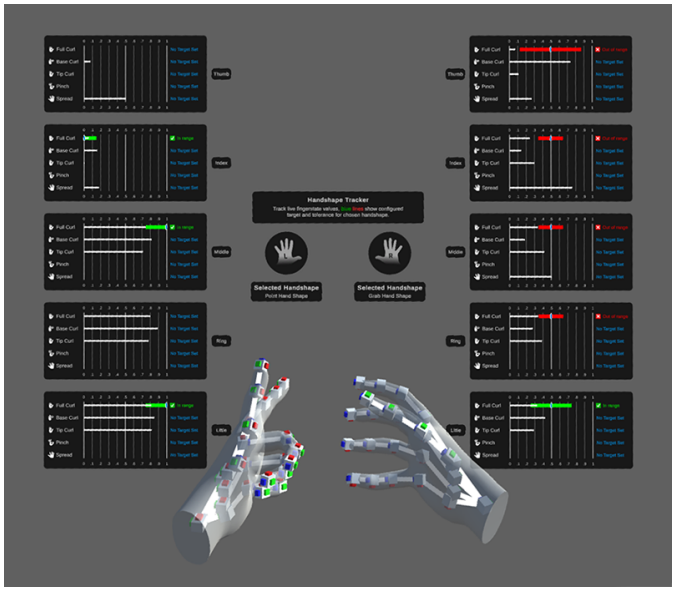


Figure 3.32: Gesture Debugger

Following initial implementation, a tuning phase adjusted threshold tolerances to optimize the recognition logic. This ensured gestures were not only precise but also comfortable to perform—reducing false positives while accommodating natural variation in users' hand size, posture, and motion.

3.8.2 MVVM for UI Decoupling

The **Model-View-ViewModel (MVVM)** pattern is implemented throughout the application to decouple the user interface (View) from business logic (Model) and data-binding logic (ViewModel).

- **Model:** Classes such as `AppData`, `ProjectData`, `MoodboardData`, and `ImageData` encapsulate the application's data structures. These classes define access rules and ensure data integrity across sessions.
- **View:** Each major interface component is associated with a dedicated view script (e.g., `V_UserSetup`, `V_MainMenu`, `V_Moodboard`, `V_Image`, `V_CreateImage`, `V_Prompting`). These scripts manage rendering and direct user interaction, delegating logic to the corresponding ViewModel.
- **ViewModel:** ViewModel scripts (e.g., `VM_AppData`, `VM_UserSetup`, `VM_CreateImage`, `VM_Prompting`) act as intermediaries between Views and Models. They transform raw data into UI-friendly formats and encapsulate view-specific logic. This structure ensures the UI remains replaceable or upgradable without disrupting application logic.

3.8.3 Supporting Architectural Patterns and Components

- **Singleton Pattern:** Applied to ensure a single instance of core services throughout the application's lifecycle. Notable examples include:
  - `VM_AppData` – manages the application's data model.
  - `SpeechToTextManager` – handles all speech recognition operations.
  - `AudioHelper` – controls playback of audio cues and feedback.
- **Image Generation Interface:** Image generation is abstracted via an interface to allow interchangeable implementations:
  - `Folder_ImageGenerator` – a local testing implementation that retrieves random images from a designated folder, enabling rapid development without network dependencies.

  - `DM_HF_ImageGenerator` – a production-ready implementation that launches a Python script to query a remote AI image generation service. It returns the generated image via a saved file path.

- **Custom Spatial Limiters:** Custom object constraints were developed to handle specific spatial behaviors beyond what MRTK3 offers:
  - `CanvaBoundLimiter` – restricts image placement within the visual bounds of a virtual moodboard canvas.
  - `SphereBoundLimiter` – ensures objects maintain a consistent orientation toward the user and remain within a user-defined radius for comfort and visibility.
- **Gaze-Based Window Positioning:** A dedicated component dynamically recenters interface windows—such as the starting menu or various dialog panels—based on the user's current gaze direction. This approach ensures that UI elements consistently appear within the user's immediate field of view upon scene changes or interaction triggers, enhancing spatial orientation and user comfort.

Chapter 4

Experiments

4.1 Test Structure and Methodology

To evaluate the application, a sample of 21 participants was selected to test the system across three distinct input modalities. After each session, a detailed report of user actions was analyzed to identify preferences and quantify the precision and effectiveness of gesture-based interactions. Additionally, participants were asked to complete questionnaires after each test.

4.1.1 Experimental Design

The experimental design involved three primary input modalities: *Traditional*, *Gesture-Only*, and *Hybrid*.

Input Modalities

- **Traditional Mode:** Gesture input was disabled. Users interacted exclusively through buttons to navigate the UI, activate voice recognition, generate images, and create moodboards.

- **Gesture-Only Mode:** Buttons were visible (with labels) but disabled. Users performed all interactions—voice activation, image generation, and board creation—through specific hand gestures.

- **Hybrid Mode:** Both input methods were enabled. Users could freely choose between gesture and button interactions for each action.

User Assignment and Training

To mitigate potential biases related to test order, participants were divided as evenly as possible: 11 users began with the **Traditional mode**, and 10 with the **Gesture-Only mode**. A dedicated in-app tutorial guided users through the gesture set, requiring successful completion of each gesture before proceeding.

All participants concluded with the **Hybrid mode**. This order was intentional: rather than identifying a universally superior input method, the study aimed to understand contextual preferences. Familiarity with both modalities was deemed essential for users to make informed choices during the Hybrid phase.

Each test followed a scripted scenario to ensure procedural consistency, which was essential for generating valid comparative metrics regarding task completion time, interaction preferences, and system performance.

Test Environment

All sessions were conducted using a Meta Quest 2 HMD connected via cable to a laptop. The environment was quiet, well-lit, and free from visual obstructions to ensure optimal hand tracking.

Figure 4.1: Test Environment

4.1.2 Scripted Interaction Flow

Each test session followed a fixed script to ensure consistency across participants and interaction modalities. The script was read aloud by the facilitator and divided into five sequential phases, introduced through voice instructions and guided steps.

General Introduction

Participants were informed that they would perform a series of tasks while wearing a VR headset, using only hand tracking—no physical controllers were allowed. They were also informed that the controls and interaction methods would vary slightly across the three test sessions, and that a short questionnaire would be completed at the end of each session.

Moodboard Context

To ensure understanding, the concept of a moodboard was briefly introduced as a visual tool used by designers and creatives to collect and arrange images representing ideas, emotions, and atmospheres. The goal of the session was to observe how users interact with the VR system during typical creative tasks.

Hand Tracking and Input Instructions

Participants were reminded that the headset tracks hands using front-facing cameras. Key interaction guidelines included:

- Keep hands raised and away from the face.
- Use pinch gestures (thumb + index finger) to interact with distant objects.
- Physically press virtual buttons for close-range interactions.
- All interactions and gestures can be performed using either hand.

Task Flow

The test procedure consisted of five main steps:

1. Project Creation Participants were instructed to create a new project via voice input:

- Speak a short project title (maximum 15 characters).
- Provide a vocal brief (up to 20 seconds).

2. Image Generation Participants were then asked to generate images for the project:

- Activate voice input and describe the desired image content.
- Customize the image style by selecting one label per image category via virtual buttons.
- Wait approximately 15 seconds for image generation (feedback was provided).
- Select one of the three generated images and instantiate it in the scene.
- Resize the image using two-hand scaling gestures.

A minimum of two images were required before proceeding.

3. Moodboard Management Participants were asked to:

- Create a new moodboard.
- Move an image from the first moodboard to the second.

4. Image Deletion To delete an image, participants had to:

- Simulate a "post-it removal" gesture by pulling downward forcefully.
- Observe floor color feedback (the floor turns red when an image is being dragged).
- Drop the image onto the floor to initiate deletion.
- Confirm the deletion via a pop-up dialog.

5. Exit and Wrap-Up To exit the session:

- Turn one palm toward the face to open the system menu.
- Use the other hand to select the "Back to Menu" button.

