

ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ  
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ  
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

**КВАЛІФІКАЦІЙНА РОБОТА**  
на тему: «Розробка програмного забезпечення для  
прототипування інтерфейсів з VR/AR»

на здобуття освітнього ступеня бакалавра  
зі спеціальності 121 Інженерія програмного забезпечення  
освітньо-професійної програми «Інженерія програмного забезпечення»

*Кваліфікаційна робота містить результати власних досліджень.  
Використання ідей, результатів і текстів інших авторів мають посилання  
на відповідне джерело*

\_\_\_\_\_ Діана СТОРЧАК  
(підпис)

Виконав: здобувачка вищої освіти група ПД-41

\_\_\_\_\_ Діана СТОРЧАК

Керівник: \_\_\_\_\_ Андрій БОНДАРЧУК  
д.т.н., професор

Рецензент: \_\_\_\_\_

**Київ 2024**

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**  
**Навчально-науковий інститут інформаційних технологій**

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти Бакалавра

Спеціальність 121 Інженерія програмного забезпечення

Освітньо-професійна програма «Інженерія програмного забезпечення»

**ЗАТВЕРДЖУЮ**

Завідувач кафедри

Інженерії програмного забезпечення

\_\_\_\_\_ Ірина ЗАМРІЙ

«\_\_\_\_\_» \_\_\_\_\_ 2024 р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

\_\_\_\_\_  
Сторчак Діані Олександрівні

1. Тема кваліфікаційної роботи: Розробка програмного забезпечення для прототипування інтерфейсів з VR/AR

керівник кваліфікаційної роботи д.т.н., професор Андрій БОНДАРЧУК,

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «27» лютого 2024 р. № 36.

2. Строк подання кваліфікаційної роботи «28» травня 2024 р.

3. Вихідні дані до кваліфікаційної роботи: науково-технічна література, протоколи передачі даних, моделі машинного навчання та клієнт-серверної взаємодії.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Аналіз сучасних вимог до інтерфейсів

2. Дослідження інструментів для розробки прототипів інтерфейсів

### 3. Розробка програмного забезпечення для прототипування інтерфейсів з VR/AR

#### 5. Перелік графічного матеріалу: презентація

1. Таблиця аналізу аналогів.
2. Програмні засоби реалізації.
3. Діаграма варіантів використання.
4. Діаграма пакетів.
5. Діаграма класів.
6. Діаграма діяльності.
7. Екранні форми.
8. Апробація результатів дослідження.

6. Дата видачі завдання «01» березня 2024 р.

### **КАЛЕНДАРНИЙ ПЛАН**

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз наявної науково-технічної літератури	01.03-07.03.24	
2	Аналіз сучасних вимог до інтерфейсів	08.03-10.03.24	
3	Дослідження настанов з доступності веб-контенту	11.03-14.03.24	
4	Дослідження інструментів для розробки прототипів інтерфейсів	15.03-20.03.24	
5	Розробка програмного забезпечення для прототипування інтерфейсів з VR/AR	21.03-01.04.24	
6	Реалізації інтерфейсів з підтримкою VR/AR	02.04-07.04.24	
7	Оформлення роботи: вступ, висновки, реферат	08.04-14.04.24	
8	Розробка демонстраційних матеріалів	15.04-25.04.24	
9	Попередній захист роботи	13.05-31.05.24	

Здобувачка вищої освіти

\_\_\_\_\_ (підпис)

Діана СТОРЧАК

Керівник кваліфікаційної роботи

\_\_\_\_\_ (підпис)

Андрій БОНДАРЧУК





## РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 64 стор., 14 рис., 26 джерела.

*Мета роботи* – метою даної роботи є підтримка процесів прототипування інтерфейсів з використанням технологій віртуальної реальності (VR) та доповненої реальності (AR) в ігрових застосунках. Основна задача полягає у розробці ефективного програмного забезпечення, яке дозволить розробникам створювати, тестувати та вдосконалювати інтерфейси для VR/AR ігор, забезпечуючи високий рівень інтерактивності та користувацького досвіду. Це включає розробку інструментів для швидкого прототипування, інтеграції з різними VR/AR платформами та забезпечення гнучкості в адаптації до різних ігрових сценаріїв.

*Об'єкт дослідження* – об'єктом дослідження є процеси прототипування інтерфейсів для VR/AR в контексті ігрових застосунків. Це охоплює всі аспекти створення та впровадження інтерфейсів, від концептуальних моделей до інтерактивних прототипів, що використовуються в ігровій індустрії. Особлива увага приділяється методологіям та інструментам, які використовуються для проектування інтерфейсів, а також специфічним вимогам та викликам, пов'язаним з VR/AR технологіями у сфері ігор.

*Предмет дослідження* – предметом дослідження є програмне забезпечення, що використовується для прототипування інтерфейсів в ігрових застосунках з використанням VR/AR технологій. Це включає аналіз існуючих програмних рішень, визначення їх можливостей та обмежень, а також розробку нових підходів і інструментів, які можуть покращити процес прототипування. Основний акцент робиться на інструментах, що дозволяють створювати інтерактивні та користувацькі інтерфейси, які забезпечують зручність та ефективність взаємодії в ігровому середовищі VR/AR.

*Короткий зміст роботи:* У роботі проведено дослідження інструментів для розробки прототипів інтерфейсів. Проаналізовано функціональні характеристики Axure, Figma, Adobe XD. Проаналізовано сучасні вимоги до інтерфейсів. Розроблено програмного забезпечення для прототипування інтерфейсів з VR/AR в ігрових застосунках.

КЛЮЧОВІ СЛОВА: ІНТЕРФЕЙС, ІГРОВІ ЗАСТОСУНКИ, ПРОТИТИПУВАННЯ VR/AR.

