

УДК 004.514

## **МЕТОДИ ВІЗУАЛІЗАЦІЇ РОЗРАХУНКІВ ВИВЕДЕННЯ НА ОРБИТУ ЛІТАЛЬНОГО АПАРАТУ**

Винокур О. О.

Науковий керівник – проф. Перова І. Г.

Харківський національний університет радіоелектроніки, кафедра СТ  
м. Харків, Україна

e-mail: [oleksii.vynokur@nure.ua](mailto:oleksii.vynokur@nure.ua)

Візуалізація відіграє критичну роль у космічній інженерії, надаючи інженерам засоби для інтуїтивного розуміння та аналізу складних розрахунків траєкторій та динаміки польотів космічних апаратів. У цьому контексті, ігрові графічні двигуни пропонують передові можливості для створення реалістичних та інтерактивних візуалізацій. Ця стаття досліджує використання ігрових графічних двигунів для підвищення ефективності візуалізації у космічній інженерії, акцентуючи на їх потенціалі для вирішення сучасних інженерних викликів.

Ігрові графічні двигуни, такі як Unreal Engine та Unity, пропонують вражаючі можливості для розробки візуалізацій, завдяки своїй спроможності створювати деталізовані 3D-сцени та підтримувати інтерактивність у реальному часі. Використання цих технологій в аерокосмічній інженерії відкриває нові перспективи для аналізу та представлення даних, відходячи від традиційних статичних методів візуалізації.

Детальний аналіз методології використання ігрових двигунів демонструє, як вони можуть слугувати інструментами для відображення розрахунків виведення на орбіту. Інтеграція реальних даних з моделюванням траєкторій у динамічні візуалізації дозволяє інженерам бачити потенційні проблеми та випробовувати різні стратегії місій у віртуальному середовищі. Такий підхід не тільки сприяє глибшому розумінню динаміки польоту але й відкриває можливості для оптимізації проектів місій.

Ігрові графічні двигуни, як Unreal Engine та Unity, відкривають нові можливості для космічної інженерії завдяки своїй спроможності до створення деталізованих 3D-сцен та інтерактивності. Однак, вони стикаються з викликами при виконанні розрахунків у великому масштабі, особливо у точному моделюванні місцезнаходження об'єктів на земній кулі для глобальних космічних місій.

Ці обмеження можуть стати критичними при інтеграції астрономічних даних та моделюванні траєкторій, де потрібна висока точність. Вирішення цієї проблеми потребує нових алгоритмів або розробки гібридних систем, які поєднують ігрові двигуни з традиційними інженерними програмами.

Можливим рішенням є використання спеціалізованого міжпрограмного інтерфейсу (API) для ефективної інтеграції ігрових двигунів з обчислювальними модулями, спеціалізованими на космічних розрахунках. Це дозволить зберегти візуальну привабливість ігрових двигунів, одночасно підвищуючи точність моделювання.

Ігрові графічні двигуни вносять значний вклад у розвиток космічної інженерії, пропонуючи інструменти для створення більш інтуїтивно зрозумілих та ефективних візуалізацій. Їх здатність інтегрувати великі обсяги даних і створювати реалістичні симуляції в реальному часі може значно покращити процеси планування та аналізу космічних місій. Однак, попри значні переваги, ігрові двигуни стикаються з обмеженнями, коли справа доходить до виконання розрахунків у великому масштабі.

Однією з основних проблем є виклик розрахунку місцезнаходження об'єктів на земній кулі в реальному масштабі. Це обмеження стає особливо помітним при спробі моделювання глобальних космічних місій, де точність і масштаб даних мають критичне значення. Ігрові двигуни, розроблені з орієнтацією на оптимізацію графічного відображення та інтерактивності, можуть виявитися не здатними ефективно обробляти величезні датасети або забезпечувати необхідну точність для космічних вимірювань і розрахунків.

Ці обмеження підкреслюють необхідність розвитку спеціалізованого програмного забезпечення або адаптації існуючих ігрових двигунів для космічної інженерії. Майбутнє досліджень у цій області може включати розробку нових алгоритмів, оптимізованих для космічних розрахунків, а також інтеграцію ігрових двигунів з традиційними інженерними інструментами для покращення точності моделювання та аналізу.

Список використаних джерел:

1. Amresh, Ashish, Okita, Alex, Unreal Game Development, Taylor & Francis, 2010, p. 424.
2. Joe Hocking, Joseph Hocking, Unity in Action: Multiplatform Game Development in C#. 1st edition, 2015, 352 p.
3. Alberto Fedele, Sanny Omar, Stefania Cantoni, Raffaele Savino, Riccardo Bevilacqua, Precise re-entry and landing of propellantless spacecraft, ScienceDirect, 2021, Vol. 68, Is. 11, P. 4336-4358.
4. Geng Chen, Xin Wang, Zhiyong Zhou, Lele Zhang, Shakedown analysis of a reusable space capsule, International Journal of Mechanical Sciences, Volume 244, 2023 URL: <https://doi.org/10.1016/j.ijmecsci.2022.108028> (дата звернення 06.03.2024).
5. Unreal Engine 5.0 Documentation URL: <https://docs.unrealengine.com/5.0/en-US/> (дата звернення 06.03.2024).