



Є. В. Левус¹, П. Я. Пустельник¹, М. Ю. Морозов², Р. О. Моравський¹

¹ Національний університет "Львівська політехніка", м. Львів, Україна

² Мюнхенський технічний університет, м. Мюнхен, Німеччина

МЕТОД КОНТЕЙНЕРИЗАЦІЇ ДЛЯ ВІЗУАЛІЗАЦІЇ ПРИРОДНИХ І АНТРОПОГЕННИХ ЛАНДШАФТІВ

Автоматизовані системи для візуалізації ландшафтів набули значного поширення порівняно із створенням тривимірних світів вручну через свою простоту та швидкість. Основними проблемами наявних рішень є їх недостатня реалістичність для великих масштабів, обмеженість засобів для деталізації результатів, продуктивність візуалізації. Вирішення таких проблем потребує значних апаратних ресурсів для забезпечення як якості, так і швидкості візуалізації. Побудовано метод економного використання обчислювальних ресурсів при візуалізації природних і антропогенних ландшафтів. Особливостями побудованого методу є використання дворівневого кешування та серіалізація контейнерів. Перший рівень кешу – оперативна пам'ять системи, де зберігаються останні згенеровані або завантажені контейнери. Другий рівень – дисковий простір системи, куди відбувається серіалізація контейнерів із кешу першого рівня. Використано компактний бінарний формат для серіалізації об'єктів, що дало змогу уникнути надлишкових даних, і як наслідок, зменшити витрати пам'яті. Рішення практично реалізовано у вигляді програмної бібліотеки із набором сервісів для контейнеризації тривимірних сцен LandscapeGen: Containerization. Візуалізовані системою антропогенні та природні ландшафти характеризуються відсутністю артефактів, адаптацією антропогенних об'єктів до природних ландшафтів. Значною перевагою розробленого програмного забезпечення є можливість використання згенерованих ним зображень у сторонніх застосунках. Проаналізовано ефективність побудованого методу для більше як 300 візуалізацій з різними вхідними даними, які є ключовими для визначення складності моделі візуалізації. Часові витрати на візуалізацію порівняно для випадків з кешуванням в оперативній пам'яті, у файлової системі та без кешування. Розроблений метод контейнеризації дає економію часу не менше як на 80 % для випадків як використання кешу першого, так і другого рівнів. Отримані результати засвідчують ефективність та масштабованість розробленого методу контейнеризації.

Ключові слова: контейнер; дворівневе кешування; серіалізація; часові витрати; ефективність методу.

Вступ / Introduction

Зі збільшенням попиту на використання віртуальних світів у ігровій, кіно-індустрії, науковій та освітніх сферах сучасних апаратних можливостей не завжди достатньо для того, щоб надавати необхідний рівень деталізації об'єктів, швидкості їх завантаження та плавності відображення. При цьому постає проблема оптимізації цього процесу, яку вирішують по-різному. Наприклад, розробляють власні закриті технології збереження моделей, використовують наявні можливості ігрових рушіїв [1, 2, 5, 9].

Об'єкт дослідження – візуалізація природних і антропогенних ландшафтів.

Предмет дослідження – модифікація методу візуалізації природних і антропогенних ландшафтів для

зменшення вимог до апаратного забезпечення.

Мета дослідження – побудувати метод економного використання оперативної пам'яті та тривалості роботи процесору при візуалізації природних і антропогенних ландшафтів.

