

І. В. ГАВРИШ

студент II курсу магістратури
за спеціальністю 121 Інженерія програмного забезпечення
Запорізький національний університет
ORCID: 0009-0007-5294-2146

А. Г. КРИВОХАТА

кандидат фізико-математичних наук,
доцент кафедри програмної інженерії
Запорізький національний університет
ORCID: 0000-0001-9664-0659

РОЗРОБКА АРХІТЕКТУРИ АДАПТИВНОГО ІНТЕРФЕЙСУ РЕДАКТОРА ВЕБ-ЗАСТОСУНКУ ДЛЯ СТВОРЕННЯ ВІРТУАЛЬНИХ ТУРІВ

Стрімкий розвиток та поширення технологій віртуальної реальності сприяє їх активному використанню у сферах туризму, нерухомості та освіти. Одним із способів представлення інформації у віртуальному середовищі є інтерактивні VR-тури. У роботі проаналізовано існуючі системи для створення та редагування віртуальних екскурсій, досліджено сучасні підходи до побудови мобільних інтерфейсів і виявлено низку суттєвих недоліків їх поточних реалізацій. Зокрема, встановлено, що більшість сервісів мають обмежений функціонал на мобільних пристроях, особливо щодо роботи з меню інструментів, керування сценою та стабільності відображення дрібних елементів на малих екранах.

Метою роботи є розробка та тестування архітектури адаптивного інтерфейсу редактора веб-застосунку віртуальних екскурсій, здатної повністю відтворювати функціональність настільної версії в умовах обмеженого екранного простору. Запропонована архітектура ґрунтується на стратегії Mobile-First, системі модульних компонентів і принципах розподілу робочих зон інтерфейсу. Вона включає структурні елементи, такі як верхня панель навігації, робочий простір 3D-сцени, bottom sheet та бічне меню інструментів. Такий підхід забезпечує раціональне використання наявної площі екрана, чіткий поділ між навігацією, інструментарієм та середовищем перегляду віртуальної сцени, а також передбачувану поведінку інтерфейсу під час зміни розміру вікна перегляду.

Для практичної перевірки архітектури було створено макет інтерфейсу з використанням фреймворку Vue.js. Запропонований підхід протестовано за допомогою набору автоматизованих UI-тестів з використанням Playwright, що охоплюють перевірку розташування, масштабування та доступності елементів під час роботи з різними розмірами вікна перегляду (viewport). Результати підтвердили відсутність конфліктів верстки, коректну роботу компонентів навігації і стабільність відображення навіть на малих екранах (наприклад, iPhone 5). Таким чином, архітектура демонструє ефективність і практичну придатність для створення інтуїтивних та зручних інтерфейсів для редакторів віртуальних екскурсій, орієнтованих на мобільні пристрої.

Ключові слова: віртуальні екскурсії, адаптивний інтерфейс, мобільний дизайн, UI/UX дизайн, автоматизоване тестування, архітектура інтерфейсу.

I. V. HAVRYSH

2nd year Master's Student
of Specialty 121 Software Engineering
Zaporizhzhia National University
ORCID: 0009-0007-5294-2146

A. G. KRYVOKHATA

Candidate of Physical and Mathematical Sciences,
Associate Professor at the Department of Software Engineering
Zaporizhzhia National University
ORCID: 0000-0001-9664-0659

DEVELOPMENT OF THE ARCHITECTURE OF AN ADAPTIVE INTERFACE OF THE EDITOR OF A WEB APPLICATION FOR CREATING VIRTUAL TOURS

The rapid development and spread of virtual reality technologies facilitates their active use in the fields of tourism, real estate, and education. VR tours are one of the ways to present information in a virtual environment. This paper analyzes existing systems for creating and editing virtual tours, examines modern approaches to building mobile interfaces, and identifies several significant limitations in their current implementations. In particular, it has been found that most services have limited functionality on mobile devices, particularly those related to tool menus, scene control, and the stability of displaying tiny elements on small screens.

The goal of this work is to develop and test the architecture of an adaptive interface for an editor of a virtual tour web application that is capable of fully replicating the functionality of the desktop version in a limited screen space. The proposed architecture is based on the Mobile-First strategy, a system of modular components, and the principles of distributing interface workspaces. It includes structural elements such as the top navigation bar, 3D scene workspace, bottom sheet, and navigation drawer. This approach ensures rational use of the available screen space, a clear separation between navigation, tools, and the virtual scene working environment, as well as predictable interface behavior when the viewport size changes.

An interface mockup was created using the Vue.js framework to test the architecture in practice. The proposed approach was tested using a set of automated UI tests with Playwright, covering the verification of the location, scaling, and accessibility of elements when working with different viewport sizes. The results confirmed the absence of layout conflicts, the correct operation of navigation components, and stable display even on small screens (e.g., iPhone 5). Thus, the architecture demonstrates its effectiveness and practical applicability for creating intuitive and convenient interfaces for virtual tour editors for mobile devices.

Key words: virtual tours, adaptive interface, mobile design, UI/UX design, automated testing, interface architecture.

Постановка проблеми

Упродовж останнього десятиліття технології віртуальної (англ. VR – Virtual Reality) та доповненої реальності (англ. AR – Augmented Reality) зазнали стрімкого розвитку й набули широкого застосування у засобах масової комунікації, навчання й цифрового маркетингу. Інтеграція цих технологій у сферу інтернет-ресурсів, освіти, архітектури та ігрової індустрії змінила способи взаємодії з контентом завдяки впровадженню нових підходів, що ґрунтуються на принципах повної або часткової присутності користувача у віртуальному просторі. VR і AR дозволяють користувачам не лише споглядати інформацію, а й взаємодіяти з нею у тривимірному середовищі, що підвищує рівень залучення, мотивації та ефективності засвоєння даних. Актуальність цих технологій підтверджується їх активним впровадженням у освітні процеси [1, 2], професійну підготовку [3, 4], медичну практику [5] та бізнес-сектор [6], де створення якісного користувацького досвіду є важливим завданням.

