

МЕТОДИ РОБОТИ З 3D-МОДЕЛЯМИ У ВЕБЗАСТОСУНКАХ

У статті проаналізовано наявні підходи та способи відображення 3D-моделей у браузері, накладання текстури на модель, а також методи створення сучасної javascript-бібліотеки, яку можна інтегрувати в будь-який вебзастосунок. Програмна частина, описана в цій роботі, являє собою фронтенд-частину для онлайн-сервісу примірювання одягу на 3D-моделях у браузері, глобальним результатом якої буде тестова версія цього сервісу.

Ключові слова: 3D-моделі у браузері, накладання текстур, javascript-бібліотека, примірювання одягу в браузері.

Вступ

Щодня безліч людей здійснюють покупки в інтернет-магазинах, де без проблем можна замовити будь-який потрібний вам товар. Проте існує проблема з товарами, для яких важливий розмір або зовнішній вигляд, а саме з одягом. Купуючи одяг в інтернеті, покупець часто не впевнений, чи підійде йому певна річ, оскільки немає способу приміряти одяг разом з іншими речами. Наприклад, покупець замовляє футболку або штани, проте він не знає, чи буде ця комбінація вдалою. Мета роботи – розв’язання цієї проблеми, а саме, дати покупцям можливість переглядати поєднання різних частин одягу (футболки і штанів) на 3D-манекені.

Постановка задачі:

- розроблення javascript-бібліотеки для відображення 3D-манекена та одягу в браузері;
- розроблення сервісу примірювання одягу на 3D-моделях у браузері, яка демонструє основні можливості бібліотеки.

Проблематика

Нині існує багато javascript-бібліотек для відображення 3D-графіки у веббраузері. Проте більшість із них потребують деяких знань у сфері 3D-візуалізації, а саме про таке:

- формати 3D-моделей;
- декартові координати у тривимірному просторі;
- карти текстур і матеріали;
- освітлення;
- камери та зону видимості.

Більшість бібліотек (детальний аналіз та порівняння – у наступному розділі) не потребують повних і глибоких знань у цій сфері, проте деякі речі можуть бути не очевидними. Через це тривіальні проблеми можуть забрати дуже багато часу, який міг піти на розроблення власного продукту.

Розв’язання проблеми

Оскільки було продемонстровано, що існує декілька сценаріїв, де ця технологія була б корисною, було вирішено створювати javascript-бібліотеку, аби цей код можна було повторно використовувати в різних проектах, тим чи тим способом пов’язаних із технологією 3D-візуалізації одягу. Також створення бібліотеки дасть змогу розділити код власного проекту на різні частини, або пришвидшити розроблення та зменшити кількість помилок під час розроблення, тестування та розгортання проекту.

Отже, було вирішено створити бібліотеку, яку було б просто інтегрувати в наявний сайт і яка б давала змогу за допомогою лише декількох методів робити таке:

- відображати 3D-моделі манекена у веббраузері;
- відправляти зображення для оброблення (видалення фону) на бекенд та отримувати готове оброблене зображення;
- відображати одяг на манекені;
- змінювати розмір одягу (футболки та штанів);
- змінювати принт (малюнок) одягу;
- прокручувати манекен навколо своєї осі, для перегляду комбінації одягу з різних сторін;
- збільшувати та зменшувати масштаб манекена та одягу.

Використання бібліотеки

Бібліотека є дуже простою. Вона, по суті, надає методи для відображення та роботи з манекеном і його одягом, тому є декілька можливих сценаріїв, де вона може бути корисною. Звісно, ці сценарії тісно пов’язані з одягом і дизайном одягу:

- інтернет-магазини одягу, де покупці відразу перед покупкою можуть переглянути ту чи ту річ, аби впевнитись, що товар їм підійде та сподобається;

- сервіс для перегляду комбінацій одягу, де, наприклад, користувач завантажує зображення власного одягу, а манекен автоматично одягається в одяг користувача;
- сервіс для створення власних принтів одягу, аби відразу перевірити, чи є принт вдалим.

Бібліотека працює разом із бекендом, який виконує такі дії:

- повертає 3D-моделі манекена та різних частин одягу через API;
- обробляє зображення одягу, видаляючи фон;
- повертає оброблені зображення на фронтенд-частину бібліотеки, для подальшого використання при відображенні одягу на манекені.

