

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра мультимедійних систем факультету інформатики

Аналіз методів роботи з 3-д моделями у веб застосуваннях.

**Текстова частина до курсової роботи за спеціальністю
„Інженерія програмного забезпечення” 121**

Керівник курсової роботи

ас. Корнійчук М. А.

(підпис)

“ ____ ” _____ 2020 р.

Виконав студент ФІ-3

Фарина О. М.

“ ____ ” _____ 2020 р.

Київ 2020

Contents

АНОТАЦІЯ.....	7
ВСТУП.....	8
РОЗДІЛ 1: БІБЛІОТЕКА ДЛЯ ВІДОБРАЖЕННЯ ОДЯГУ НА 3D МАНЕКЕНІ	10
1.1 Проблематика	10
1.2 Вирішення проблеми.....	10
1.3 Використання бібліотеки	11
1.4 Інтеграція бібліотеки.....	11
1.5 Висновки	11
РОЗДІЛ 2: АНАЛІЗ ТЕХНОЛОГІЙ, ВИКОРИСТАНИХ В РЕАЛІЗАЦІЇ БІБЛІОТЕКИ	13
2.1 Постановка задачі.....	13
2.2 Вибір технологій	13
2.2.1 Аналіз WebGL	13
2.2.2 Three.js	14
2.2.3 Babylon.js	14
2.2.4 AwayJS	15
2.2.5 Висновки та вибір бібліотеки для 3д візуалізації	15
2.2.6 Вибір технології для бекенд частини бібліотеки.....	15
2.2.7 Вибір технології обміну інформації між фронтендом та бекендом	16
2.2.8 Кодування зображення в Base64	16
2.3 Демонстрація роботи бібліотеки, на прикладі сервісу примірки одягу на 3д моделях	17
2.3.1 Постановка задачі.....	17
2.3.2 Вибір фронтенд технологій для сервісу	17
2.3.2.1 Особливості Angular.....	18
2.3.3 Вибір бекенд технологій для сервісу	19
2.3.3.1 Особливості node.js	19
2.3.4 Вибір бази даних	19

2.3.4.1 Особливості MySQL.....	20
2.4 Висновки	20
2.4.1 Бібліотека	20
2.4.2 Сервіс для демонстрації роботи бібліотеки	20
РОЗДІЛ 3: РОЗРОБКА ПРОЕКТУ, ОБЄДНАННЯ РІЗНИХ ЧАСТИН ПРОЕКТУ В КІНЦЕВИЙ ПРОДУКТ ТА ТЕСТУВАННЯ ПРОЕКТУ	22
3.1 Використання three.js	22
3.2 Архітектура сцени three.js.....	23
3.3 Обєднання фронтенду і бекенду в єдину бібліотек	24
3.4 Тестування	25
3.5 Сервіс для примірки одягу на віртуальному 3д манекені	25
3.5.1 Сторінка вибору комплекту одягу.....	25
3.5.2 Сторінка перегляду комплекту одягу.....	25
3.5.3 Сторінка адміністратора	27
3.6 Результат тестування.....	27
ВИСНОВКИ	28
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	29
ДОДАТКИ	30

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра мультимедійних систем факультету інформатики

ЗАТВЕРДЖУЮ

Олександр Петрович,

Зав.кафедри мультимедійних систем,

кандидат ф.-м.н., доцент

_____ О. О. Жежерун

(підпис)

“ ____ ” _____ 2020 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на курсову роботу

студенту 3 курсу факультету інформатики Фарина Олександр Миколайовичу

ТЕМА: Методи роботи з 3-д моделями у веб застосуваннях.

Вихідні дані:

- Дослідження можливостей Three.js в рамках створення бібліотеки веб-браузерів
- Дослідження створення javascript-бібліотеки та інтеграції бібліотеки в сайт
- Створення сервісу примірки одягу на 3д моделях в браузері для демонстрації роботи створеної бібліотеки

Зміст ТЧ до курсової роботи:

- Анотація
- Вступ
- Розділ 1. Бібліотека для відображення одягу на 3-д манекені.
- Розділ 2. Аналіз технологій, використаних в реалізації бібліотеки.
- Розділ 3. Розробка проекту, об'єднання різних частин проекту в кінцевий продукт
- Висновки
- Список використаних джерел

Дата видачі “ ____ ” _____ 2020 р.

Керівник _____
(підпис)

Завдання отримав _____
(підпис)

Тема: Методи роботи з 3-д моделями у веб застосуваннях..

