Team Member: Hossein Fathollahian, Karthik Ragi, Ahmad Albawaneh, Idunnuoluwa Adeniji
Workbook for Final Project in Human-Computer Interaction CS522 (Github)
Project output(Build Unity) LINK
Sketches
CW

Fall 2025
University of Illinois Chicago
Instructor: Prof. Debaleena Chattopadhyay
==

Helping Cosmologists explore complex scientific datasets

1- Introduction

Scientists and research students working with large and complex spatial datasets generated in cosmology face significant challenges in exploring, interpreting, and communicating their data. These datasets often contain multiple particle types (e.g., gas, stars, dark matter) and intricate spatial relationships that can be difficult to disentangle using traditional 2D desktop-based visualization tools.

To deepen understanding and foster collaboration, scientists require environments that can toggle between particle types, adjust color maps for perceptual clarity, visualize flow direction through vector arrows, and filter out noisy regions.

2- Problem Statement

Despite advances in immersive visualization technologies, many existing tools fall short in supporting the nuanced exploration and analysis required for large-scale scientific cosmological datasets. Current systems often emphasize visual immersion without offering the responsive, user-centered interaction capabilities scientists need to interpret complex relationships between data elements.

As a result, researchers struggle to efficiently isolate meaningful structures, compare spatial or temporal phenomena, and filter visual noise while maintaining focus within the immersive environment. This limits their ability to derive scientific insight and reduces the potential of immersive visualization as an analytical medium.

The core problem addressed in this project is to enable scientists to effectively explore cosmological data, identify patterns, and analyze large and complex relationships among particle types within immersive environments.

3- Data Description/Dataset

The prototype will utilize a cosmological simulation dataset represented as a point cloud, offering a complex yet meaningful environment for immersive exploration and collaboration. Additionally, it will include a stack of microscope images of fruit fly cells to broaden and diversify the types of datasets that the system can visualize.

4- Conceptual Description/Diagram

The conceptual design integrates immersive VR interaction with desktop-based scientific analysis to create a unified visualization and collaboration environment. The proposed framework could include three main conceptual layers:

1. **Data Management Component:** Handles the import, storage, and processing of point-cloud datasets (e.g., cosmological simulations). It ensures smooth data streaming between desktop and VR systems, allowing users to cut, save, and annotate specific portions of the dataset for deeper analysis.
2. **Visualization and Interaction Layer:** Enables users to manipulate 3D datasets within a VR space using controller interactions. It provides a contextual menu attached to a controller that allows users to toggle available particle types, choose perceptually uniform colormaps, overlay vector arrows that indicate directional flows, filter dense regions, or adjust opacity, point size, and other parameters through a lightweight interface.
3. **Collaboration/Multi-Device Layer**: Synchronizes activities between VR and desktop devices in real-time. Changes made in one environment (e.g., annotations, filters) are immediately reflected across connected devices, supporting continuous analysis across platforms.

4-1-Core Concepts

The diagram below (Figure 1) provides a concept of the core components of the system and workflow:

- **Dataset & Visualization Scene:** Immersive visualization of scientific data.
- **Filters, Color Maps, Camera:** Control and modify how the dataset appears.
- **VR & Desktop Interfaces:** Provide mirrored and shared control of the exact visualization.
- **Shared State Manager:** Synchronizes updates across both desktop and VR devices.
- **Session Manager & Annotations:** Save, load, and manage visualization states and user-created notes.

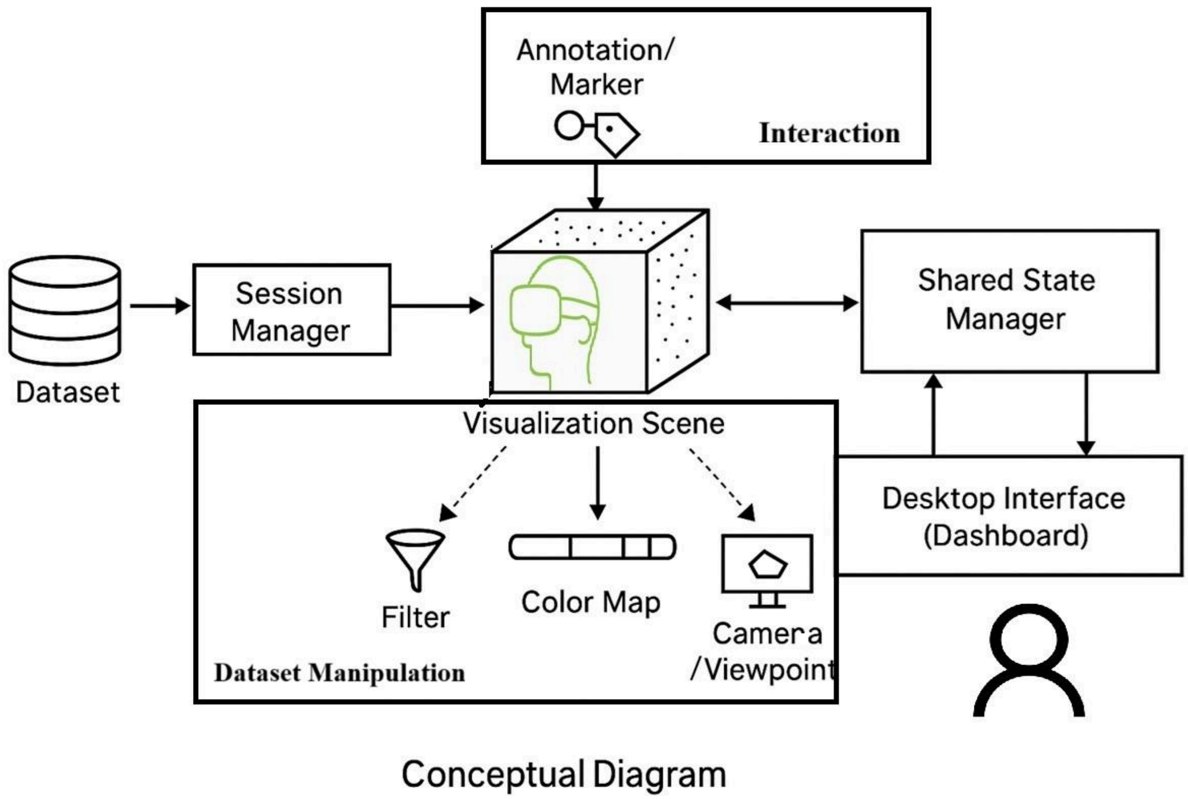


Figure 1. A conceptual diagram of the proposed system expectations and workflow

4-2-Relationships (high level)

- A Workspace contains Datasets, Sessions, Assets, and Users.
- A Session references a Scene; a Scene references a Dataset and Render Params.
- Filters, Selections, Annotations, and Camera bookmarks attach to a Scene (and optionally to a Dataset region).
- Shared State mirrors the current Session, so multiple Users see updates live. Conceptual layers (and responsibilities)

4-3-Design Rationale

This conceptual structure separates concerns of data handling, visualization, and collaboration, enabling modular and scalable development. Each layer can evolve independently, supporting future extensions such as multi-user synchronization, AI-assisted data analysis, or advanced VR interfaces.

Alignment with HCI and UX Principles

- **Learnability:** Simple, controller-based interactions for non-expert VR users.
- **Efficiency:** Streamlined workflows that minimize task interruptions.

- **Accessibility:** Compatible with affordable, widely available headsets.
- **Collaboration:** Real-time synchronization between heterogeneous devices.

Relationships (high level)

- A Workspace contains Datasets, Sessions, Assets, and Users.
- A Session references a Scene; a Scene references a Dataset and Render Params.
- Filters, Selections, Annotations, and Camera bookmarks attach to a Scene (and optionally to a Dataset region).
- Shared State mirrors the current Session, so multiple Users see updates live.

Conceptual layers (and responsibilities)

1) Data Management Component

- Import (local, remote, streaming); metadata parsing; units & coordinate frames.
- Tiling & Level-of-Detail (LOD); spatial indexing (e.g., octree).
- Cross-device streaming (desktop ↔ VR), caching, and versioned snapshots.
- Cut/save/annotate subsets; persist filters/ROIs; export views and data.

2) Visualization & Interaction Layer

- VR interaction (controllers: grab, scale, rotate; contextual radial/menu).
- Desktop interaction (mouse/keyboard: camera orbit, panels, sliders).
- Visual mappings (perceptually uniform colormaps; opacity/point size controls).
- Overlays (vector fields, density isosurfaces, legends, scale bars).
- Lightweight UI for parameter tweaks; multi-selection; comparison views (A/B).

3) Collaboration / Multi-Device Layer

- Real-time sync of scene, filters, and annotations across VR/desktop.
- Presence (who's here), cursors/laser pointers, roles & permissions.
- Session history, branching, rollbacks, and comment threads on annotations.

Key user actions (operators)

- Load dataset; configure units/scales; choose particle types.
- Navigate (teleport/grab in VR; orbit/pan/zoom on desktop).
- Filter (thresholds, boolean combinations); save named presets.
- Select region (ROI); compute stats (count, mean, variance, velocity).
- Annotate (pin note, record voice, attach image); link to selections.
- Adjust render (colormap, opacity, point size, LOD).
- Bookmark camera, capture snapshot/video, and export subset/CSV.
- Share session; invite collaborators; set roles.

System behaviors

- Maintain authoritative Shared State; broadcast diffs to all clients.
- Resolve conflicts (last-writer-wins for scalar params; CRDT/merge for lists like annotations).
- Background prefetch (tiles/LODs near camera and collaborators' cursors).
- Autosave sessions; version increments on meaningful changes.
- Validate filters/units; normalize metadata; ensure perceptual colormap defaults.