Значну роль у поширенні технологій VR та AR відіграє WebXR – відкритий стандарт, розроблений W3C, який забезпечує єдиний програмний інтерфейс для створення та відтворення XR-контенту безпосередньо у веб-браузері. Практична реалізація програмного забезпечення з підтримкою WebXR здійснюється за допомогою спеціалізованих фреймворків, зокрема A-Frame, Three.js та Babylon.js, які надають високий рівень абстракції для створення інтерактивних 3D-середовищ [7]. Ці інструменти дозволяють швидко прототипувати та розгортати складні тривимірні сцени безпосередньо у браузері, а також адаптувати їх під різноманітні бізнес-завдання. Одним із поширених варіантів застосування даних технологій стали VR-тури 360 – інтерактивні віртуальні екскурсії, що поєднують панорамні фотографії, відео та інколи тривимірну графіку з елементами навігації для створення справжнього імерсивного досвіду, достовірного відтворення реальних локацій та пам'яток культури.

Водночас активне зміщення медіаспоживання в бік мобільних пристроїв підвищує вимоги до гнучкості та зручності користувацьких інтерфейсів. Сучасні підходи до UX/UI-дизайну наголошують, що мобільний UI не може бути простою адаптацією десктопної версії – він потребує окремого проєктування за принципом Mobile-First, який ставить у пріоритет простоту, контекстність та інтуїтивну взаємодію [8]. У VR-застосунках зазначене завдання ускладнюється додатковими чинниками: тривимірним середовищем, використанням жестового та сенсорного керування, обмеженою продуктивністю мобільних пристроїв, а також відсутністю звичних фізичних інструментів введення.

Попри наявний попит на створення віртуальних екскурсій, аналіз ринку демонструє, що лише невелика кількість існуючих застосунків пропонує повнофункціональні мобільні інтерфейси для створення VR-контенту [9, 10]. Інші сервіси або обмежують функціональні можливості роботи на мобільних пристроях, або взагалі не надають їх. Ключова проблема полягає у складності адаптації багатофункціонального інтерфейсу для роботи з 3D-об'єктами до умов обмеженого екранного простору та сенсорного керування. Дослідження у галузі перевірки використовуваності (англ. usability) VR-застосунків для туризму свідчать, що навіть базові інтерфейси потребують ретельного тестування для забезпечення позитивного користувацького досвіду [11, 12]. Тому завдання проєктування повноцінного адаптивного інтерфейсу редактора для мобільних платформ залишається актуальною науково-практичною проблемою.

Аналіз останніх досліджень і публікацій

Автори дослідження [13] зазначають, що адаптація веб-інтерфейсів досягається за допомогою принципів Responsive Web Design (RWD), які передбачають використання гнучких сіток, адаптивних зображень, відносних одиниць вимірювання та CSS media queries для динамічної зміни компоновання залежно від розміру та типу екрана. Даний підхід, запропонований в [14], став стандартом для більшості сучасних веб-застосунків, оскільки забезпечує єдину кодову базу для різних типів пристроїв та мінімізує потребу у створенні окремих версій сайту. Однак під час створення складних інтерактивних систем, зокрема веб-редакторів, проста зміна компоновання елементів інтерфейсу не гарантує забезпечення оптимального користувацького досвіду, оскільки сценарії використання суттєво відрізняються між десктопними та мобільними середовищами.

На відміну від RWD, який динамічно адаптує єдину HTML-сторінку до розміру вікна перегляду (англ. viewport), Adaptive Web Design (AWD) використовує заздалегідь підготовлені версії макета сайту, щоб забезпечити його оптимальний вигляд і роботу на різних пристроях. У роботі [15] вказано, що такий підхід забезпечує точний контроль над користувацьким досвідом, але вимагає додаткових ресурсів для підтримки та оновлення декількох версій одночасно. Підхід AWD використовує серверну або клієнтську логіку для визначення характеристик пристрою і надає відповідний варіант інтерфейсу, оптимізованого під конкретний варіант взаємодії та роздільну здатність екрана. Це є особливо ефективним у складних інтерактивних системах, де якість взаємодії залежить не лише від розміру екрана, а й від способу введення, обчислювальної потужності пристрою та швидкості Інтернет з'єднання.

Підхід Desktop-First, характерний для звичних веб-застосунків, може призводити до зниження зручності, інтуїтивності та продуктивності при використанні на мобільних пристроях. Зокрема, багатоклонкові макети та статичні панелі інструментів, ефективні в десктопному середовищі, на малих екранах можуть створювати надмірне когнітивне навантаження та ускладнювати процес навігації. У дослідженні [16] зазначається, що використання архітектурних шаблонів для десктопних систем без врахування контексту мобільних пристроїв призводить до втрати інтуїтивності та ефективності користувацького інтерфейсу. Він стверджує, що сучасне проєктування має базуватися на принципах Mobile-First, які передбачають створення дизайну насамперед для мобільного середовища з подальшою адаптацією до більших екранів. Такий підхід забезпечує логічну послідовність елементів, робить навігацію веб-застосунком більш інтуїтивною та зручною, і покращує доступність контенту.

