

З. Антонов Є.В. Створення ігрових механік засобами REN'PY. Актуальні питання сучасної інформатики: Матер. доп. IV Всеукр. наук.-практ. конф. "Сучасні інформаційні технології в освіті та науці" (07-08 листопада 2019 р.). Житомир: Вид-во ЖДУ, 2019. Вип. 7. С. 109-112.

Киселевич Володимир,
здобувач першого(бакалаврського) рівня вищої освіти
фізико-математичного факультету
Науковий керівник: **Вербівський Дмитрій,**
кандидат педагогічних наук,
доцент кафедри комп'ютерних наук та інформаційних технологій,
Житомирський державний університет імені Івана Франка,
м. Житомир, Україна

ОГЛЯД ТЕХНОЛОГІЙ ДЛЯ СТВОРЕННЯ ОДНОСТОРІНКОВИХ ВЕБ-ДОДАТКІВ

До появи фреймворків та бібліотек, у всіх веб-сайтів залишалася головна проблема: витрата часу на перезавантаження сторінки при навігації по сторінкам сайту. Концепція SPA (single page application) вирішує цю проблему. Користувач завантажує лише один раз увесь веб-сайт й при навігації відбувається рендер вже завантаженого контенту.

З кожним роком популярність бібліотек та фреймворків для створення односторінкових веб-додатків (SPA) стрімко зростає. Використання їх в розробці неабияк пришвидшує сам процес розробки та спрощує підтримку додатку.

Мета статті – дослідити особливості та переваги поширених технологій розробки односторінкових веб-додатків.

За трендами та статистикою NPM Trends (Рис. 2) найпопулярніша на сьогоднішній день бібліотека – React JS. Менш розповсюджені фреймворки Angular та Vue JS. Існують також й інші фреймворки, такі як: Ember, Polymer, Riot, Knockout, але через ряд недоліків ці технології втратили свою актуальність. Тому далі мова піде саме про популярні технології: React, Vue JS та Angular, їх особливості і переваги.

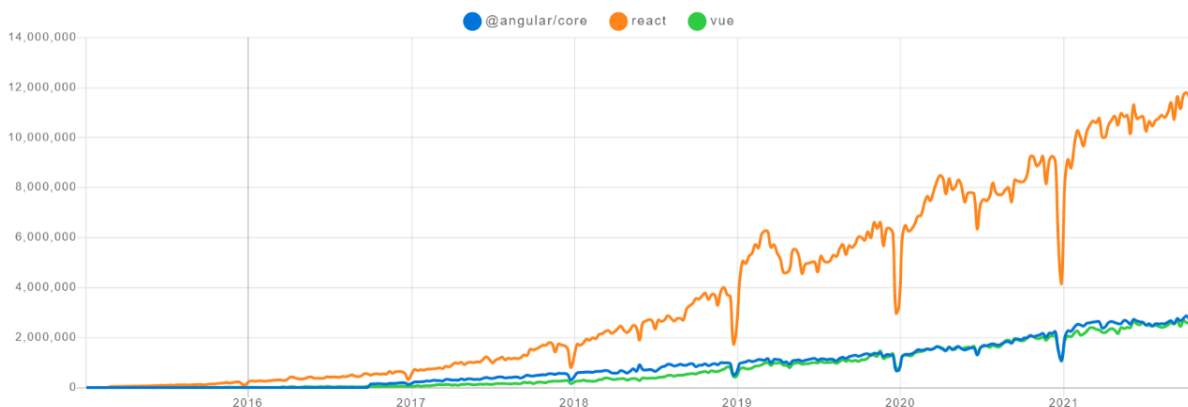


Рис. 2. Статистика завантажень фреймворків та бібліотек в NPM

React JS

React – бібліотека JavaScript, розроблена компанією Facebook. Перший реліз бібліотеки відбувся у березні 2013 року. За час існування, технологія зібрала

навколо себе багатомільйонну спільноту, завдяки чому вже існує багато користувацьких розширень, інструментів та готових рішень. React відносно простий в освоєнні та має ряд особливостей.