Календарний план виконання роботи

№ п/п	Назва етапу курсової роботи	Термін виконання етапу	Примітка
1.	Отримання завдання на курсову роботу.	02.10.2019	
2.	Огляд технічної літератури за темою роботи.	01.11.2019	
3.	Дослідити та проаналізувати існуючі технології та фреймворки, які б допомогли при розробці бібліотеки	23.12.2019	
4.	Написання фронтенд частини бібліотеки	10.02.2020	
5.	Написання бекенд частини бібліотеки	10.02.2020	
6.	Аналіз бібліотек та фреймворків для написання сервісу для примірки одягу	29.02.2020	
7.	Написання сервісу для примірки одягу	31.03.2020	
8.	Написання курсової роботи	30.04.2020	
10.	Аналіз виконаної роботи з керівником.	11.05.2020	

Студент Фарина О. М.

Керівник Корнійчук М. А.

“ _____ ”

АНОТАЦІЯ

Мета курсової роботи є аналіз роботи бібліотек для відображення 3-д моделей в браузері. Результатом роботи даної курсової роботи є створення javascript-бібліотеки для відображення 3-д моделей в браузері, а також фронт-енд для онлайн сервісу примірки одягу на 3-д моделях в браузері з використанням даної бібліотеки, глобальним результатом якої, буде тестова версія цього сервісу.

ВСТУП

Щодня безліч людей здійснюють покупки в мережі Інтернет. Для цього існують Інтернет-магазини. У них без проблем можна замовити будь-який потрібний вам товар. Проте існує проблема з товарами, для яких важливий розмір або зовнішній вигляд товару, а саме з одягом. При купівлі одягу в інтернеті, покупець часто не впевнений, чи підходить йому певна річ, оскільки немає способу глянути одяг разом з іншим одягом. Наприклад, покупець замовляє футболку чи штани, проте він невпевнений, чи буде дана комбінація вдалою чи ні. Мета роботи – вирішення даної проблеми, а саме, дати можливість покупцям можливість переглядати поєднання різних частин одягу (футболки та штанів) на 3-д манекені.

Даний проект буде дуже корисним для Інтернет-магазинів, а саме, дає можливість покупцям переглянути свої товари на віртуальному манекені, перед тим як замовити товар, що зменшить кількість повернень товару через те, що він не сподобався покупцю. Для покупця це означає, що він має змогу бути впевненим, що одяг який він замовляє, буде підходити один до одного.

Постановка задачі:

- Розробка javascript-бібліотеки для відображення 3-д манекену та одягу в браузері
- Розробка сервісу примірки одягу на 3-д моделях в браузері, яка демонструє основні можливості бібліотеки

Роботу поділено на три розділи:

Перший розділ присв'ячений підготовці до початку роботи над бібліотекою.

Проаналізовано можливі приклади використання даної бібліотеки

Другий розділ присв'ячено аналізу технологій, використаних при розробці бібліотеки та сервісу примірки.

Третій розділ присв'ячено розробці проекту, а також об'єднання різних частин проекту в кінцевий продукт та тестуванню.

В результаті роботи курсової роботи створено javascript-бібліотеку, яку легко можна інтегрувати в майже будь-який веб сайт. Також створено сервіс примірки одягу на 3-д моделях, для демонстрації роботи створеної бібліотеки.

РОЗДІЛ 1: БІБЛІОТЕКА ДЛЯ ВІДОБРАЖЕННЯ ОДЯГУ НА 3D МАНЕКЕНІ

1.1 Проблематика

На даний момент існує багато javascript-бібліотек для відображення 3-д графіки в веб-браузері. Проте більшість із них потребують деяких знань в області 3-д візуалізації, а саме:

- Формати 3-д моделей
- Декартові координати у трьохвимірному просторі
- Карти текстур та матеріали
- Освітлення
- Камери та область видимості

Хоч і більшість бібліотек (детальний аналіз та порівняння в наступному розділі) не потребують повних та глибоких знань в цій області, проте деякі речі можуть бути не очевидними. Через те, тривіальні проблеми можуть забрати дуже багато часу, який міг піти на розробку власного продукту.

