Global Orbit Visualization (GOV): A multi-modal tool for exploring large volumes of Earth-orbiting objects

Lucy Chen
Lucychen5@carleton.cmail.com
Carleton University

Alejandro Anguizola
alejandro@mersisgames.com
Mersis Games

David Thomas Senior Jr.
Dave@neuraledge.cloud
Neural Edge Solutions Inc.

Lesley Istead
LesleyIstead@cunet.carleton.ca
Carleton University

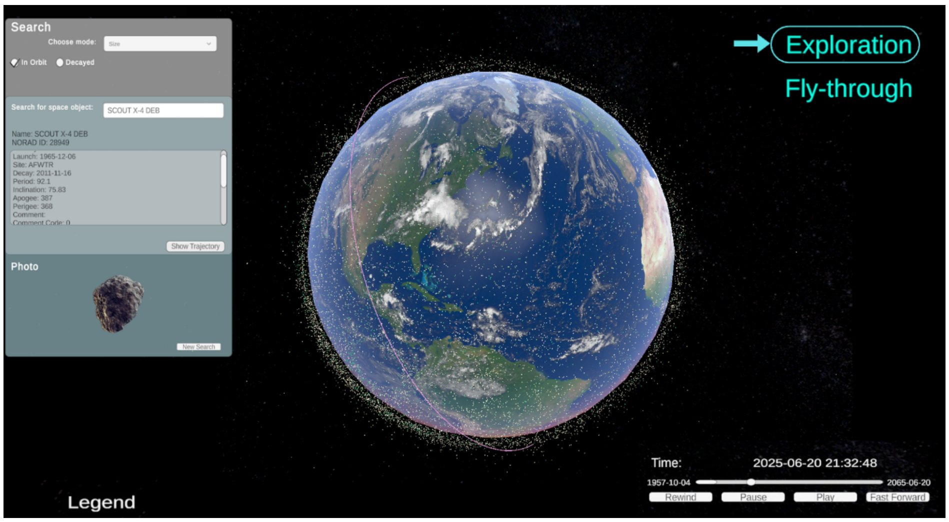

Figure 1: Trajectory visualization in Main View mode showing selected object's full orbital path around Earth activated through trajectory button in UI. Other orbiting objects depicted as tiny cubes.

Abstract

In this research we present GOV: Global Orbit Visualization tool, an interactive program that enables audiences to explore the vast number of objects (satellites, space stations, debris, etc) orbiting the Earth with a multi-modal interface. Audiences can switch between mouse-driven, text-driven, and "fly-through" interactive modes to tailor their experience and learn more about the multitude of objects in our skies. More generally, GOV demonstrates the feasibility of using gamified, multi-modal interfaces to explore large geospatial data for entertainment and education.

CCS Concepts

• **Human-centered computing → User interface design**; • **Applied computing → Interactive learning environments**.

Poster at Graphics Interface 2026 (GI '26), Waterloo, ON, Canada

Keywords

Geospatial data interaction, multi modal interaction, space education

1 Introduction

Did you know that over 57% of the nearly 58,000 tracked objects orbiting the Earth are *debris*? While these statistics are striking, visualizing the impact of this "space junk" (remnants of dead satellites, rocket bodies, and collision fragments) on future launches and existing infrastructure remains a challenge for the public. Standard 2D geospatial tools such as ArcGIS [1] and Google Maps [4] focus primarily on surface features rather than orbital data, and even robust 3D platforms like Google Earth [4] and kepler.gl [5] are not designed to render large, open orbital volumes. Additionally, contemporary research into data efficiency [2, 6] remains largely optimized for static surface data rather than the sparse, dynamic trajectories characteristic of orbital objects.

While space-specific trackers like ASTRIAGraph [3] exist, they often face performance ceilings and connectivity requirements inherent to web-based interfaces, which can limit high-fidelity rendering and interaction. To address these gaps, we present GOV: Global Orbit Visualization, an experimental interactive tool built in the Unity game engine. Using the Space-Track.org [7] datasets, GOV renders the live positions of thousands of objects in a high-performance desktop framework. This approach allows GOV to overcome the limitations of web-based tools, offering offline functionality and a multimodal experience in which users can transition among exploration, text-based metadata queries, and a first-person "fly-through" mode to create a unique educational experience.

