

УДК 004.94

Застосування 3D-графіки в мережних технологіях для вирішення практичних завдань

Проценко^f І. Ю.,Оникієнко^s Ю. О., к.т.н. доц., ORCID [0000-0001-7508-8391](https://orcid.org/0000-0001-7508-8391)

Національний технічний університет України

"Київський політехнічний інститут імені Ігоря Сікорського" ROR [00syn5v21](https://ror.org/00syn5v21)

Київ, Україна

Анотація—У роботі розглянуто застосування 3D-моделей для вирішення практичних завдань у мережних технологіях. Розглянуто основні сучасні інструменти, які дозволяють впроваджувати 3D-графіку на веб-сторінках. Виконано аналіз та порівняння інструментів з точки зору зручності, зрозумілості для користувачів, які мають деякий досвід у програмуванні, але ніколи не впроваджували 3D-графіку у браузері. Вказано переваги та недоліки кожного з інструментів. Обґрунтовано вибір інструменту на основі практичного завдання.

Ключові слова — мережеві технології; браузер, 3D-графіка; WebGL; Three JS; Babylon JS.

I. ВСТУП

Веб-сайти та інтернет технології все більше впливають на наше життя. На початку зародження мережі Інтернет, сторінки містили лише статичну гіпертекстову розмітку HTML та каскадні таблиці стилів CSS. Переважні веб-сторінки представляли собою наукові статті, в яких лише існував перехід за розділами. З появою мови програмування JavaScript, сторінки стали динамічними та інтерактивними. З'явилась можливість взаємодіяти з кожним елементом сторінки та змінювати їх вигляд або дані, що містяться в них, в залежності від дій користувача. Це посприяло на розвиток сервісів, якими ми звикли користуватися сьогодні – інтернет-магазини, браузерні ігри та інше. З появою 3D-графіки у комп'ютерних іграх та програмах постало питання про впровадження 3D-графіки і у браузерах. Спершу його впроваджували за допомогою програмного забезпечення для перегляду мультимедійного контенту у браузері. Найвідоміші з них – Adobe Flash Player [1] та Microsoft Silverlight [2]. Головною проблемою була потреба у встановленні додаткового програмного забезпечення. Почалась розробка зручного для усіх браузерів прикладного програмного інтерфейсу, який вирішував би цю проблему та не залежав від зовнішніх програм. Так з'явився сучасний WebGL [3]. На основі цього API будуються уся 3D-графіка у браузерах. Спільнота та великі компанії створюють бібліотеки, які полегшують розробку 3D-моделей та розкривають можливість використання WebGL для людей, які хочуть застосувати 3D на своїх веб-сторінках.

Мета роботи – розглянути популярні інструменти та визначити мінімальні теоретичні відомості для впровадження 3D-технологій на веб-сторінках; вико-

нати порівняння інструментів, розглянути та обґрунтувати вибір того чи іншого інструменту для вирішення практичного завдання.

II. АНАЛІЗ WebGL

Для того, щоб впровадити 3D-графіку у браузері використовують API (прикладний програмний інтерфейс) веб-графіки WebGL. Даний API дозволяє виконувати рендеринг інтерактивної 2D та 3D графіки у будь-якому сумісному веб-браузері без використання додаткових плагінів.

WebGL повністю інтегрований з іншими Web-API, що дозволяє за допомогою графічного процесору використовувати фізику, обробку зображень та ефекти як частину веб-сторінки. WebGL можна поєднувати з HTML елементами та комбінувати з іншими частинами веб-сторінки чи фону сторінки. На Рис. 1 показано приклад моделі з використанням WebGL.

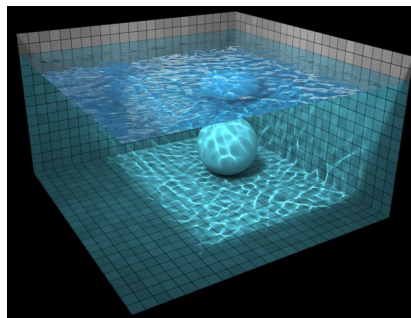


Рис. 1 Приклад WebGL



Головними перевагами WebGL є:

- Прив'язка до популярної мови програмування Javascript
- Можливість відображати контент у різних браузерах та на різних операційних системах без відмінностей
- Автоматичне управління пам'яттю
- Висока продуктивність через використання ресурсів відеокарти

Маючи зазначені переваги, WebGL є інструментом досить низького рівню та не надає можливостей для швидкого і зручного створення 3D-графіки. Це спричинило появу спеціальних бібліотек для створення таких елементів як трансформація шейдерів, можливість змінювати поле зору уявної камери, тощо. Нижче розглянуто дві популярні бібліотеки – three.js та BabylonJS.