1.2 Вирішення проблеми

Саме для цього було вирішено створити бібліотеку, яку було б просто інтегрувати в існуючий сайт, і яка б дозволяла за допомогою лише декількох методів робити наступне:

- Відображення 3-д модель манекену в веб-браузері
- Відправляти зображення для обробки (видалення фону) на бекенд та отримувати готове оброблене зображення
- Відображення одягу 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 манекену та одягу

Бібліотека є інтуїтивною в використанні, достатньо лише підключити її та використовувати описані методи.

1.3 Використання бібліотеки

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

- Інтернет-магазини одягу, де покупці відразу перед покупкою можуть переглянути ту чи іншу річ, аби впевнитись, що товар їм підійде та сподобається
- Сервіс для перегляду комбінацій одягу, де, наприклад, користувач завантажує зображення власного одягу, і манекен автоматично одягається в одяг користувача
- Сервіс для створення власних принтів одягу, аби відразу перевірити чи є принт вдалим чи ні

Бібліотека працює разом із бекендом, який виконує наступні дії:

- Повертає 3-д моделі манекену 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

1.4 Інтеграція бібліотеки

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

1.5 Висновки

Оскільки було продемонстровано, що існує декілька сценаріїв, де дана технологія була б корисною, було вирішено створювати javascript-бібліотеку, аби даний код можна було повторно використовувати в різних проектах, тим чи іншим чином зв'язаних з технологією 3-д візуалізації одягу. Також створення бібліотеки

дозволить розділити код власного проекту на різні частини, або пришвидшити розробку та зменшити кількість помилок під час розробки, тестування та розгортки проекту. У наступному розділі буде описано процес вибору тих чи інших технологій при створенні бібліотеки, а також сервісу для демонстрації роботи створеної бібліотеки.

2.1 Постановка задачі

Створення javascript-бібліотеки для відображення та роботи з 3-д манекеном та його одягом. Основні вимоги до бібліотеки мають бути наступними:

- Відображення 3-д модель манекену веб-браузері
- Відправляти зображення для обробки (видалення фону) на бекенд та отримувати готове оброблене зображення
- Відображення одягу 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 манекену та одягу

Бібліотека складається з двох частин:

- Фронтенд частини, яка займається візуалізацією 3-д моделей манекену та моделей одягу
- Бекенд частини, яка доставляє 3-д моделі на фронтенд, а також обробляє зображення, для подальшого використання на фронтенді

2.2 Вибір технологій

2.2.1 Аналіз WebGL

Для реалізації даної задачі, потрібно знайти спосіб відображення 3-д графіки в браузері. Можна обійтись стандартним javascript, проте в такому випадку веб-браузеру не вистачатиме ресурсів для відображення навіть простих сцен. Тому потрібно шукати інші способи. Одним із таких способів є **WebGL**.

WebGL – технологія, яка дозволяє впроваджувати апаратно-прискорену 3-д графіку у веб-сторінки без необхідності використовувати спеціальні плагіни.

На даний момент **WebGL** підтримується усіма сучасними. Єдина проблема **WebGL** є те, що для його використання, потрібно мати глибокі знання в області 3д візуалізації. Саме тому, існує багато бібліотек-обгортки, які використовують **WebGL**, проте є набагато легшими у використанні. При використанні цих бібліотек, недоступні усі можливості, які доступні при використанні чистого **WebGL**, проте цих можливостей буде достатньо для нашого проекту. Ось варіанти, які підходять для нашого проекту і написані на основі **WebGL**:

- Three.js
- Babylon.js
- AwayJS

2.2.2 Three.js

Three.js – javascript бібліотека для створення та відображення анімованої 3-д комп'ютерної графіки у веб-браузері. Three.js включає в собі такі можливості:

- Ортографічна та перспективна камера
- Завантаження різних форматів 3-д моделей
- Можливість додавати, редагувати та видаляти об'єкти в режимі реального часу.
- Джерела світла

Це не усі можливості бібліотеки three.js, проте для нашого проекту більше можливостей нам не потрібно, лише базові. Також є багато документації про **three.js** в Інтернеті, що пришвидшує процес розробки проектів.

2.2.3 Babylon.js

Бібліотека **Babylon.js** має такі ж можливості як і three.js, в деякій мірі навіть більші можливості, проте для нашого проекту це надлишок який не буде використовуватись, лише займатиме і збільшуватиме об'єм залежностей бібліотеки.

2.2.4 AwayJS

Бібліотека **AwayJS** не сильно відрізняється від описаних вище бібліотек. Єдиною кардинально відмінністю є те, що вона написана на typescript. Оскільки ціллю було написання бібліотеки на мові javascript, дана бібліотека не повністю підходить для нашої задачі.

