INSTITUTO TECNOLÓGICO DE AERONÁUTICA

Eduardo Moura Zindani

EDUCATIONAL ORBIT SIMULATION WITH GENERATIVE AI AGENTIC WORKFLOW AND AUGMENTED REALITY VISUALISATION

Final Paper
2025

Course of Aerospace Engineering

CDU ???.??

Eduardo Moura Zindani

EDUCATIONAL ORBIT SIMULATION WITH GENERATIVE AI AGENTIC WORKFLOW AND AUGMENTED REALITY VISUALISATION

Advisor

Prof. Dr. Christopher Shneider Cerqueira (ITA)

AEROSPACE ENGINEERING

São José dos Campos
Instituto Tecnológico de Aeronáutica

2025

Cataloging-in Publication Data
Documentation and Information Division

Zindani, Eduardo Moura
Educational Orbit Simulation with Generative AI Agentic Workflow and Augmented Reality Visualisation / Eduardo Moura Zindani.
São José dos Campos, 2025.
37p.

Final paper (Undergraduation study) – Course of Aerospace Engineering– Instituto Tecnológico de Aeronáutica, 2025. Advisor: Prof. Dr. Christopher Shneider Cerqueira.

1. AI. 2. AR. 3. Orbit. I. Instituto Tecnológico de Aeronáutica. II. Educational Orbit Simulation with Generative AI Agentic Workflow and Augmented Reality Visualisation.

BIBLIOGRAPHIC REFERENCE

ZINDANI, Eduardo Moura. **Educational Orbit Simulation with Generative AI Agentic Workflow and Augmented Reality Visualisation**. 2025. 37p. Final paper (Undergraduation study) – Instituto Tecnológico de Aeronáutica, São José dos Campos.

CESSION OF RIGHTS

Eduardo Moura Zindani
Rua H8B, Ap. 235
12.228-461 – São José dos Campos–SP

EDUCATIONAL ORBIT SIMULATION WITH GENERATIVE AI AGENTIC WORKFLOW AND AUGMENTED REALITY VISUALISATION

This publication was accepted like Final Work of Undergraduation Study

---

Eduardo Moura Zindani

Author

Christopher Shneider Cerqueira (ITA)

Advisor

Prof. Dra. Maísa de Oliveria Terra
Course Coordinator of Aerospace Engineering

São José dos Campos: June 20, 2025.

…

Acknowledgments

…

*"If I have seen farther than others,
it is because I stood on the shoulders of giants."*
— Sir Isaac Newton

Resumo

Métodos educacionais tradicionais frequentemente encontram dificuldades para transmitir conceitos complexos, espaciais e dinâmicos como os da mecânica orbital. A recente convergência entre a Realidade Aumentada (RA) de consumo e os sofisticados agentes de Inteligência Artificial (IA) Generativa apresenta uma oportunidade para criar um novo paradigma de interfaces de aprendizagem intuitivas e experienciais. Este trabalho detalha o projeto, desenvolvimento e avaliação de uma plataforma educacional interativa para a exploração dos princípios da mecânica orbital. O objetivo principal do sistema é conectar a teoria de leis físicas abstratas à compreensão intuitiva, permitindo que os usuários aprendam por meio da interação corporificada. A metodologia é centrada em uma arquitetura modular que integra três componentes principais: (1) um "cérebro" agente generativo, impulsionado por Modelos de Linguagem Abrangentes, que interpreta comandos em linguagem natural e atua como um guia educacional especializado; (2) um "mundo" de simulação em tempo real e visualização em RA, construído no motor Unity para o Meta Quest 3, que renderiza trajetórias orbitais fisicamente precisas e ancoradas no ambiente do usuário ; e (3) uma interface "corpo" corporificada, consistindo em um globo físico com um sensor rotacional, que permite o controle tangível da simulação. A plataforma facilita um ciclo de interação multimodal contínuo, onde os comandos de voz e as manipulações físicas do usuário são capturados, processados pelo agente para alterar os parâmetros da simulação e refletidos na visualização em RA com feedback auditivo conversacional. Este trabalho entrega um protótipo funcional que demonstra uma nova abordagem para a educação científica, transformando dados abstratos em uma experiência manipulável e conversacional para promover uma aprendizagem exploratória e profundamente engajadora.

Abstract

Traditional educational methods often struggle to convey complex, spatial, and dynamic concepts such as those found in orbital mechanics. The recent convergence of consumer-grade Augmented Reality (AR) and sophisticated Generative AI agents presents an opportunity to create a new paradigm for intuitive and experiential learning interfaces. This paper details the design, development, and evaluation of an interactive educational platform for exploring the principles of orbital mechanics. The system's primary objective is to bridge the gap between abstract physical laws and intuitive comprehension by enabling users to learn through embodied interaction. The methodology is centered on a modular architecture that integrates three core components: (1) a generative agent "brain," powered by Large Language Models, which interprets natural language commands and acts as an expert educational guide; (2) a real-time simulation and AR visualization "world," built in the Unity engine for the Meta Quest 3, which renders physically accurate orbital trajectories anchored to the user's environment ; and (3) an embodied "body" interface, consisting of a physical globe with a rotational sensor, that allows for tangible control over the simulation. The platform facilitates a seamless multimodal interaction loop where a user's voice commands and physical manipulations are captured, processed by the agent to alter simulation parameters, and reflected in the AR visualization with conversational auditory feedback. This work delivers a functional prototype that demonstrates a novel approach to science education, transforming abstract data into a manipulable, conversational experience to foster exploratory and deeply engaging learning.

List of Figures

List of Tables

Contents

1 Introduction

1.1 Organisation

This work is organised into three main chapters, each addressing a distinct aspect of the project. The breakdown is as follows:

- **Chapter 1: Introduction.** This chapter sets the stage for the research and development.
  - *Motivation* (§1.2): Presents the core argument that the convergence of Augmented Reality and generative AI enables a new, more intuitive paradigm for educational interfaces.
  - *Objectives* (§1.3): Defines the project's specific, actionable goals, centered on the development and evaluation of an interactive, agent-guided simulation platform.
- **Chapter 2: Literature Review.** This chapter provides the theoretical and technical foundation for the work by reviewing three key domains.
  - *Augmented and Virtual Reality* (§2.1): Reviews the evolution of immersive hardware and software ecosystems and establishes their pedagogical value for spatial learning.
  - *Generative Agents* (§2.2): Defines the architecture of modern LLM-powered agents, detailing their ability to use planning, memory, and external tools to reason through and execute complex tasks.
  - *Orbital Mechanics* (§2.3): Outlines the fundamental physics of celestial motion, including the two-body problem, classical orbital elements, and impulsive maneuvers, which form the mathematical basis for the simulation.
- **Chapter 3: Methodology.** This chapter details the practical design, implementation, and assessment of the project.

