

**ДОСЛІДЖЕННЯ МЕТОДІВ
ОПТИМІЗАЦІЇ ВІДОБРАЖЕННЯ ОБ'ЄКТІВ
У ВІДЕОІГРАХ ІЗ ВИКОРИСТАННЯМ РУШІВ
НА МОВІ ПРОГРАМУВАННЯ JAVASCRIPT**

Касперчук О.О.

e-mail: oleksandr.kasperchuk@nure.ua

Науковий керівник – доц. Назаров О.С.

Харківський національний університет радіоелектроніки, каф. ПІ
м. Харків, Україна

This work is devoted to the research of available methods of rendering optimization with the usage of game engines and libraries using JavaScript programming language. The main goal is to provide a comprehensive analysis of optimization methods for rendering, that are used in most popular video game libraries, empirically measure their performance and compare them with each other. As a result of the research, a separate library will be made, that combines the most effective methods of optimization, and it's performance would be compared against existing data.

Сьогодні розробка відеоігор є однією з найважливіших індустрій з точки зору розвитку технологій. Швидкими темпами розвивається комп'ютерна графіка, що в свою чергу спонукає розробників шукати нові методи оптимізації процесу відображення графіки. Мова програмування JavaScript є найпопулярнішою у Web середовищі, а також однією з найпоширеніших мов взагалі. Хоча існують успішні ігрові застосунки цієї мови, наразі екосистема даної мови програмування не має готових рішень, які могли б задовільнити потреби розробників та надати їм можливості для відображення найсучаснішої та найновішої графіки із достатньою швидкістю. Саме тому тема цього дослідження стає особливо актуальною, спираючись на аналіз існуючих методів оптимізації рендерингу в обраному середовищі, його аналізі, систематизації та створенні в результаті дослідження продукту, який може вирішити обрану проблему.

В якості основних були обрані емпіричні методи проведення дослідження. Це можна пояснити тим, що за їх використанням можна в рівних умовах оцінити практичну швидкість та ефективність прискорення від використання різних методів оптимізації відображення великої кількості графічних об'єктів. В процесі дослідження в рівних умовах було створено невеликий програмний застосунок, який динамічно виконує рендеринг певної кількості динамічних анімованих зображень у стані руху. Оскільки запуск та рендеринг виконуються в рівних умовах, завдяки такому методу є можливість оцінити, наскільки ефективні способи малювання використовують існуючі програмні рішення.

Для виконання дослідження було використано три найпопулярніші бібліотеки на мові програмування JavaScript за рейтингом на платформі GitHub, які підтримують малювання двовимірної графіки: Phaser.js, Pixi.js та Excalibur.js. Для кожної з них було виміряно середню кількість кадрів на секунду під час рендерингу на екрані заданої кількості спрайтів. Результати проведеного дослідження із порівнянням швидкодії при малюванні спрайтів занесені до таблиці 1.

Таблиця 1

Спрайти/Бібліотека	Phaser.js	Pixi.js	Excalibur.js
100	1020.69	3117.65	1370.85
500	1040.01	2599.27	57.43
1000	955.30	2051.55	13.54
5000	348.71	831.20	0.52
10000	139.26	481.70	0.37
50000	7.98	74.68	0.11
100000	2.00	54.26	0.01

Як можна побачити з таблиці, Pixi.js є найшвидшою з усіх трьох бібліотек, і одночасно з цим Excalibur.js є найповільнішою. Це зумовлено тим, що Pixi.js використовує декілька різних методів оптимізації, які в сукупності дають значний приріст до швидкодії. Експериментально було виявлено, що використання WebGPU надає більший приріст до швидкодії. Одночасно із цим дана бібліотека використовує такі методи, як кешування, групування об'єктів, використання атласів текстур і так-званий батчінг. Усі методи в сукупності направлені на те, щоб зменшити навантаження на GPU та кількість його викликів з програмного коду.

Варто зазначити, що використання WebGPU в Pixi.js є експериментальним. В свою чергу, Excalibur.js, як найповільніша з трьох бібліотек, наразі використовує WebGL – це менш оптимально з точки зору швидкодії. Окрім цього, батчінг в Excalibur також працює інакше, використовуючи більшу кількість невеликих за обсягом викликів для рендерингу, в той час як Pixi.js намагається якомога більше зменшити їхню кількість. Також за рахунок вищого рівня абстракції з використанням Entity-Component System, Excalibur.js створює вище навантаження на CPU, що загалом знижує швидкодію застосунку.

В результаті дослідження після проведення аналізу існуючих рішень було розроблено власну бібліотеку, яка комбінує в собі найкращі з наявних методів оптимізації, які експериментально давали найбільший прибуток до швидкодії до рендерингу з-поміж проаналізованих. Створена бібліотека повністю підтримує WebGPU, ефективно використовує шейдери, атласи текстур, батчінг, підтримує WASM і може працювати на низькому рівні з графічним процесором, при цьому надаючи зручний для розробників API.

Було проведено аналогічне дослідження з вимірюванням швидкодії. Результати порівняння швидкодії можна побачити в таблиці 2.

Таблиця 2

Спрайти/Бібліотека	Phaser.js	Pixi.js	Excalibur.js	Розроблена бібліотека
100	1020.69	3117.65	1370.85	2078.28
500	1040.01	2599.27	57.43	2201.99
1000	955.30	2051.55	13.54	2124.43
5000	348.71	831.20	0.52	1748.59
10000	139.26	481.70	0.37	1436.89
50000	7.98	74.68	0.11	589.54
100000	2.00	54.26	0.01	326.20

Розроблена графічна бібліотека ефективно інтегрує у собі методи для оптимізації з інших бібліотек, та завдяки їх суміщенню та комбінації в собі найкращих з наявних методів, дозволяє домогтися більшої швидкодії, ніж інші, значно перевищуючи швидкодію навіть якщо на екрані відображається велика кількість спрайтів.

В результаті дослідження було детально проаналізовано наявні методи оптимізації в існуючих ігрових бібліотеках для мови програмування JavaScript, виявлено найефективніші методи оптимізації в кожній з них, проведено практичну частину дослідження з їх комбінації. Завдяки використанню емпіричних методів здійснено вимірювання зростання швидкодії від їх комбінованого використання в створеній бібліотеці. З'ясовано, що обрані методи є дійсно ефективними і забезпечують значне зростання швидкодії програмних застосунків.

Список використаних джерел:

1. Назарова Н.В., Назаров О.С., Нуралієва Л.М. Дослідження методів програмної оптимізації завантаження сайтів на формування досвіду користувачів // Біоніка інтелекту. – 2025. – Т. 1, № 102. – С. 64–69. – Харків: ХНУРЕ.
2. Eric Preisz, Ben Garney "Video Game Optimization: 1st Edition" / Cengage Learning PTR – USA, 2010 – 312 с.
3. Andreas Anyuru "Professional WebGL Programming: Developing 3D Graphics for the Web: 1st Edition" / Wiley – USA, 2012 – 400 с.
4. Amy Jo Kim "Game Thinking: Innovate Smarter & Drive Deep Engagement with Design Techniques from Hit Games" / Ideapress Publishing – California, 2018 – 280 с.
5. Nicholas C. Zakas "High Performance JavaScript" O'Reilly, 2010 – 229 p.