

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

**ПОБУДОВА БАГАТЬОХ ГРАФІКІВ НА ОДНОМУ МАЛЮНКУ,
СТВОРЕННЯ НАПИСІВ ТА АНІМАЦІЯ У SPRING ПРОЄКТІ**

Текстова частина до курсової роботи
за спеціальністю «Комп'ютерні науки»

Керівник курсової роботи
професор, доктор фіз.-мат. наук

_____ Малашонок Г.І.
(підпис)

«__» _____ 2024 р.

Виконала студентка

Стецюк І.Л.

«__» _____ 2024 р.

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

ЗАТВЕРДЖУЮ

_____ Малашонок Г.І.
(підпис)

«___» _____ 2024 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
на курсову роботу

студенту Стецюк Інні Леонідівні факультету інформатики 3 курсу

ТЕМА **Побудова багатьох графіків на одному малюнку, створення написів та анімація у Spring проєкті**

Зміст ТЧ до курсової роботи:

Індивідуальне завдання

Вступ

Розділ 1: Бібліотека Two.js та її переваги

Розділ 2: Реалізація та приклади

Висновки

Список літератури

Дата видачі «___» _____ 2024 р. Керівник _____
(підпис)

Завдання отримав _____
(підпис)

Тема: Побудова багатьох графіків на одному малюнку, створення написів та анімація у Spring проєкті

Календарний план виконання роботи:

№ п/п	Назва етапу курсової роботи	Термін виконання	Примітка
1.	Отримання теми на курсову роботу.	30.10.2023	
2.	Огляд літератури за темою роботи.	листопад-грудень	
3.	Визначення структури.	січень	
4.	Виконання практичної частини.	лютий-березень	
5.	Написання першого розділу.	квітень	
6.	Написання другого розділу.	квітень	
7.	Створення презентації для захисту.	травень	
8.	Внесення змін до курсової роботи відповідно до зауважень наукового керівника.	травень	
9.	Захист курсової роботи.	23-24.05.2024	

Студент Стецюк І.Л.

Керівник Малашонок Г.І.

“ ”

Зміст

Анотація	5
Вступ.....	6
Розділ 1	7
1.1 Бібліотека Two.js.....	7
1.2 Переваги використання Two.js	8
Розділ 2	10
2.1 Backend.....	10
2.2 Frontend	11
2.3 Приклади	12
Висновки	14
Список використаних джерел	15

Анотація

Курсова робота присвячена розробці фронтенд-частини для веб-платформи “Math Partner”, яка призначена для вирішення математичних задач у різних сферах. Робота зосереджується на імплементації таких функцій як побудова декількох різних графіків на одному малюнку, додавання текстових написів та реалізація анімації з використанням бібліотеки Two.js.

Ключові слова: Math Partner, графіки, фронтенд, Two.js

Вступ

У сучасному світі зростає потреба в практичних інструментах для візуалізації даних для різних сфер таких як навчання у школах, університетах, для наукових досліджень, інженерії та іншого. Значну увагу приділяють саме ефективним інструментам для графічного представлення функцій, що є невід’ємною частиною для аналізу та інтерпретації даних. Графічне представлення дозволяє наочно демонструвати залежності та тренди. Це є ключовим елементом для розуміння складних концепцій, що має важливе значення в освітній сфері та наукових дослідженнях. Крім цього, аналіз даних за допомогою графіків може бути корисним в бізнес-аналітиці, де такі візуалізації допомагають ухвалювати обґрунтовані рішення на основі дані, для кращого розуміння тенденцій ринку та інших процесів.

Мета дослідження полягає у покращенні фронтенд-частини для графічного представлення математичних функцій на платформі “Math Partner” з використанням бібліотеки Two.js.

Об’єктом дослідження є веб-платформа “Math Partner”.

Науковою новизною отриманих результатів є розробка удосконаленого інтерфейсу використовуючи бібліотеку Two.js, що має більшу гнучкість та продуктивність для графічного представлення функцій.