- *System Architecture and Data Flow* (§3.2): Describes the end-to-end fluxogram of the system, illustrating how multimodal user input is captured, processed by the agent, and rendered in the simulation in a continuous loop.
- *Core Component Implementation* (§3.3): Details the specific development plan and tools for the three primary modules: the Generative Agent ("Brain"), the Simulation and AR Visualisation ("World"), and the Embodied Interface ("Body").
- *Evaluation Plan* (§3.5): Defines the two-pronged approach for assessment, covering the technical validation of the system's performance and a qualitative user study to gauge its potential as an effective educational tool.

1.2 Motivation

For decades, popular media and speculative fiction have envisioned futuristic interfaces for exploration and control, from holographic command centers to immersive planetary navigation tools. Films such as *Minority Report* (2002) and *Iron Man* (2008) popularized visions of humans interacting with vast information systems through gestures, speech, and spatial manipulation. These visions were once confined to science fiction, but today, the convergence of Augmented Reality (AR), Virtual Reality (VR), and Artificial Intelligence (AI) is bringing such interfaces into the realm of technological feasibility.

In particular, the past few years have seen rapid advances in consumer-grade AR/VR hardware. Devices like the Meta Quest and Apple Vision Pro represent significant milestones in accessibility and visual fidelity, enabling immersive environments that are no longer confined to laboratory research or elite applications. The implications for interface design, interaction paradigms, and knowledge acquisition are profound. AR and VR are no longer speculative technologies, they are present, evolving, and increasingly democratized.

Concurrently, the emergence of generative AI and language-based agents has introduced a paradigm shift in how humans interact with complex systems. Large Language Models (LLMs), such as those powering conversational agents, can now interpret natural language, generate multimodal content, and coordinate sequences of actions across software environments. This represents a departure from deterministic, rule-based systems toward stochastic and adaptive workflows, where agents interpret intention, negotiate uncertainty, and build dynamically responsive experiences.

When these technologies - AR/VR and generative agents - are combined, they form the foundation for a new kind of interface: one that is spatial, conversational, and adaptive. Such interfaces do not rely on code or static menus; they respond to voice, gesture, and embodied input. They transform abstract data into manipulable space, and procedural complexity into natural dialogue.

This is particularly relevant in the domain of education. Traditional educational systems remain bound to text, diagrams, and symbolic representation. While these tools are powerful, they often fall short when applied to fields that are inherently spatial, dynamic, or non-intuitive. Orbital mechanics, for example, involves motion through three-dimensional space governed by non-linear physical laws. Launch trajectories, gravitational slingshots, inclination changes, these are difficult to visualize and even harder to intuit.

In this context, immersive simulation becomes more than a visual aid: it becomes a cognitive bridge. A learner can rotate a globe, speak a question, and witness a launch trajectory materialize. They can observe orbits evolve in real time, ask about inclinations or transfer windows, and receive explanations grounded in physics. Education becomes experiential, a process of exploration rather than instruction.

Moreover, generative agents provide a layer of accessibility that is historically absent in technical domains. They can guide the learner, interpret vague queries, correct misconceptions, and explain phenomena in adaptive ways. They act as intelligent mediators between curiosity and formal knowledge.

Given these technological conditions, the maturity of AR/VR, the rise of stochastic AI agents, and the persistent limitations of traditional educational media, this project is motivated by a clear opportunity: to construct a new type of educational experience. One that is not constrained by interface conventions, disciplinary jargon, or static presentation. One that invites the user to learn by seeing, asking, moving, and listening.

The convergence of embodied interaction and generative intelligence allows for a simulation system that is not only technically rigorous, but experientially meaningful. It enables a form of learning in which the abstract becomes tangible, the distant becomes near, and the user is placed at the center of the scientific process. This project emerges from the belief that space education, and scientific education more broadly, can and must evolve to meet the possibilities of our time.

1.3 Objectives

General Objective

To develop an interactive, agent-guided simulation platform that enables users to explore and understand orbital mechanics through embodied interaction, combining physical manipulation, natural language dialogue, and real-time visualizations.

Specific Objectives

1. Design and implement a simulation environment capable of rendering orbital trajectories in real time, grounded in physically accurate models.

2. Integrate a generative agent capable of interpreting natural language input, translating it into simulation parameters, and guiding the user through explanations and interactions.

3. Develop a physical interface, centered around a globe equipped with sensors, that captures real-world rotational input and transmits it into the simulation system.

4. Enable multimodal interaction by combining physical movement, voice commands, and visual feedback to create a seamless and intuitive user experience.

5. Ensure that all components of the system, simulation, agent, and physical bridge, function coherently and communicate reliably in real time.

6. Create a system architecture that is modular and extensible, allowing for future expansion to other celestial bodies, educational modules, or mission types.

7. Evaluate the platform's potential as an educational tool for facilitating conceptual understanding of orbital mechanics through exploratory learning.

2 Literature Review

2.1 Augmented and Virtual Reality in Immersive Educational Simulation Systems

Augmented Reality (AR) and Virtual Reality (VR) are complementary immersive technologies that enrich or replace a user's perception of the world. AR overlays digital content onto the real environment in real-time, allowing virtual objects to coexist with physical surroundings (Billinghurst; Clark; Lee, 2015). In contrast, VR completely immerses the user in a fully synthetic, computer-generated environment, blocking out the physical world. Milgram's classic "Reality-Virtuality" continuum illustrates these as endpoints: AR lies near the real-world end (mixing virtual content with reality), whereas VR occupies the extreme virtual end with an entirely simulated world (Milgram; Kishino, 1994). In essence, AR adds to the user's real-world experience, while VR transposes the user into an interactive virtual scene. Both technologies share common roots in decades of research and development. The term augmented reality was first coined by Caudell and Mizell (1992) in the context of assisting Boeing manufacturing with see-through displays (Caudell; Mizell, 1992). A few years later, Azuma's influential survey defined AR by three key characteristics: combining real and virtual content, interactive operation in real time, and accurate 3D registration of virtual objects in the physical world (Azuma, 1997; Billinghurst; Clark; Lee, 2015). VR, meanwhile, has been long conceptualized as achieving presence – the feeling of "being there" in a virtual environment – by engaging multiple senses with responsive 3D graphics and audio (Johnson-Glenberg, 2018). Modern definitions emphasize that VR provides immersive first-person experiences where users can interact with simulated worlds as if they were real, inducing a strong sense of presence and agency within the virtual scene.

2.1.1 Hardware Evolution:

AR and VR technologies have evolved rapidly, enabling consumer-grade devices that support realistic immersive experiences. While early head-mounted displays date back to the 1960s (e.g., Sutherland's Sword of Damocles), the 2010s marked a turning point with modern devices. On the VR front, the Oculus Rift prototype (2010) by Palmer Luckey re-ignited interest with a wide field of view and affordable design. Crowdfunded in 2012 and acquired by Facebook in 2014, Oculus released its first consumer headset in 2016, alongside HTC's Vive, which introduced room-scale tracking. These devices brought high-fidelity visuals and motion tracking to mainstream audiences.