Для досягнення зазначеної мети визначено такі **основні завдання дослідження**:

- 1) проаналізувати переваги та ефективність наявних методів тривимірної візуалізації природних і антропогенних ландшафтів;
- 2) розробити метод економного використання обчислювальних ресурсів при візуалізації природних і антропогенних ландшафтів;
- 3) модифікувати метод візуалізації з урахуванням методу економного використання обчислювальних ресурсів;
- 4) програмно реалізувати розроблені рішення;

Інформація про авторів:

Левус Євгенія Василівна, канд. техн. наук, доцент, кафедра програмного забезпечення. Email: yelevus@gmail.com;

<https://orcid.org/0000-0001-5109-7533>

Пустельник Павло Ярославович, магістрант, кафедра програмного забезпечення. Email: yelevus@gmail.com

Морозов Микола Юрійович, магістрант, кафедри інформатики. Email: yelevus@gmail.com

Моравський Роман Орестович, магістрант, кафедра програмного забезпечення. Email: yelevus@gmail.com

Цитування за ДСТУ: Левус Є. В., Пустельник П. Я., Морозов М. Ю., Моравський Р. О. Метод контейнеризації для візуалізації природних і антропогенних ландшафтів. Науковий вісник НЛТУ України. 2021, т. 31, № 5. С. 90–95.

Citation APA: Levus, Ye. V., Pustelnyk, P. Ya., Morozov, M. Yu., & Moravskiy, R. O. (2021). Containerization method for visualization of natural and anthropogenic landscapes. *Scientific Bulletin of UNFU*, 31(5), 90–95. <https://doi.org/10.36930/40310514>

5) проаналізувати отримані результати з точки зору якості та ефективності візуалізації.

Наукова новизна отриманих результатів дослідження – розвинуто методи контейнеризації на основі використання дворівневого кешування та бінарного формату контейнерів для зменшення витрат обчислювальних ресурсів.

Практична значущість результатів дослідження – дають змогу зменшити вимоги до апаратного забезпечення для візуалізації природних і антропогенних ландшафтів, яка здійснюється на основі побудованих алгоритмів, імплементованих як у власній системі, так і у сторонніх застосунках.

Аналіз останніх досліджень та публікацій. Із розвитком віртуальних світів, поширенням їх використання збільшуються і їх розміри, наповненість, складність та досягаються щораз нові рівні у деталізації об'єктів [1, 2, 5, 9]. Усе це потребує значних апаратних ресурсів для забезпечення швидкості візуалізації. Альтернативою є створення методів для оптимізації відображення через пришвидшення завантаження тривимірних моделей та їх ефективного збереження [6, 8]. З'являються нові продукти, які надають методи із організації процесу завантаження та збереження віртуальних світів, проте і вони містять ключові недоліки.

На світовому ринку відомі продукти-аналоги, які представляють основні технології для менеджменту завантаження тривимірних моделей та їх збереження. Одним із наявних рішень є технологія Object Container Streaming компанії Cloud Imperium Games [7]. Вона дає змогу розподілити об'єкти віртуального світу на контейнери та орієнтована на використання у багатокористувачьких комп'ютерних іграх. Суть полягає у тому, що контейнер, поруч із яким немає гравців, вивантажується із пам'яті та серіалізується. Важливо відзначити, що контейнери із об'єктами можна легко перевикористовувати, при цьому самі контейнери можуть завантажуватися асинхронно. Проте основним недоліком цієї технології є неможливість її інтегрування у сторонні системи.

Іншим продуктом-аналогом є Asset Streaming від Umbra [3]. Ця технологія передбачає розподіл віртуального світу на блоки у вигляді структури *n*-арного дерева. У блоці може міститися власне тривимірний об'єкт або певні властивості (компоненти). Важливим є те, що кожен блок дерева зберігається в окремому оптимізованому файлі, відповідно, при поділі сцени на блоки відбувається створення великої кількості файлів. Перевагою цього підходу є те, що схожі властивості різних об'єктів можуть зберігатися в одному файлі, що дає змогу ефективніше використовувати пам'ять. Істотним недоліком можна виділити те, що під час процесу візуалізації сцени велика кількість файлів із об'єктами та їх властивостями будуть одночасно відкриті.

Також продуктом-аналогом можна вважати технологію Partitioned Scene Loading, яка є частиною рушія Unreal Engine [4]. Недоліком є неможливість інтегрування цієї технології у сторонні застосунки поза рушієм Unreal Engine. Partitioned Scene Loading передбачає поділ віртуальної сцени на розділи. При цьому об'єкти світу можуть бути нерівномірно розподілені між розділами, проте реалізація цієї технології є досить простою та зрозумілою.