At this point, participants completed a questionnaire evaluating the session.

4.1.3 Data Collection

Data were gathered via two main channels: post-session questionnaires and detailed Unity-generated logs.

Questionnaires

Demographic data and prior VR experience were collected at the start. After each modality session, participants completed the System Usability Scale (SUS) [45] and NASA Task Load Index (NASA TLX) [46].

In addition, at the end of all sessions, participants answered open-ended questions regarding:

- Their preferred interaction modality and why.
- The modality most suited for professional or productivity contexts.
- The modality they would prefer for casual or leisure use.

Gesture-specific evaluation was conducted using a dedicated custom questionnaire. For each gesture, participants answered the following on a 5-point Likert scale (1 = very low, 5 = very high):

- *How intuitive did this gesture feel for performing this action?*
- *How much physical effort did you feel when performing this gesture?*
- *How much fatigue did you experience when repeating the gesture multiple times?*
- *How memorable and consistently repeatable is this gesture?*
- *How well did the system respond to the gesture as expected?*

A mean score for each gesture was then computed across all participants.

Quantitative Log Data

Quantitative interaction data were automatically recorded via Unity-generated log files, structured as follows:

```
Timestamp | UserID | SystemInput | ActionPerf | Trigger |
        GestureName | SceneName | Source | Other
```

This structure allowed the extraction of:

- Task completion times.
- Input preferences in Hybrid mode.
- Gesture precision and robustness across contexts.

Saved Session Data and Moodboard Snapshots

Each user's session also saved a JSON file containing the spatial arrangement of moodboards and images, as well as the project state. Additionally, a PNG snapshot was saved for each moodboard.

Below is an example of a saved JSON structure for a session involving two moodboards, **sea** and **swimming**, each containing different image elements arranged in 3D space:

Example of stored moodboard session data (JSON):


```
{
    "projects": [
        {
            "id": "509f32dc",
            "name": "Sea",
            "brief": "Summer commercial",
            "moodboards": [
                {
                    "id": "040c9a55",
                    "name": "Sea",
                    "position": { "x": 2.96, "y": 2.08, "z": 0.41 },
                    "rotation": { "x": 0.0, "y": 0.65, "z": 0.0, "w":
                    ↪  0.76 },
                    "images": [
                        {
                            "userPrompt": "Sea",
                            "format": "Portrait",
                            "style": "Painting",
                            "mood": "Dreamy"
                        },
                        {
                            "userPrompt": "Caraibic Beach",
                            "format": "Landscape",
                            "style": "Painting",
                            "mood": "Nostalgic"
                        }
                    ]
                },
                {
                    "id": "652307df",
                    "name": "Swimming",
                    "position": { "x": -0.23, "y": 2.06, "z": 2.91 },
                    "rotation": { "x": 0.0, "y": -0.08, "z": 0.0, "w":
                    ↪  1.0 },
                    "images": [
```

```
                        {
                            "userPrompt": "Person swimming in the
                            ↪  ocean",
                            "style": "Painting",
                            "mood": "Dreamy"
                        },
                        {
                            "userPrompt": "Person swimming under water",
                            "style": "Photography",
                            "mood": "Epic"
                        }
                    ]
                }
            ]
        }
    ]
}
```


To complement the JSON, here are snapshots of the moodboards as visualized in the VR environment:

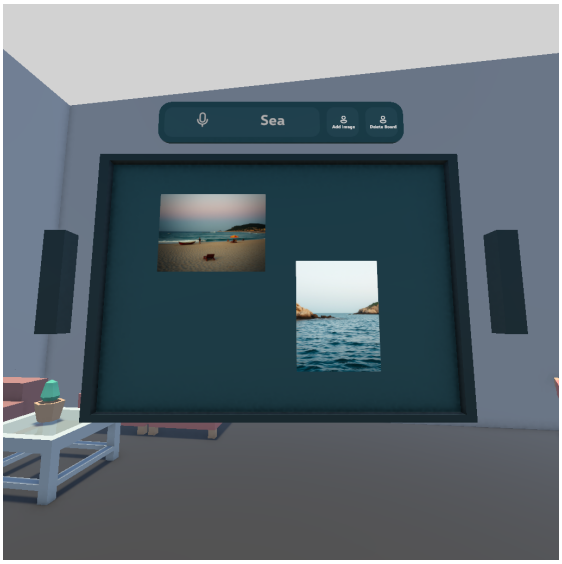


Figure 4.2: Moodboard snapshot: Sea

Figure 4.3: Moodboard snapshot: Swimming

Chapter 5

Results

5.1 Participant Demographics

The study involved **21 participants** with diverse backgrounds in terms of age, gender, and experience with VR and natural user interfaces.

Table 5.1: Demographic Overview of Participants

Attribute	Value
Number of participants	**21**
Average age	**24.5 years**
Age range	**19–29 years**
Gender distribution	**8 Female, 13 Male**
Dominant hand	**All Right-handed**
VR experience (avg on 1–5 scale)	**3.0**
Previous use of gesture/voice interfaces	**17 Yes, 4 No**

5.2 Quantitative and Qualitative Results Overview

This chapter presents both objective performance metrics and subjective evaluations of the three input modalities (**Traditional**, **Gestures**, **Hybrid**). The analysis is structured to first showcase user satisfaction (**SUS and NASA-TLX**), followed by interaction preferences, time-based efficiency, gesture-specific insights, and comparative behaviors in Hybrid use.

5.3 User Satisfaction Scores (SUS and NASA-TLX)

Perceived usability was measured using the **System Usability Scale (SUS)**, a widely adopted instrument producing scores ranging from 0 to 100. According to standard interpretation thresholds:

- Scores below 50 are considered **unacceptable**,
- Scores between 50 and 68 suggest **marginal usability**,
- Scores above 68 indicate **acceptable usability**,
- Scores above 80 are associated with **excellent usability**.

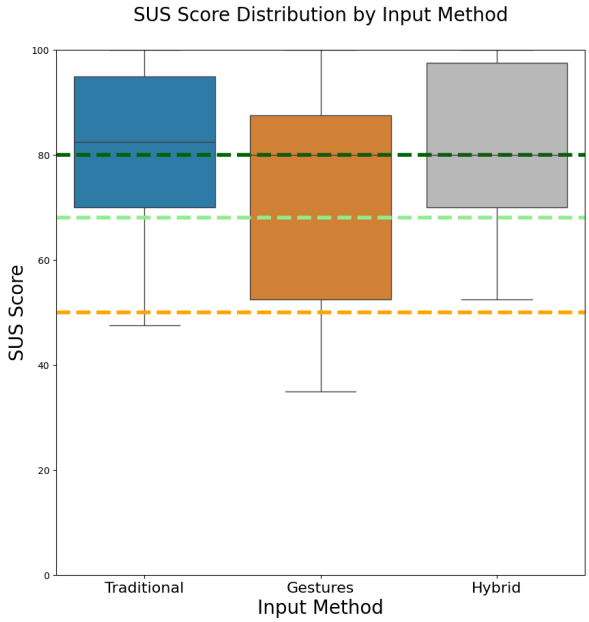


Figure 5.1: SUS score distribution by input modality

All modalities scored above the **50-point usability threshold**, indicating an overall positive user perception. A one-way ANOVA was conducted to examine whether the differences in SUS scores across the three input modalities were statistically significant. The test revealed no significant effect of input modality on usability ratings ($F(2,60) = 1.750$, $p = .1825$), suggesting that observed differences

in mean scores are not large enough to rule out random variation. Gesture mode presented the most critical usability challenges, showing greater variability and scores occasionally approaching the marginal acceptability boundary. Hybrid slightly outperformed Traditional, with both falling within the **"excellent usability"** range, reflecting a generally solid and satisfying user experience.