Робота складається з двох розділів:

Перший розділ присвячено розгляду основних концепцій Two.js, а також визначенню переваг використання бібліотеки для сформованого завдання.

В другому розділі описано реалізацію та приклади.

У висновку надано підсумок результатів проведеного дослідження.

Розділ 1

1.1 Бібліотека Two.js

Two.js – це потужна та гнучка двовимірنا графічна бібліотека, яка написана на мові JavaScript і призначена для веб-розробки. Вона надає простий та інтуїтивно зрозумілий API для малювання двовимірної графіки, яке орієнтоване на сучасні веб-браузери. Ця бібліотека дозволяє розробникам швидко створювати різноманітні візуальні елементи та анімації у веб-застосунках.

Подібно до DOM у HTML, у Two.js вся графіка організовується у вигляді дерева. Коренем цього дерева є об'єкт “Two”, який грає роль контейнера для всіх інших графічних об'єктів. Бібліотека надає широкий спектр об'єктів для створення та маніпулювання. Такі об'єкти як прями лінії, криві, прямокутники, кола, еліпси, тексти, багатокутники та інші є основними графічними об'єктами, які можуть мати стилі, такі як контур, товщина лінії, колір заливки, рівень прозорості, масштабування, кут обертання, розташування та інші. Для спрощення управління графічними об'єктами та їхньою анімацією об'єкти, які повинні бути оброблені разом, утворюють групу. Об'єкти в групі успадковують трансформації групи, такі як переміщення, обертання та масштабування і також згруповані об'єкти можуть спільно використовувати певні властивості.

Бібліотеку Two.js найчастіше використовують в освітніх застосунках, також в інтерактивних веб-інтерфейсах, рекламних кампаніях чи для створення анімаційних ефектів у веб-дизайні.

1.2 Переваги використання Two.js

Використання бібліотеки Two.js для побудови багатьох графіків на одному малюнку, додавання написів та анімації у веб-застосунку “Math Partner” є ефективним та надійним рішенням з декількох причин.

По-перше, Two.js може значно спростити процес розробки. Високий рівень абстракції, тобто представлення форм та групування об’єктів, дозволяють зменшити кількість коду, який потрібен для написання складних візуалізацій. Полегшує процес розробки також інтуїтивно зрозумілий набір правил та специфікацій, які дозволяють легко доступатися до властивостей, налаштувань вигляду та поведінки, і керувати анімацією.

По-друге, Two.js надає розробникам ряд переваг, які пришвидшують процес розробки. API можна відносно легко інтегрувати з іншими бібліотеками і фреймворками JavaScript. Це дозволяє використовувати бібліотеку у складних проєктах без значних змін в уже існуючій архітектурі. Крім цього, використовуючи Two.js розробники схильні витратити менше зусиль на підтримку коду та його оптимізацію, оскільки ця бібліотека забезпечує високу продуктивність навіть при роботі з детальними комплексними візуалізаціями та з великими обсягами даних.

По-третє, Two.js забезпечує значну гнучкість під час вибору рендерингових технологій. SVG дозволяє підтримувати високу якість зображення при будь-якому розширенні екрану та забезпечувати високу сумісність без необхідності використовувати додаткові плагіни. WebGL надає високу продуктивність для складних анімацій, які потребують великої кількості графічних об’єктів чи плавної анімації, і також добре підходить для інтерактивної взаємодії з користувачем. Підтримка різних рендерингових технологій дозволяє легко адаптуватися до різних

платформ та пристроїв, що робить цю бібліотеку чудовим вибором для кросплатформених застосунків.

Як і будь-яка інша технологія, бібліотека може має певні недоліки, наприклад зменшення продуктивності при обробці надзвичайно складних великих сцен, чи те, що бібліотека не володіє розширеними функціями для тривимірної графіки. Однак ці обмеження є досить незначними для більшості проєктів. А для роботи з 3d-графікою краще використовувати схожу спеціалізовану бібліотеку Three.js, яка створена саме для таких задач.