На основі проведеного аналізу продуктів-аналогів визначено їх недоліки, які стосуються, передусім, над-

лишковості інформації, пов'язаної з наявністю метаданих про компоненти у файлах, нерівномірного розподілу об'єктів по розділах, неможливості застосування у сторонніх програмних системах. Наведено інші важливі характеристики програмних систем візуалізації (табл. 1).

Табл. 1. Порівняння технологій контейнеризації / Comparison of containerization technologies

Критерій порівняння	Object Container Streaming від Cloud Imperium Games	Asset Streaming від Umbra 3	Partitioned Scene Loading від Unreal Engine
Підтримка рівнів деталізації об'єктів	так	ні	так
Підтримка кешування	ні	так	так
Параметричне задання відстаней завантаження	так	так	ні
Відкритий вихідний код	ні	ні	ні

Матеріали та методи дослідження. Контейнеризація сцен є складовою частиною завдання генерування та візуалізації ландшафтів, адже використовується для організації процесу збереження та завантаження об'єктів тривимірної сцени [6, 8]. Процес контейнеризації полягає у розподілі об'єктів віртуальної сцени на ієрархію контейнерів у вигляді деревовидної структури. При цьому контейнер може містити інший контейнер або власне об'єкт.

Нагадаємо, що контейнеризація (віртуалізація на рівні операційної системи, контейнерна віртуалізація, зонна віртуалізація) – метод віртуалізації, при якому ядро операційної системи підтримує кілька ізольованих екземплярів простору користувача замість одного. Ці екземпляри (зазвичай, так звані контейнерами або зонами) з погляду виконуваних у них процесів ідентичні окремому екземпляру операційної системи. Для систем на базі Unix ця технологія подібна до покращеної реалізації механізму chroot. Ядро забезпечує повну ізоляваність контейнерів, тому програми з різних контейнерів не спроможні впливати одне на іншого.

Побудований метод контейнеризації передбачає:

- 1) створення ієрархії контейнерів для заданої тривимірної сцени;
- 2) розміщення у контейнері як тривимірної моделі, так і інших контейнерів;
- 3) визначення для кожного контейнера обмежувального простору, який вміщає у собі усі тривимірні моделі контейнера;
- 4) створення дворівневого кешу для оптимізації процесу завантаження та збереження контейнерів;
- 5) зберігання об'єктів у кеші першого рівня;
- 6) зберігання об'єктів у кеші другого рівня;
- 7) зберігання у файловій системі контейнерів у бінарному форматі;
- 8) підтримка збереження та завантаження різних рівнів деталізації об'єктів всередині контейнерів.

Під час створення контейнера отримується інформація про заданий сегмент сцени із описом усіх її об'єктів. Створюється новий об'єкт контейнера і зберігається інформація про сегмент. Також для кожного об'єкта сегмента створюється власний контейнер із обрахуванням його обмежувального простору.

Алгоритм заповнення контейнера тривимірними моделями передбачає виклики методу генерування триви-

мірного об'єкта заданого рівня деталізації для контейнера та усіх його вкладених контейнерів (рис. 1).

