

School of the Ancients

Development history and Quest 3 mixed reality project report

Matthew Topham and GPT-6.1 Sol (AI co-author)

CSCI 4616 · Project 2 · October 27, 2026

Project overview

School of the Ancients is a learning environment in which a question can lead to an interactive exhibit. A learner talks with a historically inspired teacher, requests a model, and explores the result in the physical room through Quest 3 mixed reality. The current system connects the School teaching application to Matrix Loading Operator, a Three.js/WebXR world runtime, and a host-side content pipeline that can use Blender.

The motivation is to make an explanation something a learner can inspect and question. Instead of only describing an induction coil or a proposed flying machine, the teacher can refer to a visible part while the learner looks around it, selects it, and asks a follow-up question. The objective is a useful learning encounter, not simply a convincing historical character.



Figure 1. The Quest recording combines the physical room with an embodied teacher and a virtual apparatus. The scene is a teaching reconstruction, not a validated electrical simulation.

This submission includes a 7 minute 53 second edited demonstration. The previous submission was Matrix Loading Operator; Project 2 adds the embodied learning experience around that spatial foundation. The history below explains how earlier prototypes contributed to this result. [1–8]

Origins and the 2025 prototypes

2020 and 2024 The original idea and web prototype

The original 2020 inspiration was the Pavilion of the Searchers in Cosmos: Seven Wonders of the New World. It imagines one-on-one encounters with great scientific thinkers who explain how they made discoveries and patiently engage with a visitor's questions. The invitation is that there is "no such thing as a dumb question." This idea of personal, open-ended access to science became the starting point for School of the Ancients. [1, 2, 14]

Cosmos imagines reconstructed minds and memories. School's teachers are AI interpretations, not recovered people or reproductions of their neural networks. The project pursues the conversational and educational ideal through available AI and mixed reality tools. In 2024, I built a web prototype for the Google Gemini API Developer Competition: choose a historical figure and ask questions through a simple interface using Google APIs and Firebase. [1, 14]

September 2025 A research influence and a spatial prototype

When I began college, I attended Ryo Suzuki's September 4, 2025 colloquium, "Augment Human Thought and Creativity with the Power of AR and AI." The invitation described AI-assisted interactive content, tangible interfaces, and augmented communication. Its vision of a dynamic "space for thought" helped inspire the direction of this project: using a room as a place to think and explore, rather than confining the experience to a small screen. [3]

The 2025 Horizon Worlds prototype made the museum idea spatial. It featured Cleopatra, Galileo, and Leonardo, with themed zones, interactive props and button-based conversations. The environment was procedurally created and then baked into reusable assets. The project was submitted to Meta Horizon Creator Competition: Open Source Champions. The entry describes a 48-hour build and lessons about modular prefabs, interaction limits and checkpointing. [2]



Figure 2. The preserved Horizon Worlds project artwork records the earlier remixable museum concept. Source: the project's public repository. [2]

Beta work and the path to Matrix

Late 2025 SOTA alpha and beta

The alpha focused on historical-persona chat, prompt suggestions and a serverless response service. The beta, whose repository was created on September 28, 2025, expanded the experience into goals, quests, mentor generation, dialogue, assessment and history. Its React/Vite interface used Gemini integrations and Supabase-related persistence. The history records practical work on authentication, microphone capture, duplicate tool responses, mobile navigation and saving conversations. These changes mattered because an interesting conversation is not enough if the user cannot return to it reliably. [4]

2026 Rebuilds and research branches

The research repository preserves several versions of a broader educational specification: citation-aware teachers, Matrix-style world orchestration, assessment and embodied learning. These are research and design artifacts, not evidence that every proposed capability shipped. The wide v2 experiment explored multimodal classrooms, an Operator interface, quests and simulated visual providers. The subsequent text-first v2 established typed learning services, goal and curriculum flows, lesson persistence and assessment modules. A separate School candidate tested a constrained Galileo lesson and a reviewed Matrix connection. The repositories are related experiments, not one automatically migrated application. [5, 6]