NASA-TLX Workload Assessment

In addition to usability, subjective workload was measured using the **NASA Task Load Index (NASA-TLX)**. The scale considers mental, physical, and temporal demand, performance, effort, and frustration. Scores range from 0 (no workload) to 100 (extreme workload), with lower scores indicating more desirable interaction conditions.

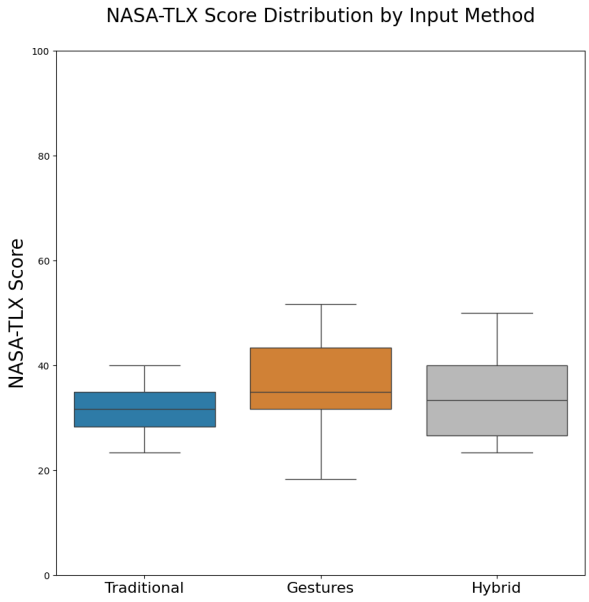


Figure 5.2: NASA-TLX workload score distribution by input modality

All workload scores ranged between **25 and 45**, indicating a moderate to moderately high level of perceived effort—neither extremely low nor extremely high—reflecting a certain degree of mental and/or physical demand. Median values were similar across modalities; however, Traditional input exhibited lower variance, while Gesture and Hybrid modes showed comparable levels of variability. Notably, Gesture interaction was associated with a slightly higher median workload compared to the other modalities.

A one-way ANOVA was conducted to assess whether the differences in NASA-TLX workload scores among the three modalities were statistically significant. The analysis yielded no significant effect of input modality ($F(2,60) = 1.626$, $p = .2052$), suggesting that the observed differences in perceived workload are not statistically meaningful and could be attributed to chance.

5.4 User Preferences and Perceived Suitability

Users were asked to select their favorite modality, the one they found most natural, and the one they considered most solid for practical work. Results were aggregated and visualized as pie charts.

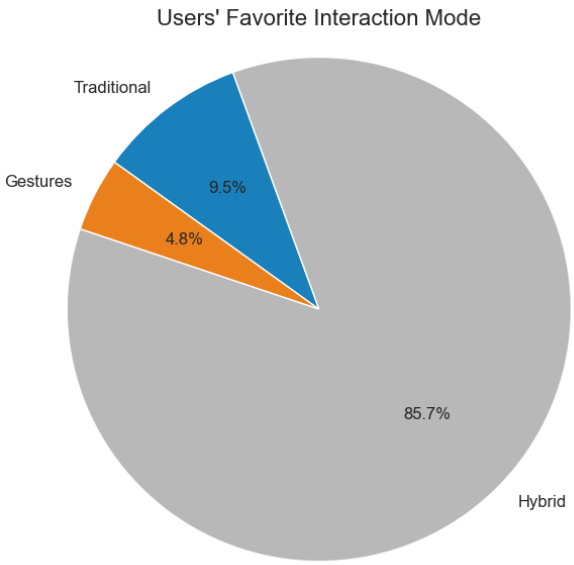
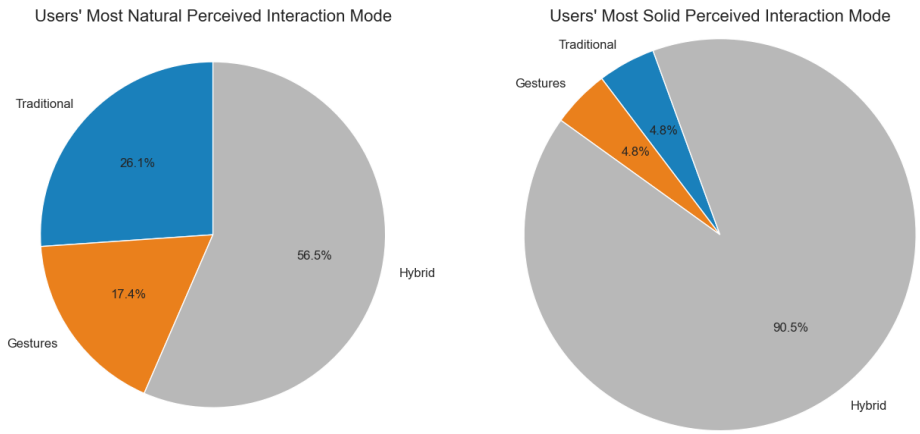


Figure 5.3: Users' favorite interaction mode

Most participants (approximately **86%**) selected **Hybrid** as their favorite mode, appreciating its flexibility and adaptability. Traditional was preferred by about **10%**, while Gestures received the lowest preference at around **5%**, likely due to perceived control issues despite being engaging.



(a) Most Natural Interaction Mode (b) Most Solid Interaction Mode

Figure 5.4: User preferences for interaction modalities.

Regarding perceived system solidity, approximately **90%** of participants favored the **Hybrid mode**, while Traditional and Gesture modes were each preferred by about **5%**, highlighting the Hybrid system's strong reliability. For naturalness, **62%** preferred Hybrid for its balance of intuitiveness and dependability, followed by Traditional at **29%**, and Gesture at **19%**, which was noted as the most natural by some users.

5.5 Task Completion Times

Efficiency Analysis: Task durations were measured and decomposed into steps, with each modality analyzed independently.

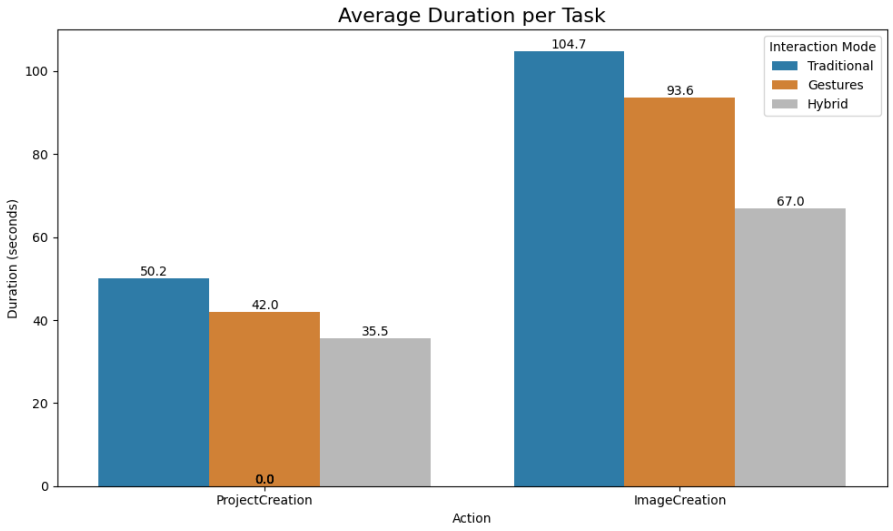


Figure 5.5: Average task duration per modality

Hybrid mode demonstrated the fastest average task completion time, likely influenced by users' accumulated experience, as it was consistently tested last. **Gesture input** slightly outperformed the **Traditional mode** in speed, suggesting that performing gestures can be quicker than targeting and activating buttons via pinch. This advantage may stem from the more natural and fluid nature of hand poses compared to precise button aiming, which can require additional time for cursor positioning and selection.