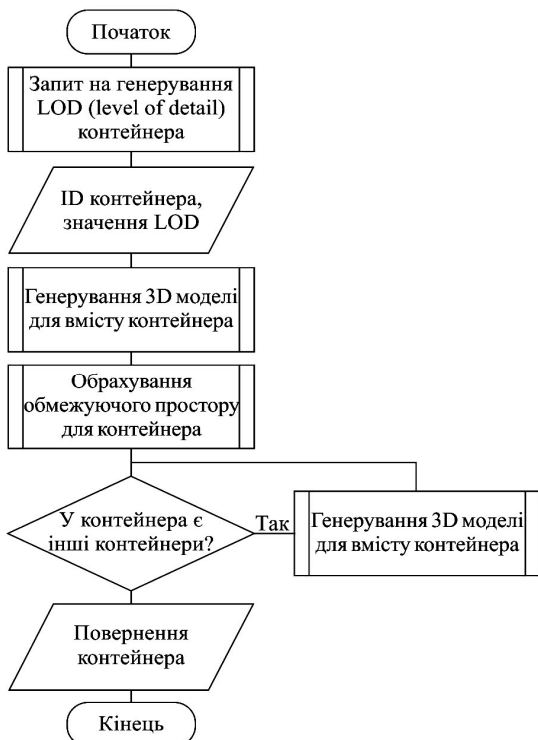


Рис. 1. Блок-схема алгоритму заповнення контейнера / Block diagram of the container filling algorithm

Контейнер подано у вигляді моделі даних, яка містить інформацію про сегмент, його тривимірну модель, поточний рівень деталізації, ідентифікатор та інші контейнери (рис. 2).



Рис. 2. Приклад ієрархії контейнерів / Container hierarchy example

Оскільки процес збереження та завантаження контейнерів є ресурсомістким, побудовано метод контейнеризації з використанням дворівневого кешування контейнерів. Перший рівень кешу – оперативна пам'ять системи, де зберігаються останні згенеровані або завантажені контейнери. Другий рівень – дисковий простір системи, куди серіалізуються контейнери із кешу першого рівня. Контейнери, які були створені чи заповнені, поміщаються у кеш першого рівня. Якщо цей контейнер певний час не запитувався із застосунку-клієнта, то його вивантажено на другий рівень. Запропоновано задавати параметри дистанції для завантаження об'єктів із різними рівнями деталізації, що забезпечує кращу оптимізацію використання оперативної пам'яті.

Процес візуалізації тривимірної сцени із контейнеризацією об'єктів з різними рівнями деталізації складається з таких кроків:

1. Візуалізація природно-антропогенного ландшафту із рухомою камерою. На ландшафті є розміщені тривимірні об'єкти, які у процесі генерації були контейнеризовані;
2. Під час руху камери відбувається перевірка того, чи хоча б частина певного контейнера знаходиться у полі зору камери;
3. Якщо частина контейнера знаходиться у полі зору камери, відбувається завантаження кореня контейнера;
4. Якщо коренем контейнера є об'єкт, відбувається його рендеринг та відображення. Якщо коренем є інший контейнер, процес повертається до кроку 3.

Розроблено програмне забезпечення для контейнеризації тривимірних сцени LandscapeGen: Containerization. Програмне забезпечення LandscapeGen: Containerization спроектовано у вигляді програмної бібліотеки із набором сервісів для контейнеризації тривимірних сцен. При цьому воно є частиною загальної системи LandscapeGen для генерування природних і антропогенних ландшафтів, хоча може використовуватись й окремо.

Архітектурою програмної системи LandscapeGen є клієнт-серверна архітектура. Серверна частина програмної системи LandscapeGen складається із трьох компонентів:

- LandscapeGen: Terrain – компонент, який містить сервіси для генерування поверхонь ландшафту за допомогою кривих і площин Без'є;
- LandscapeGen: Infrastructure – компонент, який надає сервіси для генерування антропогенних об'єктів архітектури та інфраструктури на основі готових даних про ландшафт;
- LandscapeGen: Containerization – компонент, який надає сервіси для контейнеризації тривимірних об'єктів, їх збереження та завантаження.

Описані компоненти взаємодіють між собою та надають сервіси для загального API, із яким взаємодіє клієнт. Нагадаємо, що сервіс API (від англ. *Application Programming Interface*) – це інтерфейс взаємодії між сайтом і сторонніми програмами і серверами. Програміст може скористатися API для отримання доступу до функціоналу або даних сторонньої програми. Це опис способів (набір класів, процедур, функцій, структур чи констант), якими одна комп'ютерна програма може взаємодіяти з іншою програмою. Зазвичай, входить до опису будь-якого інтернет-протоколу (наприклад, RFC), програмного каркасу (фреймворку) або стандарту викликів функцій операційної системи. Часто реалізується окремою програмною бібліотекою чи сервісом операційної системи. Використовують програмістами під час написання різноманітних додатків.