Розглянемо головні особливості бібліотеки [1]:

– Використання JSX. JSX являє собою схожий на HTML синтаксис та дозволяє його написання в JavaScript коді. Після компіляції, JSX трансформується в звичайні React елементи. Використання такого синтаксису значно спрощує сприймання коду та пришвидшує процес його написання. Приклад створення React елементів звичайним способом та за допомогою JSX синтаксису зображено на Рис. 3.

```
// Синтаксис JSX
let element = <h1 className={'title'}>Hello, world!</h1>;

// Без використання JSX
let element = React.createElement('h1', {className: 'title'}, 'Hello, world!');
```

Рис. 3. Синтаксис JSX

– Віртуальний DOM (Document Object Model). Досить часто в роботі необхідно оновлювати, змінювати, додавати та видаляти елементи на сторінці. Маніпуляція великою кількістю HTML елементів за допомогою звичайних засобів JavaScript знизить швидкість роботи додатку. На допомогу приходить концепція віртуального DOM, яка створює копію реального DOM. React звертається до віртуального DOM кожен раз, коли потрібно дізнатись стан елементів. Якщо необхідно внести зміни, то React спочатку вносить зміни у віртуальний DOM, далі порівнює віртуальний DOM з реальним та виконує мінімальну кількість маніпуляцій для оновлення до нового стану (Рис. 4). Такий підхід значно оптимізує роботу веб-додатку.

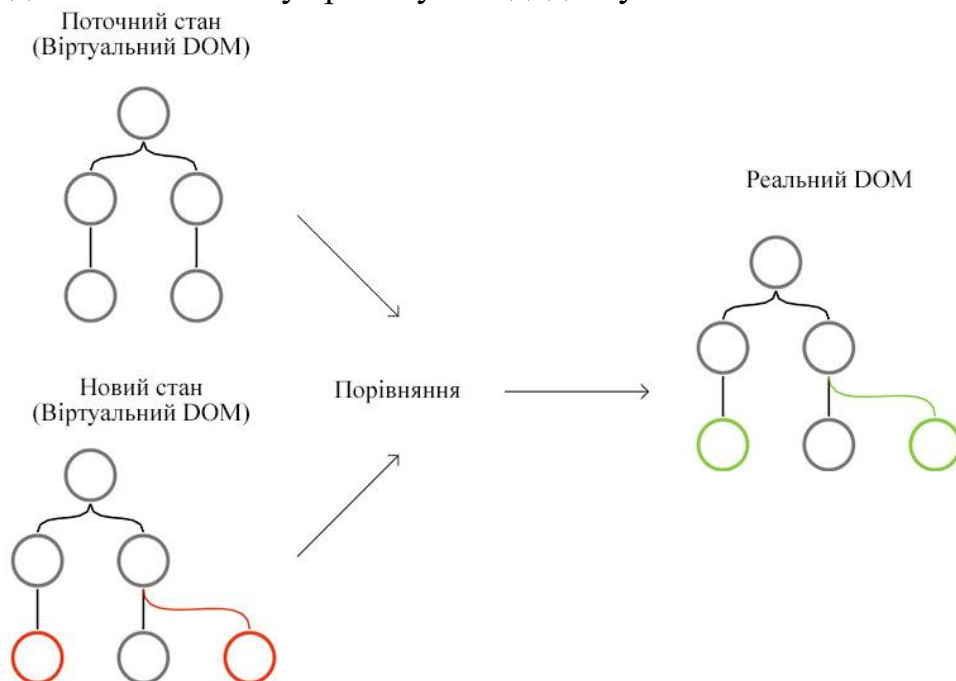


Рис. 4. Схема роботи віртуального DOM

Секція 4. Технології розробки інформаційних систем

– Компонентний підхід розробки. Такий підхід в розробці веб-додатку, дозволяє розбити елементи інтерфейсу на інкапсульовані компоненти з можливістю повторного використання в будь-якій частині коду.