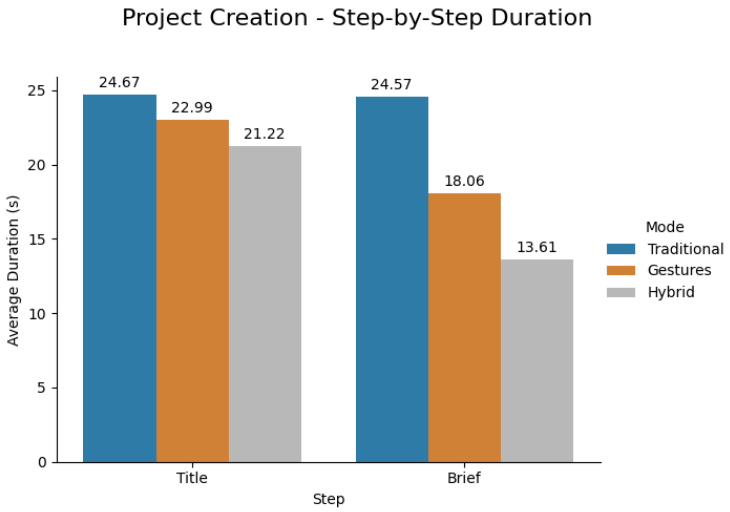


Figure 5.6: Step durations for project creation

Creating a project involves two voice inputs and two UI interactions. The shortest completion times occurred in **Hybrid mode**, with gesture-based input faster than button-based. Task durations were comparable, reflecting repeated action sequences, though a slight improvement was noted from title to brief insertion, likely due to growing user familiarity.

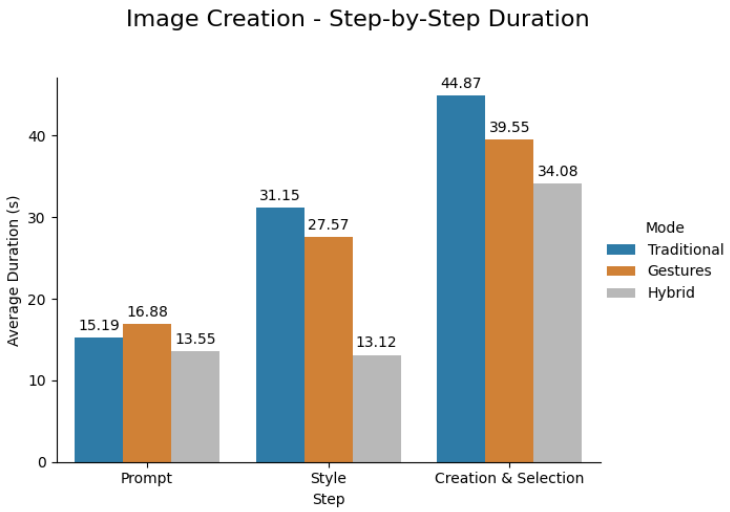


Figure 5.7: Step durations for image creation

In the image creation task, **Hybrid mode** consistently yielded the fastest completion times. The process involved three main phases: acquiring the prompt

via voice and confirming, selecting stylistic parameters using virtual buttons and proceeding, and finally selecting the generated image and confirming. While button input was faster during the initial prompt acquisition, gesture-based interaction was more efficient for style selection and image instantiation. It is worth noting that the image generation phase takes approximately **30 seconds**, which is included in the recorded completion times.

5.6 Gesture Evaluation and Feedback

Participants rated each gesture based on **intuitiveness**, **physical demand**, **fatigue**, **responsiveness** and **memorablity**.

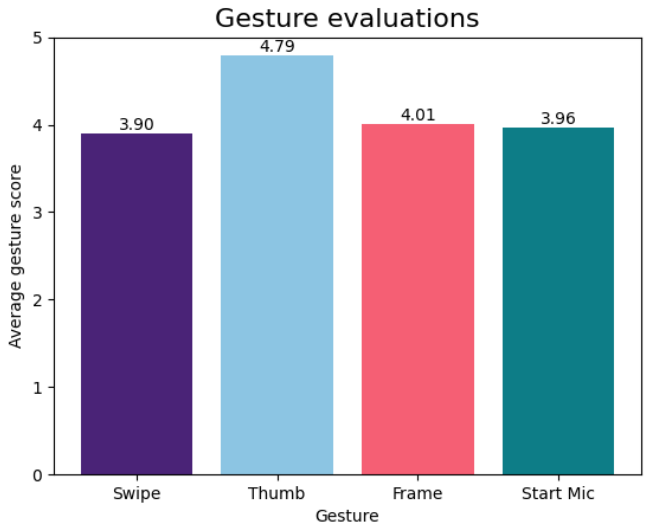
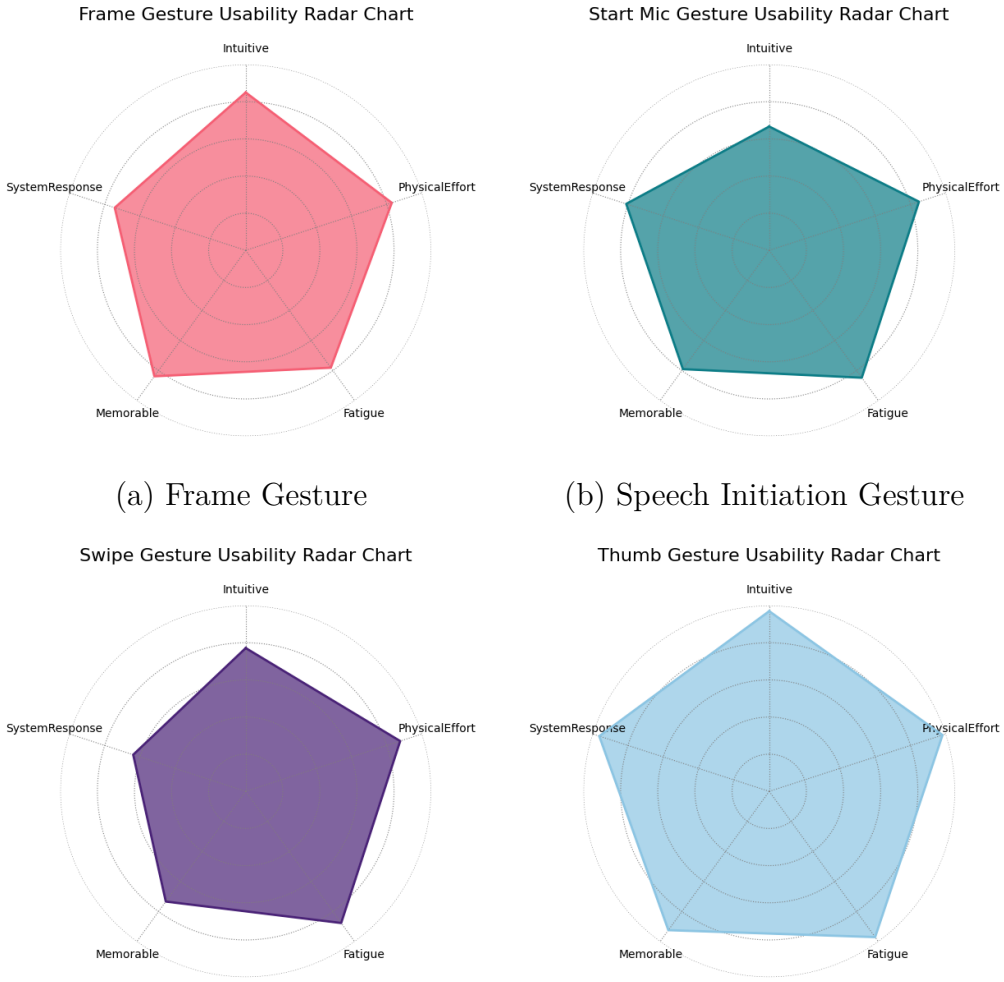


Figure 5.8: Average gesture evaluation scores across categories

Overall, all gesture types received average ratings above **4 out of 5**, indicating a strong overall perception of intuitiveness and comfort. The **Thumbs-Up gesture** received the highest average rating (**4.79/5**), reflecting excellent usability and user satisfaction. **Frame** and **Start Mic gestures** followed with similarly positive scores (**4.01** and **3.96**, respectively), suggesting they were generally well-received. **Swipe gestures** scored slightly lower (**3.9**), with some participants noting minor issues related to fatigue and responsiveness compared to the other input types.