The next major step came with standalone VR headsets. The Oculus/Meta Quest series, starting in 2019, integrated processing and inside-out tracking directly into the headset. Quest 2 (2020) and Quest 3 (2023) improved resolution, optics, and added passthrough AR capabilities (Ruth, 2024). In parallel, PC-based headsets like the Valve Index and Varjo pushed the fidelity frontier for gaming and enterprise simulation.

AR hardware followed a distinct trajectory. Initial systems used handheld or laptop setups, but the release of Microsoft's HoloLens in 2016 marked the arrival of self-contained AR headsets with spatial mapping and inside-out tracking. Magic Leap One (2018) added novel display technologies (Billinghurst; Clark; Lee, 2015), while consumer experiments like Google Glass (2013) explored heads-up interfaces before being discontinued in 2023 (Ruth, 2024).

Smartphones played a critical role in scaling AR adoption. Apps like Pokémon GO (2016) introduced mainstream users to AR through camera overlays. ARKit (Apple) and ARCore (Google), launched in 2017, enabled mobile AR with motion and depth tracking (Vieyra; Vieyra, 2018).

Most recently, the line between AR and VR is blurring. Apple's Vision Pro (announced 2023) merges high-resolution VR with passthrough AR, positioning itself as a "spatial computer." With features like dual 4K displays and hand/eye tracking, it may represent a watershed moment for XR despite its premium price (Ruth, 2024).

As of 2025, the hardware ecosystem spans from mobile-based AR apps to advanced mixed reality headsets, forming a robust toolbox for immersive educational simulations.

2.1.2 Software Ecosystems and Frameworks:

Alongside hardware, a mature software ecosystem has enabled rapid development of immersive simulations. Modern game engines such as Unity and Unreal Engine have become the de facto platforms for AR/VR content creation. These engines provide high-performance 3D graphics rendering, physics simulation, and cross-platform deployment, greatly simplifying the creation of interactive virtual environments. Unity, for example, offers an entire XR development toolkit (with support for VR headsets and AR through packages like AR Foundation) that abstracts away device-specific details and allows developers to build an application once and deploy across multiple headsets (Atta *et al.*, 2022). Unreal Engine likewise includes integrated support for VR rendering and AR (via ARKit/ARCore plugins), making high-fidelity visualization accessible to developers in academia and industry.

For mobile AR, platform-specific frameworks are key. Apple's ARKit (introduced in iOS 11, 2017) and Google's ARCore (for Android, 2017) brought advanced AR capabilities to hundreds of millions of smartphones (Vieyra; Vieyra, 2018). These software development kits handle real-time tracking of the device's position, surface detection, lighting estimation, and more, allowing apps to place and persist virtual objects in the user's environment. Thanks to ARKit/ARCore, an educator can deploy an AR simulation on standard tablets or phones – for instance, letting students point an iPad at a textbook and see 3D molecules or physical field lines appear "attached" to the pages. On the web, the WebXR API has emerged as a W3C standard enabling AR and VR experiences to run directly in web browsers using JavaScript (World Wide Web Consortium, 2021). WebXR (successor to earlier WebVR/WebAR efforts) allows an immersive educational module to be accessed with a simple URL, lowering the barrier to entry (no app install required) and ensuring compatibility across different devices (from VR headsets to phones). This is particularly relevant for broad educational deployments, where web-based delivery can be more practical. Complementing these are various supporting frameworks: for example, libraries for spatial mapping, hand tracking, and user interaction (e.g. Microsoft's Mixed Reality Toolkit for Unity, or Vuforia for image-target AR) which provide higher-level tools for common AR/VR interactions. There are also open standards like OpenXR (released by the Khronos Group in 2019) that unify the interface to VR/AR hardware – a developer can write code once against OpenXR and run on any compliant headset (Oculus, SteamVR, Windows Mixed Reality, etc.), which is increasingly adopted by engines and platforms. In summary, the software landscape – from powerful 3D engines to AR phone toolkits and web standards – has matured to a point that immersive educational simulations can be built with relatively modest effort compared to a decade ago. This thesis will leverage these tools to construct its simulation system, ensuring it is built on proven, widely supported technology.

2.1.3 Use Cases in Education:

AR and VR have shown strong potential to enhance learning, particularly in subjects involving abstract or spatial concepts. Their core strength lies in making the invisible visible and the abstract tangible. In physics education, for instance, VR has helped students visualize and manipulate 3D vectors, improving understanding of vector addition and spatial relationships (Campos; Hidrogo; Zavala, 2022). Studies show that such immersive tools can boost engagement and deepen comprehension of abstract STEM topics like electromagnetism or geometry through interactive, risk-free exploration (Campos; Hidrogo; Zavala, 2022; Johnson-Glenberg, 2018).

In astronomy and aerospace, where scales are far beyond human experience, immersive technologies offer unique advantages. VR enables virtual field trips through space — letting students stand on Mars or orbit planets — providing an intuitive grasp of scale and distance. Learners can explore the solar system with accurate proportions, making complex spatial relationships (like planetary distances or ring sizes) more comprehensible (Atta *et al.*, 2022). Astrophysical phenomena such as orbital mechanics and black hole dynamics are also made more accessible through interactive VR visualizations.

In aerospace engineering, VR and AR are increasingly used for hands-on training. Beyond traditional flight simulators, modern VR platforms allow students to perform simulated pre-flight inspections, engine maintenance, or spacecraft docking. Vaughn College, for example, uses VR for aviation trainees to practice inspecting and assembling parts, reinforcing mechanical familiarity before real-world exposure. Similarly, Atta et al. (2022) created a virtual "space lab" where students assemble a CubeSat in a simulated cleanroom, boosting their understanding of subsystem configuration through direct interaction and gamified tasks (Atta *et al.*, 2022).

AR complements this by overlaying digital instructions on real-world hardware. NASA's Project Sidekick exemplifies this: astronauts use HoloLens headsets aboard the ISS to receive real-time, spatially anchored maintenance guidance (NASA, 2015). In classrooms, AR enables students to interact with 3D models of rockets or overlay CAD designs onto physical parts, enriching theoretical lessons with live, contextual visualization (Atta *et al.*, 2022; Milgram; Kishino, 1994).