API, як інтерфейс прикладного програмування, значно спрощує процес програмування при створенні програм, абстрагуючи базову реалізацію та надаючи тільки об'єкти або дії, необхідні розробнику. Якщо графічний інтерфейс для поштового клієнта може надати користувачеві кнопку, яка виконає всі кроки для вибірки та виділення нових листів, то API для введення/виведення файлів може дати розробнику функцію, яка копіює файл з одного місця в інше, не вимагаючи від розробника розуміння операцій файлової системи, що відбуваються за лаштунками.

У реалізації об'єкта шифрування використовується бібліотека Avsc, яка містить імплементацію специфікації Avgo. Avgo використовує формат JSON для визна-

чення типів даних, проте об'єкти серіалізуються у компактному бінарному форматі, який дає змогу уникнути надлишковості даних. За цією специфікацією, для серіалізації та десеріалізації об'єктів використовуються їх схеми, що представляють собою опис усіх полів об'єктів із їх типами.

Результати дослідження та їх обговорення / Research results and their discussion

Реалізований метод контейнеризації передбачає генерацію обмежувальних просторів (рис. 3). Це дає змогу оптимізувати процес завантаження об'єктів без завантаження самих моделей, адже їх обмежувальні простори надають необхідну інформацію про позицію, відстані, розміри.

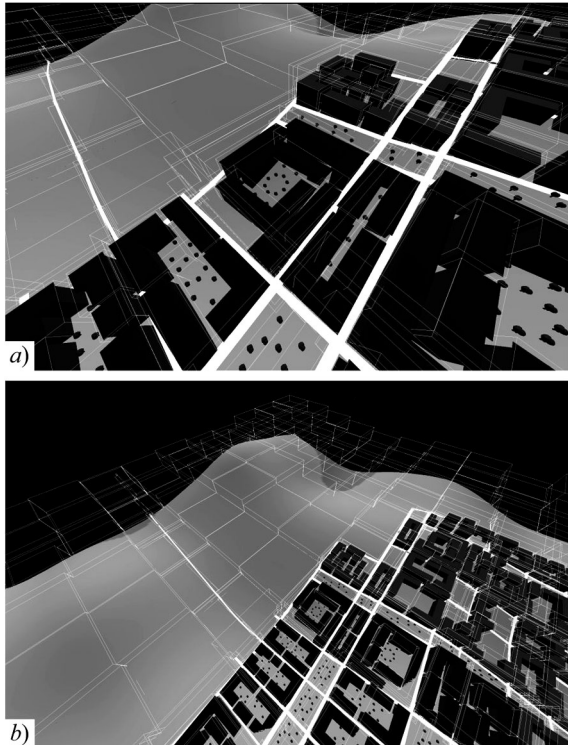


Рис. 3. Приклади відображення природних і антропогенних ландшафтів з обмежувальними просторами / Examples of reflection of small space natural and anthropogenic landscapes: a) для середньої деталізації / for medium detail; b) для низької деталізації; for low detail

Візуалізовані системою LandscapeGen антропогенні та природні ландшафти характеризуються відсутністю артефактів, адаптацією антропогенних об'єктів до природних ландшафтів. Значною перевагою розробленого програмного забезпечення є можливість використання згенерованих ним зображень у сторонніх застосунках (рис. 4).

Для аналізу часових витрат побудованого методу візуалізації проведено понад 300 різноманітних обчислювальних експериментів з візуалізації природних і антропогенних ландшафтів. Використано комп'ютер з характеристиками: CPU Intel I7 9750H, GPU Nvidia 1660TI, RAM 16GB DDR4, Drive SSD NVME2 1TB. З точки зору вимог найбільшої обчислювальної потужності локаціями, навколо яких відбувається генерація, вибрано центри міст. Серед вхідних параметрів обрано ті, які істотно впливають на час генерації, а не дають тільки візуальну різницю.