Загалом, такі основні переваги як інтуїтивно зрозуміле API, підтримка кросплатформеного рендерингу, і швидкий та ефективний процес розробки графічних візуалізацій роблять бібліотеку Two.js чудовим інструментом для покращення фронтенду на платформі “Math Partner”.

Розділ 2

2.1 Backend

У сучасних веб-застосунках клієнт-серверна архітектура є фундаментом, який дозволяє ефективно пов'язувати користувацький інтерфейс та процеси на сервері. Ця архітектура сприяє організації логіки додатку на сервері, дозволяючи клієнтам відправляти запити та отримувати потрібні дані для візуалізації.

У файлі F.java додано нову функцію showPlotsTwoJS з порядковим номером, а також додано нову функцію showPlotsTwoJs в масив усіх імен функцій FUNC_NAMES. На сайті після натиснення на кнопку showPlotsTwoJS, з'являється можливість вводу різноманітних математичних виразів та потрібних функцій мовою MathPar. Після їх введення і підтвердження команди побудувати графіки, цей запит передається безпосередньо до сервера в Page.java. Там обробляється цей запит, виймається з нього головна функція та повертається значення головної функції. В залежності від того яка це функція, запускається JavaScript файл.

Цей JavaScript файл робить запит на бекенд. В контролері на бекенді цей запит обробляється. @RequestMapping дозволяє визначити, які запити від клієнтів мають бути спрямовані до методів обробки на сервері. Використовуючи таку анотацію розробники можуть досить точно вказати, за якими URL-адресами і методами HTTP певні методи повинні бути викликані. @RequestMapping у цьому випадку використовується для визначення методу, який обробляє POST запити на api/showplots2d. Це означає, що коли користувач відправляє POST запит на цю адресу, активується описаний метод, обробляє дані, які надійшли у запиті, і повертає відповідь. Якщо в запиті немає потрібної функції, або обробка з певних причин не пройшла успішно, метод повертає відповідні статуси

HTTP, які інформують клієнта про результат обробки. На клієнтську сторону повертається масив точок, які потім будуть використані для побудови графіків на фронтенді.

2.2 Frontend

На стороні фронтенду, використання JavaScript бібліотеки Two.js дозволяє створити графіки для візуалізації різних функцій. Коли дані отримані з бекенду, скрипт аналізує відповідь та витягує колекцію точок для подальшої обробки.

Спочатку створюється екземпляр Two з переданими параметрами розміру, визначається де і як графіки будуть відображатися на сторінці. Для спрощення розуміння графіків, функції `drawGrid` та `drawAxes` відповідають за створення сітки та осей координат.

З отриманого масиву точок, будуються функції, за допомогою таких додаткових функцій як `drawLine`, `drawPlot`, `drawArrow` та інших. Кожен графік рандомно отримує колір. Додається напис функції біля кожного графіку за допомогою методу `two.makeText`.

Інтерактивність на сторінці забезпечується за допомогою функцій, які уможливають перетягування (`dragging`). Обробники подій реагують на кліки та рухи миші, дозволяючи змінювати положення осей і графіків для зручності та гнучкості у використанні.

Крім цього, функціонал масштабування дозволяє користувачам збільшувати або зменшувати малюнок. Функцій масштабування та перетягування не було в початковій версії проєкту. Їхня імплементація робить інтерфейс більш адаптивним та покращує користувацький досвід.

Для анімації використовується бібліотечний метод `two.bind('update', function(...) {...}).play()` для того, щоб оновлювати властивості графіка в кожному кадрі анімації. Можна задавати швидкість анімації та інші

властивості. Метод `two.play()` запускає анімацію. `Two.js` автоматично оновлює полотно, викликаючи подію `update` на кожному кроці анімації.

Крім цього використовуються також інші додаткові функції, такі як очищення полотна, збереження та інші.

Загалом, така інтегрована клієнт-серверна система забезпечує високу ефективність та швидкість роботи з даними, і також створює продуктивний користувацький досвід.