States

- Dataset: unloaded → loading → ready → error.
- Session: idle → editing → syncing → saved/snapshotted.
- Interaction: navigate → inspect → select → annotate → adjust → measure.
- Collaboration: solo → shared (live) → review (asynchronous).

Primary workflows

1. Import & Prepare: Load dataset → confirm units → auto-index/LOD → first Scene rendered.
2. Explore & Filter: Navigate → adjust colormap/opacity → apply filters → save preset.
3. Select & Measure: Define ROI → compute statistics → attach annotation → bookmark camera.
4. Compare: Split-view or A/B sessions → synchronized camera → delta stats and visuals.
5. Collaborate: Invite user → presence shows pointers → co-edit annotations → export snapshot.
6. Persist & Share: Save Session → export assets (images/video/CSV) → publish or handoff.

Information architecture (surface)

- Left: Project/Session browser; collaborators; dataset list.
- Center: 3D viewport (VR: world space; desktop: canvas).
- Right: Context panel (Render, Filters, ROI, Annotations, Camera, Export).
- Top: Session name, save status, presence; undo/redo; help.
- VR Controller Menu: quick toggles (particle types, filter presets, colormap, capture).

Feedback & visibility

- Load/progress bars (dataset, tiles); LOD indicators.
- Live badges for synced params ("shared") vs local staging ("draft").
- Selection overlays, filter chips, and annotation pins with visibility toggles.
- System toasts for saves/exports/errors; guidance hints in VR.

Constraints & assumptions

- Datasets can be billions of points → requires LOD & frustum-culled streaming.
- Heterogeneous devices; variable compute & bandwidth; graceful degradation.
- Perceptually uniform color maps are essential for scientific fidelity.
- VR comfort (90Hz target, comfort locomotion) and accessibility (seated modes).

Error handling & recovery

- Import errors (schema/units): show fix-up dialog; suggest defaults; allow re-map.

- Sync conflicts: Highlight conflicting fields and present options for accepting or overriding them.
- Performance drops: auto-lower point size/LOD; notify user; allow manual override.
- Safe operations: undo/redo, session checkpoints, and revert to snapshot.

Collaboration model

- Presence: avatars/labels; laser pointers; "follow camera" mode to guide tours.
- Auth & roles: owner (full), editor (scene/annotations), viewer (read-only).
- Sync primitives: parameter diffs, annotation CRUD, camera updates at a limited rate.
- History: timeline of actions; branch/fork session; merge annotations/filters.

Security & privacy

- Project-scoped access; dataset encryption at rest; signed export links.
- Activity logs for compliance; PII-free annotations by default.

Extensibility

- Plugin hooks: custom filters, analysis kernels, and new visual encodings.
- "Analysis nodes" (pipeline graph) to chain filters/statistics; cache results.
- AI-assist (optional): suggest filters/ROIs, label structures, and generate summaries.

5-Target Users and Their Characteristics

Primary Users:

Scientists and research students in fields of cosmology and spatial simulation who analyze complex 3D datasets. Most are proficient in data analysis but have varying levels of VR experience.

Secondary Users:

Collaborators, analysts, and stakeholders using desktop setups who benefit from synchronized visualization and shared annotations.

Key Characteristics:

- Intermediate to advanced technical proficiency.
- Need for immersive, intuitive interfaces for spatial data exploration.
- Desire for efficient workflows with minimal device switching.

6- User Goals

Users should be able to:

- Visualize and manipulate 3D datasets enriched with additional cues such as vector arrows indicating particle velocities.
- Toggle particle types (e.g., gas, stars, dark matter) on and off to explore relationships between subsets of the dataset and compare their distributions. For example, visualizing the relationship between gas and star particles.

- Adjust colormaps to suit different particle types and individual preferences, ensuring high legibility and accurate interpretation.
- Filter dense areas by hiding or remapping points outside selected ranges to reduce clutter and focus on busy regions without losing context.
- Interact with and filter data in real time while maintaining seamless transitions between desktop and VR environments.
- Save and load immersive work sessions that preserve filters, colormaps, and annotations, and share immersive views with desktop collaborators.

7- Example User Tasks

- Load the 3D cosmological simulation dataset containing gas, stars, and dark matter particles.
- Turn visual cues on or off, such as vector arrows, to view particle velocity and direction.
- Rotate, zoom, and navigate around the dataset to observe dynamic patterns in 3D.
- Adjust the visualization scale to highlight particle clustering or motion patterns.
- Toggle particle layers (gas, stars, dark matter) on and off to isolate or compare them.
- Overlay particle types to identify correlations between dense gas regions and star-forming zones.
- Adjust opacity or color to visualize overlaps and boundaries between components.
- Record observations by placing in-VR markers or annotations.
- Open the color map menu to customize color schemes for each particle type.
- Modify brightness and contrast for better visibility of dense or faint structures.
- Apply different color encodings (e.g., by temperature, velocity, mass) to test interpretability.
- Save preferred visualization presets for reuse in future sessions.
- Use filter controls to define value ranges of rendered points
- Hide particles outside the selected thresholds to declutter dense areas.
- Zoom into filtered regions to analyze substructures, such as halos or filaments.
- Switch between filtered and full views to retain spatial context.
- Perform real-time updates without removing the VR headset.
- Move seamlessly between desktop and VR to adjust visualization parameters and annotate findings.
- Resume exactly where the session left off after switching environments.
- Save the current visualization state (filters, color maps, and annotations).
- Load previously saved sessions for comparison or continuation of analysis.
- Export annotated 3D views or screenshots for collaboration and publication.
- Share immersive sessions with desktop collaborators for group discussion.

8- Design Challenges

1. Load datasets and visualize them

2. Manipulate, Filter, Rotate, Zoom, and Navigate
3. Save and Load session, Load previously saved sessions for comparison or continuation of analysis.
4. Perform real-time updates without removing the VR headset
5. Move seamlessly between desktop and VR to adjust visualization parameters and annotate findings.

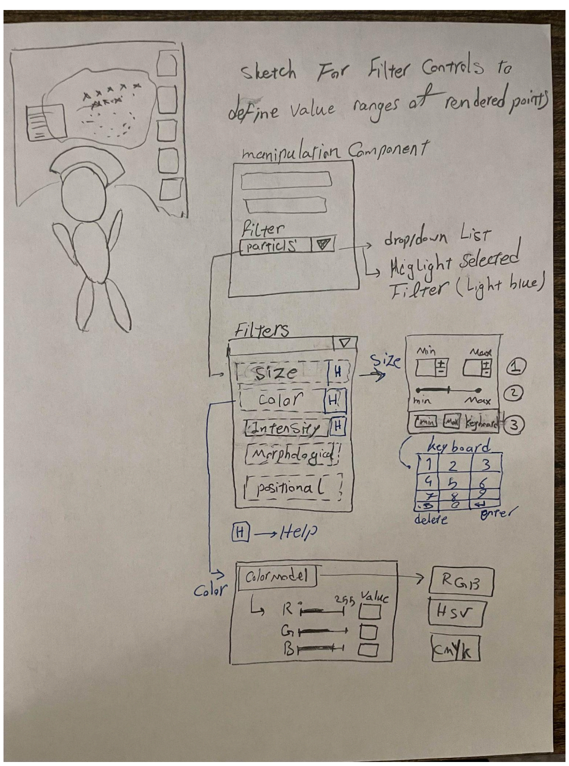
9- Sketches:

<u>Use filter controls to define value ranges of rendered points</u>

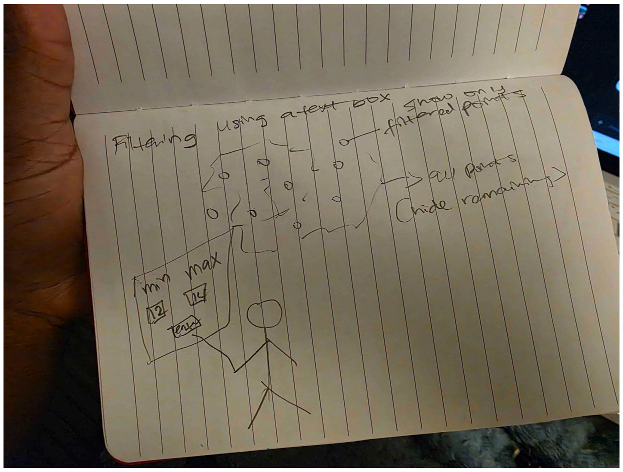
All Sketches folder links:
https://drive.google.com/drive/folders/1mNsg09_rCxjXKVI7U4Y3aeCRFkJtiGkV?usp=drive_link

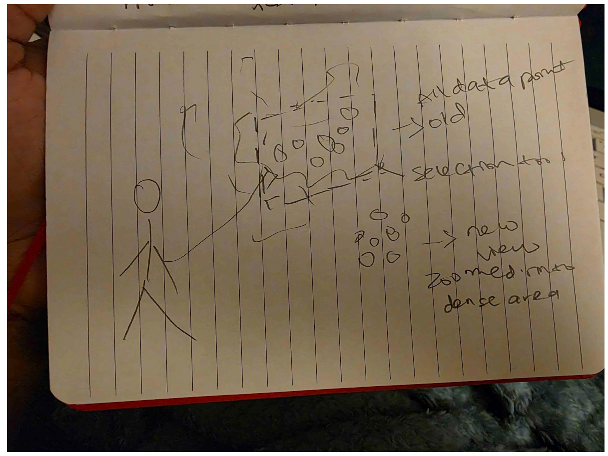
1. Hossein's Sketches

In the manipulation component, I have a filter module, which I use/drop-down menu to select one filter. Next, I choose one filter and use the keyboard to enter an exact value, or use the bar strip for left and right adjustments. The third option is the +/- button, which allows you to increase or decrease the filter value.



