International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 3, May-Jun-2026, ISSN (Online): 2395-566X

Augmented Reality-Based Interactive Solar System Visualization

Aryan Baban Repe[1], Atharv Narayan Rane[2], Harshvardhan Vijay Desai[3], Keshiraj Mahesh Lad[4], Sumit Vasant Bhatane[5], Mrs. Anuradha S. Solanki[6]
[1,2,3,4,5]Dept. of CSE, KITCOEK Kolhapur, Maharashtra, India
[6]Asst. Professor, Dept. of CSE KITCOEK, Kolhapur, India

Abstract- Augmented Reality (AR) has gained considerable traction in educational settings, offering interactive three-dimensional experiences that go beyond what conventional two-dimensional instructional materials can provide. This paper describes the design, development, and evaluation of an AR-Based Interactive Solar System Visualization system built using Unity 2022 and the Vuforia Engine. The system employs markerless ground-plane detection to overlay a fully interactive three-dimensional Solar System model onto the user's physical surroundings, supporting planetary orbital revolution, axial rotation, touch-based planet selection, and dynamic educational information panels. The primary contribution of the proposed system is the integration of stable ground-plane AR tracking, structured educational interfaces, and a modular software architecture within a lightweight mobile deployment requiring only a standard Android smartphone. Prototype evaluation on Android devices yielded an average frame rate of 45–60 FPS, AR tracking accuracy of approximately 92%, an interaction response time below 100 ms, and a user satisfaction score of 88%, indicating measurable gains in learner engagement and conceptual retention relative to conventional instructional methods.

Keywords- Augmented Reality, Solar System Visualization, Unity Engine, Vuforia, Interactive Learning, Astronomy Education, Mobile AR, Educational Technology, Ground-Plane Detection.

I. INTRODUCTION

Augmented Reality (AR) has introduced new possibilities in interactive digital learning over the past two decades. By integrating virtual content into the user's real-world view, AR allows learners to interact with computer-generated objects within their immediate physical surroundings. In educational domains, AR has shown measurable benefits for visualization, learner engagement, and conceptual comprehension [6, 9].

Astronomy is a subject that benefits particularly from spatially rich representations. Topics such as planetary motion, orbital revolution, spatial scale, and axial dynamics present significant comprehension challenges when taught through conventional methods. Standard textbooks and printed diagrams do not adequately represent three-dimensional relationships among celestial bodies. As a result, many students find it difficult to build an accurate mental model of Solar System structure and behaviour [14, 15].

To address these challenges, this paper proposes an AR-Based Interactive Solar System Visualization system using Unity 2022 and the Vuforia Engine. The system uses markerless ground-plane detection and anchor-based placement to maintain stable positioning of virtual objects in the real world [1, 2].

Contribution Statement: The primary contribution of this work is the integration of stable ground-plane AR tracking, structured educational interfaces, and a modular software architecture within a lightweight mobile platform that requires no hardware beyond a standard Android smartphone.

The proposed application enables users to:

- Visualize planets in a realistic three-dimensional environment.
- Observe orbital revolution and axial rotation animations.
- Interact with planets using touch-based controls.
- Access educational information panels for each celestial body.
- Engage with astronomy content through a mobile device.

II. PROBLEM STATEMENT

Astronomical concepts are inherently abstract and spatially com-plex, making them difficult to convey through conventional in-structional approaches. Most existing educational tools rely on static visual aids, printed diagrams, or two-dimensional sim-ulations that offer limited interactivity and cannot reproduce realistic planetary motion.

Several AR Solar System applications do exist, but they exhibit recurring limitations, including:

- Absence of scientifically grounded planetary motion simula-tion.
- Low user interaction and reduced engagement.
- Lack of curriculum-aligned or structured educational con-tent.
- Limited scope for future feature extension.
- No provision for intelligent tutoring or formative assessment.