2.2.5 Висновки та вибір бібліотеки для 3д візуалізації

Усі бібліотеки для 3-д візуалізації є дуже схожими та мають схожі можливості, проте three.js є компактнішою ніж **Babylon.js** і написана на javascript, тому це ідеальний варіант для створення нашої бібліотеки. Ось список, де бібліотека three.js допомогла при розробці:

- Відображення 3-д моделей манекену та одягу
- Накладання текстури на модель
- Поворот 3-д моделі навколо своєї осі

2.2.6 Вибір технології для бекенд частини бібліотеки

Бекенд бібліотеки, було вирішено писати на python, разом з бібліотеками **flask** та **OpenCV**, оскільки існує безліч ресурсів в Інтернеті, документації а також прикладів, що допомогло в створенні маже ідеального алгоритму обробки зображень. Також python дозволяє писати об'ємні математичні алгоритми, в тому числі і для обробки зображень:

- **Flask** – фреймворк для створення веб-застосунків на python. Він дозволяє створити API, за допомогою якого фронтенд та бекенд обмінюються інформацією.
- **OpenCV** – бібліотека комп'ютерного зору, яка дозволяє аналізувати та обробляти зображення. Завдяки OpenCV було створено алгоритм пошуку одягу на зображення, а також відокремлення та видалення фону з

зображення для подальшого використання на фронтенді у якості текстури для моделей одягу.

2.2.7 Вибір технології обміну інформації між фронтендом та бекендом

Для обміну інформації між фронтенд частиною бібліотеки та бекенд частини бібліотеки використовується API, по якому передаються http запити. Усі дані передаються у форматі JSON. Проте залишається проблема передачі зображень між фронтендом та бекендом

2.2.8 Кодування зображення в Base64

Для передачі зображення між фронтендом та бекендом, як і для передачі будь-якої інформації між фронтендом та бекендом, використовуються запити http. Інше питання в тому, в якому форматі передавати зображення. В цілях економії ресурсів та часу на розробку, було вирішено передавати зображення відразу в тілі http-запиту, всередині JSON-об'єкту. Для цього потрібно знайти спосіб, як перетворити бінарний файл (у нашому випадку це зображення) в текст, аби мати змогу передавати в тілі JSON-об'єкту. Вирішення проблеми стало використання формату **Base64**.

Base64 – формат кодування бінарних даних за допомогою 64 символів ASCII алфавіту. Цей формат кодування підтримується усіма веб-браузерами та майже усіма веб-серверами, тому був вибраний цей формат, аби не було проблем із кодуваннями при передачі файлів через http.

2.3 Демонстрація роботи бібліотеки, на прикладі сервісу примірки одягу на 3д моделях

Аби продемонструвати роботи розробленої бібліотеки, було вирішено створити простий сервіс (так звану демо-версію) де можна випробувати основні можливості бібліотеки.

2.3.1 Постановка задачі

Продемонструвати роботу бібліотеки, а саме, створити зручний користувацький інтерфейс, де продемонстровано основні можливості бібліотеки:

- Відображення 3-д модель манекену
- Відображення 3-д модель одягу футболки ти штанів в різних розмірах
- Можливість міняти розміри футболки ти штанів
- Можливість завантажувати власні зображення для обробки та накладання на модель одягу окремо для кожної частини одягу
- Можливість міняти масштаб моделей
- Можливість розглянути манекен з усіх сторін за допомогою смуги прокрутки

2.3.2 Вибір фронтенд технологій для сервісу

Оскільки більшість часу пішло на розробку і тестування бібліотеки, було вирішено використовувати технології та фреймворки з якими ми були найбільш знайомі, та які б дозволили максимально швидко створити запропонований сервіс для демонстрації роботи.

Серед запропонованих фреймворків були **Angular**, **Vue.js** та **React**. Усі три фреймворки є найбільш популярними та використовуваними для сучасної швидкої розробки **SPA**.

SPA (Single Page Application) – однострінковий веб-застосунок, в результаті якого веб-застосунок складається з єдиного HTML файлу (для економії часу завантаження сторінки у веб-браузері) та файлів CSS та Javascript.

Усі три фреймворка мають однакові можливості, проте найбільше було досвіду роботи саме з фреймворком **Angular**.