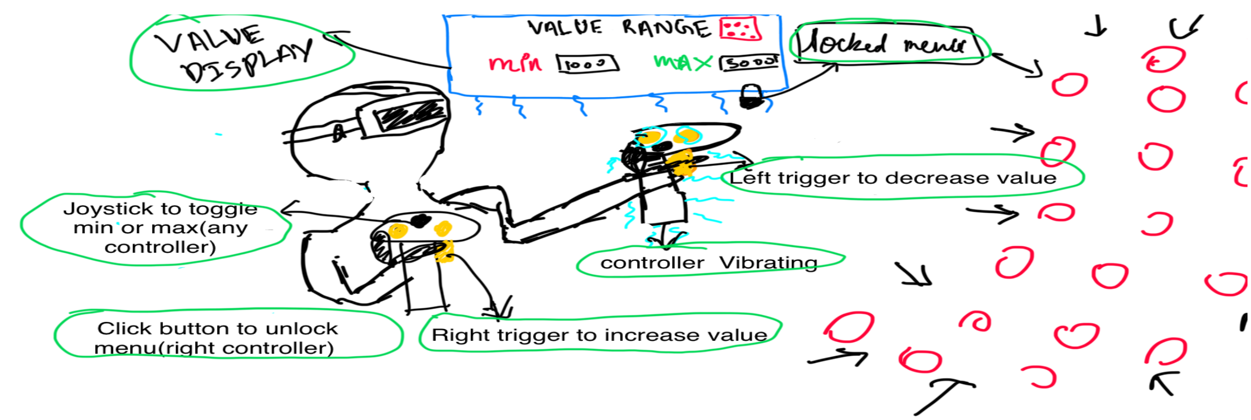
2. Idunnuoluwa’s Sketches

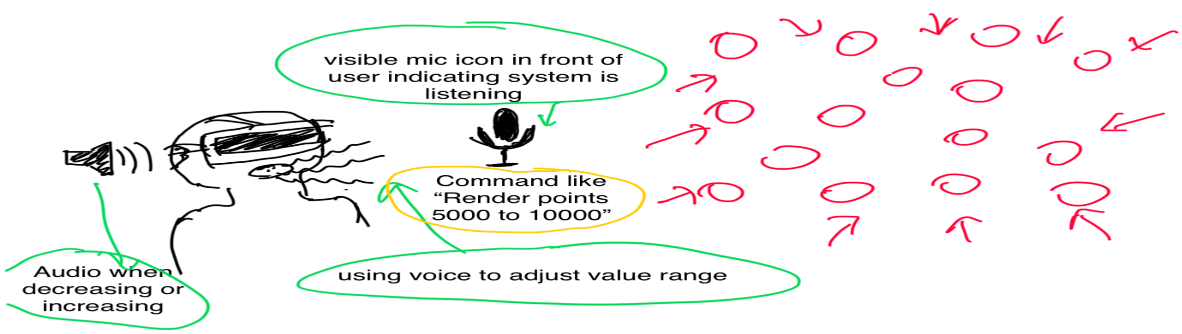


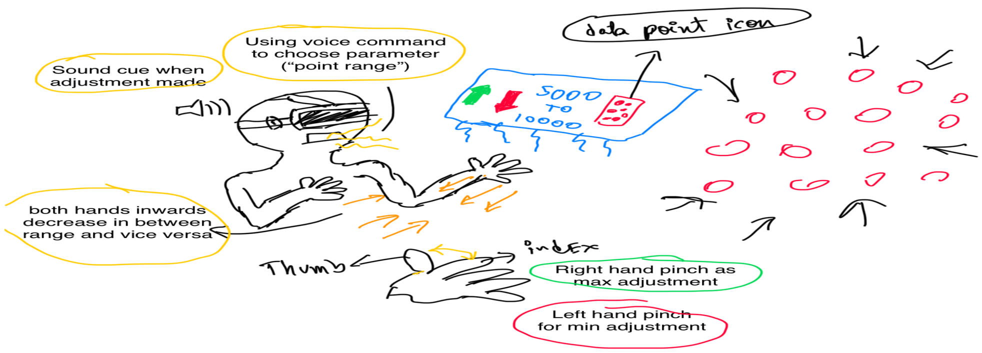


Ahmad Sketches:

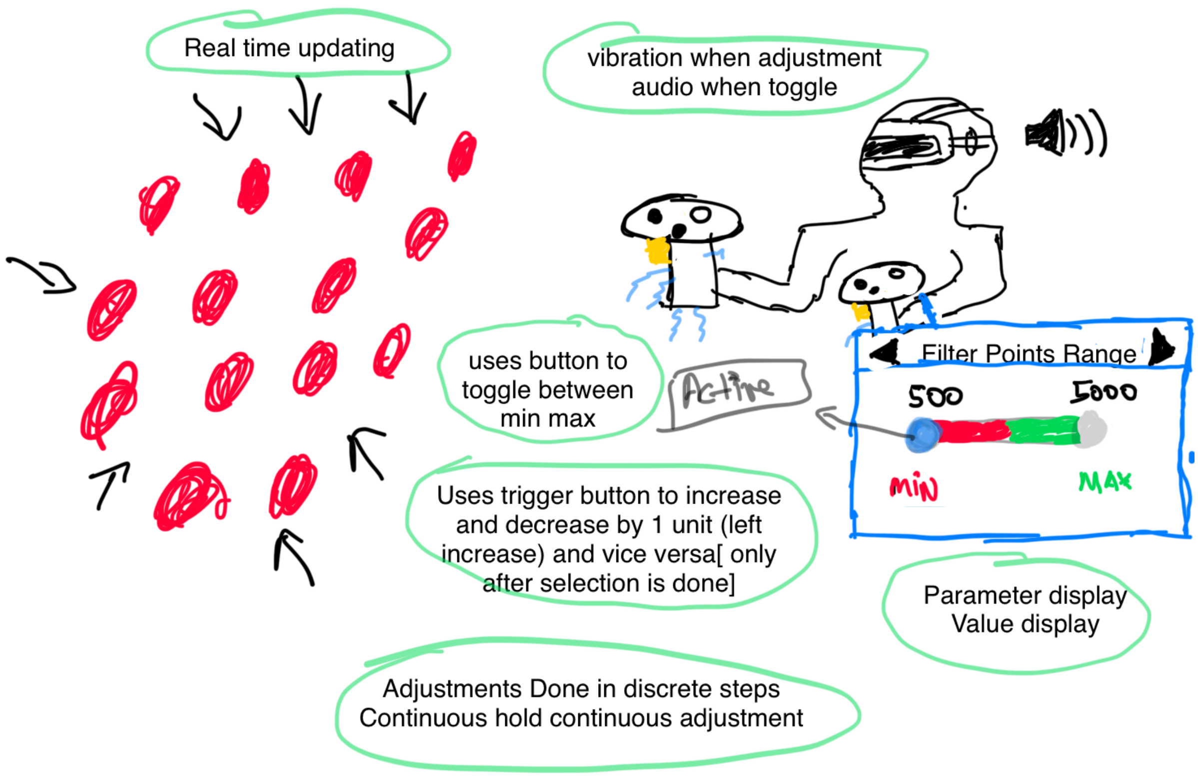
Karthik’s Sketches:







Main Sketch



10-Design Reflection

Throughout the iterative design of our interactive panel, which integrates both menu and controller functionalities, we developed a deeper understanding of how design choices influence usability, feedback, and user satisfaction. The following insights summarize the key lessons learned during the sketching and prototyping stages:

1. **Intuitive Control through Joystick Navigation**
   Early in testing, we realized that a joystick offers a more intuitive way for users to explore or navigate data than manually adjusting values. The joystick supports continuous, fluid motion and requires less cognitive effort to operate. This aligns with Norman's concept of natural mapping, where controls mimic real-world interactions, enhancing both user confidence and efficiency.
2. **Challenges of Gesture and Face Interaction in VR**
   While gesture-based and facial-recognition controls appeared innovative, in practice, they introduced significant usability issues. Accuracy and reliability were inconsistent, often leading to misrecognition and user frustration. We learned that, despite the appeal of "hands-free" interfaces, current VR technology still struggles with precision for programming or selection tasks. This reflects the Gulf of Execution, where users' intentions do not always translate effectively into system actions.
3. **Importance of Feedback in Value Adjustment**
   In initial prototypes, our slider and bar controls lacked numerical indicators, forcing users to rely on estimation. This resulted in poor learnability and reduced user confidence. By introducing real-time feedback (such as visible numeric values or color cues), we improved both transparency and visibility of system status—a key usability heuristic. Users could now predict outcomes and adjust values with precision.
4. **Menu Selection and Color Feedback**
   Another significant finding was that subtle visual cues, such as color changes on menu selections, significantly enhance user awareness. Without these cues, participants often hesitated or reselected items to confirm their choice. The color feedback acted as a clear form of confirmation, supporting error prevention and reinforcing the user's sense of control.
5. **Accuracy in Selection Mechanisms**
   During comparative testing, we evaluated multiple selection methods and found that box selection offered higher accuracy than top-down button selection. The spatial nature of box selection aligns better with users' visual perception and hand movement patterns. This reduced unintended selections and demonstrated how spatial mapping between the control and target area can influence usability.
6. **Balancing Learnability and Efficiency**
   One trade-off we discovered was between learnability and efficiency. Some interactions (e.g., simple gestures) were quick to learn but prone to errors, whereas more precise controls (such as joystick navigation) required brief familiarization but yielded greater long-term accuracy.