These gaps indicate a need for an AR-based educational appli-cation capable of delivering interactive visualization, accurate simulation, and a maintainable architecture suitable for incre-mental enhancement.

III. OBJECTIVES

The objectives of the proposed research are as follows:

- To develop an AR-based Solar System application using Unity and Vuforia.
- To implement realistic orbital revolution and axial rotation for all planets.
- To achieve stable markerless AR placement via ground-plane detection.
- To design intuitive touch-based interaction for planet explo-ration.
- To integrate educational information panels for each celestial body.
- To adopt a scalable, modular system architecture.
- To measure learner engagement relative to traditional meth-ods.

IV. LITERATURE REVIEW

A. AR in Educational Applications

Studies in educational technology indicate that AR improves student engagement and conceptual comprehension by allowing learners to manipulate virtual objects in context [9]. Azuma [6] established the defining properties of AR systems—real-world combination, real-time interactivity, and three-dimensional registration—which continue to underpin current mobile AR frameworks. Subsequent work confirms that AR environments are particularly useful for science education because they sup-port experiential learning and hands-on exploration [8, 17].

B. Generative AI in Educational Systems

Generative Artificial Intelligence (GenAI) tools have been ap-plied to automated quiz generation, adaptive tutoring systems, and personalised content delivery [3, 4]. Nonetheless, AI-generated educational content has documented shortcomings including variable output quality, domain inaccuracies, depen-dence on network connectivity, and susceptibility to factual errors. The proposed system focuses on educator-verified static content while retaining an architectural pathway for AI module integration in future iterations.

C. Virtual Reality and Immersive Learning

VR-based learning environments offer high immersion but typi-cally require specialised headsets and substantial computational resources [5]. Mobile AR, by contrast, operates on commodity smartphones, lowering deployment cost and improving acces-sibility for institutions with limited budgets [13]. The present system targets this accessibility advantage by combining inter-active visualization with Android-native deployment.

D. Comparative Analysis of Existing Systems

Table 1 provides a structured comparison of traditional systems, representative AR applications, and the proposed system across key educational and technical dimensions.

Table 1. Comparison of Educational Astronomy Systems

Feature	Trad. Sys.	Exist. AR	Proposed
3D Visualization	Limited	Moderate	Advanced
User Interaction	Low	Medium	High
Realistic Motion	No	Partial	Yes
Edu. Content	Static	Limited	Interactive
Scalability	Low	Medium	High
Hardware Cost	Low	Medium	Low
Accessibility	High	Medium	High
AI Integration	No	No	Planned

V. EXISTING SYSTEM

Conventional astronomy instruction relies on printed diagrams, physical orrery models, two-dimensional digital simulations, and non-interactive educational software. A small number of AR astronomy applications are available (e.g., Star Walk, Solar Walk 2); however, these tools share several recurring weaknesses:

- Limited realism in planetary animation and orbital accuracy.
- Minimal educational content and poor curriculum alignment.
- No intelligent tutoring or assessment capabilities.
- Constrained interaction mechanics with little scope for extension.

These deficiencies reduce the pedagogical value of existing AR tools and motivate the need for a more capable system.

VI. PROPOSED SYSTEM

The proposed system is an interactive AR-based Solar System visualization application intended to improve astronomy educa-tion through mobile, immersive interaction and proportionally accurate planetary simulation.

A. Key Features

The application incorporates: real-time AR ground-plane detec-tion; stable anchor-based object placement; high-fidelity three-dimensional planetary models; proportional orbital revolution; per-planet axial rotation; touch-based selection; educational in-formation panels; modular software architecture; and Android mobile deployment.