Бібліотека складається з двох частин:

- фронтенд-частини, яка займається візуалізацією 3D-моделей манекена та моделей одягу;
- бекенд-частини, яка доставляє 3D-моделі на фронтенд, а також обробляє зображення, для подальшого використання на фронтенді.

Інтегрування бібліотеки

Оскільки це javascript-бібліотека, вона з легкістю інтегрується у будь-який сучасний веббраузер. Потрібно лише імпортувати javascript-файл у свій проект. Разом із бібліотекою поставляються усі необхідні їй бібліотеки (описані в наступному розділі), тому проблем через те, що не вистачає певних залежностей, не виникне.

Вибір технологій

Для реалізації цієї задачі потрібно знайти спосіб відображення 3D-графіки у браузері. Можна обійтись стандартним javascript, проте в такому випадку веббраузеру не вистачатиме ресурсів для відображення навіть простих сцен. Тому потрібно шукати інші способи. Одним із таких способів є **WebGL** [4; 11].

WebGL – технологія, яка дає змогу впроваджувати апаратно-прискорену 3D-графіку у вебсторінки без необхідності використовувати спеціальні плагіни.

На цей момент WebGL підтримується усіма сучасними. Єдина проблема WebGL полягає у тому, що для його використання потрібно мати глибокі знання у сфері 3D-візуалізації. Саме тому існує багато бібліотек-обгорток, які використовують WebGL, проте є набагато легшими у використанні. Для розробки цього проекту було вирішено використати бібліотеку **three.js** [9].

Three.js – javascript-бібліотека для створення та відображення анімованої 3D-комп'ютерної графіки у веббраузері. Three.js має такі можливості [10]:

- ортографічна та перспективна камера;
- завантаження різних форматів 3D-моделей;
- можливість додавати, редагувати та видаляти об'єкти в режимі реального часу;
- джерела світла.

Вибір технологій для бекенд-частини бібліотеки

Бекенд-бібліотеки було вирішено писати на python, разом із бібліотеками **flask** (для створення вебсервісу [2]) та **OpenCV** (бібліотека комп'ютерного зору), оскільки існує безліч ресурсів в інтернеті, документації, а також прикладів, що допомогло у створенні майже ідеального алгоритму оброблення зображень для поставленої задачі [8].

Вибір технологій обміну інформації між фронтендом і бекендом

Для обміну інформацією між фронтенд- і бекенд-частинами бібліотеки використовують API, яким передають http-запити. Усі дані передають у форматі JSON. Для передавання зображень між фронтендом і бекендом зображення кодують у формат **Base64** [3].

Демонстрування роботи бібліотеки на прикладі сервісу примірювання одягу на 3D-моделях

Щоб продемонструвати роботи розробленої бібліотеки, було вирішено створити простий сервіс (так звану демо-версію), де можна випробувати основні можливості бібліотеки і створити зручний користувацький інтерфейс:

- відображення 3D-моделі манекена;
- відображення 3D-моделі одягу футболки та штанів у різних розмірах;
- можливість змінювати розміри футболки та штанів;
- можливість завантажувати власні зображення для оброблення та накладання на модель одягу окремо для кожної частини одягу;
- можливість змінювати масштаб моделей;
- можливість розглянути манекен з усіх сторін за допомогою смуги прокрутки.

Вибір фронтенд-технологій для сервісу

Оскільки більшість часу пішло на розроблення і тестування бібліотеки, було вирішено використовувати технології та фреймворки, з якими ми були найбільш знайомі та які дали б змогу

максимально швидко створити запропонований сервіс. Після аналізу технологій вибрали **Angular** [1; 5].

Вибір бекенд-технологій для сервісу

Маючи на меті зробити демонстрацію бібліотеки зручнішою, було вирішено створити бекенд, щоб зберігати комбінації одягу, без необхідності завантажувати та обробляти зображення знову і знову. Оскільки робота на бекенді є тривіальною, було вирішено написати бекенд на **node.js** [7].

Вибір бази даних

Для збереження завантажених і оброблених зображень одягу вибрали базу даних **MySQL**, оскільки був досвід роботи з нею [6].

Розроблення проекту

Бібліотека, яка виконує більшість роботи на проекті, – **three.js**, тож потрібно глибше розібратися в тому, як вона працює. Розглянемо діаграму, яка візуально показує архітектуру сцени в **three.js**.