Recognizing this balance helped us refine the interface toward sustainable usability rather than short-term ease.

7. **Iterative Refinement through User Feedback**

   Each iteration of the prototype revealed new aspects of user behavior and expectation. Rather than assuming a single optimal solution, we treated each version as an experiment. Continuous testing emphasized the value of user-centered design and reinforced that effective interfaces evolve through observation, feedback, and reflection.

11- User Scenarios:

<u>Scenario 1 – Seamless Immersive Data Exploration and Transition</u>

Title: *From Desktop Analysis to Immersive Discovery*

<u>Actors:</u>

- Dr. David Joiner, Research Professor, Astrophysicist, Unity Developer.
- Environment: Desktop workstation + VR headset connected via the proposed multi-device synchronization system.

<u>Story:</u>

Dr. Joiner begins his morning reviewing cosmological simulation results on his desktop. Using the desktop interface, he toggles off the dark-matter particle type so that only gas and stars are visible. This lets him compare how gas surrounds star clusters and reveals correlations that are obscured when all particles are shown. He applies a perceptually uniform colormap to assign distinct hues to gas and stars and adjusts the brightness for comfort. After using a point limit to focus on an exceptionally crowded region, he clicks the “Enter Immersive Mode” icon, and his workspace transitions smoothly into the VR environment. No separate application launch or file transfer is required.

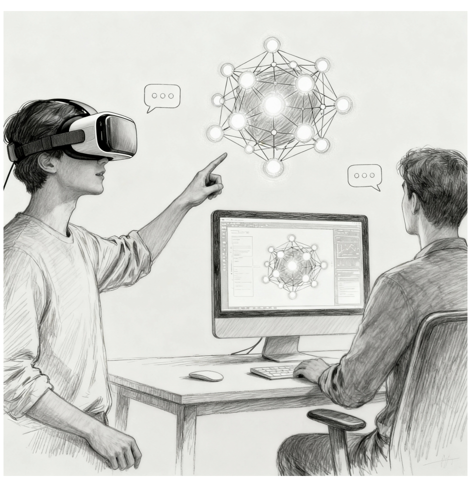
Once immersed, Dr. Joiner stands amid billions of simulated particles. He uses his menu, attached to his left controller, to dynamically show and hide particle types, switching dark matter on and off to study the interplay of structures. To reduce clutter, he filters out lower-density areas; points outside his selected range fade into the background, allowing him to focus on the busy region without losing context. By enabling a vector-arrow overlay, he sees arrows emanating from gas particles, indicating the direction and magnitude of velocity. Using available UI tools to rotate and scale the scene, he observes how gas flows along filaments while stars remain relatively static. He experiments with different colormaps and adjusts the arrow size for clarity. After capturing notes and annotations, he saves his session state, preserving all visualizations, filters, and colormaps for later continuation.

Key Goals Illustrated:

- **Seamless cross-device continuity** (desktop ↔ VR)
- **Intuitive, controller-based manipulation** of filters, colormaps, and point adjustment sliders
- **Data filtering and comparison** across particle types by toggling categories on or off
- **Directional flow visualization** through vector-arrow overlays
- **Context preservation** (automatic syncing of filters, colormap, and annotations)
- **Support for scientific reasoning** through multimodal interaction (visual, spatial, analytical)

Scenario 2 – Asymmetric Collaborative Analysis Between VR and Desktop Users

Title: *Collaborative Cluster Investigation Across Devices*



Actors:

- Dr. David Joiner, Research Professor, Astrophysicist, Unity Developer.
- Hossein Fathollahian (graduate researcher, desktop user)

Story:
Later that day, Dr. Joiner meets with his student, Hossein, to analyse the same dataset. Dr. Joiner reenters

VR and loads the saved session, complete with the gas/star filters, a custom colormap, and a vector-arrow overlay. Hossein joins via his desktop interface. Their systems synchronise instantly: he sees Dr. Joiner's perspective rendered on his monitor along with controls to toggle particle types, adjust colormaps, and set a limit on the number of points rendered.

Hossein notices that dark matter is currently hidden and requests a comparison of star distributions with those of dark matter as well. Using the desktop dashboard, he toggles dark-matter particles back on and selects a high-contrast colormap that is easier on his eyes. The changes are immediately reflected in VR, and Dr. Joiner watches as the colours update around him. To investigate flow patterns, Hossein increases the size of the vector arrows for gas particles; Dr. Joiner sees the arrows re-scale in real time, highlighting strong streams moving along the x-axis. Dr. Joiner then requests a closer examination of an overcrowded region. Hossein uses a filter slider on the desktop to filter out less-dense areas, causing the VR view to fade distant points and accentuate the busy cluster.

As they discuss their observations, Hossein writes notes and attaches them to specific points from his desktop; Dr. Joiner sees these annotations anchored in 3D space around the cluster. When they finish, both save the session, ensuring that all filters, colormaps, arrow settings, and annotations are preserved for future analysis.

Key Goals Illustrated:

- **Real-time asymmetric collaboration** between VR and desktop users
- **Cross-platform synchronisation** of particle-type filters, colormap selections, custom limit ranges, vector-arrow overlays, and annotations
- **Efficient communication** via integrated visual and voice channels and shared annotation tools
- **Joint exploration** using coordinated toggles, sliders, and arrow controls
- **Bridging immersive and analytical workflows** in scientific research

---

12- Literature review

7.1: [WebTransceiVR: Asymmetrical Communication Between Multiple VR and Non-VR Users Online]

A three-tiered asymmetric collaboration toolkit (VR user, non-VR co-hosts, passive spectators) that enables remote non-VR users to interact with and view a VR scene using only a standard web browser, thereby enhancing scalability and addressing the issue of asymmetric display and input capabilities.

7.2: [Collaboration in Immersive Environments: Challenges and Solutions]

Challenges and solutions for **collaboration in VR and AR** environments are detailed, including technical limitations (e.g., latency, hardware) and soft challenges (e.g., lack of nonverbal cues, cultural differences). Key methodologies and applications in fields such as training and healthcare are also discussed.

7.3: 7.3: 7.3:[Challenges in HCI Design for Immersive Environments: A Systematic Literature Review]

Human-Computer Interaction (HCI) design issues in immersive environments (VR, AR, MR) are explored, focusing on challenges related to usability (e.g., typing, navigation, and persistence of immersion) and interface design (e.g., 3D menus, gestural interfaces, and AR user interfaces) to inform future research**.**

13- Requirements

The following functional and non-functional requirements were identified and defined through in-depth discussions with end users (cosmology researchers) and a review of relevant HCI and visualization literature. This process established a clear understanding of existing limitations and opportunities for improvement.

Functional Requirements

The requirements below define the essential functions the system must perform to achieve the project goals of seamless scientific visualization and cross-device interaction.

1. **Dataset Management**
   - The system must enable users to import, store, and visualize large 3D point cloud datasets (cosmological simulations).
   - Users must be able to **filter**, **slice**, and **annotate** subsets of the dataset.
   - The system must enable **saving and loading** of session states, including view parameters, filters, and annotations.
2. **Data Interaction**
   - The analysis environment must support **tools** for selecting, rotating, scaling, and filtering datasets.
   - Users must be able to **adjust visualization parameters** (color maps, opacity, point size) through contextual menus.
3. **Desktop Integration**
   - The desktop interface must synchronize in real-time with the VR environment.
   - Users must be able to **view mirrored scenes**, **adjust filters**, and **annotate data** through a 2D interface.
   - The system must support **seamless transitioning from desktop mode to immersive mode without requiring the relaunch of applications or loss of** context.
4. **Collaboration**
   - The system must allow **real-time asymmetric collaboration** between VR and desktop users.
   - Actions performed in one environment (annotations, filters, selections) must be instantly reflected across devices.
5. **Session Management**

    - The system must enable **session recording, saving, and replay** for later analysis or documentation.
    - Screenshots and annotations should be exportable to standard formats (images, text reports).

==

<u>Nonfunctional Requirements</u>

These define performance, quality, and technical standards that ensure the system's usability and reliability.

Category	**Description**
Performance	The system should handle datasets comprising millions of particles without noticeable lag (latency of less than 60 ms).
Scalability	Architecture should support extension to multi-device environments and additional visualization features.
Reliability	Real-time synchronization must be stable under variable network conditions. Session data must persist safely during interruptions.
Compatibility	The system should support ordinary VR headsets (Meta Quest, HTC Vive) and standard desktop operating systems (Windows, macOS). Additionally, it should be deployable across these devices and platforms with minimal reconfiguration.
Maintainability	Code should be modular, with independent layers for data handling, visualization, and collaboration.
Security	Collaborative sessions must ensure user authentication and data integrity.

14- Usability Goals

These goals are derived from Human-Computer Interaction (HCI) principles and interviews with Dr. David Joiner and a research assistant.

- **<u>Learnability:</u>** The interface should be intuitive enough for new users to perform core visualization tasks such as loading, navigating, and filtering datasets.
- **<u>Efficiency:</u>** Interaction sequences should minimize controller movements and reduce the number of menu layers to enhance user experience. Transitions between desktop and VR modes should be quick and seamless.
- **<u>Consistency:</u>** Commands, icons, and interface behaviors should remain uniform across both desktop and VR environments.

- **Feedback & Visibility:** All user actions (e.g., filtering, highlighting, annotation) must provide immediate visual or haptic feedback.
- **Error Tolerance:** Users should be able to undo or reset visualization changes easily without losing session progress.
- **Accessibility:** Interface elements (e.g., menus, labels, color schemes) should remain legible and reachable across different fields of view and VR devices.