III. АНАЛІЗ THREE.JS

Three.js – кросбраузерна бібліотека, що використовується для створення та відображення анімованої комп'ютерної графіки при розробці веб-додатків [4].

Дана бібліотека дозволяє створювати прискорену на графічному процесорі 3D графіку, використовуючи мову програмування Javascript без підключення додаткових плагінів завдяки використанню технології WebGL. На Рис. 2 наведено приклад моделі з офіційної сторінки бібліотеки.

Особливості бібліотеки:

- Можливість додавання та видалення об'єктів у режимі реального часу.
- Використання двох видів камер – перспективної та ортогографічної.
- Різні види анімації – пряма кінематика, інверсна кінематика, покадрова анімація.
- Три джерела світла – зовнішнє, направлене, точкове.
- Два види тіней – спадаючі та отримані.
- Повний доступ до усіх OpenGJ-шейдерів (GLSL).



Рис. 2 Приклад Three.js

- Об'єкти: частинки, спрайти, лінії, скелетна анімація та інше.
- Геометрія: площина, куб, сфера, тор, 3D текст і інше; модифікатори: тканина, видавлювання
- Експорт та імпорт: сумісність з JSON файлами створених у наступних програмах - Blender, openCTM, FBX, 3D Studio Max.
- Підтримка: документація по API бібліотеки знаходиться в процесі постійного розширення і доповнення, є публічний форум і велике співтовариство.
- Приклади: на офіційному сайті можна знайти більше 150 прикладів роботи зі шрифтами, моделями, текстурами, звуком і іншими елементами сцени [5].

IV. АНАЛІЗ BABYLON JS

Babylon JS використовується для тих самих задач, що і Three.js. Як зазначалося раніше, Three.js дозволяє використовувати абстракції над чистим WebGL кодом. Babylon JS у свою чергу абстрагує деталі WebGL трохи більше ніж Three.js.

Завдяки цьому, при впровадженні 3D-графіки у веб-сайт можна швидше вчитися та створювати нові сцени за менший час. Порівнюючи час створення однієї й тієї ж сцени, Babylon JS потребує приблизно на 60 % менше часу.

Так само як і Three.js, Babylon JS використовує Canvas для рендерінгу 3D-моделей за допомогою програм шейдерів.

Для симуляції фізики, Babylon JS має можливість інтегрувати плагіни фізики. Наприклад Ammo, Cannon, Oimo [6]. Також існує можливість впроваджувати власну фізику.

V. ПОРІВНЯННЯ BABYLON JS ТА THREE JS

Головним недоліком фреймворку Three JS є те, що на його вивчення до рівня складних проектів потрібно витратити багато часу та зусиль. Набагато більше, ніж для Babylon JS.

Документація Three JS якісна та містить опис усіх доступних функцій. Інформація подається досить коротко та без зрозумілих початківцю прикладів. Для людини, що тільки почала вивчати застосування 3D графіки на веб-сторінках, інформація може здатися складною для розуміння. Виникає потреба у пошуку інформації зі сторонніх джерел. На Рис. 3 наведено приклад опису функції. На Рис. 4 наведено приклад з фрагментом коду.

Як можна побачити, опис та приклад навряд будуть зрозумілі початківцю. У прикладах комплексні проекти, в яких використовується багато інших функцій, тому складно зрозуміти застосування конкретної функції навіть на прикладі. Отже потрібно витрачати час на пошук додаткової інформації.

Three JS не має власного майданчику для тестування. Існує спеціалізований редактор [7], але його функціонал незручний для користування.