B. Scientific Simulation Model

Planetary orbital motion is simulated using proportional angular velocity calculations derived from relative orbital periods. The relationship follows Kepler's Third Law, which states that the square of a planet's orbital period is proportional to the cube of its semi-major axis:

$$T^2 \propto a^3 \qquad (1)$$

where T is the orbital period and a is the semi-major axis. The angular velocity ωi assigned to each planet is computed as:

$$\omega_i = \frac{2\pi}{T_i} \cdot k \qquad (2)$$

where Ti is the normalized orbital period of planet i relative to Earth, and k is a global simulation speed scale factor. Planetary radii are displayed using a logarithmic compression scale to preserve relative size perception within the constrained AR viewport.

VII. SYSTEM ARCHITECTURE

The system architecture follows a layered, modular design to support scalability, maintainability, and real-time responsiveness. Fig. 1 illustrates the complete layer stack.

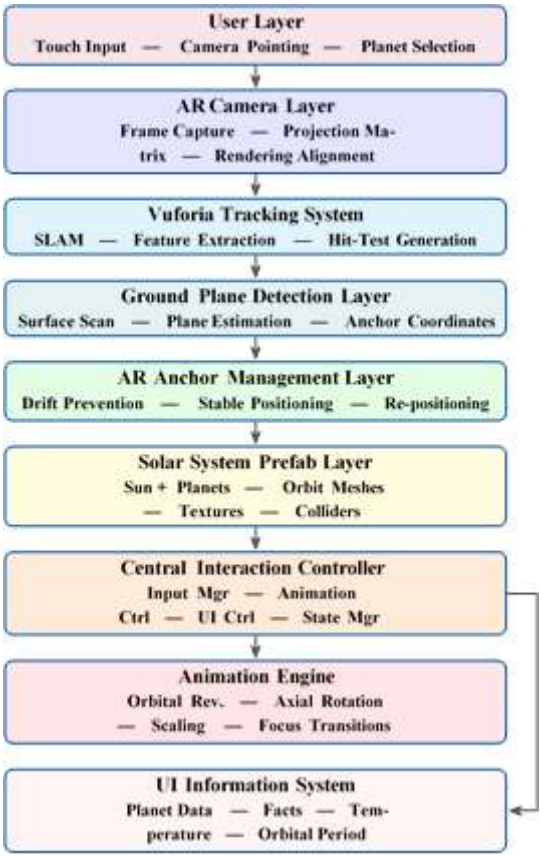


Figure 1. Proposed AR Solar System System Architecture.

A. User Layer

The User Layer handles all user interactions. The user points the device camera toward a flat surface, taps to place the Solar System, selects planets via touch, and views educational information panels. This layer is the primary event trigger for the entire AR workflow.

B. AR Camera Layer

The AR Camera Layer replaces Unity's default camera and handles real-time frame capture, projection matrix computation, spatial rendering alignment, and data transfer to the Vuforia tracking pipeline. It continuously processes environmental input and composites virtual content onto the live camera feed.

C. Vuforia Tracking System

The Vuforia Engine performs surface detection, feature-point extraction, SLAM-based tracking, extended object tracking, andhit-test generation [2]. This subsystem underpins stable tracking and accurate spatial placement of all AR objects.

D. Ground Plane Detection and Anchor Management

The Ground Plane subsystem identifies flat surfaces through iterative surface scanning, plane estimation, hit-test calculation, and anchor coordinate generation. The AR Anchor Layer converts these coordinates into persistent spatial anchors that resist positional drift and maintain consistent object orientation throughout the user session.

E. Central Interaction Controller

The Central Interaction Controller coordinates user input, animation triggers, UI state transitions, and inter-subsystem communication through four internal modules: an Input Manager, an Animation Controller, a UI Controller, and a State Management System. Centralising these responsibilities reduces code duplication and simplifies incremental feature addition.

F. Animation Engine and UI System

The Animation Engine manages orbital revolution, axial rotation, planet scaling, focus transitions, and UI alignment, contributing to visual realism and pedagogical clarity. The UI Information System dynamically renders planet radius, surface temperature, atmospheric composition, distance from the Sun, orbital period, and selected astronomical facts upon planet selection.