(a) Frame Gesture

(b) Speech Initiation Gesture

(c) Swipe Gesture

(d) Thumbs-Up Gesture

Figure 5.9: Usability radar charts for key hand gestures used in the application.

The least physically fatiguing and easiest to perform gesture was the **Thumbs-Up**, while the **Frame gesture** was the most demanding, as it requires the use of both hands—an expected outcome. Nonetheless, the Frame gesture still achieved scores around **4 out of 5**, indicating a generally positive reception. The easiest gesture to remember was the **Thumbs-Up**, followed by **Frame**, **Speech Initiation**, and **Swipe**. In terms of responsiveness, **Thumbs-Up** and **Speech Initiation** were rated highest. Interestingly, Speech Initiation was perceived as the least intuitive, with intuitiveness ratings increasing in order from **Swipe**, **Frame**, to **Thumbs-Up**.

5.6.1 Qualitative Feedback on Gesture Design

In addition to quantitative ratings, participants were encouraged to share open-ended feedback on the gesture system. These insights proved valuable in understanding how gestures were perceived and how their design could be refined.

Voice Input Gesture

The gesture used to initiate voice input received generally positive feedback, but some participants suggested more iconic alternatives—such as *mimicking the act of gripping a microphone*—indicating that **metaphor-based gestures** could improve intuitiveness and memorability.

Swipe Gestures

Swipe gestures were the most discussed interaction. While useful, their current implementation was sometimes seen as unintuitive or awkward. Participants revealed diverging mental models:

- Some found the gesture natural, likening it to *flipping through a book*.
- Others, more accustomed to digital interfaces, expected the swipe direction to match typical UI behavior—e.g., swiping right to move forward.

Suggestions to improve swipe gestures included:

- Allowing directional control based on which hand is used (right hand to go forward, left to go back)
- Letting users invert swipe direction based on personal preference
- Replacing swipes with more discrete gestures, like a *thumb-left* or *thumb-right* for navigation

Some participants also felt that the swipe was underutilized, suggesting it would be more effective for browsing large sets of content rather than triggering simple binary actions.

Frame Gesture

The **Frame gesture**, while generally understood, was considered physically demanding by some users. Suggestions included simplifying it or replacing it with a more ergonomic alternative, such as presenting open palms toward the content area to instantiate elements.

Deletion Gesture

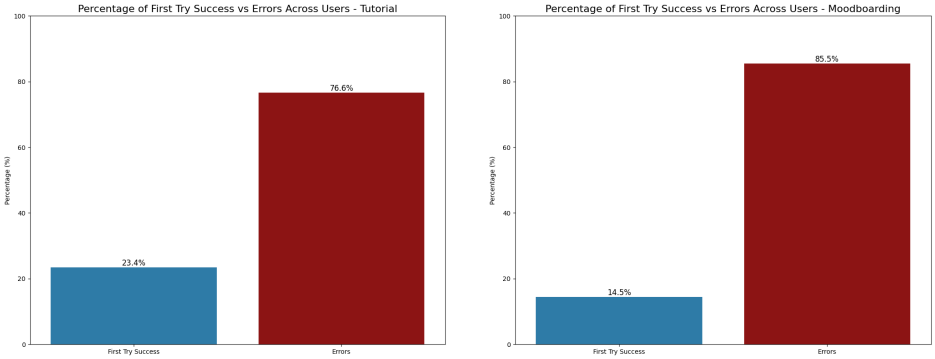
Several users expressed interest in having a **dedicated gesture for deletion**, suggesting alternatives like a *thumbs-down* to remove images or a *double thumbs-down* to delete entire moodboards. This highlights that gesture-based control was not only accepted but actively embraced, especially when tied to meaningful and context-specific actions.

Summary

Overall, this qualitative feedback emphasizes the importance of **gesture discoverability, ergonomic design**, and **metaphorical clarity**. Future iterations should consider user-driven refinements to increase both usability and the perceived naturalness of gestures.

5.7 Gesture Learning and Error Reduction

To assess gesture reliability, the proportion of correct gesture executions were tracked before and after the tutorial phase. Each attempt was categorized based on whether the gesture was successfully recognized on the first try. In the accompanying charts, successful first attempts are shown in **blue**, while failed or repeated attempts are shown in **red**.



(a) **Tutorial Phase** (b) **Moodboarding Phase**

Figure 5.10: Gesture accuracy and error rates across phases.

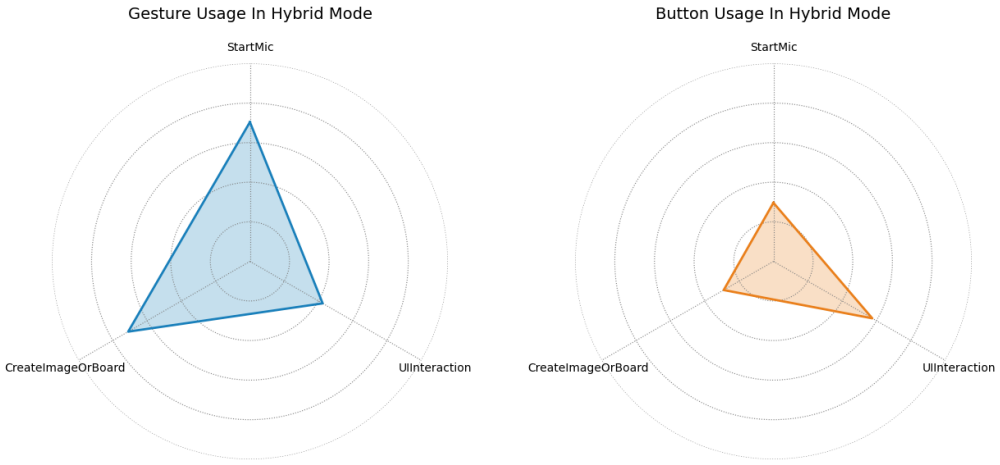
Performing the **Frame gesture** on the first try proved challenging, with only **23%** success during the tutorial and **14%** during moodboarding. Counterintuitively, tutorial statistics were better, likely because participants were asked to perform the gesture only twice in that context, compared to at least four times per test during moodboarding—amounting to a minimum of **12 attempts overall**. Although

generally perceived as intuitive and quickly recognized, many users needed multiple retries after initial failure, possibly inflating failure rates. This suggests that two-handed gestures like the Frame pose present additional difficulties, especially for users with very large or small hands. Improved hand tracking technology could help clarify whether these issues arise from the gesture design or tracking limitations.

5.8 User Behavior in Hybrid Mode

In **Hybrid mode**, we examined how often gestures were chosen over buttons for each action:

$$\text{Gesture Rate} = \frac{n_{\text{gestures}}}{n_{\text{gestures}} + n_{\text{buttons}}}$$



(a) **Gesture Usage Ratio** (b) **Button Usage Ratio**

Figure 5.11: Comparison of Gesture and Button Usage Ratios in Hybrid mode.

For operations such as instantiating images or moodboards, **gestures** were predominantly preferred by users. The same preference was observed for initiating voice input. Conversely, **virtual buttons** were favored for navigating the user interface.

5.9 Confirm Action Comparison (Proximal and Distal Windows)

Finally, the input distribution for key actions such as confirmation (**"Yes"**) and continuation (**"Next"**) was analyzed across modalities in two interface contexts: **proximal (closer)** and **distal (farther)** windows. Proximal interfaces allowed direct interaction, whereas distal interfaces required pinch gestures.