Matrix Loading Operator The previous submission

Matrix supplied the missing spatial execution layer: voice and text requests, editable objects, asset import, persistent worlds and later a shared WebXR runtime. September releases progressed through white-room composition, voice, room-aware AR, animated objects and image feedback. By v1.0/v1.1, WebXR was the forward runtime; Unity work was retained as historical material. The v1.2 checkpoint added work on controls, object locks, smaller request context, recovery and explicit saved state. [7]

The lesson for the current project was architectural: keep learning state separate from world execution. A teacher should request a bounded action and receive its result, rather than become an unrestricted second world engine. This division allows the same teaching session to survive changes to voice, avatar appearance or the scene. [7, 8]

The current learning experience

October 2026 The embodied Professor

The current Professor application owns teachers, objectives, conversation, practice, quizzes and learner records. Matrix presents the avatar and exhibits, handles XR input, and executes accepted world changes. The integration added attributed dialogue, background Operator requests, reusable appearances, local voices and lip sync, named-part selection, virtual-view references and a persistent in-world conversation panel. [7, 8]

Stage	Important change
rc.1 and rc.2	Paired Professor/Matrix package, embodied dialogue and reusable avatar cast.
rc.3 through rc.5	Interactive teaching exhibits, finite mini-lessons, reflection, saved progress, XP and achievements; control and AR recovery work.
rc.6	Optional GPT-Live conversation, Decisions-based part pointing and interruption/stale-target handling.
rc.7	Connected mini-quests, selective teaching handoffs, independent avatar jobs, saved voices, linked world/conversation restoration and remote HTTPS access.

The latest reviewed publication is v1.3.0-rc.7. A newer release does not change what an older clip proves: the submitted video is a sequence of recorded examples, not a complete test of every rc.7 feature. The separate development record lists the published releases and reviewed merged pull requests. [9]

What is new relative to the previous submission

The current contribution is the connection between creation and teaching. A learner's question can produce an exhibit request; the Professor can continue talking while the Operator works; the loaded object becomes a reference for later explanation. A selected part is shared context, so "this coil" or "that wing" can have a concrete referent. The later voice route adds more natural conversation and a distinct teacher pointer without replacing School's learning record. [8, 9]

Design lessons

The evolution also changed the pedagogy. Earlier plans leaned heavily on mandatory goals and Socratic questioning. The current design keeps free conversation available and makes a focused mini-lesson an optional activity. Explanations, demonstrations, hints and reflection can support curiosity without turning every question into a test. [4–8]

Related work and system architecture

Related products and research

Ryo Suzuki's research provides a relevant framing for AR/AI as a spatial medium for thought. Figmin XR provides spatial creation and programmable object control, including an AI-assistant workflow. ShapesXR supports collaborative spatial design and interactive prototyping. Khanmigo represents the conversational tutoring side: guiding a learner through reasoning rather than only returning an answer. These are comparisons, not claims that School is the first to combine AI and XR. [3, 10–12]

Park et al.'s Generative Agents presents a memory, reflection and planning architecture for believable agents. It is relevant to the longer-term interest in persistent teachers and inhabitants. This submission does not reproduce that paper's social simulation or inherit its evaluation results. [13]