Дизайн ефективних мобільних макетів, описаних у роботі [17], ґрунтується не лише на адаптивній верстці, а й на застосуванні розповсюджених практик до взаємодії з користувачем за допомогою таких елементів графічного інтерфейсу, як панелі навігації, меню та модальні компоненти, що забезпечують логічну структуру інтерфейсу та зручну взаємодію з застосунком. Подібні елементи, згідно з дослідженням [18], допомагають зосередити увагу користувача на основних діях, що сприяє пришвидшенню адаптації до роботи з додатком та покращує ефективність виконання цільових задач.

Водночас успішна адаптація інтерфейсу до мобільних платформ неможлива без урахування архітектурних особливостей побудови застосунків. Як зауважено у роботі [19], недостатньо забезпечити лише візуальну адаптивність елементів інтерфейсу. Архітектура мобільного UI має комплексно враховувати вимоги до продуктивності, масштабованості та доступності, оскільки саме ці характеристики безпосередньо визначають якість користувацького досвіду та стабільність роботи системи.

Отже, з огляду літератури можна зробити висновок, що сучасна адаптація веб-інтерфейсів виходить за межі класичного RWD і трансформується у комплексний підхід, що поєднує принципи дизайну Mobile-First, оптимізації структури макету в залежності від контексту взаємодії та використання типових елементів керування та навігації. Такий підхід дозволяє створювати інтерфейси, які забезпечують ефективну взаємодію з користувачем, високу продуктивність та зручність використання саме в умовах мобільних платформ.

Формулювання мети дослідження

Метою дослідження є розробка та тестування архітектури адаптивного інтерфейсу для редактора сцен VR-турів. Запропонована концепція має забезпечити ефективну взаємодію з інтерактивними та мультимедійними елементами віртуального середовища в умовах обмеженості ресурсів мобільних пристроїв. Особливу увагу приділено відображенню складних багаторівневих меню та панелей налаштувань без втрати функціональності.

Викладення основного матеріалу дослідження

Адаптація веб-інтерфейсів під мобільні пристрої вимагає не лише гнучкої верстки чи масштабування елементів, а передусім – архітектурного переосмислення взаємодії користувача з інтерфейсом. Сучасні підходи ґрунтуються на принципах Mobile-First, контекстної взаємодії та зменшення використання зайвих елементів, що є ключовими складовими нативного UX-дизайну. Відповідно до цих принципів, інтерфейс має забезпечувати інтуїтивну навігацію, зручне керування контентом із застосування сенсорного введення та адаптацію контекстних елементів до обмеженого простору екрана. Ключовим архітектурним рішенням є ієрархічна організація інформації, коли основний функціонал займає більшу частину вільного простору екрану, а вторинні дії приховуються у модальних вікнах, висувних панелях та контекстних меню. Ергономіка сенсорної взаємодії вимагає розміщення

ключових елементів керування переважно в нижній частині екрана. Окрім цього, важливу роль відіграє керування станами інтерфейсу, коли набір інструментів повністю змінюється залежно від обраного режиму (наприклад, режим перегляду проти режиму редагування). Ефективна мобільна архітектура також повинна пропагувати використання сенсорних жестів (наприклад, стискання для масштабування або свайпи) для виконання дій, що раніше вимагали окремих кнопок.

Архітектура адаптивного інтерфейсу, зображеного на рисунку 1, для мобільних пристроїв веб-застосунку редактора віртуальних турів побудована за модульно-компонентним принципом і складається з чотирьох основних структурних компонентів:

- верхня панель керування (рис. 1.а) – компонент із фіксованим розташуванням у верхній частині екрану, який відображає вміст та дії, пов'язані з поточною сценою. Панель може використовуватись для розміщення брендингу, заголовків екранів, елементів навігації та основних контекстних дій. Головне призначення панелі – забезпечення швидкого доступу до глобальних дій без перевантаження робочого простору;

- тіло, або робочий простір сцени (рис. 1.б) – центральний компонент архітектури, який відображає вміст тривимірного середовища та дозволяє користувачу взаємодіяти з 3D-об'єктами за допомогою сенсорних жестів. Компонент безпосередньо використовується як контейнер для WebXR-сумісних фреймворків (A-Frame, Three.js, Babylon.js), що дозволяє відокремити логіку побудови 3D-графіки від решти користувацького інтерфейсу. Додаткові елементи керування можна реалізувати з використанням плаваючих кнопок для основних дій або контекстних елементів, що з'являються лише при взаємодії з об'єктами сцени;

- bottom sheet (рис. 1.в) – універсальний компонент призначений для розміщення додаткового контенту, елементів меню або дій, закріплений у нижній частині екрана. Може підтримувати два стани: згорнутий – виступає в ролі елемента “footer” та відображає допоміжну інформацію, і розгорнутий – займає частковий або весь простір екрану. У згорнутому виді можна розмістити, наприклад, горизонтальний список всіх об'єктів сцени для забезпечення більш зручної та швидкої навігації у 3D-середовищі. У розгорнутому стані, bottom sheet може мати свій макет інтерфейсу, що дозволяє ізолювати додатковий або контекстно-залежний функціонал та запобігти перевантаженню основного робочого простору. Наприклад, такий підхід можна використати для відображення панелі налаштувань об'єктів сцени, реалізованої з використанням вкладок (англ. tabs) для логічного групування елементів та використання текстових полів, слайдерів та випадаючих меню для редагування параметрів налаштувань;

- бічне меню (англ. navigation drawer) (рис. 1.г) – компонент інтерфейсу, який використовується для забезпечення як основної, так і додаткової навігації застосунком та може відкриватися/закриватися за допомогою іконки меню навігації. Він є типовим для мобільних застосунків і може надавати як постійний доступ до глобальних елементів навігації, так і мати власну реалізацію в залежності від екрану. В контексті редактора віртуальних турів, бічне меню може використовуватись для відображення списку всіх сцен туру для забезпечення швидких переходів між ними.