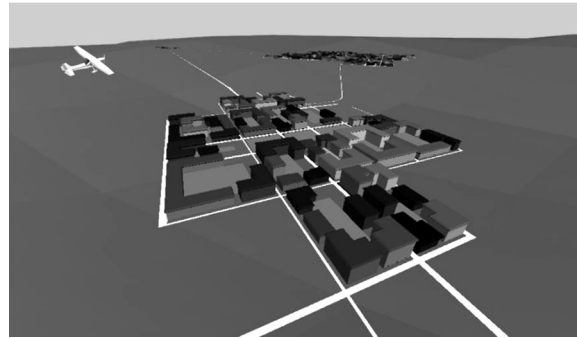


Рис. 4. Використання згенерованих антропогенних об'єктів у системі Blender / Use of generated anthropogenic objects in the Blender system

Для виявлення залежностей часу візуалізації від вхідних параметрів використано характеристику складності моделі візуалізації:

$$C_M = \frac{r \cdot d \cdot ch \cdot (lod_{max} - lod + 1)}{256}$$

де: lod_{max} – максимальний рівень деталізації; r – діапазон міста; d – максимально допустима відстань між містами у разі прокладання між ними дороги; lod – рівень деталізації; ch – кількість згенерованих сегментів.

Проведено візуалізацію для випадків: без використання кешу контейнеризації, з використанням кешу в оперативній пам'яті та на жорсткому диску. Використано $lod_{max} = 5$ як рівень перегляду композиції з найближчої точки.

Зафіксовано часові витрати для візуалізації одного сегмента для різних d, r, ch при максимально можливих їх значеннях – 4096, 1024, 25 відповідно (табл. 2). Ефект застосування вказує наскільки відсотків вдалося зменшити витрати часу на візуалізацію.

Табл. 2. Часові витрати для різних типів візуалізації / Time costs for different types of visualization

Тип візуалізації	Середній час для сегмента, ms	Ефект застосування кешу, %
Без кешу контейнеризації	68,4	-
Кеш на жорсткому диску	9,1	86
Кеш в оперативній пам'яті	0,05	99,9

Проаналізовано витрати часу на візуалізацію без кешу контейнеризації для одного сегмента при максимально допустимій кількості сегментів – 25 для різних рівнів деталізації (рис. 5) та витрати часу з кешом в оперативній пам'яті для різної кількості сегментів (рис. 6).

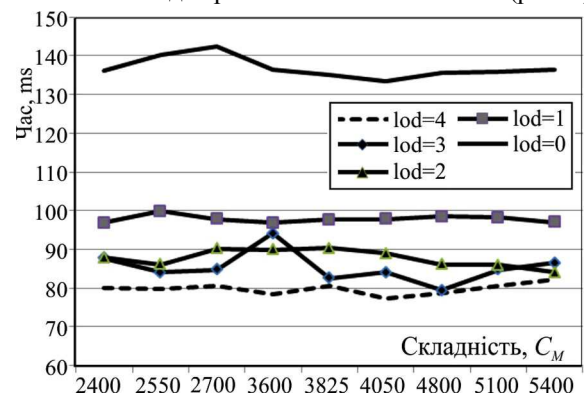


Рис. 5. Залежність часу візуалізації без кешу контейнеризації одного сегмента при візуалізації 25 сегментів від складності моделі / Dependence of visualization time without containerization cache of one segment at visualization of 25 segments on complexity of model

Нагадаємо, що кеш-пам'ять (або буферна пам'ять, буфер) – область, де зберігаються дані, які вже вважалися з вінчестера, але ще не були передані для подальшого оброблення. Там зберігається інформація, якою ОС Windows користується найчастіше. Потреба в цьому сховищі виникла через велику різницю між швидкістю зчитування даних з накопичувача і пропускною спроможністю системи. Подібним буфером мають і інші елементи комп'ютера: процесори, відеокарти, мережеві карти та ін.