2.3.2.1 Особливості Angular

Angular – фреймворк, написаний мовою typescript (мова програмування, яка має переваги над Javascript, і компілюється в Javascript для роботи в браузері), який дозволяє швидко створювати односторінкові веб-застосування (SPA).

Далі наведено переваги Angular над іншими схожими фреймворками:

- **CLI (Command Line Interface)** – інтерфейс командного рядка, за допомогою якого можна створювати компоненти на основі шаблонів, запускати тестування, компіляцію в проект для розгортки проекту на сторонній платформі та інші функції, які значно пришвидшують розробку.
- **Архітектура фреймворку – Angular** має дуже потужну архітектуру, за допомогою якої застосування має чітко розділені модулі. Ось список архітектурних модулів, які використовуються в **Angular**:
 - **Компонент** – базова одиниця в Angular, яка містить HTML, CSS та Javascript код. Компоненти використовуються як будівельні блоки, оскільки за допомогою них будується увесь проект.
 - **Сервіс** – класи в Angular, які виконують якусь специфічну функцію, як-от: отримання даних з бекенд-серверу, передача інформації між різними компонентами та модулями або ж відділення бізнес-логіки застосування.
 - **Модуль** - об'єкти в Angular, які містять в собі сервіси та компоненти. За допомогою модулів код програми можна зручно розділити на різні частини, які не заважають одна одній. Також модулі допомагають зробити застосування швидше, оскільки воно завантажується

частинами, по мірі того як користувач заходить в різні частини застосування.

За допомогою такої архітектури, можна швидко створювати швидкі та надійні застосування, тому для реалізації сервісу був вибраний Angular.

2.3.3 Вибір бекенд технологій для сервісу

Аби зробити демонстрацію бібліотеки зручнішим, було вирішено створити бекенд, для того, аби 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

2.3.3.1 Особливості node.js

Node.js – платформа для написання бекенд частин веб-застосувань на мові програмування javascript. Ось список особливостей, які виділяють node.js серед інших платформ:

- Асинхронна однопоточна модель виконання запитів
- Неблокуючий ввід-вивід
- Система пакетів **npm** (node package manager)
- Рушій Javascript Google V8, який виконує Javascript код

Недоліком node.js є те, що на ньому доволі важко писати об'ємні алгоритми (саме тому для бекенд частини бібліотеки для обробки зображень використовується python), проте для сервісу можливостей node.js буде достатньо. З іншої сторони, на node.js можна швидко створити надійний бекенд, тому була вибрана саме ця технологія.

2.3.4 Вибір бази даних

Аби зберігати завантажені та оброблені зображення одягу, було вирішено використати базу даних MySQL. Була вибрана саме ця база даних, оскільки був досвід роботи саме з цією базою даних.

2.3.4.1 Особливості MySQL

Ось деякі особливості MySQL, які допомогли в реалізації проекту:

- Висока швидкість обробки запитів
- Ціна, СУБД є безкоштовною
- Документація – це одна із найпопулярніших СУБД в світі, тому на просторах інтернету можна знайти велич документації та інформації про MySQL.

2.4 Висновки

В процесі аналізу різних технологій, було вибрано технології, які найкраще підходили під конкретні задачі. В результаті отримано такий список технологій та фреймворків, окремо для бібліотеки і окремо для сервісу примірки одягу на віртуальному манекені:

2.4.1 Бібліотека

В результаті аналізу фреймворків для 3д візуалізації було вирішено використовувати three.js для фронтенду. Для бекенду було вирішено використати python, разом із бібліотеками flask (для створення веб-серверу) та OpenCV (для створення алгоритму обробки зображень). Для передачі інформації між фронтендом та бекендом використовується API (написане на flask), а для передачі зображень всередині JSON-об'єкту використовується формат кодування двійкових даних Base64.

2.4.2 Сервіс для демонстрації роботи бібліотеки

Оскільки сервіс показує можливості бібліотеки, було вирішено використати технології, фреймворки та платформи, які максимально пришвидшили б розробку сервісу. Такими технологіями стали:

- Angular – фреймворк для швидкого створення фронтенд частини сервісу, де насамперед використовуватиметься бібліотека відображення 3д манекену та одягу
- Node.js – платформа для бекенд частини сервісу, яка обробляє запити які надходять з фронтенду

- MySQL – для зберігання даних (оброблених зображень, а також збережених комбінацій одягу).

У наступному розділі буде описано, яка саме використовувались вибрані нами технології, а також взаємодію даних технологій в кінцевому продукті.