## ЗМІСТ

ВСТУП.....	9
1 АНАЛІЗ СУЧАСНИХ ВИМОГ ДО ІНТЕРФЕЙСІВ .....	11
1.1 Аналіз інтерфейсів.....	11
1.2 WAI Ініціатива поліпшення вебсторун.....	14
1.3 Рекомендації Міністерства цифрової трансформації України до дизайну офіційних веб-сайтів.....	16
2 ДОСЛІДЖЕННЯ ІНСТРУМЕНТІВ ДЛЯ ПРОТОТИПІВ ІНТЕРФЕЙСІВ.....	22
2.1 Інструменти для прототипів інтерфейсу.....	22
2.2 Функціональні характеристики Axure .....	25
2.3 Функціональні характеристики Figma.....	27
2.4 Функціональні характеристики Adobe XD.....	36
3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ПРОТОТИПУВАННЯ ІНТЕРФЕЙСІВ З VR/AR.....	42
3.1 Технічні аспекти реалізації інтерфейсів з підтримкою VR/AR.....	42
3.2 Логічна структура програмного забезпечення для прототипування інтерфейсів з VR/AR. ....	44
3.3 Код програми.....	45
ВИСНОВКИ.....	52
ПЕРЕЛІК ПОСИЛАНЬ.....	55
ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ.....	59
ДОДАТОК Б. ЛІСТИНГИ ПРОГРАМНИХ МОДУЛІВ.....	65



## ВСТУП

У сучасному світі технології віртуальної (VR) та доповненої реальності (AR) займають важливе місце в багатьох галузях, таких як освіта, медицина, розваги та комунікації. Наприклад, VR дозволяє лікарям проводити операції на віртуальних моделях, що значно підвищує їх професійні навички." [1]. Ці технології дозволяють створювати нові види взаємодії користувачів з цифровим контентом, що відкриває широкі можливості для інновацій та розвитку. Водночас, розробка програмного забезпечення для VR/AR потребує особливих підходів та інструментів, що дозволяють швидко та ефективно створювати та тестувати інтерфейси.

Актуальність теми дослідження полягає в тому, що розвиток VR/AR технологій вимагає нових методів прототипування інтерфейсів, які б забезпечували високу якість користувацького досвіду та швидку адаптацію до змінних вимог ринку. Прототипування є важливим етапом розробки програмного забезпечення, оскільки дозволяє тестувати та оцінювати різні варіанти дизайну до їх повної реалізації, що значно зменшує ризики та витрати.

Метою даної роботи є розробка програмного забезпечення для прототипування інтерфейсів з використанням VR/AR технологій, що дозволить забезпечити ефективну та зручну роботу з цифровим контентом. Для досягнення цієї мети були поставлені наступні завдання: провести огляд сучасних технологій та інструментів для розробки VR/AR додатків, розробити методи та засоби для створення прототипів інтерфейсів, реалізувати програмне забезпечення, яке дозволить швидко та ефективно створювати та тестувати VR/AR інтерфейси, а також провести тестування та оцінку якості розробленого програмного забезпечення.

Об'єктом дослідження є технології віртуальної та доповненої реальності, а предметом дослідження – процес розробки програмного забезпечення для прототипування інтерфейсів з використанням VR/AR технологій.

Для досягнення поставлених цілей та завдань в роботі використовувалися такі методи дослідження: аналіз та синтез наукової літератури та сучасних досліджень у галузі VR/AR, проектування та розробка програмного забезпечення, методи тестування та оцінки якості програмних продуктів.

Структура роботи включає вступ, огляд літератури та сучасних технологій VR/AR, методи та засоби розробки програмного забезпечення для VR/AR, проектування та розробку програмного забезпечення для прототипування інтерфейсів, тестування та оцінку якості програмного забезпечення, практичне застосування розробленого програмного забезпечення, висновки, список використаних джерел та додатки.

# 1 АНАЛІЗ СУЧАСНИХ ВИМОГ ДО ІНТЕРФЕЙСІВ

## 1.1 Аналіз інтерфейсів

Аналіз інтерфейсу використовується для визначення, де, чому, коли, як і для кого відбувається обмін інформацією між компонентами розв'язку або через межі розв'язку.

### Опис аналізу інтерфейсів

Інтерфейс — це з'єднання між двома компонентами або рішеннями. Більшість рішень вимагають одного або декількох інтерфейсів для обміну інформацією з іншими компонентами рішення, організаційних одиниць або бізнес-процесів.

Типи інтерфейсів включають:

- інтерфейси користувача, в тому числі люди, які взаємодіють безпосередньо з рішенням в організації,
- зовнішні рішення фізичних осіб, наприклад зацікавлених сторін або контролюючих органів,
- бізнес-процеси,
- інтерфейси передачі між системами, Інтерфейси програмування • (API) і будь-які апаратні пристрої.

Аналіз інтерфейсу ідентифікує і пояснює такі дії:

- хто буде використовувати інтерфейс,
- якою інформацією буде обмінюватися через інтерфейс, а також обсяг даних,
- коли обмінюється інформацією і як часто,
- де буде відбуватися обмін інформацією,
- навіщо потрібен інтерфейс, і як інтерфейс реалізований або повинен бути реалізований.

Раннє виявлення інтерфейсів дозволяє бізнес-аналітиці створити контекст для більш детального прояснення вимог зацікавлених сторін, тим самим

визначаючи адекватне функціональне покриття рішення для задоволення потреб зацікавлених сторін.

Раннє виявлення інтерфейсів показує, які зацікавлені сторони будуть вигодувати або залежать від різних компонентів рішення, що може допомогти бізнес-аналітиці визначити, які учасники повинні бути присутні при використанні інших методів збору інформації.