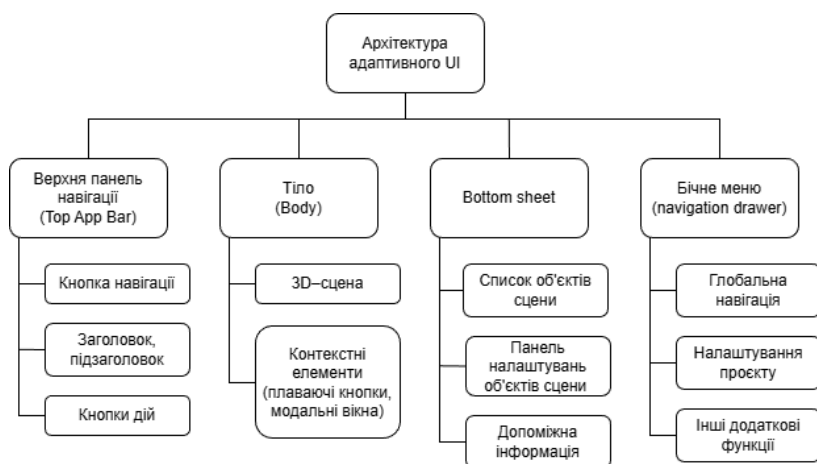


Рис. 1. Архітектура адаптивного UI: а) верхня панель навігації; б) тіло; в) bottom sheet; г) бічне меню

Таким чином, в основі архітектури лежить поєднання адаптивного та компонентного підходів. Структура інтерфейсу дотримується ключових принципів мобільного дизайну: чіткої ієрархії, модульності та відображення інформації в залежності від контексту. Особливу увагу приділено використанню звичних принципів взаємодії – основні елементи керування розміщені в нижній частині екрана для зручного доступу, а діалогові вікна реагують на жести, знайомі користувачам пристроїв з операційними системами Android та iOS. Така організація забезпечує природну взаємодію та наближає використання веб-застосунку до досвіду роботи з мобільними додатками.

Для перевірки працездатності архітектури та оцінки коректності поведінки компонентів інтерфейсу у різних умовах було створено практичний приклад реалізації редактора, який відтворює ключові сценарії взаємодії користувача. Реалізація побудована за принципом Mobile-First і відповідає вимогам до продуктивності, інтуїтивності та стабільної роботи на мобільних пристроях. Загальна структура (рис. 2) передбачає три основні складові: систему навігації, що забезпечує перехід між сценами та збереження результатів; робочий простір 3D-сцени, де користувач взаємодіє з елементами віртуальної екскурсії (наприклад, на базі A-Frame); та панель налаштувань, яка містить контекстно-залежні вкладки для керування параметрами вибраних елементів. Практичну реалізацію створено з використанням фреймворку Vue.js, однак архітектура залишається універсальною й може бути впроваджена за допомогою інших бібліотек або класичного JavaScript у поєднанні з CSS.



Рис. 2. Загальна структура інтерфейсу

Система навігації є базовим елементом інтерфейсу редактора та забезпечує доступ до ключових функцій застосунку – переходу між сценами, збереженням результатів роботи та поверненню до попереднього екрану. Одними з основних компонентів для навігації на мобільних пристроях є верхня панель керування та бічне меню. Відповідно до прототипу, зображеному на рисунку 3, панель керування складається з кнопки “Назад”, назви поточної сцени та кнопки збереження змін. Таке компонування елементів відповідає звичним практикам побудови інтерфейсів та сприяє більш швидкій адаптації користувачів до роботи з застосунком.

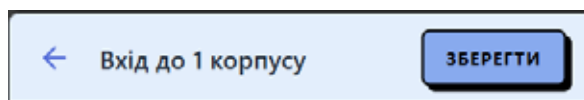


Рис. 3. Реалізація верхньої панелі керування

Бічне меню містить список всіх сцен (рис. 4), з яких складається віртуальна екскурсія, що забезпечує зручну та ефективну навігацію між ними. Кожен елемент списку відображає мініатюру сцени та її назву, що дозволяє користувачу швидко ідентифікувати потрібну локацію. Використання висувної панелі замість постійно видимого списку дозволяє оптимально використати обмежений простір мобільного екрану.

Компонент робочого простору 3D-сцени (рис. 5) є основним місцем взаємодії користувача з віртуальним середовищем для додавання, редагування та розміщення елементів сцени. За реалізацію цього компоненту відповідають фреймворки, що забезпечують створення віртуальних 3D-середовищ з підтримкою стандарту WebXR, наприклад, A-Frame. Фреймворк відповідає за такі ключові механізми взаємодії з VR-контентом, як рендеринг панорам 360 та примітивів, відображення інтерактивних елементів, зображень тощо. Користувач може повертати сцену за допомогою жестів (свайп, перетягування), змінювати масштаб та взаємодіяти з об'єктами з використанням сенсорного керування.

За додавання нових елементів до сцени відповідає плаваюча кнопка з меню (англ. floating action button menu), яка є одним з компонентів дизайн-системи Material Design 3 від Google. Такий тип кнопки активно використовується в застосунках Google, наприклад, Google Drive або Keep Notes, для відображення декількох варіантів додавання нового ресурсу. В розробленому макеті, при натисканні на кнопку розгортається список доступних

типів об'єктів: хотспоти для переходу між сценами, мультимедійні елементи (аудіо, відео, зображення) та текстові анотації. Після вибору потрібного типу, об'єкт розміщується відповідно до координат розташування віртуального курсора, який потім користувач може вільно перемістити в межах сцени.