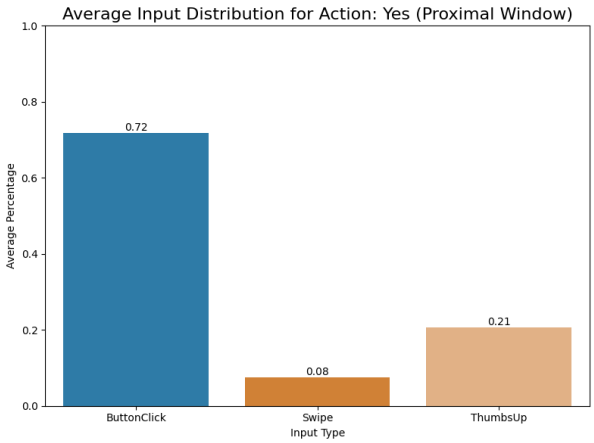


Figure 5.12: Input method distribution for confirm actions with distal windows in Hybrid mode

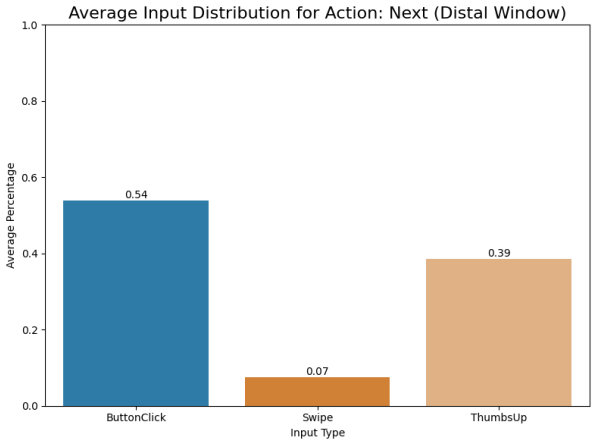


Figure 5.13: Input method distribution for confirm actions with proximal windows in Hybrid mode

Interaction with the UI consistently favored **virtual button presses** overall.

Among gestures, the **Thumbs-Up** was the most frequently used for quick confirmations, surpassing **Swipe**. A clear distinction emerged between interactions with **distant windows**—where users predominantly pressed the **"Next" button**—and **closer windows**—where the **"Yes" button** was mainly activated. While **Swipe usage** remained consistent across contexts, the **Thumbs-Up gesture** was preferred over virtual buttons, likely because it offers greater immediacy and ease compared to performing a pinch to select distant buttons.

Chapter 6

Conclusions and Future Work

6.1 Summary of Findings

This study investigated user preferences, performance, and perceptions of three interaction modalities—traditional virtual buttons, hand gestures, and a hybrid combination—within a VR moodboarding application. Key findings include:

- **Usability and Workload:** No statistically significant differences were found in overall usability (SUS) or perceived workload (NASA-TLX) between traditional button and gesture-based input modes. Both modalities were generally well received, with traditional buttons rated as more reliable and gestures as more variable but intuitive.
- **User Preferences:** Approximately 86% of participants selected the hybrid interface as their favorite. When comparing standalone modes, about 10% preferred traditional buttons for their solidity and precision, while gestures, favored by roughly 5%, were considered more natural yet sometimes fatiguing.
- **Task Efficiency:** Hybrid mode enabled the fastest task completion times, followed by gestures, which outperformed traditional buttons in some cases. Gestures facilitated quicker, more fluid actions such as image instantiation, whereas buttons excelled in precise UI navigation.
- **Gesture Evaluation:** Gestures received positive ratings overall, with the Thumbs-Up gesture scoring highest for intuitiveness and comfort. The Frame gesture—the only one requiring two hands—was found to be physically more demanding and less reliably recognized, suggesting potential for design improvements.

- **Uncertainty Regarding Users' Mental Models:** The Swipe gesture sparked mixed reactions due to ambiguity in users' mental models: some perceived it as analogous to flipping through a physical book, while others expected swipe directions to follow digital navigation conventions (e.g., swiping right to go forward). This ambiguity highlights the need for customizable or context-aware swipe interactions.
- **Hybrid Interaction Patterns:** Users naturally adopted a complementary strategy in the hybrid mode—favoring gestures for creative, initiation tasks (e.g., image creation, voice input) and buttons for navigation and confirmation—highlighting the strengths of multimodal input.
- **Design Implications:** Results underscore the importance of balancing reliability and naturalness in VR interfaces. Gestures should be designed for discoverability and ergonomic comfort, while traditional controls remain vital for tasks demanding precision and consistency.

6.2 Answering Research Questions

The study addressed the following research questions:

- **Do users prefer traditional virtual button-based interfaces or gesture-based interaction paradigms in virtual reality environments?** Users showed no statistically significant difference in overall usability (SUS) or workload (NASA-TLX) between traditional virtual buttons and gesture-based interactions. However, gestures exhibited greater variability in usability and were perceived as slightly more physically demanding. Traditional button-based interfaces were generally considered more solid and reliable, especially for UI navigation and control tasks. Gestures were appreciated for their naturalness and fluidity but sometimes raised concerns about precision and physical fatigue.
- **In a hybrid interface, do users naturally prefer to use gestures or virtual buttons for specific types of tasks (e.g., image creation, GUI navigation, text dictation)?** Within the hybrid interface, users predominantly favored gestures for tasks such as instantiating images, creating moodboards, and initiating voice input, leveraging the immediacy and natural feel of hand movements. Conversely, virtual buttons were preferred for general UI navigation and confirmation actions, where reliability and precision were more critical. This indicates a complementary use pattern, with gestures employed for creative and initiation tasks, and buttons for stable, repetitive interactions.

6.3 The Role of Hand Gestures in Multimodal Interaction Design

Building on the insights collected throughout this study, this section explores how hand gestures can meaningfully contribute to the design of creative applications in virtual reality and spatial computing contexts. Rather than offering prescriptive rules, the following considerations highlight the potential of gesture-based interaction as a core component of flexible, intuitive, and user-centered multimodal systems:

- **Prioritizing Direct Hand Interaction**: Emphasizing natural, hand-based input may help reduce abstraction and improve intuitiveness, particularly for users less familiar with VR environments.
- **Supporting Input Flexibility**: Allowing users to switch between gestures and virtual buttons can accommodate different contexts, personal preferences, and levels of physical effort.
- **Providing Clear Affordances and Feedback**: Making gesture possibilities visually evident and ensuring appropriate feedback (visual, auditory, or haptic) upon successful recognition may increase user confidence and reduce ambiguity.
- **Introducing Gestures Progressively**: Gradual onboarding through guided tutorials or visual prompts may lower the entry barrier and mitigate the risk of misinterpreted gestures.
- **Aligning Interaction Modes with Task Types**: Different modalities may suit different purposes—for instance, gestures might be more effective for expressive or symbolic actions, while buttons could be preferable for precise or repetitive tasks.

6.4 Limitations of the Study

Several limitations must be acknowledged:

- The **sample size**, while sufficient for initial insights, limits generalizability. More participants with diverse backgrounds would strengthen the results.
- The **gesture recognition system**, while functional, may not reflect the robustness of more advanced tracking setups, leading to higher error rates or gesture fatigue.

- The study focused on a **specific domain** (moodboarding). Results may not transfer directly to other VR applications (e.g., data visualization, simulation).
- No **haptic feedback** or **eye tracking** was integrated, limiting the exploration of truly multimodal and immersive interaction paradigms.

6.5 Future Work

Future directions can be structured across several axes of development:

1. Interaction Modalities and Multimodal Systems

- **Integration of Eye Tracking**: Using gaze to direct focus or disambiguate input could enhance speed and reduce effort in multimodal contexts, particularly in complex UI layouts.
- **Advanced Voice Input**: Beyond basic prompt acquisition, integrating natural language understanding (**NLU**) could allow users to refine prompts, chain commands, or reference visual elements contextually.
- **Gesture Library Expansion and Detection Improvements**: Exploring a wider range of gestures—including bi-manual or posture-based ones—and improving recognition fidelity through better sensors or **ML-based classifiers** could increase expressiveness and reduce frustration.
- **Haptic Feedback Integration**: Incorporating haptics (e.g., vibration, resistance) could reinforce gesture success and reduce cognitive demand, particularly for abstract or confirmation-based interactions. Future implementations should consider wearable or environmental haptic systems that do not obstruct hand mobility or interfere with gesture execution, unlike traditional controllers.