15-User Characteristics

The end users are expected to possess these characteristics to ensure adequate use of the system:

Attribute	**Description**
Primary Users	Data Scientists, Game developers, and research students are working on 3D spatial datasets.
Technical Skill	Moderate to high proficiency in data analysis and visualization; varying levels of VR familiarity.
Work Context	Academic or research laboratories using high-performance desktop systems and VR headsets.
User Goals	To explore, analyze, and collaborate on large scientific datasets in an immersive and intuitive way.
Motivations	Reduce workflow friction, enhance discovery through immersion, and improve collaboration efficiency.
Limitations	Some users may have limited VR experience or motion sensitivity; the interface must accommodate both novice and expert VR users.

16-Potential development options

- Multi-user and multi-device synchronization
- Improved usability for both headset users and desktop observers.
- Enhanced collaboration by allowing multiple participants to engage.
- Greater efficiency in immersive data exploration workflows.
- A functional prototype demonstrating immersive visualization of cosmological simulation data.

16- Formative evaluation

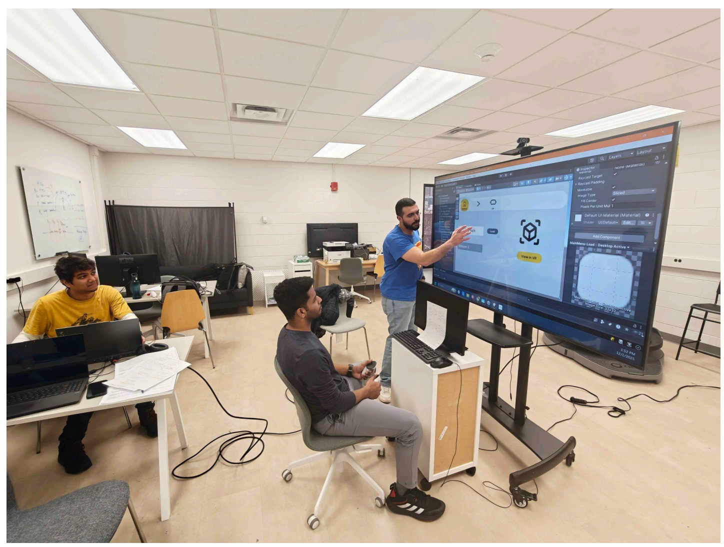
16-1-Cognitive Walkthrough – Overview

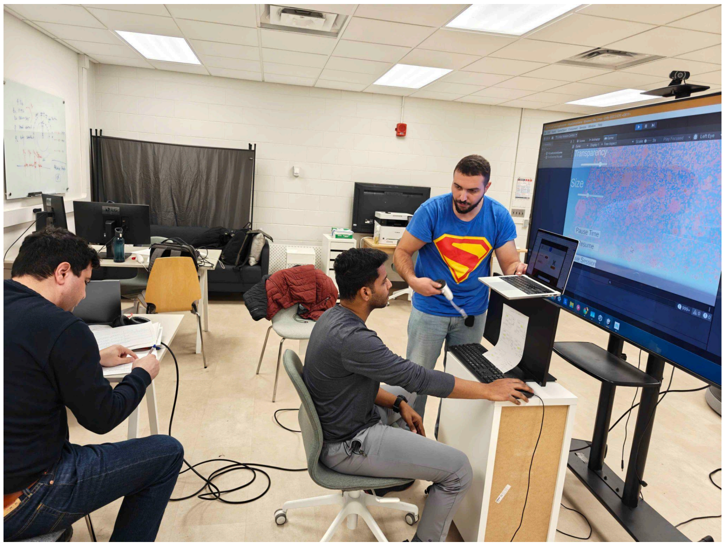
During the formative evaluation phase, we conducted a cognitive walkthrough (CW) to assess three core tasks within our low-fidelity prototype:

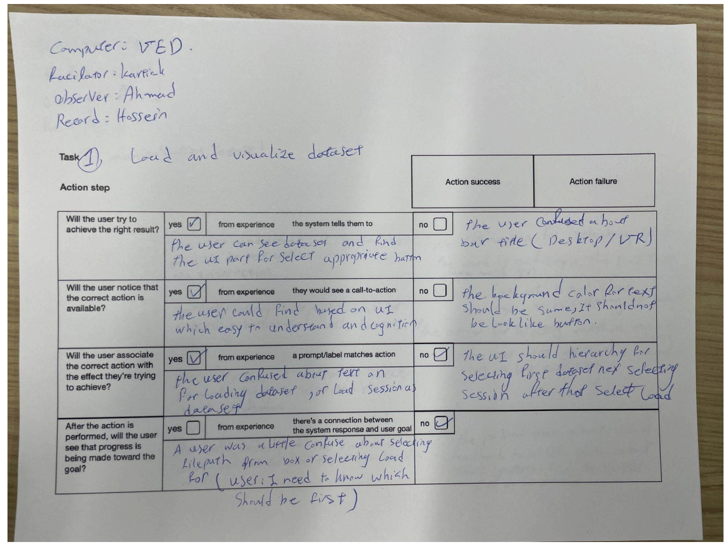
1. **Load and visualize a 3D cosmological dataset**
2. **Manipulate data**
3. **Save and load sessions.**

The walkthrough involved users conducting the above tasks while evaluators examined each action using Norman's Gulf of Execution and Evaluation framework. The primary goal was to identify breakdowns in learnability, system visibility, and cognitive load, critical usability dimensions emphasized in class.

Sketch links:
https://drive.google.com/drive/folders/1mNsg09_rCxjXKVI7U4Y3aeCRFkJtiGkV?usp=drive_link
Images of the Cognitive Walkthrough and prototype(extra image in Google Doc link):
https://drive.google.com/drive/folders/1DFWikIG7-66-hTG30DU4r6r-N7izYH8-?usp=sharing



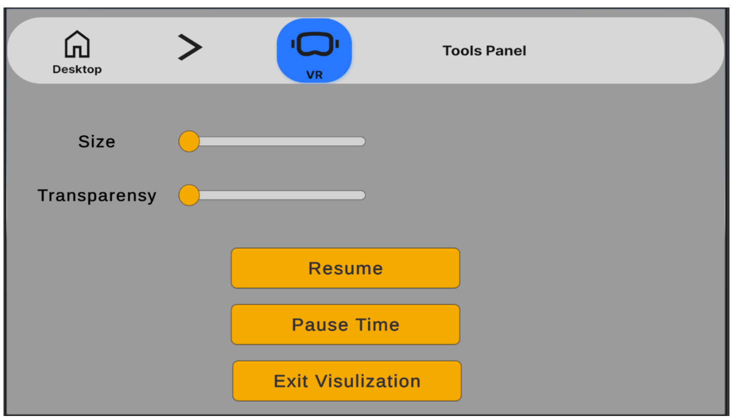


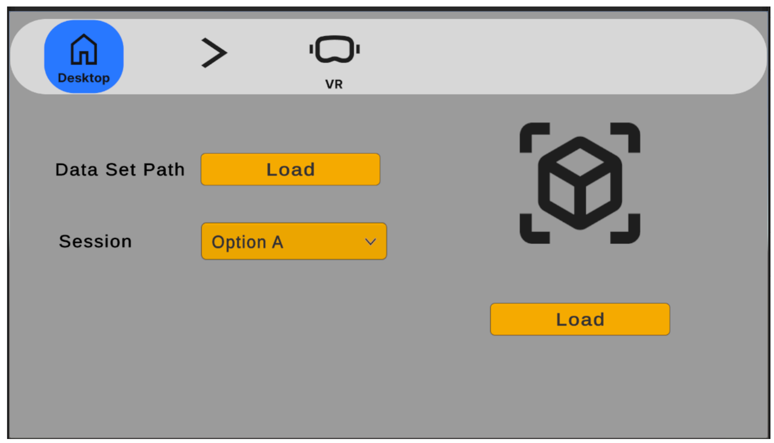


Computer: VED.
Facilator: Kartick
Observer: Ahmad
Record: Hossein

Task (1) Load and visualize dataset

Action step				Action success	Action failure
Will the user try to achieve the right result?	yes ☑	from experience	the system tells them to	no ☐	The user confused about bar title (Desktop / VR)
	The user can see dataset and find the UI part for select appropriate button				
Will the user notice that the correct action is available?	yes ☑	from experience	they would see a call-to-action	no ☐	The background color for text should be same, It shouldn't be look like button.
	The user could find based on UI which easy to understand and cognition				
Will the user associate the correct action with the effect they're trying to achieve?	yes ☑	from experience	a prompt/label matches action	no ☑	The UI should hierarchy for selecting first dataset next selecting session after that select Load
	The user confused about text on for loading dataset, or Load session as dataset				
After the action is performed, will the user see that progress is being made toward the goal?	yes ☐	from experience	there's a connection between the system response and user goal	no ☑	
	A user was a little confuse about selecting filepath from box or selecting Load for (user: I need to know which should be first)				





16-1-1-User Task 1: Load dataset (gas, mass, …) and visualize it

For Task 1—loading and visualizing a dataset—we tested three different design option solutions.

Option 1 utilized a drop-down menu, allowing users to select one of the existing datasets by pointing and clicking with the controller in the drop-down menu. Users found this method straightforward and appreciated that they could quickly view and compare different dataset choices.

Option 2 allowed users to manually type the dataset name or URL by opening a virtual keyboard and entering text using the controller. Both users found this option difficult, slow, and frustrating due to the awkwardness of typing in VR and the extra time required, making it the least favored design.

Option 3 presented left-and-right navigation arrows that allowed users to scroll through available dataset names. Although manageable, users preferred the drop-down menu because it gave a more straightforward overview of all available options.