2.3 Приклади

Перший приклад побудови трьох графіків на одному малюнку: функція, котра задана таблицею значень, простої функції й параметричної.

```
\set2D(-10, 10, -10, 10);
f1 = \plot(\sin(x));
f2 = \tablePlot([[ -8, 2, 6], [-7, 3, 6]]);
g = \sin(x) * 3; k = \cos(x) * 3 ;
f3 = \paramPlot([g, k], [0, 4\pi]);
\showPlots([f1, f2, f3]);
```

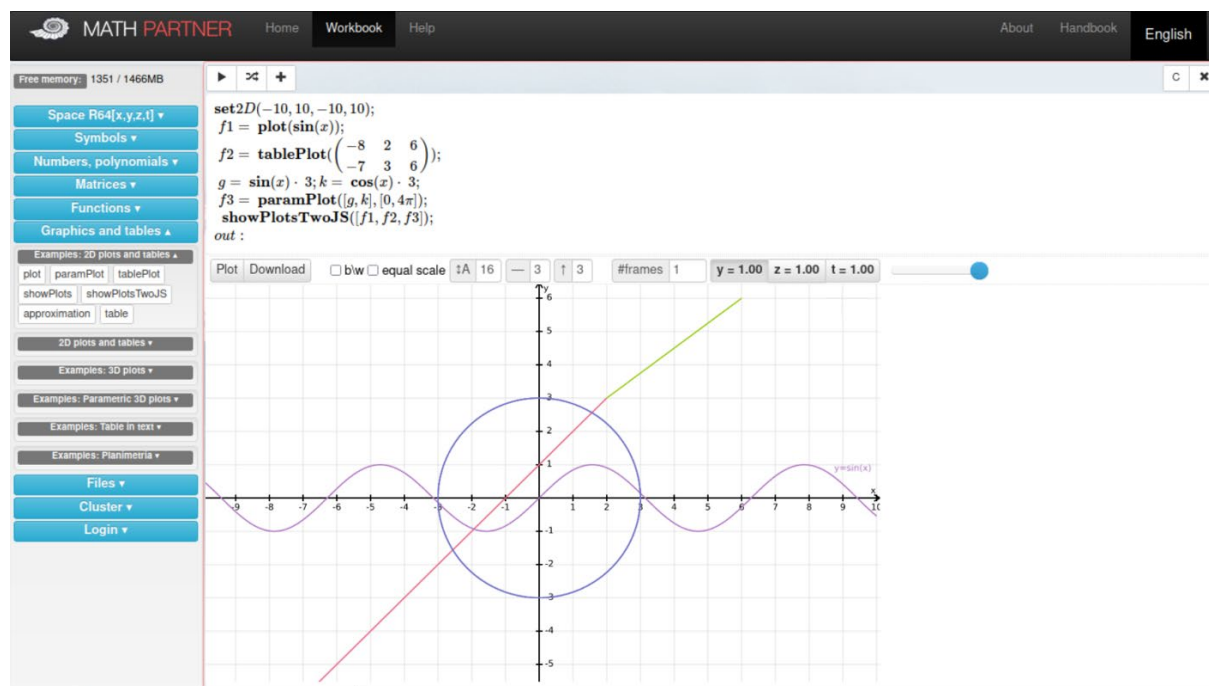


Рис.1 – Перший приклад

Другий приклад побудови двох графіків на одному малюнку: простої функції та параметричної.

```
\set2D(-30, 30, -17, 17);
f1 = \plot(\abs(x+3));
g = \sin(x) * x;
k = \cos(x) * x ;
f2 = \paramPlot([g, k], [0, 5.8\pi]);
\showPlotsTwoJS([f1, f2])
```