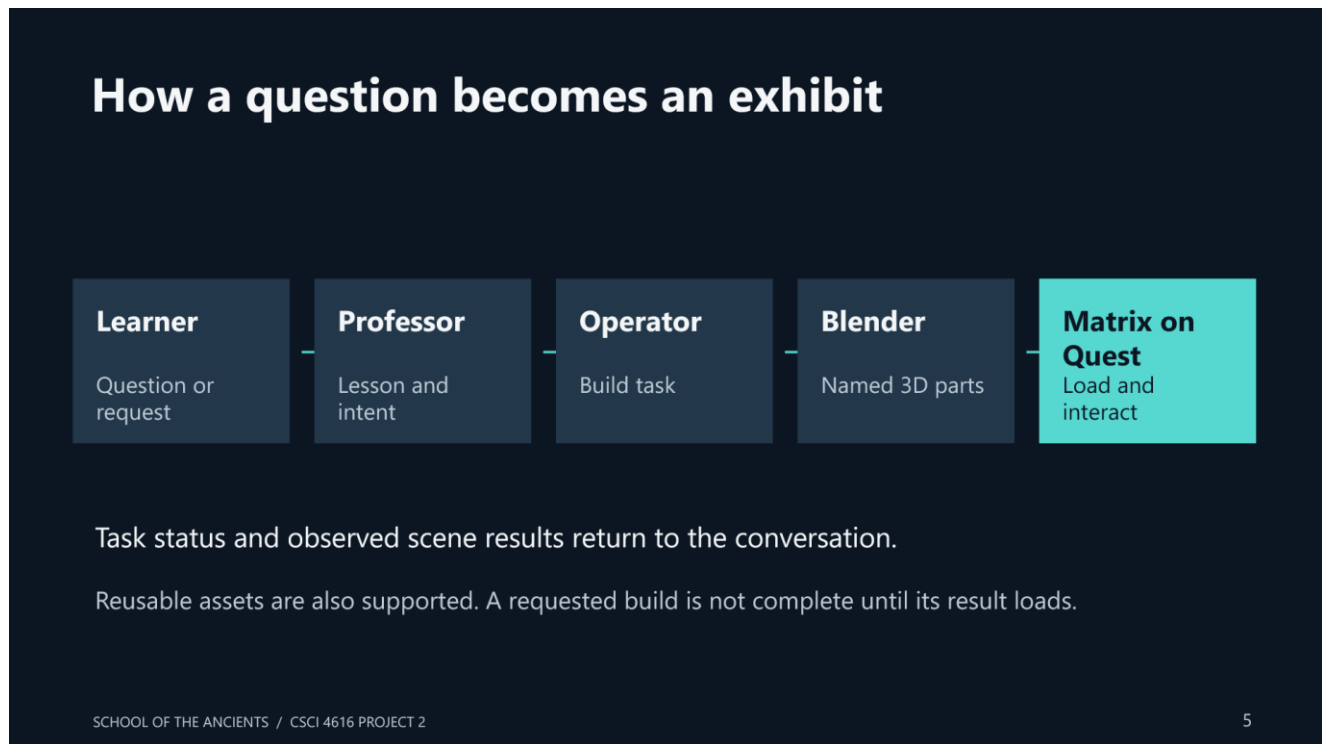


Figure 3. The request path. Host-side creation and verified world results connect the teaching session to the Quest client. Reusable assets and procedural generation are additional Operator paths.

Responsibilities and data flow

The Quest browser renders the Three.js/WebXR scene and captures allowed input. School maintains the teacher, lesson and attributed journal. The Operator accepts scoped work, tracks the task and coordinates content preparation. Blender can create geometry and animation on the host PC; export produces a GLB with identifiable parts. Matrix validates, registers and loads the asset. Task and runtime results return to the teaching conversation. A completed request, a registered asset and an observed loaded object are different states. [7–9]

Implementation and reproduction

The application is a PC-hosted WebXR system used from Quest Browser, not a standalone Quest APK.

Reproducing the complete workflow requires both the headset client and the host services. Use a pinned paired package so the School and Matrix versions match. The instructions below describe the reviewed release path, rather than guessing the exact commit of every source recording. [7–9]