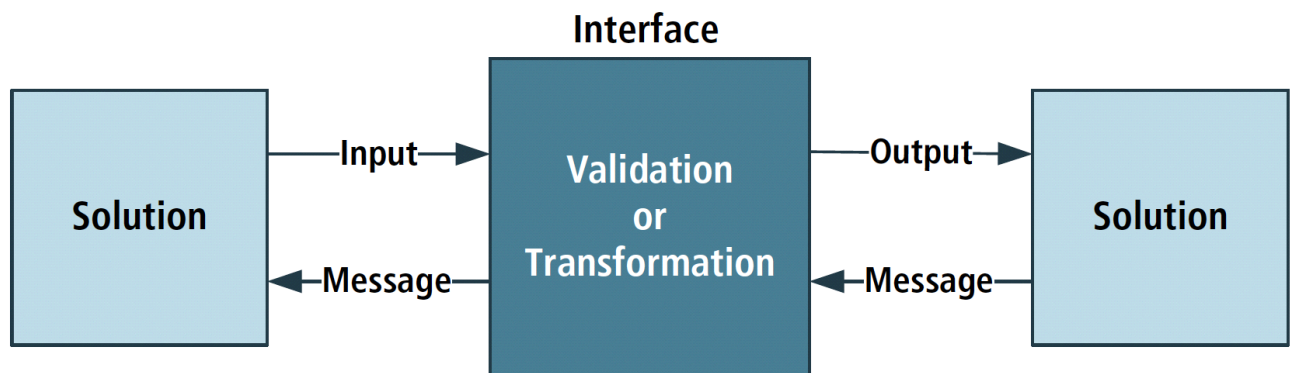


Рис. 1.1 Аналіз інтерфейсу

### Елементи Аналізу інтерфейсів

Бізнес-аналітик може використовувати інші методи, такі як аналіз документів, спостереження, моделювання сфери програм та інтерв'ю, щоб зрозуміти, які інтерфейси визначити. Контекстна діаграма може виявляти інтерфейси високого рівня між людьми, організаційними одиницями, бізнес-процесами або іншими компонентами рішення.

Результати цього аналізу можуть показати частоту використання доступних інтерфейсів, а також будь-які проблеми з ними, які можуть підсилити аргумент на користь змін. Результати також можуть допомогти визначити будь-які ключові проблеми, які потрібно вирішити для створення інтерфейсу.

Бізнес-аналітики визначають, які інтерфейси потрібні в майбутньому для кожної зацікавленої сторони або системи, яка взаємодіє з системою. Взаємовідношення між зацікавленими сторонами і інтерфейсами може бути багато

до багатьох або, в деяких випадках, один до одного.

Деякі інтерфейси можуть бути менш очевидними або менш частими, наприклад, інтерфейс, який використовується для регуляторних функцій, аудиту або навчання працівників. Ідентифіковані інтерфейси можуть включати інтерфейси з інших рішень, окрім операційного рішення.[3]

Для кожного інтерфейсу бізнес-аналітики

- описує функцію інтерфейсу
- оцінює частоту використання інтерфейсу
- оцінити, який тип інтерфейсу може бути підходящим, і
- Визначте початкові подробиці щодо інтерфейсу.

Требования інтерфейсу зосереджуються на описі вхідних і виходів цього інтерфейсу, будь-яких правил перевірки, які керують цими вхідними і виводами, а також на подіях, які спричиняють сумісність. виходи, а також події, які можуть спричинити сумісність.

Існує велика кількість можливих типів взаємодії, кожен з яких слід описати. Взаємодія може бути спричинена типовим або альтернативним потоком вхідних і вихідних даних у бізнес-рішення або винятковими подіями, такими як порушення.

Бізнес-аналітики розглянуть, хто буде використовувати інтерфейс, яка інформація передається через інтерфейс, а також коли і де відбувається взаємодія.

Інтерфейс визначає процес роботи користувача між системами, ролями і правами доступу користувача, а також будь-якими цілями керування інтерфейсом. Визначення інтерфейсу залежить від таких принципів, як вимоги щодо доступності або загальні вимоги щодо процесу роботи.

Для того, щоб виявити будь-які серйозні проблеми дизайну, інтерфейси між компонентами розв'язання або процесу та людьми потребують детального аналізу інтерфейсу, який буде виконано заздалегідь. Визначення інтерфейсу включає

- назва інтерфейсу;
- покриття або діапазон інтерфейсу;
- метод обміну між двома об'єктами;
- формат повідомлення;

- курс обміну.

Розгляди щодо використання аналізу інтерфейсу

- Залучення до аналізу інтерфейсу на ранніх стадіях дозволяє розширити функціональне покриття.
- Чітка специфікація інтерфейсів забезпечує структурований спосіб розподілу вимог, бізнес-правил та обмежень прийняття рішень.
- Завдяки широкому застосуванню, це дозволяє уникнути надмірного аналізу дрібних деталей.
- Не дає розуміння інших аспектів рішення, оскільки аналіз не оцінює внутрішні компоненти.[1]

## 1.2 WAI – Ініціатива поліпшення вебдоступу

Web Accessibility Initiative (WAI) – це ініціатива Консорціуму World Wide Web (W3C), спрямована на покращення доступності Всесвітньої павутини (WWW або Web) для людей з інвалідністю. WAI надає рекомендації, інструменти та ресурси для розробників, проектувальників та користувачів, щоб забезпечити доступність веб-контенту для всіх.

Чому доступ до інтернету важливий?

Люди з інвалідністю можуть мати труднощі з використанням комп'ютерів та інтернету. Їм можуть знадобитися спеціальні пристрої та браузері, що робить веб-сайти більш доступними для ширшого кола користувачів та пристроїв, включаючи мобільні.

Основні принципи веб-доступності

WAI визначила чотири основні принципи доступності веб-сайтів:

- Адаптивність: Веб-контент повинен відповідати потребам користувачів з різними сенсорними здібностями, такими як зір, слух і когнітивні здібності.
- Продуктивність: Користувачі повинні мати можливість взаємодіяти з веб-контентом за допомогою різних пристроїв введення, таких як

клавіатури, миші та голосового управління.