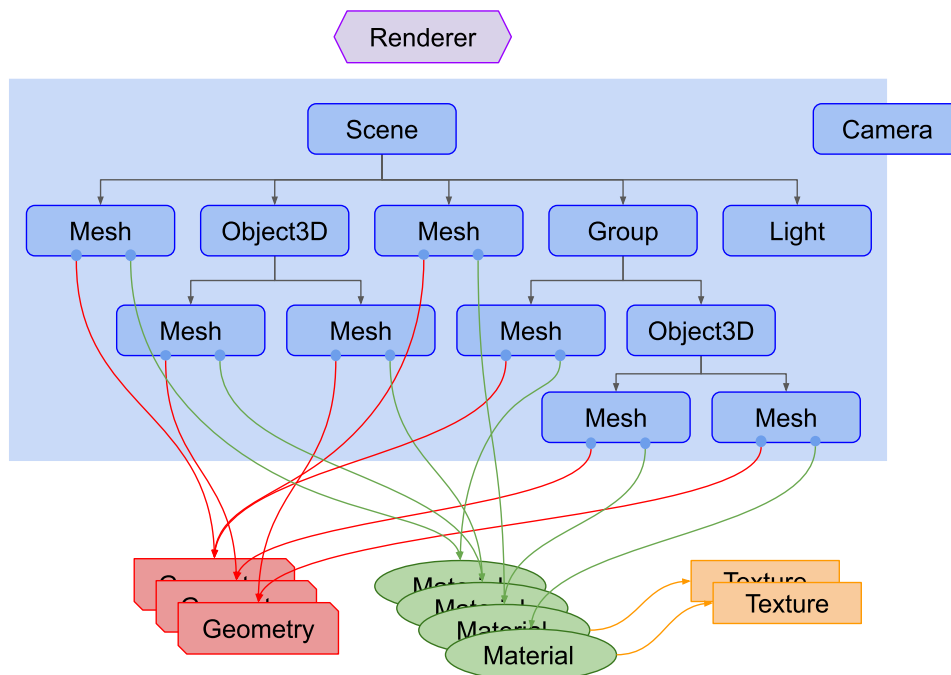
РОЗДІЛ 3: РОЗРОБКА ПРОЕКТУ, ОБ'ЄДНАННЯ РІЗНИХ ЧАСТИН ПРОЕКТУ В КІНЦЕВИЙ ПРОДУКТ ТА ТЕСТУВАННЯ ПРОЕКТУ

3.1 Використання three.js

Перед початком розробки, потрібно повторити, які ж вимоги до розробленої нами бібліотеки:

- Відображення 3-д модель манекену
- Відображення 3-д модель одягу футболки ти штанів в різних розмірах
- Можливість міняти розміри футболки ти штанів
- Можливість завантажувати власні зображення для обробки та накладання на модель одягу окремо для кожної частини одягу
- Можливість міняти масштаб моделей
- Можливість розглянути манекен з усіх сторін за допомогою смуги прокрутки

Бібліотека яка виконує більшість роботи на проекті є three.js. Для початку розробки, потрібно глибше розібратися в тому, як працює three.js. Ось діаграма, яка візуально показує архітектуру сцени в three.js:



3.2 Архітектура сцени three.js

Бібліотека three.js містить наступні об'єкти:

- Рендерер – глобальний об'єкт, який безпосередньо відображує сцену в певний DOM елемент на веб сторінці
 - Сцена – глобальний об'єкт, в якому дочірні елементи є 3-д об'єктами, які Рендерер буде відображувати на певному елементі. Для кожної сцени прив'язується камера, яка виступає в ролі спостерігача.
 - Камера – об'єкт, який замається усією математикою для того, аби отримати матрицю, яка позначає положення та поворот камери в 3д сцені. Усі операції виконуються автоматично, тому робота з камерою є інтуїтивною та не вимагає глибоких знань в математиці.
 - Текстура – зображення, яке завантажується для відображення на поверхні певного об'єкту
 - Матеріал - іншими словами, зовнішній вигляд поверхні для певного 3д об'єкту. Матеріал складається з багатьох параметрів, таких як:
 - Текстура
 - Блиск
 - Колір
 - Прозорість
- Ці параметри дозволяють міняти зовнішній вигляд кожного об'єкту в сцені, в тому числі і текстуру (зображення для одягу).
- Освітлення – об'єкти, які дозволяють створювати, налаштовувати та видаляти джерела світла. Саме завдяки освітленню, 3-д моделі виглядають по-справжньому та мають тінь.