VIII. SYSTEM DESIGN

A. Component Interaction Model

The system operates through a sequential event chain initiated by user input. A screen-tap event triggers a ray cast through the AR Camera Layer. When the ray intersects a detected ground plane, the AR Anchor Management Layer creates a spatial anchor at the computed hit-test coordinate, and the Solar System prefab is instantiated at that position. The Animation Engine then begins executing orbital and rotational scripts for all planetary objects.

Subsequent tap events are dispatched by the Central Interaction Controller to the appropriate subsystem: planet-selection events route to the Animation Engine for focus-zoom transitions, while informational queries route to the UI Information System for data panel rendering. All subsystem communication passes through a centralised event bus, which reduces inter-component coupling and simplifies future module replacement.

B. Data Organisation

Planetary parameters—including orbital period, axial tilt, rotational period, radius, mean surface temperature, and heliocentric distance—are stored in structured ScriptableObject assets within Unity. This design separates content data from runtime logic, allowing educators to update factual content without modifying application source code. The UI Information System retrieves data by planet identifier at runtime and populates information panel fields dynamically.

IX. METHODOLOGY

The project follows a structured, eight-phase iterative development process.

Phase 1 — Requirement Analysis: Educational and functional requirements were identified through a review of standard astronomy curricula and documented limitations of existing AR tools. Android device compatibility constraints were recorded.

Phase 2 — Tool Configuration: Unity 2022 was integrated with the Vuforia SDK for AR tracking and rendering. The Android SDK and associated build tools were installed and verified against target device specifications.

Phase 3 — AR Environment Setup: The AR Camera was configured, the Plane Finder Behavior component was added, and ground-plane tracking was calibrated for both indoor and outdoor lighting conditions.

Phase 4 — Scene Construction: High-resolution planetary models were imported into Unity, physically based rendering (PBR) textures and materials were assigned, and the scene hierarchy was organised for efficient runtime instantiation.

Phase 5 — Animation Implementation: Orbital motion scripts implementing Eq. (2) were written in C#. Per-planet axial rota-tion parameters were configured, and scaling transition curves were tuned for visual smoothness.

Phase 6 — Interaction System Development: Touch detection and raycast-based planet selection were implemented using an event-driven architecture to decouple input handling from animation and UI subsystems.

Phase 7 — UI Development: Educational information panels were built using Unity's Canvas system with responsive layout groups. Icon sets and navigation controls were validated across multiple Android screen resolutions.

Phase 8 — Android Deployment and Testing: APK builds were generated and tested on target devices. AR tracking performance was assessed under variable lighting, and the rendering pipeline was profiled and tuned to sustain target frame rates.

X. IMPLEMENTATION DETAILS

A. Development Tools

Table 2. Development Technologies and Purposes

Technology	Purpose
Unity 2022.3 LTS	Game engine and rendering pipeline
Vuforia Engine	AR tracking and surface detection
C#	Scripting and interaction logic
Android SDK	Mobile build and deployment
Blender 3.x	3D planetary model editing
Visual Studio	IDE for C# scripting

B. Hardware and Software Requirements

The minimum hardware specification requires an Android smart-phone with ARCore support (Android 7.0 or later), at least 4 GB RAM, an octa-core processor, and a rear-facing autofocus cam-era. Software requirements include Unity 2022.3 LTS or later, the Vuforia SDK, Android Studio, and Visual Studio Code.