Рис. 4. Реалізація бічного меню



Рис. 5. Макет компоненту 3D-сцени

Панель налаштувань, зображена на рисунку 6, виконує функцію контекстного інструменту редагування, який динамічно змінює вміст в залежності від обраного об'єкта сцени. Для відображення такого складного елементу інтерфейсу використовується компонент bottom sheet. У згорнутому стані (рис. 6.а) він відображає горизонтальний список всіх об'єктів сцени, що дозволяє швидко переключатись між ними. У розгорнутому (рис. 6.б) – надає доступ до параметрів налаштувань обраного об'єкту. Всі доступні налаштування логічно згруповані та розділені на окремі вкладки. Це сприяє структурованості інтерфейсу та зменшує когнітивне навантаження при роботі з великою кількістю параметрів. Для кожного типу параметрів використовуються оптимальні елементи керування, що забезпечують баланс між точністю налаштування та зручністю взаємодії. Зокрема, позиціонування об'єктів у сцені реалізовано з використанням комбінації слайдера та числового поля введення для координат X, Y та Z. Для параметрів, що мають обмежений набір варіантів, таких як колір тексту, або цільова сцена для переходу, використано випадаючі списки. Всі зміни автоматично застосовуються до обраного об'єкта та відображаються у вікні сцени. Користувач може змінювати стан панелі за допомогою свайпу вгору/вниз або натискання на відповідний маркер.

Важливим аспектом реалізації є оптимізація продуктивності при роботі з панеллю налаштувань на мобільних пристроях з обмеженими ресурсами. Вміст вкладок має завантажуватись динамічно лише при їх активації, щоб зменшити початкове навантаження на систему та прискорити відкриття панелі. Також можливе застосування

техніки дебаунсингу (англ. debouncing) для обробки змін параметрів, що запобігає надмірному оновленню сцени при швидкому оновленні значень за допомогою слайдерів чи текстових полів.

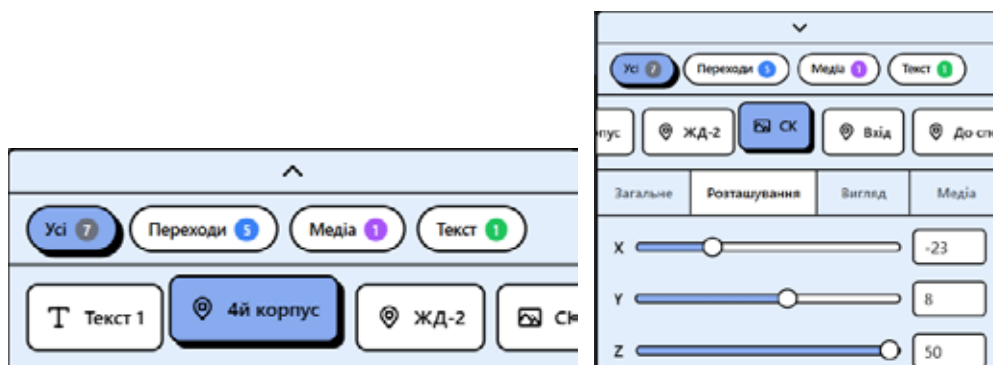


Рис. 6. Макет панелі налаштувань об'єктів сцени: а) згорнутий вид; б) розгорнутий вид

Тестування є одним з етапів циклу розробки програмного забезпечення та необхідне для забезпечення стабільності, передбачуваності роботи та якості кінцевого продукту. Тестування дозволяє виявити помилки на стадії реалізації застосунку, зменшити ризики помилок та перевірити відповідність системи функціональним та нефункціональним вимогам. Автоматизоване тестування забезпечує швидку та систематичну перевірку функціональності застосунку шляхом автоматичного виконання попередньо визначених тестових сценаріїв, що зменшує потребу в ручному тестуванні. Автори дослідження [20] відзначають, що автоматизація суттєво знижує витрати на підтримку програмного забезпечення та розширює тестове покриття, що є критичним для комплексних та масштабних проєктів. Існують різні види автоматизованого тестування, зокрема модульне тестування, інтеграційні тести, UI-тести та тестування продуктивності, кожен з яких призначений для перевірки окремих компонентів системи або їх взаємодії. Для тестування фронтенд-застосунків та мобільних веб-інтерфейсів широко застосовуються фреймворки Jest, Vitest, Mocha, Playwright, Cypress, Selenium WebDriver, Appium тощо, які дозволяють автоматизувати як перевірку логіки, так і змодельовати взаємодію користувача з системою. Використання зазначених інструментів для тестування адаптивних інтерфейсів дозволяє перевіряти коректність роботи UI при різних розмірах екранів, конфігураціях пристроїв та сценаріях взаємодії.

Для перевірки запропонованої архітектури адаптивного інтерфейсу було розроблено набір автоматизованих тестів з використанням Playwright [21]. Playwright – це фреймворк з відкритим кодом для автоматизації комплексного та кроссбраузерного тестування веб-застосунків. Серед особливостей цього фреймворку можна виділити підтримку емуляції мобільних браузерів Google Chrome та Safari, необхідних для перевірки роботи адаптивного інтерфейсу. Метою тестування є оцінка ефективності обраного архітектурного підходу для створення мобільного інтерфейсу редактора VR-турів. Тести охоплюють валідацію таких елементів UI, як панель навігації, бічне меню, сцена та компоненти bottom sheet, і спрямовані на перевірку роботи інтерфейсу, його адаптивності для типових сценаріїв використання користувачем на різних мобільних пристроях. Тестування дозволить виявити такі проблеми у структурі інтерфейсу, як накладання елементів один на одного, неправильне масштабування компонентів або порушення їхнього розташування. Крім того, перевірка переходів між екранами та зміни відображуваних даних спрямована на оцінку роботи системи навігації застосунку та взаємодії його компонентів.