Ознайомившись із документацією бібліотеки, було створено власну бібліотеку на базі three.js, яка виконує усі перераховані вище можливості.

3.3 Об'єднання фронтенду і бекенду в єдину бібліотек

Процес роботи бібліотеки описаний в наступних рядках:

1. Фронтенд відправляє до бекенду зображення одягу на однорідному фоні. Відправляється 2 зображення: одне спереди та одне ззаді.
2. Отримавши фотографії через http запит, бекенд виділяє та замінює фон на прозорий, залишаючи лише сам одяг:



Зображення перед обробкою на сервері



Зображення після обробки на сервері

В свою чергу фронтенд, отримавши оброблені зображення, використовує three.js і відображує модель манекену, та модель одягу з отриманими текстурами. Як уже було зазначено, зображення кодуюються в текстовому форматі Base64 для простішої і більш зручної передачі.

3.4 Тестування

Для тестування створеної бібліотеки, було вирішено створити простий сервіс для примірки одягу на віртуальному 3-д манекені

3.5 Сервіс для примірки одягу на віртуальному 3д манекені

Сервіс складається лише з трьох сторінок:

- Сторінка вибору комплекту одягу
- Сторінка манекена, одягненого в певний комплект
- Сторінка адміністратора, де можна добавляти нові комплекти одягу, для перегляду згодом

3.5.1 Сторінка вибору комплекту одягу

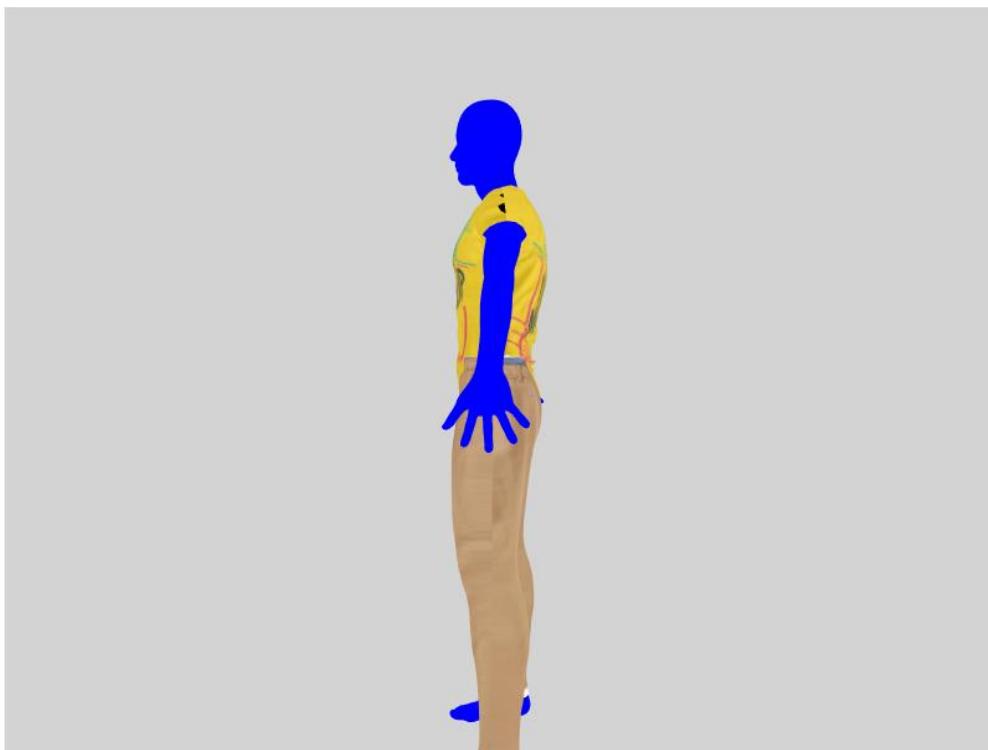
На цій сторінці відображається список, де користувач може вибрати окремий комплект. При натисканні на комплект, користувача переадресовує на сторінку перегляду одягу.