- Зрозумілість: Веб-контент повинен бути зрозумілим і простим у використанні для всіх користувачів, незалежно від їхнього рівня освіти або знань.
- Стабільність: Веб-контент повинен залишатися доступним для користувачів незалежно від використовуваної технології.

### Типи інвалідності

Існує багато типів інвалідності, які можуть вплинути на доступність веб-сайтів. Ось деякі з найпоширеніших:

- Слабкий зір або сліпота
- Глухота або порушення слуху
- Когнітивні порушення
- Фізичні вади
- Фоточутливість

### Доступні технології

Існує широкий спектр доступних технологій, які можуть допомогти людям з інвалідністю отримати доступ до Інтернету. Ці технології включають:

- Спеціальне програмне забезпечення для читання екрана
- Підвищена контрастність кольорів
- Поради щодо мови для елементів веб-сайту
- Відео субтитри

### Переваги веб-доступності

Забезпечення доступу до Інтернету має багато переваг, зокрема:

- Розширення аудиторії та охоплення більшої кількості потенційних клієнтів та користувачів
- Покращення сприйняття бренду та репутації

- Відповідність законним вимогам
- Створення більш інклюзивного та справедливого цифрового середовища

WAI - це важлива ініціатива, яка відіграє ключову роль у забезпечення доступності веб-вмісту для всіх. Застосовуючи основні принципи доступності веб-сайтів, розробники та дизайнери можуть зробити Інтернет більш інклюзивним і доступним для людей з інвалідністю.

### Веб-доступність W3C

Розробники переймаються багатьма речами, але універсальний дизайн все ще можна розглядати як додаткову річ - щось необов'язкове, коли все інше робиться. Хоча є виклики на зручному інтерфейсі з усіх сторін, це часто не зручно для всіх.

У світі є багато людей з певними психічними або фізичними розладами, які також використовують програмне забезпечення. Це величезна аудиторія, тому нерозумно забувати про це, як етично, так і ділово. Крім того, в деяких країнах це заборонено законодавством.

Про скільки людей насправді говорять? Наприклад, за даними ВООЗ, щонайменше 2,2 мільярди людей мають певні порушення зору або сліпі. 430 мільйонів людей у світі мають глухий або важкий слух. Очікується, що до 2050 року 2,5 мільярди людей матимуть певну ступінь втрати слуху. Якщо ви подивитесь на дані W3C, загалом 15-20% населення світу має певні фізичні або психічні порушення.

Для того, щоб сайти та програми були доступні для різних користувачів, у багатьох країнах існують обов'язкові стандарти та закони. Вони, як правило, засновані на технологічних стандартах та рекомендаціях, розроблених міжнародною організацією World Wide Web Consortium (W3C).

За ними веб-доступність (Веб-доступність) повинна враховувати всі аспекти, які впливають на використання Інтернету. Зокрема, порушенням є:

- Візуальна (це може бути кольорова сліпота, часткова або повна втрата зору);



- Слухання;
- Когнітивні (наприклад, порушення пам'яті, логіка, проблеми з концентрацією уваги тощо);
- Неврологічні та фізичні (як проблеми координації, моторики, дислексії);
- Мовлення.

Стандарт консорциуму називається WCAG 2.0, і він є найбільш всюдисущим, але не єдиним. Є й інші правила та правила, які відрізняються від країни до країни, але ми обговоримо їх трохи пізніше.

### Догляд за доступністю веб-сайту (WAI)

Ініціатива покращення веб-доступу (Ініціатива доступності Wit, WAI) — це спроба Всесвітнього веб-консорциуму (W3C) поліпшити доступність Всесвітньої мережі (WWW або Web) для людей з інвалідністю. Люди з інвалідами можуть стикатися з труднощами при використанні комп'ютерів та Інтернету. Їм потрібні нестандартні пристрої та браузері, які роблять веб-сайти більш доступними завдяки ширшому спектру агентів та пристроїв користувача, включно з мобільними пристроями.

W3C має декілька робочих груп, які розробляють пов'язані документи, технічні звіти, навчальні матеріали та інші документи, що стосуються різних аспектів доступності веб-сайтів. Ці компоненти включають веб-сторінки, веб-браузери та медіапрогравачі, інструменти розробки та інструменти оцінки.

Посібники доступності веб-контенту (WCAG) 2. 1 охоплюють широкий спектр рекомендацій щодо того, як зробити веб-вміст більш доступним. Дотримуючись цих інструкцій, можна зробити контент більш доступним для широкого кола людей з інвалідністю, зокрема адаптувати його до сліпих та слабких зором, тих, хто має порушення слуху, тих, хто має порушення мобільності, порушення мовлення та світлочутливість, та комбінації цих станів, і забезпечують адаптацію для людей з труднощами в навчанні та з когнітивними порушеннями. Однак, все ще неможливо задовольнити потреби всіх користувачів з такими порушеннями. Ці інструкції стосуються доступності веб-вмісту на настільних комп'ютерах, ноутбуках, планшетах та мобільних пристроях. Дотримуючись цих

рекомендацій, зазвичай, можна зробити веб-вміст більш дружнім до користувача.[2]

Критерії успіху WCAG 2.1 записуються як виписки, які не залежать від технологій і передаються на перевірку. Рекомендації щодо виконання критеріїв успіху в конкретних технологіях, а також загальна інформація про інтерпретацію критеріїв успіху наведено в окремих документах. Загальна інформація та посилання на технічні та навчальні матеріали WCAG можна знайти у Загальному огляді Руководящих принципів доступності веб-контенту (WCAG).