Для тестування коректної роботи адаптивності інтерфейсу було підібрано набір поширених значень розмірів вікна перегляду, які відповідають розмірам екранів існуючих мобільних пристроїв (табл. 1) – від компактних смартфонів (iPhone 5/SE, iPhone 8) до сучасних фаблетів (iPhone 12 Pro Max). Увагу приділено критичним точкам, де найчастіше виникають проблеми з розташуванням елементів при адаптації веб-сторінки до поточної ширини екрану. Такий підхід дозволяє перевірити поведінку інтерфейсу в умовах різної щільності пікселів, співвідношення сторін і фізичних розмірів екрану, що забезпечує комплексну оцінку адаптивності інтерфейсу.

Таблиця 1

Використанні значення розміру вікна перегляду для тестування адаптивності [22]

Пристрій	Розмір екрану в пікселях	Розмір вікна перегляду
iPhone 5/SE	640 x 1136	320 x 568
iPhone 8	750 x 1334	375 x 667
iPhone 12 Pro Max	2778 x 1284	428 x 926

На рисунку 7 представлено приклад автоматизованого тесту, який перевіряє адаптивність бічного меню при зміні розміру вікна перегляду. Тест має забезпечити, що всі пункти меню залишаються видимими, не перекривають одне одного та не виходять за межі екрану. Така перевірка гарантує, що навігація залишається стабільною та доступною незалежно від розміру пристрою.

```
test('Drawer Menu adapts to different screen sizes', async ({ page }) => {
  const drawerElements = ['mobile-drawer-overlay', 'mobile-drawer-header', 'mobile-drawer-scenes-list'];

  for (const viewport of viewports) {
    await page.setViewportSize(viewport);

    // Open the drawer
    await page.getByTestId('drawer-trigger').click();

    const drawerMenu = page.getByTestId('mobile-drawer-menu');
    await expect(drawerMenu).toBeVisible();

    await Promise.all(
      drawerElements.map(elementTestId =>
        expect(page.getByTestId(elementTestId)).toBeVisible()
      )
    );

    console.log('Drawer menu adapts correctly to viewport ${viewport.width}x${viewport.height}');

    // Close the drawer before next iteration
    await page.getByTestId('drawer-close-button').click();
    await expect(drawerMenu).toBeHidden();
  }
});
```

Рис. 7. Приклад автоматизованого тесту

Результати виконання повного набору автоматизованих тестів підтвердили стабільність роботи мобільного інтерфейсу редактора (рис. 8). Валідацію компонентів – верхньої панелі навігації, бічного меню, робочої області сцени та панелі bottom sheet – на трьох цільових розмірах viewport було виконано без виявлення конфліктів верстки чи втрати функціональності. Інтерфейс демонстрував коректну поведінку при зміні станів, зокрема під час навігації, відкриття меню та трансформації панелей. Отримані дані свідчать про ефективність розробленої архітектури адаптивного інтерфейсу у забезпеченні передбачуваного користувацького досвіду на мобільних пристроях.

Simulator Mobile UI - Adaptive UI		
✓ Bottom Sheet adapts to different screen sizes	Mobile Chrome	24s
✓ Drawer Menu adapts to different screen sizes	Mobile Chrome	25s
✓ Header adapts to different screen sizes	Mobile Chrome	16s
✓ Scene Body adapts to different screen sizes	Mobile Chrome	11s
Simulator Mobile UI - Bottom Sheet		
✓ Opens and closes to fixed heights correctly	Mobile Chrome	11s
✓ Contains expected content	Mobile Chrome	19s
✓ Shows important UI elements when closed	Mobile Chrome	7s
✓ Covers less important UI elements when opened	Mobile Chrome	6s
Simulator Mobile UI - Drawer Menu		
✓ Contents are scrollable when expanded	Mobile Chrome	12s
✓ Contents fit within bottom sheet width	Mobile Chrome	16s
Simulator Mobile UI - Drawer Menu		
✓ Toggles correctly	Mobile Chrome	14s
✓ Clicking on the overlay area closes the drawer	Mobile Chrome	16s
✓ Contains header, scenes list, and close button	Mobile Chrome	14s
✓ Displays over other content	Mobile Chrome	15s
✓ Clicking on a scene navigates to that scene	Mobile Chrome	15s

Рис. 8. Результати виконання повного набору тестів

Висновки

У роботі проаналізовано проблему відсутності єдиної архітектури користувацького інтерфейсу для створення та редагування VR-турів 360 на мобільних пристроях. Виявлено, що доступні рішення обмежені у використанні через складність адаптації комплексних інтерфейсів до екранів смартфонів та особливостей сенсорного керування, пропонують лише часткову функціональність або взагалі не підтримують адаптивний дизайн. Для вирішення цієї задачі запропоновано та обґрунтовано архітектуру адаптивного інтерфейсу, що базується на підході Mobile-First та використанні контекстно-залежних елементів, таких як компонент bottom sheet, модальні вікна та меню. Розроблена архітектурна модель спрямована на забезпечення повної підтримки функціональності редактора віртуальних екскурсій без перевантаження доступного екранного простору. Перевірку запропонованого

рішення було здійснено шляхом реалізації прототипу та проведення серії автоматизованих тестів на різних розмірах екрану. Результати тестування підтвердили коректність розташування та масштабування елементів, а також надійність механізмів навігації на мобільних пристроях. Таким чином, розроблене рішення забезпечує надійну основу для створення сучасного та масштабованого адаптивного інтерфейсу для редактора віртуальних екскурсій.