Бібліотека React сконцентрована на використанні JavaScript при написанні коду, розмітки та навіть стилів. В більшій мірі спрямована на ефективний рендер елементів інтерфейсу, але сама по собі не має розширеного функціоналу, такого як: менеджер стану, валідація форм, маршрутизатор та інших інструментів. Проте, все це вже створено React спільнотою, і ми маємо можливість додатково підключити інструменти до свого проекту, обмежившись лише необхідним.

Angular

Angular – JavaScript фреймворк з відкритим кодом, створений у 2010 році компанією Google. Розробники класифікують Angular як повноцінну платформу розробки, яка включає в себе весь необхідний функціонал для розробки та не потребує підключення сторонніх інструментів

Розглянемо особливості фреймворку [2]:

– TypeScript. Написання програмного коду за замовчуванням статично типізованою мовою програмування TypeScript.

– Звичайний DOM. В порівнянні з віртуальним DOM, звичайний DOM працює повільно, адже буде оновлювати все дерево елементів поки не дійде до необхідного елемента.

– Компонентний підхід у зв'язці з шаблонами HTML і TypeScript.

Angular має розмір 566KB, що являється головною вразливістю фреймворку. Велика кількість інтегрованих інструментів для розробки, тестування та налагодження у невеликому веб-додатку відіграють роль баласту. Написання коду за замовчуванням статично типізованою мовою програмування TypeScript, може стати ще одним бар'єром при прийнятті рішення на користь Angular. Проте, всі перераховані недоліки у масштабних проектах стають безумовними перевагами.

Vue JS

Vue JS – прогресивний JavaScript фреймворк з відкритим кодом, спрямований на розробку користувацьких інтерфейсів. Творцем фреймворку є Еван Ю, який до того ж, працював над фреймворком Angular. Розробка Vue JS почалася в 2013 році, і вже в 2014 відбувся перший реліз. На сьогоднішній день, актуальною версією є Vue 3, яка повністю написана на статично типізованій мові TypeScript.

Фреймворк Vue JS об'єднує в собі декілька ідей з бібліотеки React та фреймворку Angular, додаючи від себе нові можливості.

Розглянемо головні особливості [3]:

– віртуальний DOM. Робота віртуального DOM аналогічна бібліотеці React (Рис. 4), але віртуальний DOM VueJS має невелику перевагу в оптимізації. Фреймворк автоматично відслідковує залежності компонента і точно знає, які компоненти дійсно потребують рендер, на відміну від React, який запускає рендер усього піддерева, починаючи з себе.

– використання HTML шаблонів. Фреймворк Vue використовує у своїй екосистемі шаблони, які дозволяють писати традиційну HTML розмітку,

Секція 4. Технології розробки інформаційних систем

ізолювані CSS стилі в тезі Style та JavaScript код в тезі Script. При необхідності, Vue дає можливість використання JSX синтаксису.

Незважаючи на відносно невелику спільноту користувачів, фреймворк має досить детальну та інформативну документацію, яка компенсує цю різницю. Vue JS в порівнянні з React та Angular важить всього 18-21KB. Швидкість роботи фреймворку, простота інтеграції в проект та можливість відразу почати працювати, не вивчаючи складних конструкцій для простих речей – головна перевага технології.

Підсумки дослідження.

Підводячи підсумки дослідження, можна зробити висновок, що не існує найкращої технології. Кожен засіб являє собою потужний інструмент. Обираючи технологію, варто оцінити яка перед вами поставлена задача. Angular має багато інтегрованих інструментів та підходить для великомасштабних проектів з великою командою розробників, яка вже використовує TypeScript. React та Vue JS являються гнучкими та швидкими технологіями зі своїми особливостями та перевагами, які варто оцінити під конкретну задачу.

Список використаних джерел та літератури

1. Документація бібліотеки React. URL: <https://reactjs.org/> (дата звернення 28.10.2021).
2. Документація фреймворку Angular. URL: <https://angular.io/> (дата звернення 02.11.2021).
3. Документація фреймворку Vue JS. URL: <https://vuejs.org/> (дата звернення 30.10.2021).