Geometry

Geometry is a user-friendly alternative to [BufferGeometry](#). Geometries store attributes (vertex positions, faces, colors, etc.) using objects like [Vector3](#) or [Color](#) that are easier to read and edit, but less efficient than typed arrays.

Prefer [BufferGeometry](#) for large or serious projects.

Example

[WebGL / geometry / minecraft](#)

[WebGL / geometry / minecraft / ao](#)

Рис. 3 Приклад документації Three JS (опис функції)

```
var geometry = new THREE.Geometry();

geometry.vertices.push(
  new THREE.Vector3( -10, 10, 0 ),
  new THREE.Vector3( -10, -10, 0 ),
  new THREE.Vector3( 10, -10, 0 )
);

geometry.faces.push( new THREE.Face3( 0, 1, 2 ) );

geometry.computeBoundingSphere();
```

Рис. 4 Приклад документації Three JS (фрагмент коду)

Щоб додати нову сцену або камеру, користувачам кожен раз доводиться копіювати у редактор багато шаблонного коду.

Three JS це досить старий на сьогоднішній день фреймворк з досить неактивною спільнотою. То ж чекати відповіді на питання доведеться досить довго.

Three JS налічує близько 500+ помилок на сьогоднішній день. Вирішенням проблем займаються досить повільно. В той час як Babylon JS налічує близько 50 помилок. Фреймворк підтримується командою розробників та ініціативною спільнотою, тому помилки активно вирішуються.

На відміну від Three JS, код фреймворку Babylon JS написаний на мові програмування TypeScript, яка є абстракцією над Javascript. Це дозволяє уникати помилок та швидше виправляти баги. Код написаний на TypeScript [8] транспілюється у Javascript.

Також у Three JS відсутня повна підтримка NPM (Node Packet Manager) [9].

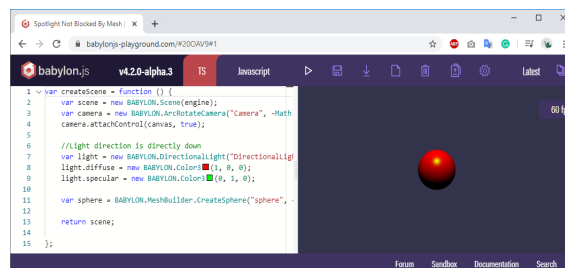


Рис. 5 Вигляд сторінки Babylon Playground

Babylon JS має власний ресурс Babylon Playground, який дозволяє створювати та тестувати 3D моделі одразу у браузері. Це дуже зручно і позитивно впливає на спільноту фреймворку. Якщо у людини що хоче впровадити 3D на свою сторінку виникають запитання або цікаві ідеї, вона може надіслати посилання, за яким іншим користувача одразу буде представлений код та результат виконання коду.

Наприклад, на Рис. 5 можна побачити модель, у якій застосовано направлене світло (світло випромінюється у визначеному напрямку та має нескінченний діапазон).

Babylon дозволяє підключати плагіни, які задають закони поведінки та взаємодії об'єктів. Без встановлення додаткових плагінів, можна одразу працювати з двигунами як Cannon JS та Oimo JS. Є можливість додавати інші двигуни, та навіть власне створену «фізику».

Отже, можна зробити наступні висновки що до двох бібліотек.

Переваги Three JS:

- Потужний та легкий за розміром фреймворк.
- Якісна підтримка графічних шейдерів.
- Багато демонстраційних прикладів.
- Якісна документація.
- Підтримка багатьох типів форматів моделей.
- Можливість розширити функціонал, додавши фізичні двигуни, такі як Cannon JS або Ammo JS.

ТАБЛИЦЯ 1 ПОРІВНЯЛЬНІ ХАРАКТЕРИСТИКИ БІБЛІОТЕК

| Назва бібліотеки | Характеристики | | | | | |
|------------------|------------------------|-----------------------------|----------------------|---|--|--------------------|
| | Мова програмування | Інтегровані фізичні двигуни | Версія WebGL | Імпорт форматів | Експорт форматів | Ліцензія |
| Three.js | JavaScript | Ні | Native (1.0 and 2.0) | glTF, DRACO, FBX, OBJ, STL, MMD, PRWM, PCD, PDB | OBJ, glTF, PLY, Collada | MIT |
| Babylon.js | JavaScript, TypeScript | Так | Native (1.0 and 2.0) | OBJ, FBX, STL, Babylon, glTF | Усі формати, що підтримуються Blender and 3dsMax (доступні для експорту) | Apache License 2.0 |



Недоліки Three JS:

- Потребує на вивчення багато часу та зусиль.
- Потреба у сторонніх ресурсах для початківців.
- Часте написання шаблонного коду.
- Старий фреймворк з неактивною спільнотою.