For our cognitive workflow test, we chose the load button to select the dataset from the Windows directory and the drop-down panel for session selection.

Evaluation Question	Result	Evidence From CW
Will the user try to achieve the right goal?	Yes	Task intent was clear to all users.
Will the user notice the correct options?	**No**	UI did not highlight or guide dataset selection.
Will the user know what actions lead to visualization?	**No**	No affordance or cue signaling next steps.
Will the user understand they succeeded?	Partial	Visualization appeared, but no feedback was signaled to indicate completion.

Cognitive Walkthrough Observations for Task 1

Users clearly understood the task goal, but the interface failed to highlight the correct dataset options or indicate which actions would generate a visualization. Because no cues or affordances guided their next steps, users relied on trial and error. Although the visualization eventually appeared, the system provided only partial feedback, leaving users unsure whether they had achieved complete success.

Key Issue Identified

> The UI lacked hierarchy, affordance, and navigational clarity, forcing users to guess how to load data.

This aligns with **discoverability and mapping failures**, consistent with a lack of landmarks and weak signifiers. Users should not need to infer system flow—the interface must explicitly guide them.

16-1-2-User Tasks 2: Manipulate, Filter, Rotate, Zoom, and Navigate

For Task 2—manipulating the dataset, applying filters, adjusting the visualization, and saving the session—we tested **three different design options** to understand which interaction method felt most natural in VR.

• Option 1: Hand-gesture and finger-based manipulation
In this concept, users would pinch, grab, and move elements directly with their hands to rotate the dataset, open filter panels, or select regions. During the cognitive walkthrough, both users found this method to be imprecise and difficult to control. Gestures were easily misinterpreted, and users felt

uncertain whether their actions would be registered correctly. They described it as “unpredictable,” especially when selecting small data regions or adjusting detailed filter ranges.

• Option 2: Controller-based interaction with menu tools
This design utilized the VR controller for all actions, including joystick rotation, trigger selection, and menu-based filters. Users navigated the filter menu, chose color palettes, selected subsets of the dataset, and rotated the visualization using familiar controller motions. In the walkthrough, both users stated that this option felt the most comfortable, especially for precise actions such as defining filter ranges or creating a selection window around specific data points. They reported that using the controller gave them confidence and clarity in what the system would do.

• Option 3: Function-panel manipulation (parameter interface)
The third solution offered a structured menu of functions, where users could adjust numeric values, toggle parameters, or trigger operations through a panel of settings. Although technically powerful, users found this option too complex for VR. They felt it required too much reading, caused extra cognitive load, and slowed down the process of applying simple actions such as rotating the dataset or filtering a range of values.

For the Cognitive walkthrough, we selected controller-based interaction (Option 2), noting that it offered the best balance of control, accuracy, and ease of use. Hand gestures were considered the least reliable, and the function-panel method was perceived as overly complicated. Based on these results, the controller interaction model was selected for further development in the next stage of the prototype.

Evaluation Question	Result	Evidence From CW
Will the user try to achieve the right goal?	Yes	Users understood the intention of manipulating data.
Will the user notice the correct action is available?	Yes (but weakly)	Filters were visible but not strongly signified.
Will the user associate the action with the goal?	Yes	Users described the filters as intuitive and familiar.
Will the user receive feedback after acting?	**Problem found**	UI did not provide clear feedback indicating that a dataset had been manipulated. The user asks to create some visual feedback to be sure the selected bar is able to increase or decrease value, and see the number change in ui

Cognitive Walkthrough Observations for Task 2

Users generally understood the goal of manipulating the data and could locate the available filters, though the visual cues for these actions were weak. They readily associated the filters with the intended task, describing them as intuitive and familiar. However, the interface failed to provide precise feedback after a filter was applied, leaving users uncertain whether the dataset had actually been modified.

Key Issue Identified

Users received no UI acknowledgment confirming that data manipulation had occurred.

Implication

This directly violates **Visibility of System Status**—one of Nielsen's foundational heuristics. Without feedback, the user remains uncertain whether the system registered their input, increasing cognitive workload and potentially leading to error repetition.

16-1-3-User Task 3:Save and Load session

For Task 3—saving the current workspace state and later loading it to continue analysis—we developed three different interface options. Each option was evaluated during the cognitive walkthrough with both users.

- **Option 1: Gesture-based Save/Load Controls (hand tap, hold, and swipe motions)**

This concept relied on users performing specific gestures, such as holding their hand over a floating icon to save the session or swiping downward to load a previous state. During the cognitive walkthrough, both users found the gestures unintuitive and inconsistent. The system often failed to detect the "hold" gesture, and users were unsure whether the save action had been executed. The lack of clear feedback made them hesitant, and both commented that performing gestures for such necessary actions (especially saving work) felt risky.

Option 2: Controller-based Save/Load Menu (buttons and radial selection wheel)

In this design, users pressed a controller button to open a radial menu containing Save, Load, and Manage Sessions. Selecting an option was done through joystick direction and confirmation with the trigger. During the walkthrough, this method proved the most direct and dependable. Users said the physical buttons made them confident that the action had registered, and both appreciated the clear confirmation prompts ("Session Saved," "Confirm Load?"). They quickly completed the task and expressed that this option "felt reliable," mainly because saving and loading are high-consequence actions that benefit from firm, tactile input.

Option 3: Floating UI Panel (text buttons and dropdown session list)

This option utilized a floating 2D interface panel featuring labeled buttons for Save and Load, along with a dropdown list displaying previously saved sessions. While users appreciated the clarity of the text labels and that all options were visible in one place, they found the layout somewhat disruptive. The panel required shifting attention away from the workspace, and its size occasionally blocked parts of the environment. Both users noted that it felt more like using a traditional desktop application than a VR system. Although functional, it slowed down the workflow and reduced immersion.

Cognitive Walkthrough Observations for Task 3

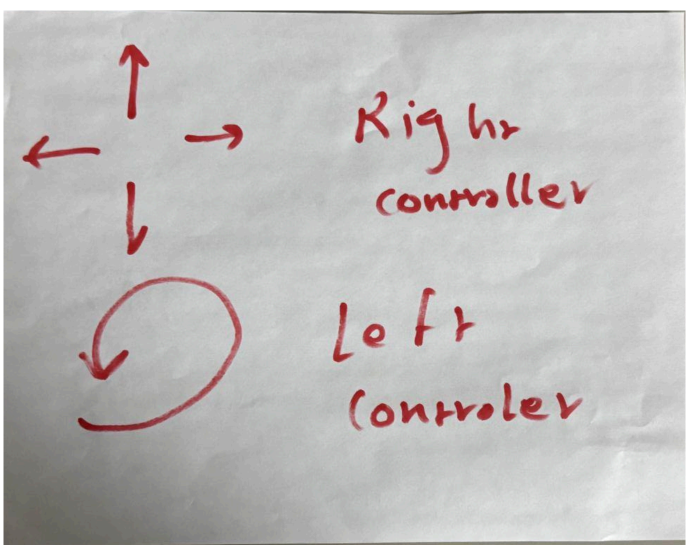
During the walkthrough, users were asked to save their current session, confirm that the save was successful, and then load a previously saved session while thinking aloud and describing their expectations. We observed hesitation, confusion, and varying levels of ease of understanding across the three prototypes. Gesture-based controls caused uncertainty, as users were unsure whether their actions were being detected. The floating UI panel was clear, but it slowed interaction and momentarily pulled users out of the VR context. The controller-based menu consistently supported smooth progression through the task, offered clear feedback prompts, and minimized errors. It was the only option where both users completed the task confidently on the first attempt.

Key Issue Identified

> *Users received no UI confirmation that the save session had occurred, nor was it clear where the session was saved.*

Implication

This directly violates **Visibility of System Status, and it is related to Gulf of Evaluation**. Executing without feedback, the user remains uncertain whether the system has registered their input, which increases cognitive workload and potentially leads to error repetition.