WCAG 2.1 є продовженням Руководств щодо доступності веб-контенту 2.0 [WCAG20], опублікованих у грудні 2008 року. у формі рекомендацій W3C. Вміст, який відповідає WCAG 2.1, також відповідає WCAG 2.0. РГ прагне переконатися, що Руководящие принципы WCAG 2.1 можуть слугувати альтернативним засобом перевірки відповідності правилам, які повинні відповідати WCAG 2.0. Публікація WCAG 2.1 не скасовує або не замінює WCAG 2.0. WCAG 2.0 залишається рекомендацією W3C, однак, в той же час W3C радить використовувати WCAG 2.1 для максимального застосування заходів доступності в майбутньому. W3C також рекомендується використовувати останню версію WCAG при розробці або оновленні політики доступу до веб-сайтів/вебдоступності.

### **1.3 Рекомендації Міністерства цифрової трансформації України до дизайну офіційних веб-сайтів**

Дизайн-код офіційного веб-сайту

1. Офіційний веб-сайт повинен відповідати таким вимогам:

1) Верхня частина офіційного веб-сайту:

розташований горизонтально у верхній частині кожного веб-сайту офіційного веб-сайту і повинен містити:

Елемент графічного інтерфейсу з написом «Дія», який містить посилання на Єдиний державний веб-портал електронних послуг. Завантажити логотип можна на <https://thedigital.gov.ua/logo>;

малий Державний Герб України у лівій частині верхньої частини сайту;  
найменування центрального органу виконавчої влади або найменування офіційного веб-сайту справа малий Державний Герб України у центрі верхньої частини веб-сайту;

у разі горизонтального розташування повинен містити горизонтальний зміст офіційного веб-сайту (елемент інтерфейсу), що містить не більше восьми позицій (гіперпосилання на розділи офіційного веб-сайту);

верхня частина офіційних веб-ресурсів, пов'язаних з діяльністю центральних органів виконавчої влади, може відрізнятися від цієї структури та порядку її розміщення;

2) всі веб-сайти офіційного веб-сайту повинні містити навігаційний ланцюжок компонент, який розміщений горизонтально під верхньою частиною веб-сайту і вказує на розміщення активного веб-сайту в структурі офіційного веб-сайту і містить гіперпосилання на веб-сторінки з верхніх рівнів ієрархії формування структури офіційного веб-сайту;

3) використовувати системи тегів для розповсюдження матеріалів за категоріями на офіційному веб-сайті з урахуванням обсягу повноважень відповідних органів виконавчої влади;

4) використовувати для офіційних сторінок веб-сайтів шрифт e-Ukraine. Ви можете завантажити шрифт і побачити приклади використання на <https://thedigital.gov.ua/fonts>;

5) використовувати кольорову палітру для елементів інтерфейсу відповідно до положень розділу IV цих Вимог;

6) Текст повинен мати співвідношення контрастності не менше 4,5:1 відносно тла.

## 2. Винятки можуть бути:

великий текст, такий як заголовки, може мати коефіцієнт контрастності 3:1;  
вторинний текст, неактивні компоненти інтерфейсу користувача або декоративний текст не вимагають коефіцієнта контрастності;

мінімальні вимоги щодо читання контрастності не застосовуються до логотипа.

### Офіційний сайт Елементи інтерфейсу

1. Типографія офіційного веб-сайту повинна відповідати таким вимогам:

1) використовувати заголовки рівнів від h1 до h7 при формуванні структури веб-сторінки офіційного веб-сайту, і кожна веб-сторінка повинна містити лише один заголовок рівня h1;

2) Використовуйте такі значення розміру цегли та інтервалу ліній: 38/40, 28/32, 24/28, 20/24, 18/28, 16/24. Фарба - звичайна;

3) Базовий текстовий шрифт становить 16 пікселів, інтервал між рядками 24 пікселів. Малюнок - світло;

4) додатковий (пояснювальний) текстовий шрифт - 13 пікселів, інтервал рядків - 18 пікселів. Малюнок - регулярний;

5) опорний шрифт - 12 пікселів, інтервал ліній - 16 пікселів. Креслення - регулярне;

{Підпункт 5 пункту 1 розділу III із змінами внесеними згідно з Наказом Міністерства цифрової трансформації № 67 від 21.07.2022}

6) вирівнювання основного тексту - по лівому краю;

7) не дозволяється використовувати одночасно підкреслення, курсив і великі літери;

8) Рекомендована максимальна ширина веб- сторінки становить 1680 пікселів. Якщо роздільна здатність пристрою більша, поля буде додано на веб-сторінку.

2. Розмір шрифту тексту без заголовків повинен змінюватися в межах 200 відсотків без використання допоміжних технологій та втрати інформаційного наповнення чи функціональності офіційного веб-сайту.

3. Елементи інтерфейсу розміщуються на сторінці офіційного сайту через серію рядків і стовпців за допомогою модульної сітки. Рекомендується використовувати 12- стовпчикову адаптивну сітку.

4. Сітка побудована з базових 8-піксельних модулів. Отже, розміри елементів і відступів повинні бути кратними 8:16, 24, 32, 40, 48.

### Кольори

1. Кольори офіційного веб-сайту повинні бути контрастними і використовуватися в поєднанні з формою і текстом. Колір не повинен бути єдиним візуальним засобом передачі інформації, що вказує на дію або розрізняє елемент серед інших.

Основними кольорами повинні бути:

кольори тексту: #000000, #808080, #8080 (прозорість 50%), #FFFFFF;

{Абзац перший пункту 1 розділу IV із змінами внесеними згідно з Наказом Міністерства цифрової трансформації No 67 від 21.07.2022}

Кольори основних елементів: #E7EEF3, #000000, #FFFFFF;

Абзац перший пункту 1 розділу IV із змінами внесеними згідно з Наказом Міністерства цифрової трансформації No 67 від 21.07.2022}

допоміжні кольори: #E6E6E6, #FFF7E3, #CCCCCC, #FFF4D7, #F1F1F1;

{Абзац п'ятий пункту 1 розділу IV із змінами внесеними згідно з Наказом Міністерства цифрової трансформації No 67 від 21.07.2022}

Кольори для системних станів: #FF3A44, #E6E6E6, #FFD600.