Переваги Babylon JS:

- Більша абстрактність над WebGL ніж у Three JS.
- Створення сцен займає менше часу ніж з Three JS.
- Простіший до засвоєння для новачків.
- Детальна документація з прикладами.
- Зручний ігровий майданчик.
- Активна спільнота, швидкість виправлення помилок.

Недоліки Babylon JS:

- Менш компактний ніж Three JS.
- Програє за популярністю Three JS.
- Вірогідність припинення підтримки через непопулярність.

На Рис. 6 представлено динаміку популярності [10] у період з березня 2019 по березень 2020 року.

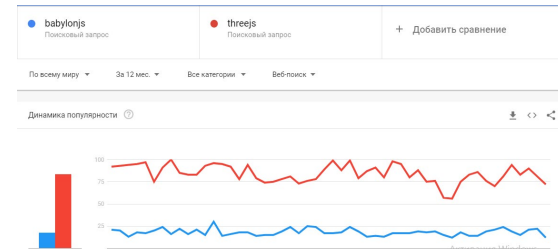


Рис. 6 Популярність запитів Three JS та Babylon JS

VI. КОНЦЕПЦІЯ САЙТУ

Застосування 3D моделей може бути корисним для вирішення практичних завдань. Концепцією пропонуваного проекту є освітня платформа для навчальних закладів.

В якості прикладу можна обрати інтерактивний посібник з геометрії для 11 класу та тему стереометрії. Платформа надає можливість обрати потрібну дисципліну. На Рис. 7 зображено макет сторінки інтерактивного посібника.

Сторінка містить навігацію по освітній онлайн платформі у верхній частині. В основній частині інтерактивного підручника знаходяться правила, закони, теореми з прикладами та практичні завдання, які полегшують розуміння матеріалу з застосуванням 3D моделей. Наприклад, користувач хоче відпрацювати завдання на знаходження об'ємів стереометричних фігур. Будується 3D модель, яка складається з компонентів, при наведенні на які можна побачити зазначені у умові параметри.

Чому доцільно використовувати 3D для вирішення подібних завдань? На прикладі освітньої платформи можна зазначити які переваги надає використання 3D технологій у веб-сторінках. Подібний освітній матеріал більше затримує увагу людей, з ним можна взаємодіяти та гратися, що позначається на якості засвоєння знань.

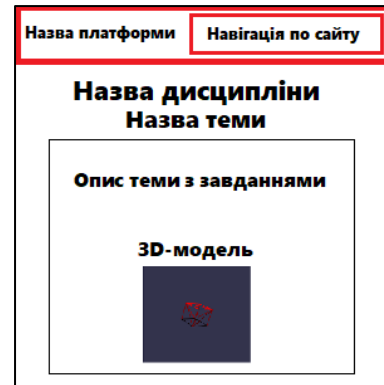


Рис. 7 Макет сторінки

ВИСНОВКИ

Для відображення 3D у браузері слугує WebGL. Головним недоліком цього API є те, що створення найпростішої моделі потребує часу та навичок для написання низькорівневого коду.

Цю проблему вирішують бібліотеки, код та команди в яких є абстракцією над кодом WebGL. Типові бібліотеки – Three JS та Babylon JS.

Переваги та недоліки обох бібліотек у розділі V. Для впровадження 3D моделей для освітньої платформи обрано бібліотеку Babylon JS, тому що дана бібліотека більш проста для освоєння новачкам, розробка займає менше часу, а також існує багато прикладів, які можна модифікувати та одразу інтегрувати у проєкт. Варто зазначити, що Babylon JS більш сучасний фреймворк, у якому регулярно виправляються помилки та існує активна спільнота. У разі виникнення запитань, можна сподіватись на відповідь інших користувачів. Використання ігрового майданчику Babylon Playground дозволяє виконати тестування коду зручно та наявно одразу у браузері.