1. Obtain the paired build. Download the v1.3.0-rc.7 Windows/WebXR ZIP and its checksum file from the Matrix release. Extract to a new directory. Compare the ZIP SHA-256 with the published checksum before using it; the manifest identifies the paired source commits.
 2. Install host prerequisites. Use the Node.js, Python and native Codex versions accepted by the bundled release guide; the inspected source uses Node.js 24 and a native Codex compatibility check. Configure the host's native teaching sign-in. Install Blender for fresh model authoring. VRoid is optional for new avatar work; the package includes a reusable cast.
 3. Prepare speech. For the original voice route, run the bundled local speech and Professor voice setup scripts. Local speech weights are separate downloads. For optional GPT-Live, configure the Professor server's private environment with OPENAI_API_KEY and SCHOOL_LIVE_ENABLED=1, and set an explicit spending allowance. Keys stay on the host.
 4. Start the services. Run Start-Professor.ps1 from the extracted package. The paired local configuration serves Matrix on port 8765 and School on 3110. Keep the launcher running. If the ports are occupied, use the existing intended instance or resolve the conflict; do not start two competing world clients.
 5. Connect Quest 3. Enable developer access and USB debugging for the headset, connect it to the PC, and run python Tools/connect_professor_quest.py. Open <http://127.0.0.1:8765/web/> in Quest Browser, grant the requested microphone/XR permissions, and enter AR. rc.7 also provides a separately configured authenticated HTTPS route.
 6. Start a fresh demonstration. Archive the existing world before starting an empty one. Start a new Professor conversation and select the intended mentor. A new world, a new teaching conversation and a new Operator session are separate choices; preserve old work before resetting any of them.
 7. Request an exhibit. Ask a learning question, then explicitly request an exhibit. One recorded example is "Can you ask the Matrix Operator to build it in Blender, with the spiral canvas rotor that slowly turns? Load it here when ready." The general pipeline may reuse a suitable asset; request a fresh Blender build when that is the capability being tested.
 8. Check the result and interact. Wait for task status and inspect the actual loaded object. Confirm scale, parts and any supported animation. In the documented control scheme, grip talks, trigger activates, trigger plus grip grabs, A/X selects a teaching point, and B/Y clears it. The live-voice route is explicitly started and stopped through its own controls.
 9. Save and evaluate. Save the world and retain the linked lesson. Reopen and confirm the intended scene, conversation and room placement. Test interruption, duplicate requests, selection changes and recovery separately. A successful demo run does not replace these checks.
- If the teacher is silent, check microphone permissions and the active voice route. If a build appears stuck, inspect its task status before retrying. If the room anchor is unavailable, use the documented archive-and-reposition workflow rather than clearing all browser data. [7–9]

The submitted demonstration

The opening explains the intended fresh-world workflow: exhibits are created and loaded on demand in response to a learner’s request. This describes the demonstration setup, not a claim that the application has no reusable code, avatars or asset library. The edited video combines earlier learner audio with an approved loaded scene, then compares original and live-voice examples. It does not show an uninterrupted recording of Blender generating every mesh.

Approximate time	What the audience sees
0:00–0:26	Opening, fresh-world explanation, the learner’s question and Blender request.
0:26–0:53	Leonardo with the loaded ornithopter and aerial screw, with a short explanation.
0:53–2:38	Original-mode title and Dyson/Faraday examples.
2:38–3:21	Live-voice title and Dyson example.
3:21–7:49	The full Tesla conversation: explanation, selected-part context and learner follow-up.
7:49–7:53	Closing title.



Figure 4. The Faraday exhibit in passthrough AR. Virtual objects are situated in the visible physical room.

The recordings show embodied explanation, loaded objects, part references and follow-up dialogue. They do not establish that every requested design is buildable, that the mechanisms are physically validated, or that the experience improves learning outcomes. The slow/fast labels identify the author’s original and live-voice versions; this is not a controlled speed benchmark.

Evaluation challenges and future work

Engineering challenges

The difficult part is coordinating asynchronous systems without losing the conversation. A build can finish after the learner changes topic; a speech interruption can arrive while a pointer request is pending; a new world can have a different anchor or teacher binding. The implementation therefore separates teaching state, task state, asset state and observed world state. Stable identifiers, explicit statuses and stale-result rejection are as important as the visible model. [7–9]

Usability work also mattered: microphone transcription can mishear a mentor’s name; a grab should not accidentally interrupt the teacher; the transcript must remain readable in AR; and an apparent room reset must not silently erase a saved lesson. The build history records fixes to these problems. The current package is still a development checkpoint.