Перспективи подальших досліджень пов'язані з оптимізацією продуктивності рендерингу 3D-сцен та компонентів інтерфейсу на мобільних пристроях із низькою обчислювальною потужністю, що залишається важливим технічним обмеженням. Крім того, перспективним напрямком є дослідження можливостей масштабування запропонованої архітектури для планшетів або пристроїв зі складними екранами. Також доцільно провести розширені емпіричні дослідження адаптивного інтерфейсу за участі реальних користувачів для оцінки його використовуваності під час тривалої роботи з редактором. Актуальним напрямком є й дослідження розробки інтерфейсу редактору віртуальних екскурсій для використання у режимі повного занурення безпосередньо у VR-гарнітурах.

Список використаної літератури

1. Burke D., Crompton H., Nickel C. The Use of Extended Reality (XR) in Higher Education: A Systematic Review. *TechTrends*. 2025. № 69. P. 998-1011. URL: <https://doi.org/10.1007/s11528-025-01092-y>
2. Thangavel S., Sharmila K., Sufina K. Revolutionizing Education Through Augmented Reality (AR) and Virtual Reality (VR): Innovations, Challenges and Future Prospects. *Asian Journal of Interdisciplinary Research*. 2025. Vol. 8, Iss. 1. P. 1-28. URL: <https://doi.org/10.54392/ajir2511>
3. Wang K. Research and Application of VR in Training and Learning. *Highlights in Science, Engineering and Technology*. 2024. Vol. 85. P. 1306–1313. URL: <https://doi.org/10.54097/anhqyy16>
4. Shahu A., Kinzer K., Michahelles F. Enhancing Professional Training with Single-User Virtual Reality: Unveiling Challenges, Opportunities, and Design Guidelines. *MUM '23: Proceedings of the 22nd International Conference on Mobile and Ubiquitous Multimedia (Vienna, 3-6 December 2023)*. Vienna, 2024. P. 224-256. URL: <https://doi.org/10.1145/3626705.3627791>
5. Virtual reality and augmented reality in medical education: an umbrella review / T. Tene, D. F. V. Lopez, P. E. V. Aguerre et al. *Frontiers in Digital Health*. 2024. Vol. 6:1365345. DOI: 10.3389/fdgh.2024.1365345
6. What is XR? Towards a Framework for Augmented and Virtual Reality / P. A. Rauschnabel, R. Felix, C. Hinsch et al. *Computers in Human Behavior*. 2022. Vol. 133. P. 107289. URL: <https://doi.org/10.1016/j.chb.2022.107289>
7. Macario G. WebXR, A-Frame and Networked-Aframe as a Basis for an Open Metaverse: A Conceptual Architecture. 2024. URL: <https://arxiv.org/abs/2404.05317>
8. Govindaraju M. The Impact Of Mobile-First Development Approaches On User Experience And Development Efficiency. *International Journal of Creative Research Thoughts*. 2025. Vol. 13, no. 3. URL: <https://www.ijcrt.org/papers/IJCRT25A3445.pdf>
9. Kuula: #1 Virtual Tour Software for any business. *Kuula*. URL: <https://kuula.co/> (дата звернення: 20.09.2025).
10. Virtual Tour Easy. *360° Virtual Tour software*. URL: <https://virtualtoureasy.com/> (дата звернення: 20.09.2025).
11. Fang Y.-M., Lin C. The Usability Testing of VR Interface for Tourism Apps. *Applied Sciences*. 2019. Vol. 9(16). P. 3215. URL: <https://doi.org/10.3390/app9163215>
12. Kamińska D., Zwoliński G., Laska-Leśniewicz A. Usability Testing of Virtual Reality Applications—The Pilot Study. *Sensors*. 2022. Vol. 22(4). P. 1342. URL: <https://doi.org/10.3390/s22041342>
13. Bhanarkar N., Paul A., Mehta A. Responsive web design and its impact on user experience. *International Journal of Advanced Research in Science, Communication and Technology*. 2023. Vol. 3, Is. 4. P. 50-55. DOI: 10.48175/ijarsct-9259
14. Marcotte E. *Responsive Web Design*. 2nd ed. New York : A Book Apart, 2014. 153 p.
15. Responsive web design: principles and implementation / R. Rupali, D. P. Mankame, B. Patil et al. *International Journal of Research Publication and Reviews*. 2024. Vol. 5, Is. 7. P. 4291-4293. URL: <https://ijrpr.com/uploads/V5ISSUE7/IJRPR31795.pdf>
16. Prakash B. Retrofitting Mobile First Design, Responsive Design: Driving Factors, Approach, Best Practices and Design Considerations. *Current Trends in Computer Sciences & Applications*. 2020. Vol. 2, Is. 2. DOI: 10.32474/CTCSA.2020.02.000131
17. Kim H., Moritz D., Hullman J. Design Patterns and Trade-Offs in Responsive Visualization for Communication. 2021. URL: <https://arxiv.org/abs/2104.07724>
18. Understanding Visual Saliency in Mobile User Interfaces / L. A. Leiva et al. 2021. URL: <https://arxiv.org/abs/2101.09176>
19. Bilohub D., Skrypchenko M., Tytenko S. Quality attributes and architectural patterns of modern mobile apps. *Modern Engineering And Innovative Technologies*. 2023. № 1(29-01). P. 33-38. URL: <https://doi.org/10.30890/2567-5273.2023-29-01-056>
20. Gu R., Rojas J. M., Shin D. Software testing for extended reality applications: a systematic mapping study. *Automated Software Engineering*. 2025. № 32(56). URL: <https://doi.org/10.1007/s10515-025-00523-7>