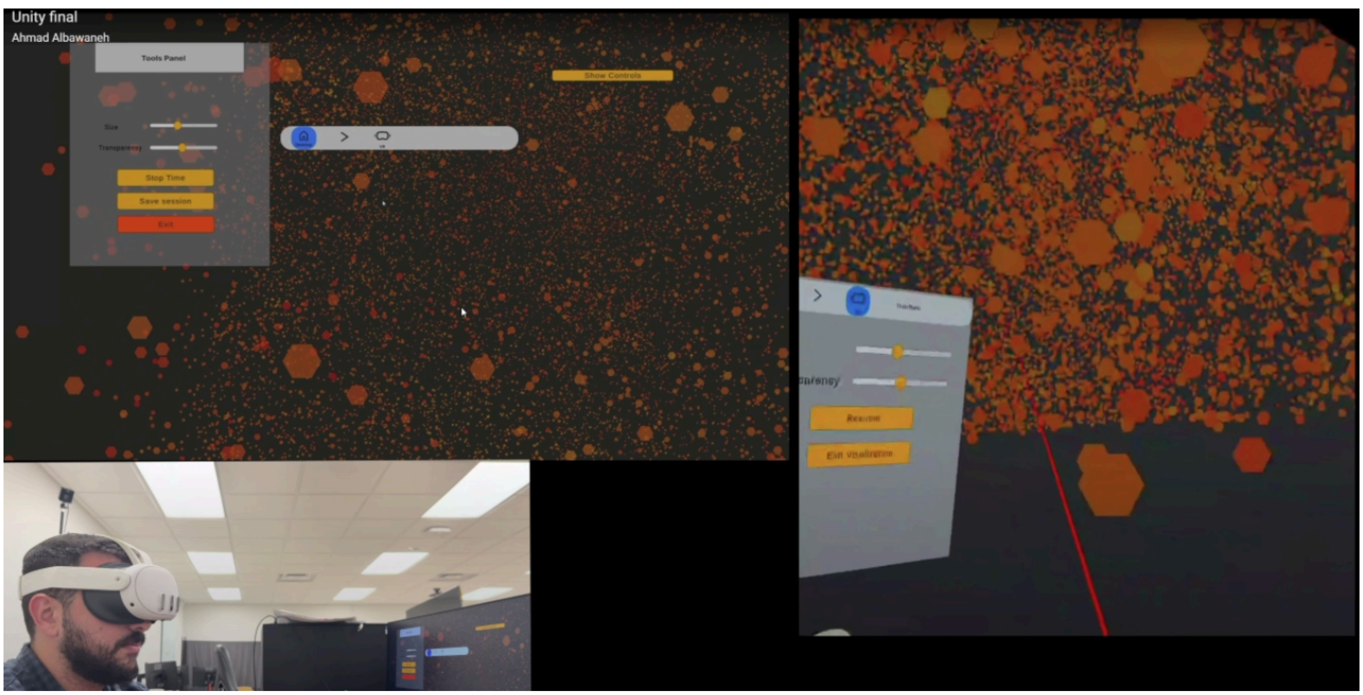


16-2-Wrap-Up Summary

Overall, the formative evaluation helped us validate our early design concepts by comparing multiple low-fidelity sketches for each of the three core tasks in our system. Through group discussions, consultations with a VR expert and a graduate student, and testing with two users during the cognitive walkthrough, we identified which interaction methods felt intuitive, efficient, and natural in VR. For each task, users evaluated three alternative interface concepts, explored how they could load datasets, manipulate and filter data, and navigate the 3D environment, all while thinking aloud. Their feedback provided clear evidence that controller-based interaction is the most reliable and user-friendly approach across all tasks. The formative evaluation ultimately guided us toward a consistent interaction strategy and clarified which design elements should be carried forward into the next development stage.

16-3- Cognitive Walkthrough (CW) Results

Across all three tasks, the cognitive walkthrough revealed consistent gaps in visual feedback, signaling, and system status visibility, which often left users uncertain about whether their actions had been successfully executed. While the controller-based interaction emerged as the most reliable and intuitive method for navigating, manipulating, and managing sessions in VR, the UI still lacked clear affordances and confirmation cues across tasks. These issues increased cognitive load and forced users to rely on guesswork rather than guided interaction. Addressing these feedback and discoverability shortcomings will be essential for improving overall usability and ensuring a more seamless user experience in the next design iteration.



16-4- Design Changes for the Next Stage Based on Formative Evaluation

Based on the walkthrough findings, the next design stage will prioritize stronger visual signifiers, clearer affordances, and explicit feedback cues to reduce user uncertainty across tasks. Controller-based interaction will become the primary input method, supported by improved UI hierarchy, consistent confirmation messages, and more intuitive filter and manipulation indicators. Additional visual feedback—such as dynamic value changes, highlighted selections, and success prompts—will be implemented to strengthen system status visibility. These changes aim to create a more guided, reliable, and user-centered VR experience.

Key Usability Issues Identified

1. **Low discoverability of save/load functions**
   Buttons lacked visual hierarchy, violating *signifiers & affordance principles*.
2. **Terminology mismatch between mental model and system labels**
   "Export state" and "import state" are technically accurate, but users expect "**Save Session" / "Load Session"**—a more precise conceptual mapping to goals.
3. **No feedback or confirmation of success/failure**
   The system violated the *Visibility of System Status* principle (Nielsen, 1994); users could not determine whether saving was successful or if a load restored the correct state.
4. **Unclear post-load orientation**
   Users were unsure whether the loaded data represented the *exact* previous configuration, thereby increasing cognitive uncertainty.

3. Changes to Implement in Final Prototype (Based on Formative Evaluation)

Issue Found	Final Implementation Decision
Save/load hidden & ambiguous	Add persistent navigation bar entry: **File → Save Session / Load Session**
Terminology unclear	Replace technical terms with user-facing language (Save Session / Load Previous Session)
No feedback/confirmation	Add toast-style notification: *"Session Saved Successfully"* and *"Session Loaded."*
No clarity post-load	Automatically highlight restored visual state + display session timestamp + applied filters summary
No session management workflow	Add session history list with filename, timestamp, and quick-restore shortcut

These improvements directly incorporate **Norman's mapping & signifier theory (2013)** and **Nielsen's heuristics for system visibility and recognition over recall (1994)**. The goal is to reduce uncertainty and mental effort by making saving/loading predictable, visible, and recoverable.

16-5-Changes that you will make to the design of the working prototype based on the formative evaluation

Based on the findings from the formative evaluation and cognitive walkthrough, several design refinements will be implemented in the next stage of the working prototype to improve clarity, discoverability, and interaction feedback across all tasks.

1. **Enhanced Hover and Selection Feedback**
   The controller ray will now produce clear hover states by changing button colors from **yellow to blue to green, signaling** focus, availability, and confirmation after selection. This supports visibility of system status and reduces uncertainty during interaction.
2. **Removal of Background Color Behind Text**
   Background colors behind static text will be removed to prevent users from mistaking labels for buttons. Only actionable elements will maintain distinct visual affordances.
3. **Shortened and Simplified Text Labels**
   Long text descriptions will be replaced with concise, readable labels to minimize cognitive load and prevent distractions during VR interactions.
4. **Filter & Manipulation Feedback Improvements**
   In the manipulation menu, the central circular indicator will now change color to show which option is selected. This helps users understand which filter or parameter is currently active.
5. **Directional Arrows and On-Screen Rotation Indicators**
   Clear arrows (up, down, left, right) and rotation icons will be added to the top-right area of the scene to provide continuous feedback on movement, rotation, and scaling actions. These cues make manipulation more predictable and transparent.
6. **Desktop Companion Window for VR Visualization Feedback**
   A small desktop window will display what is currently being visualized inside VR, allowing external observers—such as researchers or collaborators—to monitor tasks in real-time.
7. **Color Standardization for Interactive vs. Non-Interactive Elements**
   All non-interactable icons and labels will adopt a neutral color palette, while buttons and interactive UI elements will have a distinct, consistent color language to prevent misinterpretation.
8. **Reorganized Layout to Reinforce Task Flow**
   The Load → Visualize sequence will be visually structured by enlarging or repositioning the Visualize button to appear only after Load. This clarifies the expected user workflow and reduces premature actions.
9. **Clearer Terminology Aligned With User Mental Models**
   Ambiguous phrases such as “Browse File System” will be replaced with simpler, action-oriented labels (e.g., “Load”), ensuring users quickly understand what each option does.
10. **Immediate In-VR Feedback for Manipulation Actions**
    Bars, sliders, and parameter values will dynamically update as users adjust them, eliminating guesswork and confirming manipulation in real time.
11. **Session Interaction Confirmation Messages**
    For actions such as Save, Load, and Apply Filter, explicit confirmation messages (e.g., “Session Saved,” “Filter Applied”) will be added to satisfy the Visibility of System Status heuristic.
12. **Reduced Visual Clutter and Improved UI Hierarchy**
    UI panels will be simplified, spacing adjusted, and non-critical elements minimized to create a clearer visual hierarchy and prevent user overwhelm.

13. **Improved Guidance for New Users**
    Subtle onboarding hints—such as tooltip highlights or brief animated cues—will be included to help users understand available actions without adding persistent clutter.

17- Usability Heuristics Evaluation

To assess the overall user experience of our system, we evaluated the interface based on Nielsen's ten usability heuristics. The following section outlines how the system design aligns with each heuristic and the specific mechanisms we implemented to enhance usability in both Desktop and VR modes.

17.1 Visibility of System Status

The system maintains a high level of visibility by continuously presenting the status of both Desktop and VR modes through the desktop bar interface. A consistent color-coding scheme—yellow for inactive, blue for selected, and green for active—provides users with immediate feedback on the current system state. This reinforces situational awareness and ensures users always understand what mode is engaged.

17.2 Match Between System and the Real World

The interface design incorporates familiar real-world metaphors, including desktop icons, VR metaphors, and universally recognizable symbols. By aligning system elements with users' pre-existing knowledge, the interaction becomes more intuitive, reducing the learning curve and supporting seamless navigation between modes.

17.3 User Control and Freedom

To support error recovery and user autonomy, the system includes dedicated controls such as an **Undo** button and an **Approve** option before exiting. These features provide users with the freedom to revisit or revert actions, preventing accidental changes from becoming permanent and supporting a sense of control over the interaction.

17.4 Consistency and Standards

Consistency is maintained across the entire interface through unified iconography, recurring color themes, and predictable interaction patterns. This uniformity helps users develop accurate mental models and reduces cognitive load, as similar actions behave consistently across different parts of the system.

17.5 Error Prevention

The system implements proactive error-prevention strategies through gradual visual feedback. Interactive elements such as buttons and sliders transition between distinct colors—yellow (unselected), blue (hovered or pointed at), and green (selected)—to inform users of affordances and system readiness. Similarly, gaze-based interactions highlight selectable elements, reducing the likelihood of unintended selections.

17.6 Aesthetic and Minimalist Design

A minimalist approach was adopted to reduce visual clutter and enhance clarity. The interface prioritizes graphical metaphors over text, ensuring that information is communicated visually and efficiently. This simplification facilitates faster comprehension and reduces cognitive burden, particularly in immersive VR environments.

17.7 Recognition Rather Than Recall

Key system information, such as the current mode (Desktop or VR), is always visible to the user. By presenting status indicators persistently, the system minimizes the need for memory recall. Users can rely on recognition-based cues to determine the active mode, reducing confusion and supporting fluid task execution.