2. User Experience and Design Strategies

- **Augmented Reality Extensions**: Porting the tool to AR contexts could enable in-situ moodboarding with tracked surfaces or physical references, bridging the physical-virtual divide.
- **Skeuomorphic Interfaces**: Further exploration of skeuomorphism—interfaces that visually or behaviorally mimic real-world tools—may enhance learnability and immersion, particularly if combined with haptics or spatialized audio.
- **Long-Term Use Studies**: Studying how user behavior evolves over longer periods would provide deeper insights into learning curves, gesture memorability, and interface fatigue.

3. Moodboarding-Specific Enhancements

- **Stereoscopic Image Presentation**: Converting flat 2D images into stereoscopic visuals could increase immersion and visual realism in moodboards.
- **Multimedia Enrichment**: Embedding ambient sounds, voice notes, or even spatial audio elements could allow users to build moodboards that reflect tone and mood more richly.
- **Local AI Model Inference**: Running image generation models locally, rather than relying on external inference services, could allow for more customizable prompt parameters and finer control over image creation. This approach would also improve latency, privacy, and potentially offline usability.
- **Prompt Chaining and Image History**: Allowing users to iteratively modify generated images and track the evolution of a concept across prompt versions could support a more fluid creative process.
- **External Image Search Integration**: Integrating web-based image search functionality—e.g., through APIs from **Google Images**, **Pinterest**, or other visual platforms—could greatly enhance inspiration workflows, allowing users to enrich their boards with familiar or culturally contextual visuals.

In conclusion, this work highlights the potential and challenges of designing multimodal creative tools for VR. By combining **gesture**, **voice**, and **traditional input** in user-centered ways, future systems can foster expressiveness, accessibility, and immersion—essential qualities for virtual creativity.

Bibliography

[1] L. Tremosa. *Beyond AR vs. VR: What is the Difference between AR vs. MR vs. VR vs. XR?* Accessed: 2025-07-01. Interaction Design Foundation - IxDF. Mar. 2025. URL: https://www.interaction-design.org/literature/article/beyond-ar-vs-vr-what-is-the-difference-between-ar-vs-mr-vs-vr-vs-xr (cit. on pp. 3, 4).

[2] P. Milgram and F. Kishino. «A taxonomy of mixed reality visual displays». In: *IEICE Transactions on Information and Systems* E77-D.12 (1994), pp. 1321–1329 (cit. on pp. 3, 4).

[3] I. E. Sutherland. «A head-mounted three dimensional display». In: *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I*. ACM. 1968, pp. 757–764 (cit. on pp. 3, 5).

[4] Doug A. Bowman, Ernst Kruijff, Joseph J. LaViola Jr, and Ivan Poupyrev. *3D User Interfaces: Theory and Practice*. Addison-Wesley, 2004 (cit. on pp. 4, 7, 8).

[5] R. T. Azuma. «A survey of augmented reality». In: *Presence: Teleoperators & Virtual Environments* 6.4 (1997), pp. 355–385 (cit. on p. 4).

[6] Sharon Oviatt. «Ten myths of multimodal interaction». In: *Communications of the ACM* 42.11 (1999), pp. 74–81 (cit. on pp. 4, 7–10).

[7] Ismo Rakkolainen, Ahmed Farooq, Jari Kangas, Jaakko Hakulinen, Jussi Rantala, Markku Turunen, and Roope Raisamo. «Technologies for Multimodal Interaction in Extended Reality—A Scoping Review». In: *Multimodal Technologies and Interaction* 5.12 (2021). ISSN: 2414-4088. DOI: 10.3390/mti5120081. URL: https://www.mdpi.com/2414-4088/5/12/81 (cit. on p. 7).

[8] Sharon Oviatt. «Advances in robust multimodal interface design». In: *Proceedings of the IEEE* 91.9 (2003), pp. 1397–1410 (cit. on p. 7).

[9] R. A. Bolt. «Put-that-there: Voice-and-gesture at the graphics interface». In: *Computer Graphics* 14.3 (1980), pp. 262–270 (cit. on pp. 7, 10).

[10] D. Jurafsky and J. H. Martin. *Speech and Language Processing*. Pearson Prentice Hall, 2009 (cit. on p. 8).

[11] Lizhou Cao, Huadong Zhang, Chao Peng, and Jeffrey T. Hansberger. «Real-time multimodal interaction in virtual reality - a case study with a large virtual interface». In: 82.16 (2023). URL: https://doi.org/10.1007/s11042-023-14381-6 (cit. on pp. 9–11, 34, 37).

[12] M. Kolsch, R. Bane, T. Hollerer, and M. Turk. «Multimodal interaction with a wearable augmented reality system». In: *IEEE Computer Graphics and Applications* 26.3 (2006), pp. 62–71 (cit. on p. 9).

[13] Augusto Esteves, Yonghwan Shin, and Ian Oakley. «Comparing selection mechanisms for gaze input techniques in head-mounted displays». In: *International Journal of Human-Computer Studies* 139 (2020), p. 102414. ISSN: 1071-5819. DOI: https://doi.org/10.1016/j.ijhcs.2020.102414. URL: https://www.sciencedirect.com/science/article/pii/S1071581920300185 (cit. on p. 9).

[14] Eduardo Velloso and Marcus Carter. «The Emergence of EyePlay: A Survey of Eye Interaction in Games». In: *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play*. CHI PLAY '16. Austin, Texas, USA: Association for Computing Machinery, 2016, pp. 171–185. ISBN: 9781450344562. DOI: 10.1145/2967934.2968084. URL: https://doi.org/10.1145/2967934.2968084 (cit. on p. 9).

[15] Ed Kaiser, Alex Olwal, David McGee, Hrvoje Benko, Andrea Corradini, Xiaoguang Li, Phil Cohen, and Steven Feiner. «Mutual disambiguation of 3D multimodal interaction in augmented and virtual reality». In: *Proceedings of the 5th International Conference on Multimodal Interfaces*. ICMI '03. Vancouver, British Columbia, Canada: Association for Computing Machinery, 2003, pp. 12–19. ISBN: 1581136218. DOI: 10.1145/958432.958438. URL: https://doi.org/10.1145/958432.958438 (cit. on p. 10).

[16] Adam S. Williams, Jason Garcia, and Francisco Ortega. «Understanding Multimodal User Gesture and Speech Behavior for Object Manipulation in Augmented Reality Using Elicitation». In: *IEEE Transactions on Visualization and Computer Graphics* 26.12 (2020), pp. 3479–3489. DOI: 10.1109/TVCG.2020.3023566 (cit. on p. 10).

[17] Yukang Yan, Chun Yu, Xin Yi, and Yuanchun Shi. «HeadGesture: Hands-Free Input Approach Leveraging Head Movements for HMD Devices». In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2.4 (Dec. 2018). DOI: 10.1145/3287076. URL: https://doi.org/10.1145/3287076 (cit. on p. 10).

[18] Logan D. Clark, Aakash B. Bhagat, and Sara L. Riggs. «Extending Fitts' law in three-dimensional virtual environments with current low-cost virtual reality technology». In: *International Journal of Human-Computer Studies* 139 (2020), p. 102413. ISSN: 1071-5819. DOI: https://doi.org/10.1016/j.ijhcs.2020.102413. URL: https://www.sciencedirect.com/science/article/pii/S1071581920300173 (cit. on p. 11).

[19] Lisa Aufegger, Natasha Elliott-Deflo, and Tim Nichols. «Workspace and Productivity: Guidelines for Virtual Reality Workplace Design and Optimization». In: *Applied Sciences* 12 (July 2022), p. 7393. DOI: 10.3390/app12157393 (cit. on pp. 12, 13).