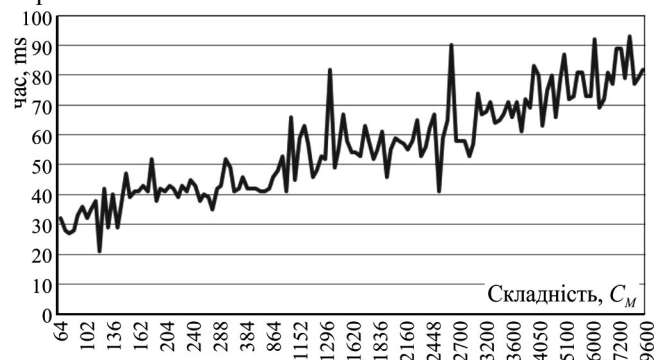


Рис. 6. Залежність часу візуалізації для одного і більше сегментів з кешуванням в оперативній пам'яті від складності моделі / Dependence of visualization time for one and more segments with caching in RAM on complexity of model

Обговорення результатів дослідження. Отримані результати роботи програмної системи LandscapeGen засвідчують ефективність розробленого методу візуалізації з використанням контейнеризації.

Середня тривалість візуалізації для проведених експериментів з кешуванням на жорсткому диску приблизно в 7 разів менша, ніж тривалість візуалізації без кешування. Водночас використання кешування в оперативній пам'яті зменшує тривалість візуалізації, порівняно з використанням жорсткого диску, приблизно у 180 разів. Ефективність застосування кешування у методі контейнеризації більше 80 %.

Найбільше часу витрачається на випадки без кешування для мінімального рівня деталізації (див. рис. 5).

Про масштабованість методу контейнеризації свідчить те, що значне збільшення складності моделі візуалізації не впливає істотно на тривалість візуалізації одного сегмента (див. рис. 6). Зокрема, збільшення складності моделі візуалізації в 100 разів впливає на збільшення часу тільки удвічі.

Запропонований метод дає змогу ефективно зберігати та завантажувати тривимірні об'єкти, що особливо важливо для врахування тенденцій збільшення обсягу тривимірних сцен та вимог до деталізації об'єктів.

Висновки / Conclusions

Актуальним дослідженням є оптимізація процесу відображення тривимірних світів, що містить ефективне зберігання та швидке завантаження урбаністичних і природних об'єктів. У роботі описано рішення, яке враховує тенденцію збільшення розміру віртуальних тривимірних світів та їх складності, деталізації, наповненості. Розроблене рішення дає змогу зменшити вимоги до обчислювальних можливостей апаратних систем.

1. Проаналізовано наявні методи контейнеризації, що використовуються у системах автоматизованої візуалізації

віртуальних світів. Серед основних недоліків методів – надлишковість інформації, нерівномірний розподіл об'єктів, що створює проблеми для оптимального використання ресурсів.

2. Розроблено метод візуалізації природних і антропогенних ландшафтів з використанням контейнеризації із дворівневим кешуванням. Метод контейнеризації містить, окрім кешування, серіалізацію та взаємодію з LandscapeGen: Infrastructure для генерування тривимірних моделей та отримання інформації про об'єкти сцени.
3. Рішення програмно імплементовано у вигляді застосунку LandscapeGen: Containerization, який функціонує як частина загальної системи генерування природних і антропогенних ландшафтів LandscapeGen, так і як самостійний застосунок, який можна легко інтегрувати у сторонні продукти.
4. Отримані результати для різних типів візуалізації (без кешу, з кешом в оперативній пам'яті, з кешом на дисковому просторі) засвідчують ефективність та масштабованість розробленого методу контейнеризації.
5. Розвиток застосунку пов'язано з легкістю розширення базового функціоналу. У майбутньому метод контейнеризації може бути розширений для розподілення окремих моделей на ієрархії контейнерів.