2.1.4 Embodiment, Interaction, and Spatial Cognition:

A recurring theme in the educational use of AR/VR is the role of embodied and spatial learning. Immersive technologies engage the human sensorimotor system – users move their bodies to navigate virtual spaces, use gestures to interact with virtual objects, and perceive environments at true scale. This physicality supports cognitive processing by leveraging innate spatial reasoning and muscle memory. The theory of embodied cognition holds that learning is grounded in the body's interactions with its environment, and AR/VR extend this principle digitally. Johnson-Glenberg (2018) highlights the pedagogical value of 3D gestures: when learners rotate a virtual object or walk through a graph, they build stronger memory links (Johnson-Glenberg, 2018). Her research shows that full-motion VR, where body movements align with abstract concepts, can deepen understanding and recall. Complementary studies (e.g., Liu et al., 2020) found improved retention when students enacted phenomena physically, and also noted increased presence and agency—factors tied to motivation (Campos; Hidrogo; Zavala, 2022; Johnson-Glenberg, 2018).

Spatial cognition benefits are also well-documented. VR's stereoscopic depth and six degrees of freedom help learners perceive complex spatial relationships, vital in subjects like anatomy, geography, and engineering. Students exploring a molecule or a solar system in VR can shift perspective freely, activating spatial memory and supporting what researchers call "situated learning" – knowledge acquired in rich spatial contexts becomes more intuitive and transferable. Campos et al. (2022), for example, found that immersive 3D interaction notably enhanced vector learning tasks requiring spatial reasoning (Campos; Hidrogo; Zavala, 2022). Similarly, in astronomy, VR's ability to scale from the Milky Way to Earth provides concrete visualizations of abstract systems (Kersting et al., 2024).

While AR/VR offer compelling tools, they are not magic bullets – user comfort, software complexity, and thoughtful pedagogical integration remain critical (Johnson-Glenberg, 2018). Still, evidence shows that immersive simulations can enhance traditional teaching, especially for learning goals involving visualization, experimentation, or embodied experience. In the context of this thesis, the implications are clear: AR and VR form a foundational layer. They enable students to interact with simulations of aerospace systems—such as satellites or orbital dynamics—in an intuitive and experiential manner. As hardware becomes lighter and more capable, and software ecosystems more robust, immersive tools are becoming increasingly viable in education. With spatial computing platforms entering mainstream use (Ruth, 2024), AR and VR are poised not just as delivery platforms but as new paradigms for engaging with knowledge.

2.2 Generative Agents

Traditional software and simulations have been predominantly *deterministic*—given the same inputs, they yield the same outputs. Modern AI systems built on *generative* models, by contrast, introduce stochasticity and creativity. Large Language Models (LLMs) do not follow hard-coded rules; instead, they sample from probability distributions learned from vast textual corpora. Consequently, an LLM can produce context-dependent, varied responses rather than a single predetermined answer. This marks a paradigm shift from scripted to emergent behaviour. In recent work, advanced LLMs such as GPT-4 have even outperformed traditional reinforcement-learning agents in complex environments by reasoning through text rather than executing pre-programmed control policies (Carrasco; Rodriguez-Fernandez; Linares, 2025). While stochastic generation entails some unpredictability, it is precisely this creativity that lets *generative agents* adapt to scenarios beyond their designers' foresight.

At a conceptual level an LLM is a statistical language engine: given a textual history, it predicts the most plausible continuation one word at a time. Because it is trained on heterogeneous data, a single model can answer coding questions, analyse legal texts, or reason about orbital mechanics when prompted appropriately. This broad, generative capability underpins the rise of *LLM-powered agents* (Anthropic, 2024).

LLM-based agents are autonomous software entities that embed an LLM as their core "brain." An agent senses its environment, reasons about goals, and acts—iteratively—until a task is complete. Industry definitions describe such an agent as "a system that uses an LLM to reason through a problem, create a plan, and execute that plan with tools" (Chen, 2023; Huang; Grady, *et al.*, 2024). The LLM supplies the reasoning; auxiliary modules provide planning, memory, and tool use (Anthropic, 2024). Crucially, the agent—not the user—controls the loop: it may decide which function to call, when to revise a plan, or whether to request clarification (OpenAI, 2023). Hence an agent is more than a single LLM invocation; it is a continual perceive–think–act cycle.

Architectural Components

Generative-agent designs typically comprise five interacting elements (Anthropic, 2024; Huang; Grady, *et al.*, 2024):

- **Planning and reasoning.** The agent decomposes high-level goals into actionable steps, often prompting the LLM to produce an internal plan or "chain of thought."
- **Memory.** Short-term context (recent turns) and long-term knowledge (summaries or retrieved documents) are stored externally—e.g. in a vector database—and injected into prompts as needed.
- **Tool use and APIs.** Through structured outputs (JSON function calls, shell commands, *etc.*) the agent invokes external tools to compute, query, or effect changes in its environment (OpenAI, 2023).
- **Iterative control loop.** The agent cycles through *observe* → *reason* → *act* → *observe*, optionally reflecting or self-critiquing between steps to improve reliability.
- **Autonomy and adaptation.** Equipped with the above, the agent can switch strategies, recover from errors, and pursue its objective with minimal human micromanagement.

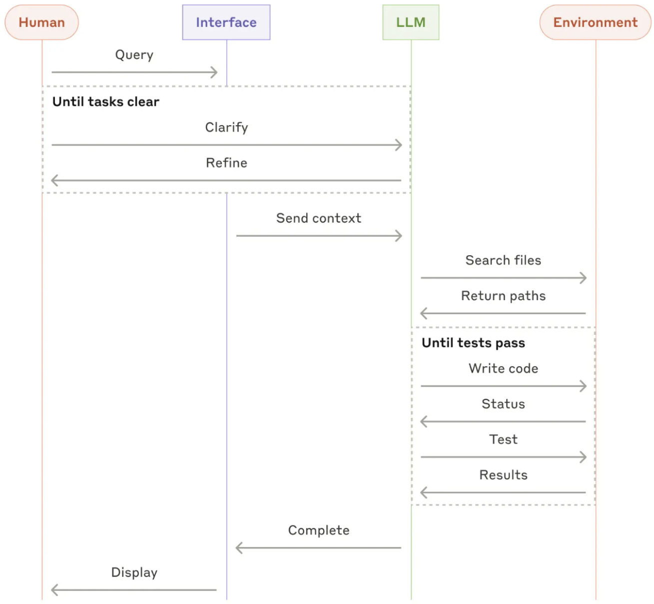

FIGURE 2.1 – End-to-end agentic workflow from Anthropic's "Building Effective Agents." The human issues a query through an *interface*; the LLM asks clarifying questions until the task is precise, receives contextual files, iteratively writes and tests code against the environment, and finally returns results for display (Anthropic, 2024).