[20] Watanabe, Yudai and Cohen, Michael. «Intuitive space texture generation using hand tracking, speech recognition, and generative AI». In: *SHS Web Conf.* 194 (2024), p. 03003. DOI: 10.1051/shsconf/202419403003. URL: https://doi.org/10.1051/shsconf/202419403003 (cit. on p. 15).

[21] Tracy Cassidy. «The Mood Board Process Modeled and Understood as a Qualitative Design Research Tool». In: *Journal of Fashion Practice* 3 (Nov. 2011), pp. 225–252. DOI: 10.2752/175693811X13080607764854 (cit. on pp. 16, 17).

[22] Manuela Celi. «Design, metadesign and the importance of vision». In: *Strategic Design Research Journal* 5.2 (2012), pp. 84–90. DOI: 10.4013/sdrj.2012.52.04. URL: https://revistas.unisinos.br/index.php/sdrj/article/view/sdrj.2012.52.04 (cit. on p. 16).

[23] Andrés Lucero. «Framing, aligning, paradoxing, abstracting, and directing: how design mood boards work». In: *Proceedings of the Designing Interactive Systems Conference*. DIS '12. Newcastle Upon Tyne, United Kingdom: Association for Computing Machinery, 2012, pp. 438–447. ISBN: 9781450312103. DOI: 10.1145/2317956.2318021. URL: https://doi.org/10.1145/2317956.2318021 (cit. on pp. 17, 23, 55).

[24] Andrés Lucero, Dzmitry Aliakseyeu, and Jean-Bernard Martens. «Funky wall: presenting mood boards using gesture, speech and visuals». In: *Proceedings of the Working Conference on Advanced Visual Interfaces*. AVI '08. Napoli, Italy: Association for Computing Machinery, 2008, pp. 425–428. ISBN: 9781605581415. DOI: 10.1145/1385569.1385650. URL: https://doi.org/10.1145/1385569.1385650 (cit. on pp. 17, 20).

[25] Andrés Lucero. «Funky-Design-Spaces: Interactive Environments for Creativity Inspired by Observing Designers Making Mood Boards». In: vol. 9298. Sept. 2015, pp. 474–492. ISBN: 978-3-319-22697-2. DOI: 10.1007/978-3-319-22698-9_32 (cit. on pp. 17, 20).

[26] Fausto Brevi, Manuela Celi, and Flora Gaetani. «Creating Moodboards with Digital Tools: A New Educational Approach». In: June 2019. DOI: 10.36315/2019v1end115 (cit. on p. 17).

[27] Stefan Schmager, Fabian Schöttle, and Ralph Tille. *The moodCave - An immersive interactive environment to create Mood Boards*. https://citizendrain.de/bits/wp-content/uploads/2016/05/Schmager_Schoettle_Paper.pdf. Interface Design, Information Design, Hochschule der Medien Stuttgart. 2016 (cit. on p. 20).

[28] Vincent Rieuf, Carole Bouchard, and Améziane Aoussat. «Immersive moodboards, a comparative study of industrial design inspiration material». In: *J. of Design Research* 13 (Jan. 2015), p. 78. DOI: 10.1504/JDR.2015.067233 (cit. on pp. 20, 21).

[29] Juuso Mikkonen. «Advent of GAN: How does a generative AI create a moodboard?» In: *Nordes 2023: This Space Intentionally Left Blank*. Ed. by Stefan Holmlid, Vânia Rodrigues, Clara Westin, Peter G. Krogh, Maarit Mäkelä, Dag Svanaes, and Åsa Wikberg-Nilsson. Norrköping, Sweden: Linköping University, 2023. URL: https://doi.org/10.21606/nordes.2023.114 (cit. on pp. 21–24).

[30] Hugging Face. *Diffusers Library Documentation*. https://huggingface.co/docs/diffusers/v0.14.0/index. Accessed: 2025-07-01. 2023 (cit. on pp. 30, 31).

[31] Jakob Nielsen. «Heuristic evaluation». In: *Usability Inspection Methods*. USA: John Wiley & Sons, Inc., 1994, pp. 25–62. ISBN: 0471018775 (cit. on p. 34).

[32] Mica Endsley. «Situation Awareness and Human Error: Designing to Support Human Performance». In: (Jan. 1999) (cit. on p. 34).

[33] Steven Vi, Tiago Da Silva, and Frank Maurer. «User Experience Guidelines for Designing HMD Extended Reality Applications». In: Aug. 2019, pp. 319–341. ISBN: 978-3-030-29389-5. DOI: 10.1007/978-3-030-29390-1_18 (cit. on pp. 34–41, 44, 49).

[34] Ferran Argelaguet and Carlos Andujar. «A Survey of 3D Object Selection Techniques for Virtual Environments». In: (May 2013) (cit. on pp. 34, 38).

[35] Martin Bellgardt, Sebastian Pick, Daniel Zielasko, Tom Vierjahn, Benjamin Weyers, and Torsten W. Kuhlen. «Utilizing immersive virtual reality in everydaywork». In: *2017 IEEE 3rd Workshop on Everyday Virtual Reality (WEVR)*. 2017, pp. 1–4. DOI: 10.1109/WEVR.2017.7957708 (cit. on p. 34).

[36] N. Donorman. *The design of everyday things*. Basic books, 2013 (cit. on pp. 36, 37).

[37] Microsoft. *MRTK3 UX Components*. Retrieved June 27, 2025. 2023. URL: https://learn.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/mrtk3-uxcomponents/packages/uxcomponents/overview (cit. on pp. 39, 45).

[38] Thiago Campos, Maria Castello, Eduardo Damasceno, and Natasha Valentim. «An Updated Systematic Mapping Study on Usability and User Experience Evaluation of Touchable Holographic Solutions». In: *Journal on Interactive Systems* 16.1 (Jan. 2025), pp. 172–198. DOI: 10.5753/jis.2025.4694. URL: https://journals-sol.sbc.org.br/index.php/jis/article/view/4694 (cit. on p. 41).

[39] Perception. *Iron Man 2 Technology Design*. https://www.experienceperception.com/work/iron-man-2/. Accessed July 6, 2025; progetto di motion graphic e interfacce per Tony Stark (telefono trasparente, tavolo olografico, schermi windows-feed, ecc.) :contentReference[oaicite:0]index=0. 2010 (cit. on p. 41).

[40] Microsoft. *MRTK3 Figma Toolkit*. Retrieved June 27, 2025. 2022. URL: https://www.figma.com/community/file/1145959192595816999 (cit. on p. 45).

[41] Meta Platforms. *Meta Quest 2 Tech Specs*. 2021. URL: https://www.meta.com/it/quest/products/quest-2/tech-specs/ (cit. on p. 47).

[42] Jason Jerald. «The VR Book: Human-Centered Design for Virtual Reality». In: (2015) (cit. on p. 47).

[43] Mitsuhiro Kubo. *Guide for Spatial Design of VR*. https://www.figma.com/community/file/1304562419848737242/guide-for-spatial-design-of-vr. Figma Community Resource. 2023 (cit. on p. 47).

[44] Unity Technologies. *Unity XR Hands (com.unity.xr.hands) Manual*. 2024. URL: https://docs.unity3d.com/Packages/com.unity.xr.hands@1.5/manual/index.html (cit. on p. 57).

[45] John Brooke. «SUS: A quick and dirty usability scale». In: *Usability Eval. Ind.* 189 (Nov. 1995) (cit. on p. 64).

[46] Sandra G. Hart and Lowell E. Staveland. «Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research». In: *Human Mental Workload*. Ed. by Peter A. Hancock and Najmedin Meshkati. Vol. 52. Advances in Psychology. North-Holland, 1988, pp. 139–183. DOI: https://doi.org/10.1016/S0166-4115(08)62386-9. URL: https://www.sciencedirect.com/science/article/pii/S0166411508623869 (cit. on p. 64).