Королук Валерія,
здобувач першого (бакалаврського) рівня вищої освіти
фізико-математичного факультету
Науковий керівник: Жуковський Сергій,
кандидат педагогічних наук,
доцент кафедри комп'ютерних наук та інформаційних технологій,
Житомирський державний університет імені Івана Франка,
м. Житомир, Україна

ОСНОВИ СТВОРЕННЯ ПЕРСОНАЖУ ДЛЯ КОМП'ЮТЕРНОЇ 2D ГРИ

Актуальність навчальної теми пов'язана з популярністю комп'ютерних ігор, яка у сучасному світі росте з кожним днем. Щороку на ринок випускаються десятки тисяч нових відеоігор, що розходяться мільйонами копій по всьому світу та приносять десятки мільярдів доларів їх розробникам. Варто взяти до уваги те, що комп'ютерні ігри останнім часом перестали позиціонуватися тільки, як програми для відпочинку та розваг. Наприклад, завдяки використанню ігрових технологій, можна створювати симуляції для навчання робітників різних сфер: лікарів, пілотів та навіть космонавтів [2].

Зародження комп'ютерних ігор бере свій початок ще у 50-ті роки XX століття. Тоді А. С. Дугласом була написана перша гра «ОХО», що являла собою

звичайні "хрестики нулики". Далі Вільям Хігінботем створив комп'ютерну гру «Теніс» в яку могло грати дві людини [3]. Тепер цим вже нікого не здивувати, оскільки написати подібні ігри може кожен звичайний користувач маючи при собі ПК та трохи терпіння.

Хоча індустрія 2D ігор з'явилася набагато раніше ніж сучасні, деталізовані 3D розробки, але вона досі залишається актуальною серед юзерів та прибутковою серед компаній. Де доводить популярність таких ігор, як: Hollow Knight, Limbo, Don't Starve, Cuphead, Rayman та багато інших. Отже, для створення хорошої комп'ютерної гри важливим є не тільки використання сучасної графіки, а й сюжет, привабливі та цікаві персонажі, атмосфера і інші деталі розробки.

Для створення такої відеогри зазвичай користуються готовими рішеннями – ігровими платформами [1]. На даний момент одне з найпопулярніших і ефективних рішень для створення 2D і 3D ігри – Unity. Але є і інші актуальні засоби розробки, а саме: GameMaker 2, Godot Engine, Unreal Engine 4, Construct 3 та ін [4].

Тому було обрано саме цю тему, щоб дізнатися, як створювати одну з важливіших деталей комп'ютерної 2D гри, а саме – 2D персонажа.

Метою статті є: Розкрити основи та особливості створення цікавого і запам'ятовуючого персонажа для 2D гри.

Для ґрунтового аналізу обраної теми спочатку розглянемо такі поняття:

Комп'ютерна гра – це програма, яка призначена для організації ігрового процесу, зв'язку з партнерами по грі або сама виступає в ролі партнера і її метою є забезпечити хороше дозвілля із застосуванням комп'ютера [5].

3D графіка – це вид саме комп'ютерної графіки, тобто комплекс методів та інструментів, що застосовуються для створення зображень 3D об'єктів на комп'ютері [6].

Персонаж – це герой з індивідуальною зовнішністю, характером і звичками, дійова особа або істота мистецького твору [7].

З попереднього визначення виходить, що **2D персонаж** – це той самий герой, але створений за допомогою спеціальних програм та інших засобів створення 2D графіки.

2D графіка – це основа всієї комп'ютерної графіки (у тому числі і 3D-графіки). Вона головним чином складається з двовимірних моделей (таких як геометричні 2D моделі, текст і цифрові зображення) і методів, визначених для них [8-9].

Під словом «модель» частіше розуміють деякий образ об'єкта (зокрема, умовний чи уявний), що нас цікавить, з цього випливає, що **2D модель** – це цифрове зображення цього образу, створюваного в спеціальній програмі [10].

Деталі створення 2D персонажу.

Процес створення будь-якого персонажу, особливо запам'ятовуючого буде займати час, оскільки потрібно треба спочатку розібратися з деталями. Початок потрібно брати з оригінальної ідеї та концепції. Далі описуються основні характеристики героя, такі як стать, характер, вік, зовнішність та інші