Applications and Relevance

- **Simulations and interactive worlds.** Park *et al.* created "Generative Agents" that populate a sandbox town with virtual characters who plan, remember, and socially interact—producing emergent storylines never scripted by the developers (Park *et al.*, 2023).
- **Aerospace guidance and control.** Carrasco *et al.* demonstrated an LLM agent piloting a spacecraft in the *Kerbal Space Program* simulation by iteratively reading textual telemetry and issuing control actions, matching classical controllers without explicit orbital equations (Carrasco; Rodriguez-Fernandez; Linares, 2025).
- **Legal reasoning.** Harvey AI equips law-firm associates with an agent that drafts memos, retrieves precedents, and iteratively refines analyses through dialogue—illustrating agentic workflows in language-dense tasks (Chen, 2023).
- **Education.** Khan Academy's *Khanmigo* employs GPT-4 as a Socratic tutor that adapts explanations to each learner, providing hints rather than answers and thereby personalising study sessions at scale (Academy, 2023).

2.3 Orbital Mechanics: The Physics of Celestial Motion

The intuitive, visual understanding of orbital motion is a primary objective of this project. While the generative agent will handle the underlying calculations, a firm grasp of the governing principles is essential to frame the simulation's logic and appreciate its educational value. Orbital mechanics is the study of the motion of bodies under the influence of gravity. For missions in Earth's orbit and for most interplanetary transfers, the foundational principles discovered by Isaac Newton and Johannes Kepler provide a remarkably accurate framework for describing and predicting these celestial paths. This section outlines the core concepts that form the physical basis of the simulation system.

2.3.1 The Fundamental Law: Gravity and the Two-Body Problem

At the heart of all orbital motion lies gravity. In the 17th century, Sir Isaac Newton formulated the Law of Universal Gravitation, stating that any two bodies attract each other with a force proportional to the product of their masses and inversely proportional to the square of the distance between them (Curtis, 2020). This is expressed mathematically as:

$$F = G\frac{m_1 m_2}{r^2}$$

where F is the gravitational force, G is the gravitational constant, m_1 and m_2 are the masses of the two bodies, and r is the distance between their centers.

When applied to a satellite orbiting a celestial body like Earth, this law simplifies into the cornerstone of astrodynamics: the **two-body problem**. This model makes a critical assumption: it considers only the gravitational force between the satellite and the primary body (e.g., Earth), ignoring all other perturbations such as atmospheric drag, solar radiation pressure, and the gravitational pull from other bodies like the Moon or the Sun (Vallado, 2013). While these forces are significant for high-precision, long-term trajectory prediction, the two-body model provides an elegant and highly accurate approximation for most foundational analysis and educational purposes. The resulting equation of motion is:

$$\ddot{\vec{r}} + \frac{\mu}{r^3}\vec{r} = 0$$

Here, $\vec{r}$ is the position vector of the satellite relative to the primary body, $\ddot{\vec{r}}$ is its acceleration, and μ (mu) is the standard gravitational parameter of the system ($\mu = G(m_1 + m_2)$). The solution to this equation reveals that the satellite's path must be a conic section: a circle, ellipse, parabola, or hyperbola (Curtis, 2020). For a satellite captured in orbit, its path will be an ellipse, or a circle as a special case of an ellipse.

2.3.2 The Language of Orbits: The Classical Orbital Elements

While the equation of motion describes the physics, it does not provide an intuitive description of an orbit's path. To define the size, shape, and orientation of an orbit in three-dimensional space, a set of six parameters, known as the **classical Keplerian orbital elements**, is used. These elements provide a unique and static description of the orbit that results from solving the two-body problem (Bate; Mueller; White, 1971). They serve as the precise, non-ambiguous instruction set that a generative agent can use to calculate and render any given orbit within the simulation.

The six elements are (Curtis, 2020):

- **Semimajor Axis (a):** Defines the size of the orbit. It is half of the longest diameter of the ellipse.
- **Eccentricity (e):** Defines the shape of the orbit. For a bound orbit, e ranges from 0 for a perfect circle to less than 1 for an ellipse.
- **Inclination (i):** Defines the tilt of the orbital plane with respect to a reference plane (typically Earth's equatorial plane). An inclination of $0°$ is an equatorial orbit, while $90°$ is a polar orbit.
- **Right Ascension of the Ascending Node (Ω):** Defines the orientation, or swivel, of the orbital plane in space. It is the angle measured in the reference plane from a reference direction (the vernal equinox) to the point where the satellite crosses the equator from south to north (the ascending node).
- **Argument of Perigee (ω):** Defines the orientation of the ellipse within its orbital plane. It is the angle measured from the ascending node to the orbit's point of closest approach to the primary body (the perigee).
- **True Anomaly (ν):** Defines the position of the satellite along its elliptical path at a specific time. It is the angle from the perigee to the satellite's current position vector.

2.3.3 Changing Orbits: Impulsive Maneuvers and Delta-V

Orbits are not always static. To move a satellite from one orbit to another—for example, from a low parking orbit to a higher operational orbit—it must change its velocity vector ($\vec{v}$). In practice, this is achieved by firing a thruster. For mission planning and simulation, these burns are often modeled as **impulsive maneuvers**, which are assumed to be instantaneous changes in velocity (Bate; Mueller; White, 1971). This simplification is highly effective when the burn time is short compared to the orbital period.

The "cost" of performing such a maneuver is measured by **delta-v** (Δv), which is the total change in velocity required. Delta-v is the fundamental currency of spaceflight; every orbital maneuver has a Δv budget, which ultimately dictates the amount of propellant required (Vallado, 2013).

A classic and highly efficient example of an orbital maneuver is the **Hohmann transfer**, used to move between two circular, coplanar orbits. It consists of two impulsive maneuvers:

1. A first burn (Δv_1) is performed to increase the satellite's speed, placing it into an elliptical transfer orbit that is tangent to both the initial and final orbits.
2. Upon reaching the highest point of the transfer orbit (apoapsis), a second burn (Δv_2) is performed to increase speed again, circularizing the path into the final, higher orbit.

This is precisely the type of task an agent could execute in the simulation: a user could request a transfer between two orbits, and the agent would calculate the required Δv and visualize the two-burn Hohmann transfer trajectory.

2.3.4 The Initial Step: From Surface to Orbit

Before a satellite can orbit, it must first get there. The launch phase involves a complex journey through the atmosphere to achieve the required altitude and velocity for orbit insertion. The fundamental challenge is to provide the launch vehicle with enough energy to overcome two primary obstacles: Earth's gravitational pull and atmospheric drag.

Conceptually, a launch can be viewed as a process of gaining both vertical and horizontal velocity. The vehicle must ascend vertically to clear the densest part of the atmosphere, after which it performs a "gravity turn" to begin building horizontal speed. The objective is to reach a target altitude with a velocity vector that is nearly horizontal and has a magnitude equal to that required for a stable orbit. At this point, known as orbit insertion, the engines cut off, and the spacecraft begins its free-fall journey governed by the principles of orbital mechanics. For the purpose of this simulation, the complex atmospheric ascent can be abstracted, with the interactive experience beginning at the moment of orbit insertion, allowing the user to focus on the orbital dynamics that follow.