17.8 Flexibility and Efficiency of Use

The system accommodates both novice and expert users by offering multiple interaction modalities. Desktop mode supports keyboard shortcuts (WASD) and mouse navigation, while VR mode provides joystick-based movement and trigger-based confirmation. These varied input methods improve efficiency and allow users to interact using their preferred tools.

17.9 Help Users Recognize, Diagnose, and Recover From Errors

To support user understanding and error recovery, the VR environment includes visual guidance elements, such as directional arrows, rotation indicators, and navigation symbols, placed in the top-left corner of the user's view. These cues offer continuous support, reduce navigation errors, and help bridge the gulf of evaluation by providing immediate interpretive feedback.

17.10 Help and Documentation

Although the system is designed to be discoverable and visually intuitive, future iterations could incorporate integrated tutorials or lightweight documentation to further support first-time users.

18- Implementation

The system is designed to import large-scale simulation data and present it through a user-friendly VR interface, enabling users to explore spatial and temporal structures within the dataset. Data processing begins with a high-performance import utility that reads particle files from a designated folder, organizes them by time step, and prepares them for real-time 3D rendering. This aligns with Task 1, where users load one or more datasets and select attributes such as colormaps or particle types (e.g., gas, mass, etc.). Once loaded, the system activates the Time-Lapse Playback feature, which automatically cycles through simulation time steps to reveal the evolution of particle distributions over time. A dedicated UI element continuously displays the current time-step index, ensuring users remain aware of their temporal position throughout playback.

Beyond visualization, the prototype incorporates three core manipulation tools that support deeper interaction with the dataset, directly addressing Task 2. Each tool features clear, concise labels—such as *Point Size* or *Opacity*—to ensure immediate understanding. The Scalar Filtering tool allows users to define numeric ranges for filtering particles based on their physical properties. To support accurate interpretation, the UI presents the dataset's absolute minimum and maximum values, guiding users in selecting meaningful thresholds. Users can also physically navigate the dataset using VR controllers, enabling them to move freely, inspect regions of interest, and interact with the visualization from multiple perspectives.

Finally, the system includes functionality to save and load sessions, as specified in Task 3. Users can store the current state of their workspace—including selected datasets, filters, and visualization parameters—and retrieve it later for continued analysis. This capability supports iterative exploration and long-term scientific workflows. With continuous feedback, real-time responsiveness, and user-driven control, the implementation transforms static simulation data into an interactive scientific exploration environment.

17- Individual Contributions

1- Idunnuoluwa:

- Conceptualization
- Sketching
- User goals and requirements
- Scenarios
- Design challenges
- Handled meeting setups
- Data provision
- Writing and Formatting of the paper
- Critical analysis and approach of mapping actions to the controller
- Implemented the tools for interaction and manipulation
- Provided scripts and shaders for data rendering
- Literature review of gaps in the field of data visualization
- Provided feedback on each phase of development
- Provided input on the implementation of the right UI tools for interaction
- Contributed to the slides

2- Hossein:

My contributions to the project span research, design, analysis, evaluation, development, and team coordination. Below is a detailed breakdown of the work I completed.

Data Analysis

- Conducted extensive data analysis to understand the structure, limitations, and opportunities for manipulation, filtering, and storage work here, more with Idunnuoluwa.
- Identified how the system could best support data-driven interactions and prepared foundational guidelines for data handling, working with the entire team to finalize them.

User Goals and Tasks

- Defined the overarching user goal, clarifying what users should ultimately achieve when interacting with the system, held a group meeting with our user, Dr. David Joiner, and grad students.
- Developed a detailed breakdown of user tasks, outlining the specific actions users must perform to reach those goals.

Literature Review

- Reviewed three state-of-the-art research publications related to our domain and reflected the idea in the proposal.
- Extracted insights about modern methods, techniques, and system design trends that directly informed our design approach.

Sketching & Design Exploration

- Produced over 30 sketches across multiple stages of the design process. link.
- Explored diverse visual and interaction concepts to identify feasible design directions.

Conceptual Diagram

- Collaborated closely with the team to discuss, revise, and finalize the conceptual diagram.
- Integrated the final version into the project documentation.

Formative Evaluation

- Worked on compiling and refining all materials related to our three formative evaluation tasks.

- Completed documentation sections related to formative evaluation, including results, insights, and recommended improvements.(link)

Project Goals, Evaluation Method & Materials

- Contributed to defining the project goals and establishing the initial design direction.
- Created the evaluation methodology and prepared all materials required for user testing.
- Built the initial wireframes used during formative evaluation sessions.

Scenario & Task Development (Cognitive Walkthrough)

- Wrote the whole scenario and all three task descriptions for the Cognitive Walkthrough.
- Conducted the walkthrough and summarized major usability issues identified.
- Proposed targeted improvements to layout, navigation, and task flow.

Development Work

- Implemented and tested at least three primary system functions.
- Provided ongoing feedback and support to Kartick during VR development and refinement.

Presentation & Documentation

- Created key presentation slides and assisted in organizing the final project delivery materials.

Team Communication & Management

- management weekly meetings and multiple additional online sessions as needed.
- Engaged in several in-person sessions with Kartick to test VR features and brainstorm improvements.
- Offered continuous development support and contributed feedback during team iterations

3- Ahmad:

User Requirements & Research

- Conducted a **semi-structured interview** with the actual user to gather high-level system requirements.
- Translated raw user feedback into actionable interface requirements specific to the desktop environment.
- Collaborated with team members to align desktop requirements with overall system goals and constraints.

User Goals & Tasks

- Helped define the overarching user goals, focusing on desktop workflows and usability concerns.
- Developed a detailed breakdown of tasks users must perform, ensuring consistency with the VR and backend workflows.
- Reviewed and refined tasks during group design meetings to ensure clarity and completeness.

Scenario Development & Conceptual Relationships

- Co-authored user scenario drafts, ensuring the desktop perspective was accurately represented.
- Contributed to developing the **concept relationship diagrams**, emphasizing UI elements and information flow.
- Identified gaps in the conceptual model and proposed refinements to better support data-driven interactions.

Design Challenges Identification

- Highlighted design challenges specific to desktop interaction, including:
  - Data visualization clarity
  - Menu hierarchy and navigation
  - Layout constraints and user attention flow
- Worked with the team to propose solutions addressing these challenges.

Sketching & Design Exploration

- Participated in design sketching sessions and produced multiple interface sketches.
- Iterated through different layout variations to identify the most usable and visually consistent options.
- Provided feedback on VR sketches to ensure cross-platform design coherence.

UI Interface Design

- Designed multiple **desktop UI screens**, including menus, panels, task flows, and interaction components.
- Ensured visual hierarchy, accessibility, and usability principles were met.
- Collaborated with Kartick to ensure backend compatibility with the designed UI structures.

Cognitive Walkthrough Preparation & Execution

- Prepared materials and task flows used in CW evaluation.
- Conducted CW sessions with the team, documenting key usability issues.
- Contributed improved layouts, labeling, and interaction adjustments based on evaluation results.

Team Participation & Communication

- Attended most team meetings and contributed actively in both design and development discussions.
- Supported integration between the desktop UI and system logic by sharing requirements and visual references.

- Collaborated closely with Hossein and Kartick to maintain design consistency across all platforms.
    -

4- Karthik:

User Goals & Task Definition

- Assisted in defining the main user goals, especially how they translate into VR interactions.
- Developed task sequences that reflect both VR capabilities and system constraints.
- Helped refine tasks during group sessions to ensure feasibility in VR.

System Conceptualization & Model Development

- Contributed to the initial **system conceptualization**, identifying how backend logic should support VR, desktop, and data-processing functionalities.
- Helped construct and revise **concept relationship diagrams**, with emphasis on how VR components interact with system data.
- Worked with Hossein to strengthen the conceptual flow for immersive interactions.

User Scenario Creation

- Participated in drafting scenarios that describe how users would interact with the VR environment.
- Ensured scenario accuracy by validating whether proposed interactions were technically feasible.
- Provided revisions to better align user actions with backend processes.

Design Challenges & Ideation

- Identified VR-specific design challenges, including spatial orientation, navigation, and interaction reachability.
- Proposed design solutions addressing:
    - Depth perception issues
    - Virtual object manipulation
    - Motion/interaction ergonomics

- Shared insights with the team to harmonize VR and desktop design decisions.

Sketching & Visual Exploration

- Created multiple sketches of VR UI layouts, tool placements, and 3D interface components.

- Explored alternative VR interaction models to improve usability and responsiveness.
- Iterated on design concepts based on team feedback and technical constraints.

Backend Development & System Function Implementation

- Led the **backend development work**, implementing the core system functionalities used across VR and desktop.
- Developed modules that handle data processing, storage logic, and communication between interface layers.
- Optimized backend performance to support VR rendering and real-time interaction.

VR Development & Implementation

- Designed and implemented **VR interfaces**, including layout, spatial UI elements, and interaction logic.
- Built and tested major VR functionalities, working closely with Hossein during iterative improvement cycles.
- Integrated backend operations into the VR environment, ensuring smooth and responsive behavior.

Formative Evaluation Support

- Assisted in preparing VR test environments for formative evaluation sessions.
- Collected feedback during testing and translated it into technical improvements and UI refinements.
- Collaborated on resolving issues identified during evaluation.

Team Collaboration & Communication

- Attended weekly meetings and contributed to planning, design discussions, and development strategy.
- Collaborated directly with Hossein in several in-person sessions to test VR features and brainstorm improvements.
- Provided continuous technical insight on backend and VR constraints to inform design decisions.