{Абзац шостий пункту 1 розділу IV із змінами внесеними згідно з Наказом Міністерства цифрової трансформації No 67 від 21.07.2022}

Подібні та яскраві кольори не допускаються.

2. При використанні світлих відтінків кольору непрозорість повинна бути 75 і/або 50 відсотків.

3. Веб-сторінки офіційного веб-сайту повинні бути адаптовані для перегляду без прив'язки до кольору.[3]

## 2 ДОСЛІДЖЕННЯ ІНСТРУМЕНТІВ ДЛЯ ПРОТОТИПІВ ІНТЕРФЕЙСІВ

### 2.1 Інструментів для прототипів інтерфейсів

Кожен дизайнер і розробник знає, що детальні брендмауери є основою будь-якого цифрового проекту. Незалежно від того, чи ви звикли малювати ескіз олівцем на папері чи створювати повноцінний динамічний прототип, вам буде корисно знати про інструменти, які можуть позбавити вас від непотрібних головних біль, безсонних ночей.

Кожен проект унікальний, має свою форму і розміри. Інструменти прототипу допоможуть вам вирішити будь-яке поставлене вами завдання. Незважаючи на багато інструментів і різноманітність, у цій статті я спробую мінімізувати кількість найкращих. Небагато їх зібрали до 20) Всі ці інструменти охоплюють цілий ряд різних методів прототипування, оптимізують час, дозволяють знайти слабкі сторони інтерфейсу і поліпшити логіку майбутнього сайту або мобільного додатку.

#### 1. Marvel

Marvel - один з нових гравців, але досить великий на ринку прототипних послуг. Зрозумілий і зручний інтерфейс Marvel допомагає перетворити всі ваші макети, ескізи, малюнки для інтернету або мобільного додатку в динамічний презентаційний матеріал для клієнта. Виходячи з досвіду, більшість клієнтів люблять дивитися на "живу" структуру сайту, ніж на малюнки на папері. За допомогою цього інструменту ви навіть можете створювати прототипи програм для Apple Watch – новий тренд.

#### 2. InVision

InVision готова допомогти вам у веб- або мобільному проекті. Це працює з будь-якою платформою, Android або iOS. За допомогою InVision ви можете створювати інтерактивні прототипи та компонування. InVision можна сміливо класифікувати як цілу команду. І це тому, що прототипи можна легко поділитися з членами вашої команди.

За допомогою InVision ви можете створювати інтерактивні прототи та макети.

### 3. Proto.io

Для використання Proto.io не потрібні спеціальні коди навичок. Навпаки, інтерфейс та логіка послуг прості, що робить його чудовим інструментом для створення інтерактивних прототипів для мобільних програм.

Для використання Proto.io не потрібні спеціальні коди навичок.

### 4. Moqups

Moqups - це ще один не менш відомий інструмент для прототипів. Але трохи інакше, ніж попередня. Платформа заснована на HTML5. Це ідеальний інструмент для створення незалежних від SVG мокапів та пожежних кадрів.

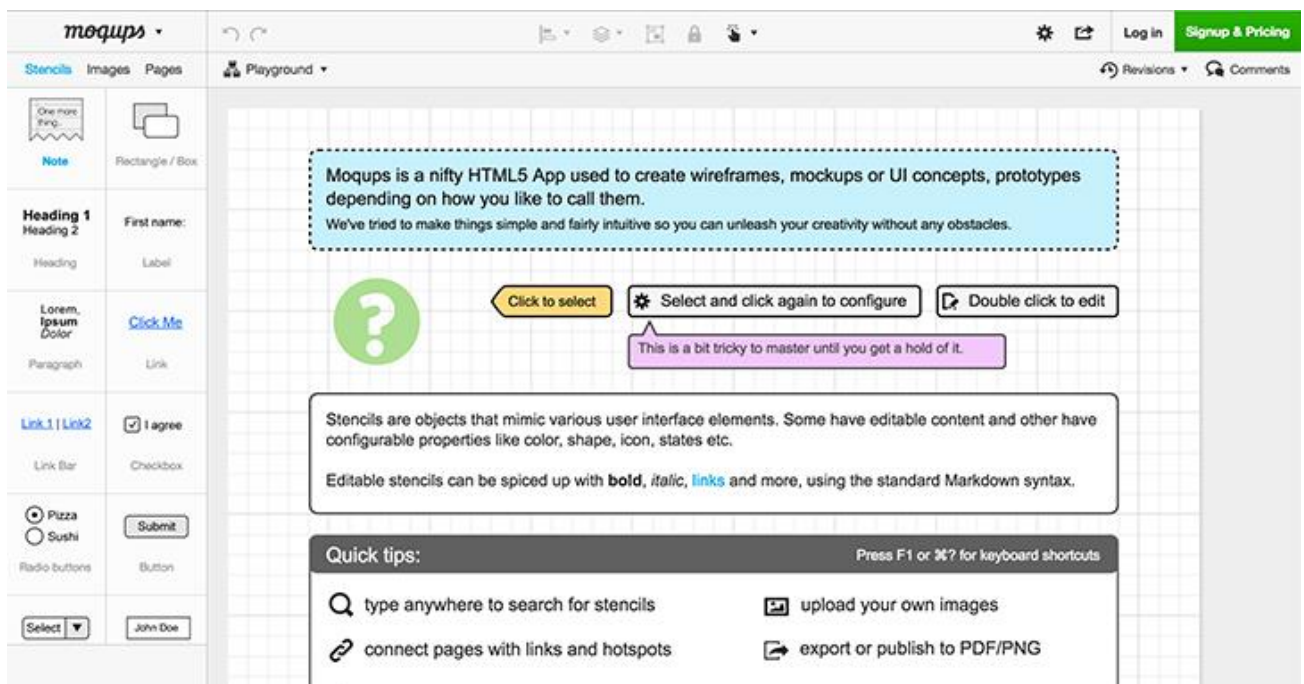


Рис. 1.2 Платформа заснована на HTML5

### 5. Fluid

Fluid - це додаток для мобільних прототипів. У середовищі, яке знає про це, існує багато думок про цей онлайн інструмент, і багато користувачів кажуть, що він дуже швидкий, зручний та інтуїтивно зрозумілий.