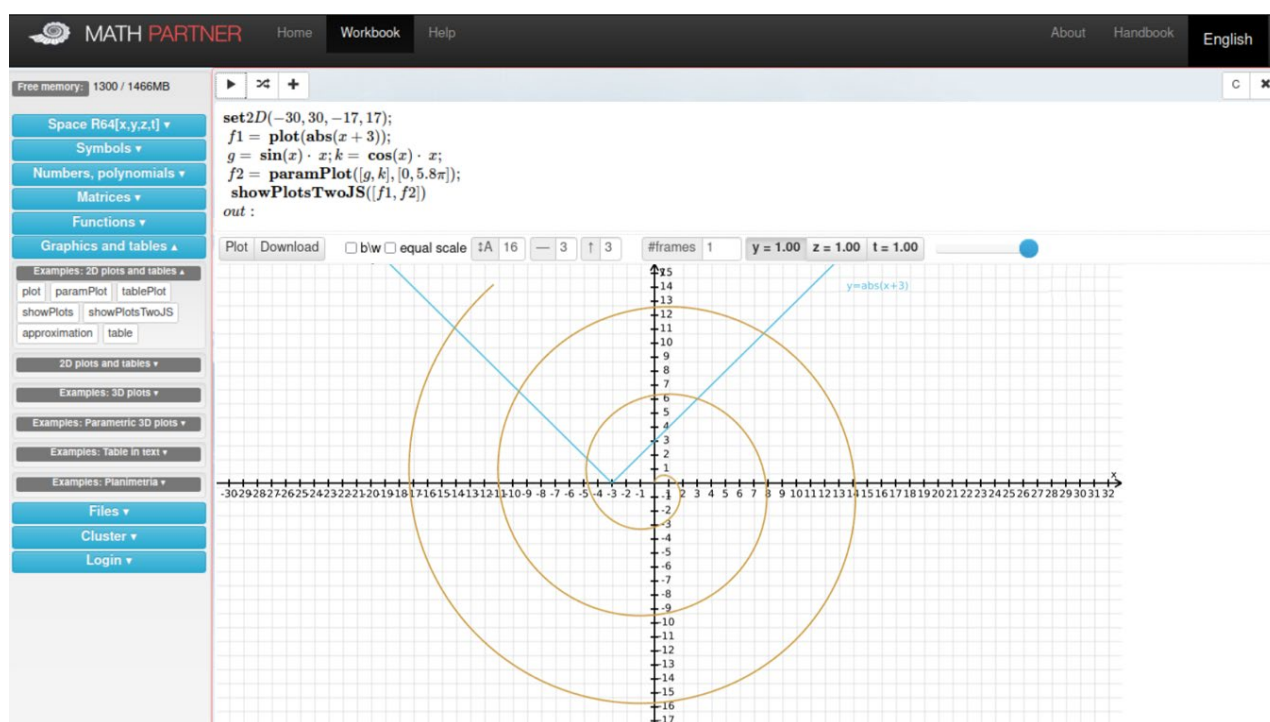


Рис.2 – Другий приклад

Висновки

Дослідження та робота над проєктом “Math Partner” з використанням бібліотеки для двовимірної графіки Two.js дозволила певним чином покращити користувацький інтерфейс. Через удосконалення фронтенд-частини, користувач може користуватися новими додатковими можливостями такими як перетягування малюнку та збільшення чи зменшення масштабу. Таким чином, користувач може використовувати платформу для своїх цілей продуктивно, адже спеціалізована бібліотека Two.js забезпечує зручність та адаптивність інтерфейсу, що є дуже важливим для широкого спектру юзерів.

Крім цього, архітектура даного проєкту є гнучкою та легко масштабованою. Це забезпечує можливість легшого оновлення, додавання нових функцій та покращення в майбутньому як веб-застосунку в цілому, так і побудови різноманітних графіків, створення написів та анімації.

Список використаних джерел

1. Документація бібліотеки Two.js

<https://two.js.org/docs/two/>

2. @RequestMapping

<https://docs.spring.io/spring-framework/reference/web/webmvc/mvc-controller/ann-requestmapping.html>

3. Math Partner Handbook

<https://mathpar.ukma.edu.ua/en/welcome.html>

4. Math Partner Help

<https://mathpar.ukma.edu.ua/en/help/index.html>