Архітектура сцени three.js

Бібліотека **three.js** містить такі об'єкти (рисунк):

- рендерер – глобальний об'єкт, який безпосередньо відображує сцену в певний DOM елемент на вебсторінці;
- сцена – глобальний об'єкт, у якому дочірні елементи є 3D-об'єктами, які рендерер відобразить на певному елементі. Для кожної сцени прив'язується камера, яка виконує роль спостерігача;

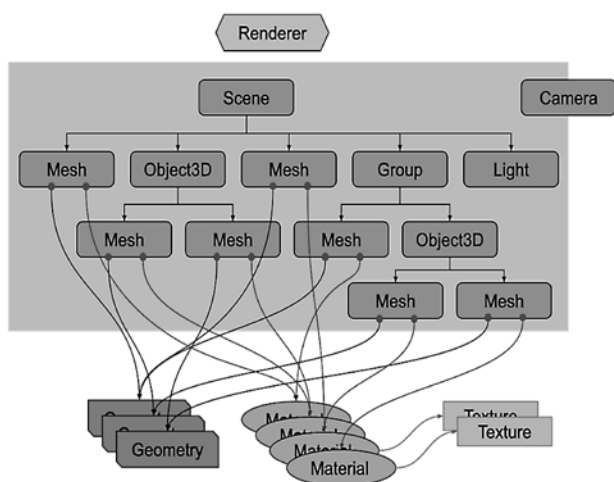


Рисунок. Архітектура сцени three.js

- камера – об'єкт, який займається усією математикою для того, щоб отримати матрицю, яка позначає положення та поворот камери у 3D-сцені. Усі операції виконуються автоматично, тому робота з камерою є інтуїтивною та не потребує глибоких знань із математики;
- текстура – зображення, яке завантажується для відображення на поверхні певного об'єкта;
- матеріал – поверхня для певного 3D-об'єкта, що має зокрема такі параметри: текстура, блиск, колір, прозорість. Ці параметри дають змогу змінювати зовнішній вигляд кожного об'єкта у сцені (зокрема зображення для одягу);
- освітлення – об'єкти, які дають змогу створювати, налаштовувати та видаляти джерела світла. Саме завдяки освітленню 3D-моделі виглядають по-справжньому та мають тінь.

Ознайомившись із документацією бібліотеки, було створено власну бібліотеку на базі **three.js**, яка має усі перелічені можливості.

Об'єднання фроненду і бекенду в одну бібліотеку

Фроненд відправляє до бекенду два зображення одягу на однорідному фоні: одне спереду та одне ззаду.

Отримавши фотографії через **http**-запит, бекенд виділяє та замінює фон на прозорий, залишаючи лише одяг. Своєю чергою фронтенд, отримавши оброблені зображення, використовує **three.js** і відображує модель манекена й модель одягу з отриманими текстурами. Як було зазначено, зображення кодуються в текстовому форматі **Base64** для простішого і зручного передавання.

Висновки

Під час виконання роботи було досліджено методи для відображення 3D-графіки у веббраузерах. У результаті було створено **javascript**-бібліотеку, написану з використанням бібліотеки для полегшення 3D-відображення **three.js**, а також бекенду, який обробляє зображення для моделей. Для тестування створеної бібліотеки створено сервіс для примірювання свого одягу на віртуальних 3D-моделях. Бібліотека може бути корисною в тих сферах, де потрібна візуалізація одягу, щоб покращити взаємодію з користувачами та клієнтами.

Список літератури

1. Angular.io [Electronic resource]. Angular – Explore angular resources. Mode of access: <https://angular.io/resources?category=development>.
2. Flask user's guide [Electronic resource]. Welcome to Flask – Flask Documentation (1.1.x). – Mode of access: <https://flask.palletsprojects.com/en/1.1.x>.
3. MDN web docs [Electronic resource]. Base64 encoding and decoding. – Mode of access: https://developer.mozilla.org/ru/docs/Web/API/WindowBase64/Base64_encoding_and_decoding.
4. MDN web docs [Electronic resource]. WebGL Web API interface. – Mode of access: https://developer.mozilla.org/uk/docs/Web/API/WebGL_API.
5. Metanit.com [Electronic resource]. Introduction to angular. What is Angular, the first project. – Mode of access: <https://metanit.com/web/angular2/1.1.php>.
6. MySQL docs [Electronic resource]. MySQL :: MySQL documentation. – Mode of access: <https://dev.mysql.com/doc>.
7. Node.js docs [Electronic resource]. Node.js documentation. – Mode of access: <https://nodejs.org/uk/docs>.
8. OpenCV docs [Electronic resource]. OpenCV: OpenCV modules. – Mode of access: <https://docs.opencv.org/4.2.0>.
9. Three.js docs [Electronic resource]. Three.js manual: Creating a scene. – Mode of access: <https://threejs.org/docs/index.html#manual/en/introduction/Creating-a-scene>.
10. Threejsfundamentals [Electronic resource]. Three.js Fundamentals. – Mode of access: <https://threejsfundamentals.org/threejs/lessons/threejs-fundamentals.html>.
11. Webglfundamentals [Electronic resource]. Introduction to WebGL. – Mode of access: <https://webglfundamentals.org/webgl/lessons/ru>.