What has and has not been evaluated

The rc.7 release notes report automated backend, browser, School and packaging tests, plus bounded wearer observations. Those are publisher-reported checks, not tests rerun for this written report. The release still identifies device checks for next-goal transitions, reward visibility, restoration and long sessions. The current video does not show XP or level-up overlays. It also contains no participant study, pre/post learning assessment, validated physics experiment or quantitative latency comparison. [9]

Future work

First, test the complete setup with other users: can they connect the headset, ask a useful question, understand build status and recover from a failed request? Second, measure end-of-speech to first audible response and request-to-loaded-exhibit time under comparable hardware, content and provider conditions. Third, build a small reviewed curriculum and source-grounded content library while retaining free exploration. Fourth, evaluate learning through explanation and transfer tasks, not engagement alone.

Longer-term work includes more reliable generated mechanisms, accessible text/mobile paths, reusable lesson packages, stronger content provenance, and multi-user learning. The research archive includes broader ideas such as LMS integration and persistent inhabitants; those remain separate development goals rather than claims about this submission. [5]

Conclusion

School of the Ancients has evolved from a future-museum idea into multiple working prototypes and now an embodied Quest learning workflow. Project 2 connects the earlier Matrix world-creation foundation to a teacher and a learner record. Its strongest result is concrete: a learner can ask about an idea, request a representation, and use the loaded object as the subject of further conversation. The next challenge is making that encounter dependable and educationally useful for people other than its creator.

References and project record

[1] Early web prototype and project history.

<https://github.com/thetopham/schooloftheancients>

[2] Horizon Worlds source and competition entry.

<https://github.com/School-of-the-Ancients/school-of-the-ancients-vr> ; <https://devpost.com/software/school-of-the-ancients>

[3] Ryo Suzuki, CU Boulder. Colloquium: Augment Human Thought and Creativity with the Power of AR and AI, September 4, 2025.

Title/date from the invitation supplied by the author; research profile below..

<https://www.colorado.edu/atlas/ryo-suzuki>

[4] SOTA alpha and beta.

<https://github.com/School-of-the-Ancients/sota-alpha> ; <https://github.com/School-of-the-Ancients/sota-beta>

[5] Research archive and versioned specifications.

<https://github.com/School-of-the-Ancients/research>

[6] V2 experiments and separate School candidate.

<https://github.com/School-of-the-Ancients/sota-v2-feature-creeped> ; <https://github.com/School-of-the-Ancients/sota-v2> ;

<https://github.com/School-of-the-Ancients/school-of-the-ancients>

[7] Matrix Loading Operator source and guides.

<https://github.com/School-of-the-Ancients/matrix-loading-operator>

[8] SOTA Professor source and implementation checkpoint.

<https://github.com/School-of-the-Ancients/sota-professor>

[9] Paired rc.7 release and recorded validation limits.

<https://github.com/School-of-the-Ancients/matrix-loading-operator/releases/tag/v1.3.0-rc.7>

[10] Figmin XR scripting and spatial creation.

<https://www.figmin.com/api>

[11] ShapesXR product documentation.

<https://learn.shapesxr.com/>

[12] Khanmigo product description.

<https://www.khanmigo.ai/>

[13] Park et al. Generative Agents: Interactive Simulacra of Human Behavior, UIST 2023.

<https://arxiv.org/abs/2304.03442>

[14] Cosmos Seven Wonders of the New World, Pavilion of the Searchers excerpt. Video and transcript supplied by the author..

<https://www.youtube.com/watch?v=DKd-m-X1lyM>

Authorship and evidence note

Matthew Topham is the project creator and supplied the recordings, personal history and submission requirements. GPT-6.1 Sol is credited at the author's request as an AI co-author for development and preparation of this submission. The report uses the supplied media, public project documentation, GitHub release/PR records and cited related work. Private meeting links, credentials and learner archives are not included.

The companion GitHub development record provides the dated release table and merged pull-request index. Snapshot reviewed October 26, 2026. Repository descriptions and historical plans are attributed as such; they are not treated as proof that every proposed feature shipped.