21. Fast and reliable end-to-end testing for modern web apps. *Playwright*. URL: <https://playwright.dev/> (date of access: 07.11.2025).
22. Common Viewport Sizes for Mobile. *Award Winning Web Design Newport – Digital Marketing*. URL: <https://www.icwebdesign.co.uk/common-viewport-sizes> (date of access: 07.11.2025)

References

1. Burke, D., Crompton, H., & Nickel, C. (2025). The Use of Extended Reality (XR) in Higher Education: A Systematic Review. *TechTrends*, (69), 998-1011. <https://doi.org/10.1007/s11528-025-01092-y>
2. Thangavel, S., Sharmila, K., & Sufina, K. (2025). Revolutionizing Education Through Augmented Reality (AR) and Virtual Reality (VR): Innovations, Challenges and Future Prospects. *Asian Journal of Interdisciplinary Research*, 8(1), 1-28. <https://dx.doi.org/10.2139/ssrn.5195697>
3. Wang, K. (2024). Research and Application of VR in Training and Learning. *Highlights in Science, Engineering and Technology*, 85, 1306-1313. <https://doi.org/10.54097/anhqyy16>
4. Shahu, A., Kinzer, K., & Michahelles, F. (2024). Enhancing Professional Training with Single-User Virtual Reality: Unveiling Challenges, Opportunities, and Design Guidelines. *Proceedings of the 22nd International Conference on Mobile and Ubiquitous Multimedia*, 224-256. <https://doi.org/10.1145/3626705.3627791>
5. Tene, T., Vique López, D. F., Valverde Aguirre, P. E., Orna Puente, L. M., & Vacacela Gomez, C. (2024). Virtual reality and augmented reality in medical education: an umbrella review. *Frontiers in Digital Health*, 6:1365345. <https://doi.org/10.3389/fdgth.2024.1365345>
6. Rauschnabel, P. A., Felix, R., Hinsch, C., Shahab, H., & Alt, F. (2022). What is XR? Towards a framework for augmented and virtual reality. *Computers in human behavior*, 133, 107289. <https://doi.org/10.1016/j.chb.2022.107289>
7. Macario, G. (2024). WebXR, A-Frame and Networked-Aframe as a Basis for an Open Metaverse: A Conceptual Architecture. <https://arxiv.org/abs/2404.05317>
8. Govindaraju, M. (2025). The Impact Of Mobile-First Development Approaches On User Experience And Development Efficiency. *International Journal of Creative Research Thoughts*, 13(3). <https://www.ijcrt.org/papers/IJCRT25A3445.pdf>
9. Kuula: #1 Virtual Tour Software for any business. (n.d.). Kuula. <https://kuula.co/>
10. Virtual Tour Easy. (n.d.). *360° Virtual Tour software*. <https://virtualtoureasy.com/>
11. Fang, Y.-M., & Lin, C. (2019). The Usability Testing of VR Interface for Tourism Apps. *Applied Sciences*, 9(16), 3215. <https://doi.org/10.3390/app9163215>
12. Kamińska, D., Zwoliński, G., & Laska-Leśniewicz, A. (2022). Usability Testing of Virtual Reality Applications – The Pilot Study. *Sensors*, 22(4), 1342. <https://doi.org/10.3390/s22041342>
13. Bhanarkar, N., Paul, A., & Mehta, A. (2023). Responsive Web Design and Its Impact on User Experience. *International Journal of Advanced Research in Science, Communication and Technology*, 3(4), 50-55. <https://doi.org/10.48175/ijarsct-9259>
14. Marcotte, E. (2014). *Responsive web design* (2nd ed.). New York : A Book Apart.
15. R, R., P Mankame, D., Patil, B., & Dhavalgi, V. (2024). Responsive Web Design: Principles and Implementation. *International Journal of Creative Research Thoughts*, 5(7), 4291-4293. <https://ijrpr.com/uploads/V5ISSUE7/IJRPR31795.pdf>
16. Prakash, B. (2020). Retrofitting Mobile First Design, Responsive Design: Driving Factors, Approach, Best Practices and Design Considerations. *Current Trends in Computer Sciences & Applications*, 2(2). <https://doi.org/10.32474/ctcsa.2020.02.000131>
17. Kim, H., Moritz, D., & Hullman, J. (n.d.). Design Patterns and Trade-Offs in Responsive Visualization for Communication. <https://arxiv.org/abs/2104.07724>
18. Leiva, L. A., Xue, Y., Bansal, A., Tavakoli, H. R., Köroğlu, T., Dayama, N. R., & Oulasvirta, A. (n.d.). Understanding Visual Saliency in Mobile User Interfaces. <https://arxiv.org/abs/2101.09176>
19. Bilohub, D., Skrypchenko, M., & Tytenko, S. (2023). QUALITY ATTRIBUTES AND ARCHITECTURAL PATTERNS OF MODERN MOBILE APPS. *Modern engineering and innovative technologies*, 1(29-01), 33-38. <https://doi.org/10.30890/2567-5273.2023-29-01-056>
20. Gu, R., Rojas, J. M., & Shin, D. (2025). Software testing for extended reality applications: a systematic mapping study. *Automated Software Engineering*, 32(56). <https://doi.org/10.1007/s10515-025-00523-7>
21. Fast and reliable end-to-end testing for modern web apps. (n.d.). Playwright. <https://playwright.dev/>
22. Common Viewport Sizes for Mobile. (n.d.). Award Winning Web Design Newport – Digital Marketing. <https://www.icwebdesign.co.uk/common-viewport-sizes>

Дата першого надходження рукопису до видання: 25.11.2025
 Дата прийнятого до друку рукопису після рецензування: 12.12.2025
 Дата публікації: 31.12.2025