3 Methodology

3.1 Design Philosophy and Approach

The development of this project is fundamentally an exploratory research endeavour into a new paradigm of human-computer interaction for educational purposes. Given the innovative and complex nature of integrating generative AI, augmented reality, and embodied interfaces, a rigid, waterfall-style development plan would be inappropriate. Instead, the methodology is guided by a philosophy that embraces iteration and modularity to navigate the technical challenges and discovery process inherent in such work.

The approach is defined by three core principles:

- **Prototype-Driven:** The primary goal is the creation of a functional prototype that demonstrates the feasibility and potential of the proposed system. This approach prioritizes implementing the core functionalities of the user experience over exhaustive feature development, allowing for tangible and testable results that can validate the project's central thesis.
- **Iterative Development:** The project will be built in iterative cycles, following a process of building a core feature, testing its performance and usability, and refining it based on the results. This allows for flexibility in the implementation details, acknowledging that the optimal solutions for sensor integration, agent prompting, and user interaction will be discovered and improved upon throughout the development lifecycle.
- **Modular Architecture:** The system is designed as a collection of distinct yet interconnected modules: the generative agent (the "brain"), the simulation and visualisation engine (the "world"), and the physical interface (the "body"). This modularity, a key objective of this project, makes the complex system manageable, facilitates parallel development and testing of components, and ensures the final architecture is extensible for future work.

3.2 System Architecture and Data Flow

This section outlines the high-level system architecture, describing the flow of information between the user, the physical interface, the software components, and the generative agent. This end-to-end process, or ”fluxogram,” illustrates how the various components work in concert to create a seamless, real-time interactive experience. The architecture is designed as a continuous ”perceive-think-act” cycle, mirroring the agentic workflows described in the literature.

The data flow for a single user interaction can be detailed in the following sequence:

1. **User Input:** The interaction begins with the user issuing a multimodal command, such as speaking a request (e.g., ”Show me an orbit with an inclination of 45 degrees”) while simultaneously rotating the physical globe to a desired orientation.

2. **Physical Interface Capture:** The system's hardware interfaces capture the raw input data. The microphone on the AR/VR headset records the user's voice, while the Arduino microcontroller reads the rotational data from the sensor attached to the globe.

3. **Data Transmission to Core Application:** The captured audio stream and the serialized rotational data from the Arduino are transmitted in real-time to the central application running in the Unity engine.

4. **The Agentic Core (Reasoning and Planning):** Within Unity, the core logic orchestrates the agent's reasoning process. The application sends the transcribed user query and the relevant contextual data (e.g., globe orientation) to the OpenAI API. The generative agent, guided by its engineered prompt, interprets the user's intent, formulates a plan, and identifies the appropriate ”tool” to use—a predefined C# function within the Unity simulation.

5. **Simulation and Visualisation:** The agent invokes the corresponding function in the simulation engine, passing the translated parameters (e.g., $i = 45°$, $\Omega = \text{calculated\_value}$). The Unity engine calculates the orbital trajectory based on these parameters and renders the path as a 3D visualisation on the Meta Quest 3, correctly synchronised with the virtual Earth's orientation.

6. **Auditory Feedback Loop:** To complete the interaction, the agent generates a textual confirmation or explanation (e.g., “Certainly, here is the 45-degree inclination orbit you requested.”). This text is sent to the ElevenLabs API, which synthesizes a natural-sounding voice response that is played back to the user through the headset, providing coherent, conversational feedback (Anthropic, 2024).

3.3 Core Component Implementation

The architecture described above is realized through the implementation of three distinct, yet deeply integrated, core components. This section details the specific role, planned tools, and development process for each component, clarifying how they contribute to the project's objectives.

3.3.1 The Generative Agent (The "Brain")

Role The generative agent serves as the central intelligence of the system. Its primary role is to act as a natural language interface and reasoning engine, translating the user's high-level, often ambiguous, spoken intent into the precise, deterministic commands required by the simulation engine. It functions as an interactive, educational guide for the user.

Tools The agent's capabilities will be powered by a suite of Application Programming Interfaces (APIs). The core reasoning and language understanding will be handled by the **OpenAI API**, leveraging a powerful Large Language Model (LLM) with function-calling capabilities. The agent's voice, providing auditory feedback, will be synthesized using the **ElevenLabs API**, chosen for its ability to generate high-quality, low-latency speech.

Process The development will focus on prompt engineering to define the agent's persona and behaviour as an expert aerospace tutor. The agent will be provided with a schema of the available C# functions within the Unity environment, effectively giving it a set of "tools" it can use. When a user issues a command, the agent will reason about the intent and select the appropriate function to call with the correct parameters. The challenge of LLM hallucination is acknowledged; mitigation strategies, such as providing contextually relevant information within the prompt and validating outputs, will be explored and refined during the iterative development cycles.

3.3.2 The Simulation and AR Visualisation (The "World")

Role This component is responsible for creating a real-time, physically-grounded, and visually intuitive representation of the orbital environment. It must accurately simulate celestial motion and render the results in an interactive, three-dimensional space for the user.

Tools The simulation and visualisation will be built using the **Unity** 3D development engine, chosen for its robust cross-platform capabilities and extensive support for XR development. The target hardware for deployment is the **Meta Quest 3**, selected for its high-resolution, full-colour passthrough, which is ideal for Augmented Reality (AR), and its powerful standalone processing capabilities.

Process The simulation's physics will be implemented in Unity using C# scripts. The core logic will be based on the principles of astrodynamics, primarily the two-body problem, to calculate orbital trajectories as described in the foundational literature(Curtis, 2020). The primary development goal is an AR experience, where the orbital visualisations are overlaid onto the user's real-world environment and anchored to the physical globe. The modular nature of the design, however, allows for the future extension to a fully immersive Virtual Reality (VR) mode within the same application.

3.3.3 The Embodied Interface (The "Body")

Role The embodied interface is the physical bridge between the user and the digital simulation. Its role is to capture the user's physical actions—specifically, the rotation of a globe—and convert them into a stream of digital input, enabling a tangible and intuitive method of controlling the simulation's orientation.

Tools The hardware for this interface will consist of a physical **Globe**, an **Arduino** microcontroller (or a compatible equivalent) to process sensor data, and a rotational sensor.

Process A rotational sensor will be physically integrated with the globe's axis of rotation. The initial development will explore the use of a **Potentiometer** for its simplicity in measuring single-axis rotation. Concurrently, an **Inertial Measurement Unit (IMU)** will be investigated as a more capable alternative that could provide multi-axis rotational data (pitch, roll, and yaw), offering a more expressive degree of control. The Arduino will be programmed to read the data from the chosen sensor and transmit it via a serial (USB) connection to the Unity application. This data stream will directly and continuously control the orientation of the virtual celestial body in the simulation, ensuring a one-to-one correspondence between the user's physical action and the digital visualisation.