ПЕРЕЛІК ПОСИЛАНЬ

- [1] “Adobe AIR and Adobe Flash Player Technologies,” *Adobe Labs*. [Online]. Available: <https://labs.adobe.com/technologies/flashruntimes/>.
- [2] “Microsoft Silverlight,” *Microsoft*. [Online]. Available: <https://www.microsoft.com/silverlight/>.
- [3] “WebGL 2.0 Specification,” *Khronos Group*. [Online]. Available: <https://www.khronos.org/registry/webgl/specs/latest/2.0/>.
- [4] “Three JS Documentation,” *three.js*. [Online]. Available: <https://threejs.org/docs/>.
- [5] “Three JS Examples,” *three.js*. [Online]. Available: https://threejs.org/examples/#webgl_animation_cloth.

- [6] “Babylon JS Documentation: Use a Physics Engine,” *babylon.js*. [Online]. Available: https://doc.babylonjs.com/how_to/using_the_physics_engine.
- [7] “Babylon JS Playground,” *babylon.js*. [Online]. Available: <https://playground.babylonjs.com/>.
- [8] “Typescript: The starting point for learning Typescript,” *Microsoft*. [Online]. Available: <https://www.typescriptlang.org/docs/home>.

- [9] “What is npm?,” *Refsnes Data*. [Online]. Available: https://www.w3schools.com/whatis/whatis_npm.asp.
- [10] “Google Trends: Babylon JS & Three JS,” *Google*. [Online]. Available: <https://trends.google.com/trends/explore?q=babylonjs,threejs>.

Надійшла до редакції 23 березня 2020 року.

UDC 004.94

3D-Graphics Adaptation in Network Technologies for Practice Tasks Solving

I. Yu. Protenko^f,

Yu. O. Onykiienko^g, PhD Assoc.Prof., ORCID [0000-0001-7508-8391](https://orcid.org/0000-0001-7508-8391)

National technical university of Ukraine "Igor Sikorsky Kyiv polytechnic institute" ROR [00syn5v21](https://ror.org/00syn5v21)
Kyiv, Ukraine

Abstract—This article contains a short story of technologies development and tools, which were used for displaying 3d content for web sites. WebGL, which is a basic and only one tool for displaying 3d graphics on web pages, was reviewed. An analysis of advantages and disadvantages of this tool was made. A review of modern tools for applying 3d graphics to web pages to ease interaction with webgl was made. A review of theoretical knowledge of Three JavaScript library was made. Main advantages and disadvantages were highlighted and possibilities of this library were described. A review of theoretical knowledge of Babylon JavaScript library was made. Main advantages and disadvantages were highlighted and possibilities of this library were described. A comparison between Three JavaScript and Babylon JavaScript in terms of convenience of development and implementation of 3d graphics to web pages for a beginner web developer was made. Main advantages and disadvantages were highlighted. A graph of popularity of both libraries was composed. A comparison table, which contains data about specifications, was constructed. Such as, export-import format, physical engine support, programming languages support, et cetera. A model of practice task or project, that contains 3d models, was made. The essence of the task is to create an educational online platform in which 3D graphics are implemented. This allows students to better understand the learning material. It also extends the capabilities of learning educational platforms and web pages in general. An analysis on which library is better to use for solving a task was carried out. Coming from the fact that WebGL is used to display 3D in the browser and it is difficult to work with it directly, it was decided to choose a library that simplifies the work. Despite the fact that both libraries are able to accomplish this task, Babylon JS was chosen. Babylon JavaScript library was chosen for the implementation of 3D models for the educational platform because this library is easier to learn for beginners, development takes less time, and there are many usefull examples that can be modified and integrated into the project immediately. It is worth noting that Babylon JavaScript is a more modern library, which is supported by well-known companies engaged in 3D graphics. There is also an active community that reports and suggests ways to resolve errors. So, if the user has a question, he or she can hope for help to solve the problem. Also, worth mentioning is a tool like Babylon Playground, which allows to test 3D models directly in your browser.

Key words — network technologies; web browser; 3D-Graphics; WebGL; Three JS; Babylon JS.