## 6. Prott

Prott - це метод швидкого прототипу з інструменту співпраці з місцем. Він візуалізує ваші ідеї від першого ескізу до інтерактивного прототипу, який працює як реальна програма. Крім того, Prott має чудовий дизайн і підходить для спільної роботи з іншими членами команди.

## 7. Флінто

Флінто швидко і легко створює прототипи, які виглядають акуратно і працюють належним чином. Флінто вражає свій ідеальний баланс між швидким прототипом та реалізмом.

## 8. POPs

Якщо ви надаєте перевагу старим шкільним методам, коли прототипи були намальовані вручну через відсутність сучасних інструментів, POPs будуть правильним вибором для вас. Все спрощено до мінімального значення: кидання ідей до вашої ескізної книги, фотографування і додавання посилань між кнопками і розкладками. Потім ви синхронізуєте прототипи з Dropbox і voila – готовий прототип) І все це лише з одним мобільним телефоном. За допомогою цієї програми ви можете створювати прототипи навіть стоячи в трафіці.

## 9. Pih

Pihate — це абсолютно новий гравець. Ця платформа для дизайну мобільної сумісності. Можливо розробники врахували досвід попередніх платформ і зробили його ще простішим все що вам потрібно це завантажити шари і додати взаємодію або анімацію кількома кліками мишки.

## 10. Axure

Axure - мій найперший інструмент, з яким ми пройшли крізь вогонь і воду. Дуже цікаво і зручно. Мені знадобилося не менше тижня щоб опанувати його основні можливості але час іде і раз ми повинні були попрощатися бо порівняно з часом з яким я познайомився на ринку навіть не пахло чимось простим і інноваційним Моя любов до Axure залишається як етикетка на робочому столі. комп'ютери) Можливо, таке ставлення пов'язане з тим, що можливості продукту не зовсім візуально красиві, але це лише моя думка, тому я буду тримати її при собі.



## 2.2 Функціональні характеристики Axure

Axure RP (Rapid Prototyping) є одним з найбільш популярних інструментів для UX/UI дизайнерів, який дозволяє створювати детальні, інтерактивні прототипи без необхідності програмування. Його можливості широко використовуються для планування, прототипування і спілкування дизайну великих і складних систем. Таке різноманіття функцій робить Axure незамінним інструментом в арсеналі сучасного дизайнера.[4]

### Детальний огляд функціональності

#### 1. Створення інтерактивних прототипів

Axure дозволяє створювати комплексні інтерактивні прототипи, які можуть включати переходи, скачки, зміни станів об'єктів та інші динамічні елементи. Це допомагає дизайнерам демонструвати і тестувати складні взаємодії та потоки користувачів, забезпечуючи більше контексту для розробників і зацікавлених сторін.

#### 2. Використання змінних і логіки

Змінні в Axure можуть бути використані для створення умовної логіки та моделювання справжньої поведінки системи. Наприклад, змінні можна використовувати для підтримки станів між сторінками, дозволяючи імітувати вхід, покупку, налаштування користувача тощо.

#### 3. Динамічний вміст

Динамічні панелі в Axure дозволяють створювати багат шарові інтерфейси, де один шар може бути відображений або прихований залежно від дій користувача. Цей параметр особливо корисний для створення вкладок, акордеонів, анімованого вмісту та інших інтерактивних елементів.

#### 4. Розширена бібліотека віджетів

Axure надає велику бібліотеку попередньо встановлених віджетів, таких як кнопки, списки, текстові поля і радіокнопки. Користувачі також можуть створювати власні бібліотеки віджетів, які забезпечують велику гнучкість і швидкість їх дизайну.

## 5. Командна робота та коментарі

Axure RP підтримує можливості командної роботи, що дозволяє декільком користувачам працювати над одним проектом одночасно. Це включає інтеграцію з Slack для спілкування в реальному часі та систему коментарів, яка дозволяє залишати нотатки та обговорення прямо у файлі проекту.

## 6. Експорт і документація

Axure дозволяє експортувати прототипи у формат HTML, який можна переглянути у будь-якому переглядачі Тенет без потреби у встановленні додаткового програмного забезпечення. Крім того, інструмент здатний генерувати специфікації, що описують сумісність, використання віджетів та логіку, що значно спрощує передачу проекту розробникам.

## 7. Інтеграція з іншими інструментами

Axure може імпортувати файли з інших популярних інструментів дизайну, таких як Ескіз і Adobe XD. Це дозволяє дизайнерам легко інтегрувати Axure у поточний процес роботи і використовувати його разом з іншими інструментами.

## 8. Адаптивний дизайн

Axure підтримує адаптивний дизайн, який дозволяє створювати прототипи версій для різних розмірів екрана та орієнтації. Це важливо для розробки веб-сайтів та програм, оптимізованих для мобільних пристроїв і планшетів.

## 9. Підтримка спільноти

Axure має активну міжнародну спільноту, де користувачі можуть знайти підтримку, обмінюватися знаннями та взаємодіяти з іншими професіоналами. Завдяки форумам, вебінарам, блогам та конференціям користувачі можуть ділитися досвідом та найкращими методами, забезпечуючи постійне навчання та розвиток.[5]

## Заклучення

Axure RP - це не просто інструмент для прототипів, а повноцінна платформа для розробки досвіду користувача, яка забезпечує глибоке розуміння та тестування дизайну перед його впровадженням. Його гнучкість, інтегровані можливості співпраці та великі функціональні можливості роблять його ідеальним вибором для

комплексного дизайну великих систем. Ці характеристики допомагають переконатися, що кінцевий продукт не тільки відповідає технічним вимогам, але й враховує потреби кінцевих користувачів.