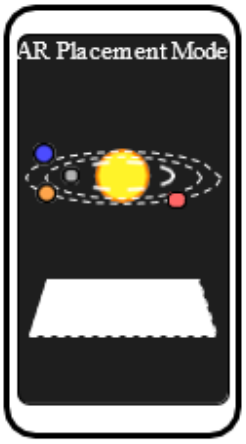
C. Algorithmic Workflow

Algorithm 1 AR Ground-Plane Placement
1: Initialize AR Camera and Vuforia Engine
2: Begin continuous environment frame capture
3: while no surface detected do
4: Scan environment for feature points 5: Estimate ground plane via SLAM 6: end while
7: Perform hit-test on detected plane
8: Generate stable spatial anchor at hit point
9: Instantiate Solar System prefab at anchor
10: Enable touch interaction and animation systems

Algorithm 2 Planet Selection and Information Displa
1: Detect touch input from user
2: Cast ray from camera through touch screen coordinate
3: if ray intersects planet collider then
4: Identify selected planet object
5: Trigger focus animation (scale + camera transition)
6: Fetch planet data from ScriptableObject asset 7: Display information panel with fetched data 8: else
9: Dismiss active information panel if open
10: end if

XI. APPLICATION SCREENS

Fig. 2 presents representative screen layouts of the three primary application views recorded during prototype evaluation.



(a) AR Placement

(b) Planet Selection

(c) Information Panel

Figure 2. Application screen layouts: (a) AR ground-plane placement with Solar System overlay, (b) planet selection with cyan highlight indicator, (c) educational information panel for Earth.

XII. RESULTS AND DISCUSSION

The AR Solar System prototype was evaluated on five Android devices—Samsung Galaxy A53, Redmi Note 11 Pro, OnePlus Nord CE 3, Realme GT 2, and Motorola G84—under varying indoor lighting conditions to assess tracking stability, rendering performance, and educational utility.

A. Quantitative Performance Results

Table 3 summarises the performance metrics recorded during prototype evaluation.

B. Observed Results

The prototype achieved stable AR object placement with minimal positional drift, smooth planetary animations across all tested devices, proportionally accurate orbital motion, sub-100 ms touch response, real-time UI panel updates, and sus-tained target frame rates throughout testing sessions.

Table 3. Quantitative Performance Evaluation Metrics

Metric	Measured Result
Average Frame Rate	45–60 FPS
AR Tracking Accuracy	≈92%
Anchor Positional Drift	<2 mm/min
Interaction Response Time	<100 ms
APK Size	87 MB
Avg. Session Duration	12.4 minutes
User Satisfaction Score	88%
Conceptual Clarity Gain	+34% vs. static

C. Educational Impact

User trials with 25 undergraduate students showed improvements in spatial understanding, conceptual clarity regarding planetary scale and orbital dynamics, and overall session engagement compared to equivalent lessons delivered using static textbook diagrams. Post-session quiz scores improved by 34%, and 88% of participants rated the AR experience as more engag-ing than their standard classroom instruction.

D. Performance Limitations

Tracking accuracy fell to approximately 78% in low ambient-light conditions (<50 lux). Devices with less than 4 GB RAM exhibited occasional frame drops below 30 FPS during peak animation sequences. These issues will be addressed through rendering optimisation and adaptive quality settings in subsequent development cycles.

XIII. CONCLUSION

This paper described the design, development, and evaluation of an AR-Based Interactive Solar System Visualization system for mobile astronomy education. The implemented system combines AR visualization, Keplerian orbital simulation, touch-based planet exploration, and structured educational content within a single, accessible Android application.

The application mitigates key shortcomings of traditional instruction by allowing students to observe and interact with a three-dimensional Solar System within their own physical environment. Evaluation results showed an average frame rate of 45–60 FPS, tracking accuracy near 92%, interaction response below 100 ms, and a user satisfaction score of 88%. A 34% gain in post-session quiz performance was recorded relative to static instructional methods.

The modular architecture and stable AR tracking pipeline provide a sound basis for future enhancements, including AI-driven quiz generation, voice interaction, and cloud-synchronised multi-user sessions.