References

1. Biljecki, F., Kumar, K., & Nagel, C. (2018). CityGML Application Domain Extension (ADE): overview of developments. *Open Geospatial Data, Software and Standards* 3.1, 13 p. <https://doi.org/10.1186/s40965-018-0055-6>
2. Brasebin, M., Christophe, S., Jacquiod, F., Vinesse, A., & Mahon, H. (2016). 3D Geovisualization & Stylization to Manage Comprehensive and Participative Local Urban Plans. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, IV-2/W1.*, 83–91. <https://doi.org/10.5194/isprs-annals-IV-2-W1-83-2016>
3. Bushnaief, J., & Czatrowski, P. (2014). Solving Visibility and Streaming in The Witcher 3: Wild Hunt with Umbra 3. GDC Vault. Retrieved from: www.gdcvault.com.
4. Level Streaming Overview. (2021). Unreal Engine. Retrieved from: <https://docs.unrealengine.com/en-US/BuildingWorlds/LevelStreaming/Overview/index.html>.
5. Merino, L., Fuchs, J., Blumenschein, M., Anslow, C., Ghafari, M., et al. (2017). On the impact of the medium in the effectiveness of 3D software visualizations. *IEEE Working Conference on Software Visualization (VISOFT)*, 11–21. <https://doi.org/10.1109/VISOFT.2017.17>
6. Mostaghimi, H., Amour, B. St., & Abdul-Kader, Walid. (2017). Three-dimensional container loading: A simulated annealing approach. *International Journal of Applied Engineering Research*, 12(7), 1290–1304.
7. Object Container Streaming. (2021). *Star Citizen*. Retrieved from: https://starcitizen.tools/Object_Container_Streaming
8. Zhao, X., Bennell, J. A., Bektaş, T., & Dowsland, K. (2014). A comparative review of 3D container loading algorithms. *International Transactions in Operational Research*, 23(1–2), 287–320. <https://doi.org/10.1111/itor.12094>
9. Morozov, M. Iu., Levus, Ye. V., Moravskii, R. O., & Pustelnik, P. Ia. (2020). Generuvannia landshaftiv dlia sferichnikh poverkhon: analiz zavdannia ta varianti virishennia. *Scientific Bulletin of UNFU*, 30(1), 136–141. <https://doi.org/10.36930/40300124>

CONTAINERIZATION METHOD FOR VISUALIZATION OF NATURAL AND ANTHROPOGENIC LANDSCAPES

The tendency to increase the size of virtual three-dimensional worlds and their complexity, detail, and fullness causes an increase in the requirements for computing resources. Containerization typically solves the problem of optimal use of memory and processor in the three-dimensional visualization. In the course of research we analyzed the containerization methods used in computer graphics such as Container Object Streaming, Asset Streaming, and Partitioned Scene Loading. Among the disadvantages we can distinguish the following: redundancy in files, uneven distribution of objects by partitions, the impossibility of use in third-party software systems. Constructed method of containerization provides as follows: the creation of a hierarchy of containers for a given three-dimensional scene; placement in the container of both three-dimensional model and other containers; determination of a limited space for each container; creation of two-level caching to optimize the process of loading and storing containers; serialization of containers; support for different levels of detail of objects inside containers. It is proposed to set the parameters of loading distances of objects for different levels of detail, which optimizes the use of RAM. We also used a compact binary format to serialize objects, which helps avoiding redundant as well as reduced memory consumption data. Software of visualization of natural and anthropogenic landscapes that includes usage of the developed method of containerization is developed. The solution is implemented in the form of the application LandscapeGen: Containerization, which functions as part of the overall LandscapeGen system for generating natural and anthropogenic landscapes. It is possible to apply LandscapeGen: Containerization separately in cases of integration into third-party products. LandscapeGen can also be used for visualization for different cases, in particular without cache, with cache in RAM, with cache on disk space. There were analyzed time costs for more than 300 visualizations with different input data (the number of generated segments, level of detail, city range, the maximum allowed distance between cities in case of road construction). The obtained results testify to the efficiency and scalability of the developed method of containerization. It was possible to reduce time costs by more than 80 %, which reduces the requirements for computing capabilities of hardware systems.

Keywords: container; two-level caching; serialization; time costs; method efficiency.