References

- Angular.io. (n.d.). Angular – Explore angular resources. Retrieved from <https://angular.io/resources?category=development>.
- Flask user's guide. (n.d.). Welcome to Flask – Flask Documentation (1.1.x). Retrieved from <https://flask.palletsprojects.com/en/1.1.x>.
- MDN web docs. (n.d.). Base64 encoding and decoding. Retrieved from https://developer.mozilla.org/ru/docs/Web/API/WindowBase64/Base64_encoding_and_decoding.
- MDN web docs. (n.d.). WebGL Web API interface. Retrieved from https://developer.mozilla.org/uk/docs/Web/API/WebGL_API.
- Metanit.com. (15.02.2020). Introduction to angular. What is Angular, the first project. Retrieved from <https://metanit.com/web/angular2/1.1.php>.
- MySQL docs. (n.d.). MySQL :: MySQL documentation. Retrieved from: <https://dev.mysql.com/doc>.
- Node.js docs. (n.d.). Node.js documentation. Retrieved from: <https://nodejs.org/uk/docs>.
- OpenCV docs. (n.d.). OpenCV: OpenCV modules. Retrieved from: <https://docs.opencv.org/4.2.0>.
- Three.js docs. (n.d.). Three.js manual: Creating a scene. Retrieved from <https://threejs.org/docs/index.html#manual/en/introduction/Creating-a-scene>.
- Threejsfundamentals. (n.d.). Three.js Fundamentals. Retrieved from <https://threejsfundamentals.org/threejs/lessons/threejs-fundamentals.html>.
- Webglfundamentals. (n.d.). Introduction to WebGL. Retrieved from <https://webglfundamentals.org/webgl/lessons/ru>.

O. Faryna, M. Korniiuchuk

METHODS OF 3D RENDERING IN MODERN WEB-BROWSERS

This paper reviews methods of rendering 3d models in modern web-browsers, texture mapping for the model and methods of creating and integrating a javascript-library into a modern web-application. The result of this paper, is a web-application for an online service to view 3d models of clothes on a 3d model of a human in a web-browser and a demonstration of the usage of the library.

Every day, many people shop online. There are online stores for this. You can easily order any product you need from them. However, there is a problem with products for which the size or appearance of the product is important, namely clothing. When buying clothes online, the buyer is often not sure if a certain thing will suit him, because there is no way to look at clothes with other clothes. For example, the buyer orders a T-shirt or pants, but he is not sure whether the combination will be successful or not. The purpose of the work is to solve this problem, namely, to give customers the opportunity to view a combination of different pieces of clothing (T-shirts and pants) on a 3-d mannequin.

Currently, there are many javascript libraries for displaying 3D graphics in a web browser. However, most of them require some knowledge in the field of 3-d visualization, namely: formats of 3-d models; cartesian coordinates in three-dimensional space; texture maps and materials; lighting; cameras and field of view.

Although most libraries (detailed analysis and comparison in the next section) do not require complete and in-depth knowledge in this area, some things may not be obvious. Because of this, trivial problems can take a lot of time that could go into developing your own product.

As it is shown that there are several scenarios where this technology would be useful, it has been decided to create a javascript library so that this code could be reused in various projects, in one way or another related to the technology of 3D visualization of clothing. Also, creating a library will allow you to divide the code of your own project into different parts, or speed up development and reduce the number of errors during the development, testing and deployment of the project.

Keywords: 3d models in web-browsers, texture mapping, javascript-library, clothes fitting in web-browsers.



Creative Commons Attribution 4.0 International License (CC BY 4.0)

Матеріал надійшов 10.06.2020