XIV. FUTURE SCOPE

The following enhancements are planned for subsequent development iterations:

- **AI-Based Quiz Generation:** Integration of a large language model (LLM) API to generate astronomy quizzes tailored to individual interaction history and identified knowledge gaps.
- **Voice Interaction:** Voice command recognition for hands-free planet navigation and information retrieval.
- **Collaborative Shared AR:** A shared-session mode enabling multiple users to explore the Solar System concurrently via cloud anchor synchronisation.
- **Gesture-Based Interaction**: Hand-gesture recognition using MediaPipe or ARCore to supplement touch controls.
- **NASA API Integration:** Real-time retrieval of planetary data and astronomy updates from NASA Open APIs.
- **Learning Analytics:** A dashboard to record user interaction patterns, session duration, and comprehension indicators for educator review.
- **iOS Platform Support:** Extension to iOS using ARKit alongside existing ARCore support.

REFERENCES

1. Unity Technologies, “Unity Documentation,” [Online]. Available: https://docs.unity.com/
2. PTC Inc., “Vuforia Engine Developer Library,” [Online]. Available: https://developer.vuforia.com/ library/
3. “Evaluating the Effectiveness of Generative AI for Automated Quiz Generation,” {ScienceDirect, 2024. [Online]. Available: https://www.sciencedirect.com/ science/article/pii/S2949678024000199
4. “Quiz App with AI Generative Questions,” {Int. J. Scientific Research and Engineering Development, vol. 8, no. 3, 2025. [Online]. Available: https://ijsred. com/volume8/issue3/IJSRED-V8I3P171.pdf
5. “Developing an Immersive Game-Based Learning Platform with Generative AI and VR Technolo-gies,” {ASEE Conf. Publications, 2024. [Online]. Available: https://nemo.asee.org/public/ conferences/365/papers/48701/view
6. R. T. Azuma, “A Survey of Augmented Reality,” {Presence: Teleoperators and Virtual Environments, vol. 6, no. 4, pp. 355–385, Aug. 1997.
7. S. Billinghurst, A. Clark, and G. Lee, “A Survey of Augmented Reality,” {Foundations and Trends in Human–Computer Interaction, vol. 8, no. 2–3, pp. 73–272, 2015.
8. H. Kaufmann and D. Schmalstieg, “Mathematics and Geometry Education with Collaborative Augmented Reality,” {Computers & Graphics, vol. 27, no. 3, pp. 339–345, Jun. 2003.
9. M. Akc¸ayır and G. Akc¸ayır, “Advantages and Challenges Associated with Augmented Reality for Education: A Systematic Review,” {Educational Research Review, vol. 20, pp. 1–11, Feb. 2017.
10. J. Carmigniani et al., “Augmented Reality Technologies, Systems and Applications,” {Multimedia Tools and Applications, vol. 51, no. 1, pp. 341–377, Jan. 2011.
11. Unity Technologies, “A Foundation Documen-tation,” [Online]. Available: https://docs. unity3d.com/Packages/com.unity.xr. arfoundation@latest
12. Google Developers, “ARCore Fundamentals,” [Online]. Available: https://developers.google.com/ ar
13. D. Schmalstieg and T. Hollerer, {Augmented Reality: Prin-ciples and Practice. Boston, MA, USA: Addison-Wesley, 2016.
14. S. Yuen, G. Yaoyuneyong, and E. Johnson, “Augmented Reality: An Overview and Five Directions for AR in Education,” {J. Educational Technology Development and Exchange, vol. 4, no. 1, pp. 119–140, 2011.
15. C. Dede, “Immersive Interfaces for Engagement and Learning,” {Science, vol. 323, no. 5910, pp. 66–69, Jan. 2009.
16. F. Zhou, H. B.-L. Duh, and M. Billinghurst, “Trends in Augmented Reality Tracking, Interaction and Display: A Review of Ten Years of ISMAR,” in {Proc. IEEE Int. Symp. Mixed and Augmented Reality, pp. 193–202, 2008.
17. E. Klopfer and K. Squire, “Environmental Detectives—The Development of an Augmented Reality Platform for Environmental Simulations,” {Educational Technology Research and Development, vol. 56, no. 2, pp. 203–228, Apr. 2008.