2 Method

GOV is developed in the Unity game engine to leverage its real-time 3D rendering, interaction, and VR toolkits. The environment features an accurately scaled Earth model rotating at a sidereal rate of $0.004178°/s$ with a $23.5°$ axial tilt. The system ensures real-world alignment by computing Greenwich Sidereal Time (GST) from the current Coordinated Universal Time (UTC) time upon application launch.

2.1 Data and Position

We utilize the Space-Track.org [7] gp and `satcat` datasets to provide orbital elements (Keplerian elements, epoch) and metadata (origin, size, type). To maintain high performance and offline functionality, data is cached in a local JSON file, though live positions can be updated hourly through the API. Spatial positions are computed using classical orbital mechanics to convert Keplerian elements into Earth-centered inertial (ECI) Cartesian coordinates. The orbital radius r is determined by the conic section equation:

$$r = \frac{a(1 - e^2)}{1 + e \cos \nu} \tag{1}$$

Calculations are executed in real-time using Unity's multithreaded Jobs System and Burst Compiler, ensuring frame-rate stability across all active objects.

2.2 Rendering and Interaction

GOV offers two distinct visualization modes:

- **Main View:** A global perspective where objects are rendered as colour-coded cubes to optimize rendering time. Users can toggle views based on size, type, or region of origin, and use a text-based search to highlight specific objects and their trajectories. Clicking any orbiting object highlights its trajectory and displays its detailed metadata in an information card (see Figure 1).
- **Fly-Through:** A first-person, physics-based exploration mode (Figure 2). Objects transition from simple Level-of-Detail (LOD) meshes to detailed 3D models as the user approaches. To navigate the sparsity of space, a heads-up-display (HUD) guides users toward the nearest object, and selection triggers an automatic camera pan.

2.3 Technical Challenges

Managing high amounts of entities required several optimization strategies. Positional updates are parallelized across CPU cores to prevent performance drops. To eliminate initialization lag, objects are instantiated across multiple frames using coroutines. Finally, we utilize frustum culling and LOD meshes to reduce rendering overhead, ensuring smooth interaction during complex metadata queries.

Figure 2: Close-up rendering of a satellite in Fly-through mode with HUD campus guiding users to the nearest object to camera.

3 Conclusion

GOV demonstrates the efficacy of using high-performance game engines to bridge the gap between complex orbital mechanics and accessible geospatial visualization. By transforming abstract statistics into an immersive 3D environment, GOV combines first-person navigation with real-time data to turn static orbital information into an intuitive and engaging educational tool. Overall, GOV provides a scalable framework for space situational awareness, offering a unique perspective on the growing density of Earth's orbit and the importance of debris management for future space exploration.

References

[1] ArcGIS. [n. d.]. arcgis.com. https://www.arcgis.com/index.html. [Accessed 31-05-2025].
[2] Hannah Bast, Patrick Brosi, Johannes Kalmbach, and Axel Lehmann. 2023. Efficient Interactive Visualization of Large Geospatial Query Results. In *Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems* (Hamburg, Germany) *(SIGSPATIAL '23)*. Association for Computing Machinery, New York, NY, USA, Article 51, 4 pages. doi:10.1145/3589132.3625594
[3] Maria Esteva, Weijia Xu, Nevan Simone, Amit Gupta, and Moriba Jah. 2020. Modeling Data Curation to Scientific Inquiry: A Case Study for Multimodal Data Integration. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020* (Virtual Event, China) *(JCDL '20)*. Association for Computing Machinery, New York, NY, USA, 235–242. doi:10.1145/3383583.3398539
[4] Google Inc. [n. d.]. Google Earth — earth.google.com. earth.google.com. [Accessed 31-05-2025].
[5] Kepler.gl. [n. d.]. Large-scale WebGL-powered Geospatial Data Visualization Tool — kepler.gl. https://kepler.gl/. [Accessed 01-06-2025].
[6] Olufunso Oje, Tashi Stirewalt, Ofer Amram, Perry Hystad, Solmaz Amiri, and Assefaw Gebremedhin. 2025. HierGP: Hierarchical Grid Partitioning for Scalable Geospatial Data Analytics. *ACM Trans. Spatial Algorithms Syst.* 11, 2, Article 7 (April 2025), 20 pages. doi:10.1145/3699511
[7] U.S. Space Command. 2025. Space-Track.org. https://www.space-track.org/auth/login Accessed: 2025-06-05.

Received 30 April 2026; accepted 18 May 2026