3.5.2 Сторінка перегляду комплекту одягу

На цій сторінці відображається основна робота розробленої нами бібліотека. Посередині сторінки видно манекен, одягнений в комплект одягу. Також на цій сторінці можна вибрати розмір одягу:



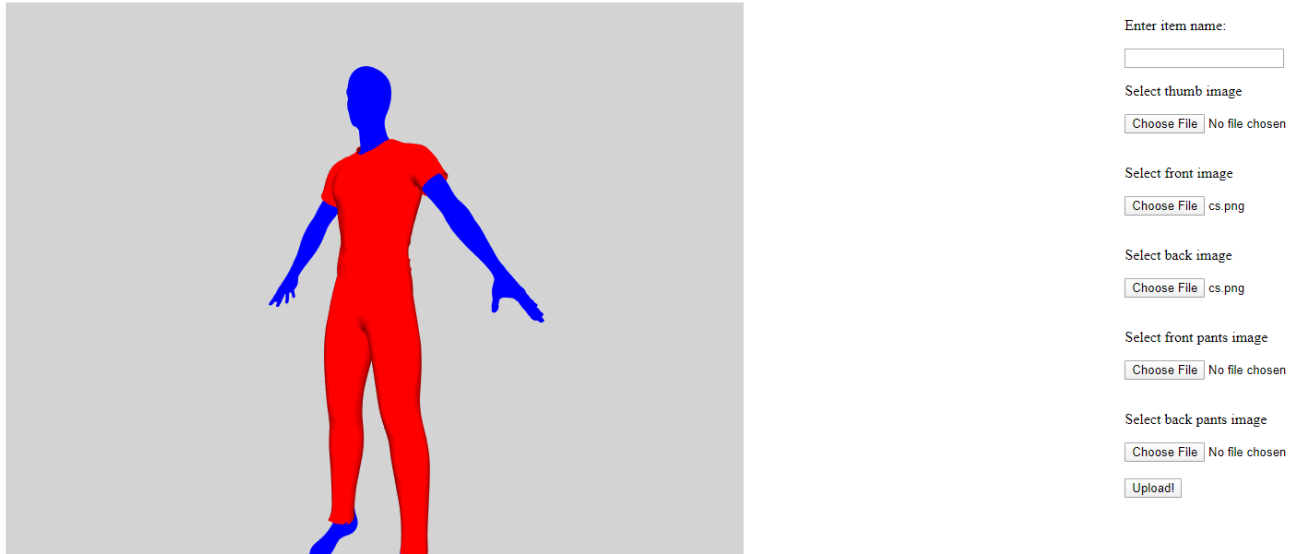
Манекен, одягнений в комплект одягу



Манекен, одягнений в штани меншого розміру, повернутий вбік

3.5.3 Сторінка адміністратора

На цій сторінці, адміністратор може добавляти новий комплект одягу в базу даних, який згодом можна буде переглядати на манекені. Адміністратору пропонується надати назву комплекту, зображення комплекту, а також 4 зображення для футболки та штанів, спереду і ззаду.



Користувацький інтерфейс адміністратора, де можна добавити повну інформацію про комплект одягу

3.6 Результат тестування

В результаті розробки сервісу для примірки одягу на манекені, було продемонстровано роботу розробленої нами бібліотеки. Також успішно протестовано усі функції бібліотеки.

ВИСНОВКИ

В ході виконання роботи, було досліджено методи для відображення 3д графіки в веб-браузерах. В результаті було створено javascript-бібліотеку, написану з використанням бібліотеки для полегшення 3-д відображення three.js, а також бекенду, який обробляє зображення для моделей. Для тестування створеної бібліотеки було створено сервіс для примірки свого одягу на віртуальних 3-д моделях. Створена нами бібліотека може бути корисною в тих областях, де потрібна візуалізація одягу, аби покращити взаємодію з користувачами та клієнтами.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

<https://threejs.org/docs/index.html#manual/en/introduction/Creating-a-scene>

https://developer.mozilla.org/ru/docs/Web/API/WindowBase64/Base64_encoding_and_decoding

<https://threejsfundamentals.org/threejs/lessons/threejs-fundamentals.html>

<https://metanit.com/web/angular2/1.1.php>

<https://angular.io/resources?category=development>

<https://doc.babylonjs.com/>

<https://flask.palletsprojects.com/en/1.1.x/>

<https://docs.opencv.org/4.2.0/>

<https://dev.mysql.com/doc/>

<https://nodejs.org/uk/docs/>

<https://github.com/awayjs/awayjs-full>

https://developer.mozilla.org/uk/docs/Web/API/WebGL_API

<https://webglfundamentals.org/webgl/lessons/ru/>

ДОДАТКИ