3.4 Development and Version Control

To ensure a systematic and traceable development process, the project will be managed using modern software engineering practices. The primary tool for this purpose is

GitHub, a distributed version control system. All source code, including the C# scripts for the **Unity** application and the C++ code for the **Arduino** firmware, will be stored in a centralized repository on GitHub. This approach provides a complete history of all changes, facilitates branching for experimental features without compromising the stability of the main project, and establishes a foundation for potential future collaboration. Regular commits will document the incremental progress, aligning with the iterative development philosophy outlined in Section 3.1.

3.5 Evaluation Plan

A critical component of this project is to measure its success, not only as a functional piece of software but also as a potentially effective educational tool. The evaluation plan is therefore designed to address two distinct aspects: the technical validation of the system and a qualitative assessment of its educational potential, directly addressing the final specific objective of this thesis 1.3.

3.5.1 Technical Validation

The first phase of evaluation will focus on verifying that the system's components function correctly, reliably, and efficiently. This involves a series of tests to measure key performance indicators:

- **Agent Accuracy:** Testing the generative agent's ability to correctly parse a range of spoken commands and accurately invoke the corresponding simulation functions with the correct parameters.
- **Simulation Fidelity:** Verifying that the rendered orbital trajectories are mathematically correct according to the implemented physics models.
- **System Latency:** Measuring the end-to-end latency, from user input (voice and gesture) to the corresponding visual and auditory feedback, to ensure the interaction feels responsive and seamless.
- **Interface Precision:** Assessing the accuracy of the physical interface by ensuring that the rotation of the physical globe maps precisely and smoothly to the orientation of the virtual model.

3.5.2 Educational Potential Assessment

The second phase of evaluation aims to gather qualitative data on the platform's potential as a tool for facilitating conceptual understanding of orbital mechanics. This will be achieved through a small-scale, qualitative user study.

Participants A small group of target users (e.g., 3-5 undergraduate students in aerospace engineering or a related field) will be invited to participate.

Procedure Each participant will be given a brief introduction to the system's controls and features. They will then be asked to complete a series of exploratory tasks, such as creating a geostationary orbit, visualizing a polar orbit, or performing a Hohmann transfer between two altitudes.

Data Collection Following the interactive session, feedback will be collected through a semi-structured interview and a short questionnaire. Questions will focus on the user's experience regarding engagement, ease of use, and perceived value. Specifically, participants will be asked whether the platform helped them visualize or understand orbital concepts more intuitively compared to traditional learning methods like textbooks and diagrams.

The goal of this assessment is not to achieve statistical significance, but to gather rich, qualitative insights that can validate the educational premise of the project and inform future improvements.

3.6 Chronogram

This section outlines the planned development schedule for the project. The work is broken down into five distinct phases, mapping the key implementation and evaluation tasks to a five-month timeline from July 2025 to November 2025. This chronogram serves as a high-level project plan to guide and track development progress.

Phase	Key Tasks	Timeline	Status
Phase 1: Core Simulation Engine (The "World")	• Project Setup (GitHub, Unity Project for Quest 3) • Implement Core Physics (Two-Body Problem) • Develop Visualizer & Debug UI • First AR/VR Deployment & Test	**Completed**	✓
Phase 2: Embodied Interface (The "Body")	• Hardware Prototyping (Sensor mounting) • Firmware Development (Arduino coding) • Unity-Arduino Integration (Serial communication)	**July 2025**	☐
Phase 3: Generative Agent (The "Brain")	• API "Plumbing" (OpenAI, ElevenLabs, Speech-to-Text) • Implement Agent "Tool Use" (Function calling) • Close the Interaction Loop (Voice $\rightarrow$ Agent $\rightarrow$ Sim)	**Aug - Sep 2025**	☐
Phase 4: Refinement & User Experience (UX)	• UI/UX Development (Data display, visual feedback) • AR Experience Enhancement (Anchoring, stability) • Agent Persona Refinement (Improve tutoring)	**October 2025**	☐

Phase	Key Tasks	Timeline	Status
Phase 5: Evaluation & Finalization	• Technical Validation (Testing against metrics) • Conduct User Study • Final Project Documentation & Video Demo	**November 2025**	☐

References

ACADEMY, K. **Khanmigo: Khan Academy's AI-Powered Tutor (GPT-4 Pilot)**. [*S. l.: s. n.*], 2023. `https://openai.com/blog/khan-academy/`. Khan Academy Press Release (via OpenAI), Mar. Cit. on p. 23.

ANTHROPIC. **Building Effective Agents**. [*S. l.: s. n.*], 2024. `https://www.anthropic.com/engineering/building-effective-agents`. Anthropic Engineering Blog, Dec 19. Cit. on pp. 22, 23, 28.

ATTA, G.; ABDELSATTAR, A.; ELFIKY, D.; ZAHRAN, M.; FARAG, M.; SLIM, S. O. Virtual Reality in Space Technology Education. **Education Sciences**, v. 12, n. 12, p. 890, 2022. DOI: `10.3390/educsci12120890`. Cit. on pp. 19, 20.

AZUMA, R. T. A Survey of Augmented Reality. **Presence: Teleoperators & Virtual Environments**, v. 6, n. 4, p. 355–385, 1997. Cit. on p. 17.

BATE, R. R.; MUELLER, D. D.; WHITE, J. E. **Fundamentals of Astrodynamics**. New York, NY: Dover Publications, 1971. ISBN 978-0-486-60061-1. Cit. on p. 25.

BILLINGHURST, M.; CLARK, A.; LEE, G. A Survey of Augmented Reality. **Foundations and Trends in Human–Computer Interaction**, v. 8, n. 2-3, p. 73–272, 2015. DOI: `10.1561/1100000049`. Cit. on pp. 17, 18.

CAMPOS, E.; HIDROGO, I.; ZAVALA, G. Impact of Virtual Reality Use on the Teaching and Learning of Vectors. **Frontiers in Education**, v. 7, p. 965640, 2022. DOI: `10.3389/feduc.2022.965640`. Cit. on pp. 19–21.

CARRASCO, A.; RODRIGUEZ-FERNANDEZ, V.; LINARES, R. Large Language Models as Autonomous Spacecraft Operators in Kerbal Space Program. **Advances in Space Research**, 2025. Preprint: arXiv:2505.19896. Cit. on pp. 21, 23.

CAUDELL, T. P.; MIZELL, D. W. Augmented Reality: An Application of Heads-Up Display Technology to Manual Manufacturing Processes. *In:* PROC. 25th Hawaii Int. Conf. on System Sciences (HICSS). [*S. l.*]: IEEE, 1992. p. 659–669. Cit. on p. 17.

CHEN, D. **Exploring Autonomous Agents: A Semi-Technical Dive**. [*S. l.: s. n.*], 2023. https://www.sequoiacap.com/article/autonomous-agents-perspective/. Sequoia Capital Blog, May 11. Cit. on pp. 22, 23.

CURTIS, H. D. **Orbital Mechanics for Engineering Students**. 4th. Oxford, UK: Elsevier/Butterworth-Heinemann, 2020. ISBN 978-0-08-102927-1. Cit. on pp. 24, 25, 30.

HUANG, S.; GRADY, P., *et al.* **Generative AI's Act 1: The Agentic Reasoning Era Begins**. [*S. l.: s. n.*], 2024. https://www.sequoiacap.com/article/generative-ais-act-o1/. Sequoia Capital Essay, Oct 9. Cit. on p. 22.

JOHNSON-GLENBERG, M. C. Immersive VR and Education: Embodied Design Principles That Include Gesture and Hand Controls. **Frontiers in Robotics and AI**, v. 5, p. 81, 2018. DOI: 10.3389/frobt.2018.00081. Cit. on pp. 17, 20, 21.

MILGRAM, P.; KISHINO, F. A taxonomy of mixed reality visual displays. **IEICE Trans. Information and Systems**, E77-D, n. 12, p. 1321–1329, 1994. Cit. on pp. 17, 20.

NASA. **NASA and Microsoft Collaborate to Bring Science Fiction to Science Fact**. [*S. l.: s. n.*], 2015. Press Release 15-139 (June 25, 2015). Available: https://www.nasa.gov/news-release/nasa-microsoft-collaborate-to-bring-science-fiction-to-science-fact. Cit. on p. 20.

OPENAI. **Function Calling and Other API Updates**. [*S. l.: s. n.*], 2023. https://openai.com/blog/function-calling-and-other-api-updates. OpenAI Blog, June 13. Cit. on p. 22.

PARK, J. S.; O'BRIEN, J. C.; CAI, C. J.; MORRIS, M. R.; LIANG, P.; BERNSTEIN, M. S. Generative Agents: Interactive Simulacra of Human Behavior. *In:* PROCEEDINGS of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST '23). [*S. l.: s. n.*], 2023. Cit. on p. 23.

RUTH, J. S. **Has Apple Changed VR and AR's Trajectory with the Vision Pro?** [*S. l.: s. n.*], 2024. InformationWeek (Feb. 16, 2024). Available from: https://www.informationweek.com/data-management/has-apple-changed-vr-and-ar-s-trajectory-with-the-vision-pro-. Cit. on pp. 18, 21.

VALLADO, D. A. **Fundamentals of Astrodynamics and Applications**. 4th. Hawthorne, CA: Microcosm Press and Springer, 2013. ISBN 978-1-881-88318-0. Cit. on pp. 24, 26.

VIEYRA, R.; VIEYRA, C. **Comparing Google ARCore and Apple ARKit**. [*S. l.: s. n.*], 2018. Vieyra Software Blog (Sep. 23, 2018). Available from: https://www.vieyrasoftware.net/single-post/2018/09/23/comparing-google-arcore-and-apple-arkit. Cit. on pp. 18, 19.

WORLD WIDE WEB CONSORTIUM. **WebXR Device API (Editor's Draft, May 2021)**. [*S. l.: s. n.*], 2021. W3C Working Draft. Available from: `https://www.w3.org/TR/webxr/`. Cit. on p. 19.

FOLHA DE REGISTRO DO DOCUMENTO

1. CLASSIFICAÇÃO/TIPO	2. DATA	3. DOCUMENTO Nº	4. Nº DE PÁGINAS
TC	25 de março de 2015	DCTA/ITA/DM-018/2015	37

5. TÍTULO E SUBTÍTULO:

Educational Orbit Simulation with Generative AI Agentic Workflow and Augmented Reality Visualisation

6. AUTOR(ES):

Eduardo Moura Zindani

7. INSTITUIÇÃO(ÕES)/ÓRGÃO(S) INTERNO(S)/DIVISÃO(ÕES):

Instituto Tecnológico de Aeronáutica – ITA

8. PALAVRAS-CHAVE SUGERIDAS PELO AUTOR:

AI; VR

9. PALAVRAS-CHAVE RESULTANTES DE INDEXAÇÃO:

AI; VR

10. APRESENTAÇÃO: (**X**) **Nacional** ( ) **Internacional**

ITA, São José dos Campos. Curso de Mestrado. Programa de Pós-Graduação em Engenharia Aeronáutica e Mecânica. Área de Sistemas Aeroespaciais e Mecatrônica. Orientador: Prof. Dr. Adalberto Santos Dupont. Coorientadora: Profª. Drª. Doralice Serra. Defesa em 05/03/2015. Publicada em 25/03/2015.

11. RESUMO:

Métodos educacionais tradicionais frequentemente encontram dificuldades para transmitir conceitos complexos, espaciais e dinâmicos como os da mecânica orbital. A recente convergência entre a Realidade Aumentada (RA) de consumo e os sofisticados agentes de Inteligência Artificial (IA) Generativa apresenta uma oportunidade para criar um novo paradigma de interfaces de aprendizagem intuitivas e experienciais. Este trabalho detalha o projeto, desenvolvimento e avaliação de uma plataforma educacional interativa para a exploração dos princípios da mecânica orbital. O objetivo principal do sistema é conectar a teoria de leis físicas abstratas à compreensão intuitiva, permitindo que os usuários aprendam por meio da interação corporificada. A metodologia é centrada em uma arquitetura modular que integra três componentes principais: (1) um "cérebro" agente generativo, impulsionado por Modelos de Linguagem Abrangentes, que interpreta comandos em linguagem natural e atua como um guia educacional especializado; (2) um "mundo" de simulação em tempo real e visualização em RA, construído no motor Unity para o Meta Quest 3, que renderiza trajetórias orbitais fisicamente precisas e ancoradas no ambiente do usuário ; e (3) uma interface "corpo" corporificada, consistindo em um globo físico com um sensor rotacional, que permite o controle tangível da simulação. A plataforma facilita um ciclo de interação multimodal contínuo, onde os comandos de voz e as manipulações físicas do usuário são capturados, processados pelo agente para alterar os parâmetros da simulação e refletidos na visualização em RA com feedback auditivo conversacional. Este trabalho entrega um protótipo funcional que demonstra uma nova abordagem para a educação científica, transformando dados abstratos em uma experiência manipulável e conversacional para promover uma aprendizagem exploratória e profundamente engajadora.

12. GRAU DE SIGILO:

(**X**) **OSTENSIVO** ( ) **RESERVADO** ( ) **SECRETO**