### 2.3 Функціональні характеристики Figma

Розробляючи веб, дизайнери використовують різноманітні інструменти для покращення та оптимізації своєї роботи. Одним з найпоширеніших сьогодні є Figma, звичайний інструмент дизайну інтерфейсу, який надає широкий спектр функцій, щоб полегшити життя дизайнеру.

Створення стилів кольорів допоможе вам успішно і швидше виконувати завдання. Застосування єдиної кольорової системи допомагає об'єднати всі компоненти і зробити їх частиною єдиного цілого. Саме тому спільно з UI\UX дизайнером Владиславом Федоренком ми створили покрокову інструкцію для дизайнерів-початківців по створенню кольорової системи Figma. Крім того, ми описуємо способи побудови базових і розширених моделей кольорів і поділу їх на функціональні групи.

#### Мета створення кольорової системи у Figma

Систематизація і класифікація в будь-якому проекті допомагає оптимізувати процеси і налаштувати взаємодію. Саме тому дизайнери прагнуть систематизувати кольори, які використовуються в проекті, і зробити їх швидшою.

Однак, збереження кількох графічних палітр кольорів і подальше їх використання недостатньо для по-справжньому ефективної роботи. Кожен колір повинен мати свою функцію і виконувати її.

Ця схема полегшує життя дизайнеру наступним чином:

- допомагає орієнтуватися і відстежувати кольори;
- ділить композицію кольорів на групи і створює окрему сітку;
- сприяє підбору додаткових кольорів.

В результаті навігація між елементами інтерфейсу стає простішою, а згруповані палітри зручніші у використанні, особливо для створення нових

інтерфейсів.

Настав час перейти до практичних кроків, здійснивши які дизайнер отримає безкоштовну систему, яку можна перекосити у випадку масштабних проектів. Далі ми розглянемо чотири основні кроки.

### Крок 1: Вибір колірної моделі

Перше, що потрібно вибрати - колірну модель. Вибирати кольори для виробу можна тільки серед цих відтінків, хоча це зручніше робити з чіткою системою.

Кольорова модель — це спосіб передачі даних кольорів за допомогою цифр і літер через систему координат кольорів.

У цій похіді ми використовуємо модель HSL разом з Hex, RGB, CSS, HSB.

Модель HSL (відтінок, насиченість, легкість) використовується як альтернатива моделі RGB. Він був розроблений з урахуванням особливостей візуального сприйняття, щоб полегшити читання та розпізнавання кольорів.

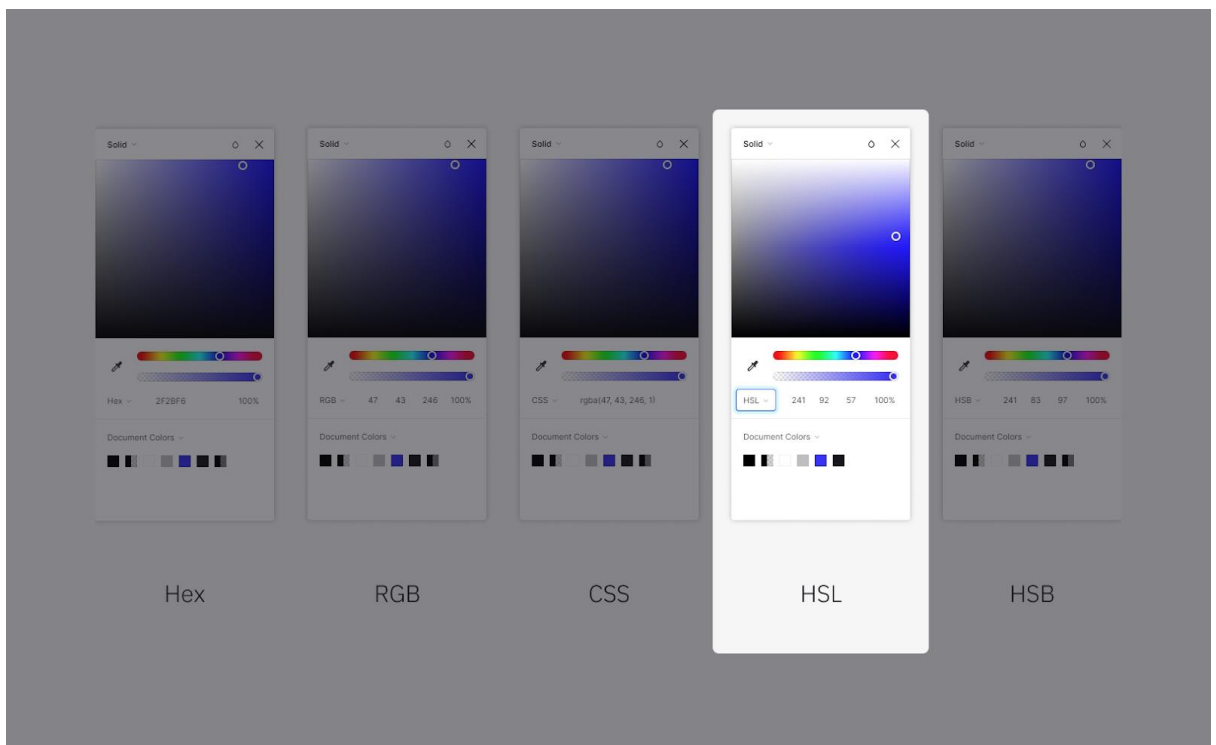


Рис. 2.1 Вибір кольору: Hex, RGB, CSS, HSL, HSB.

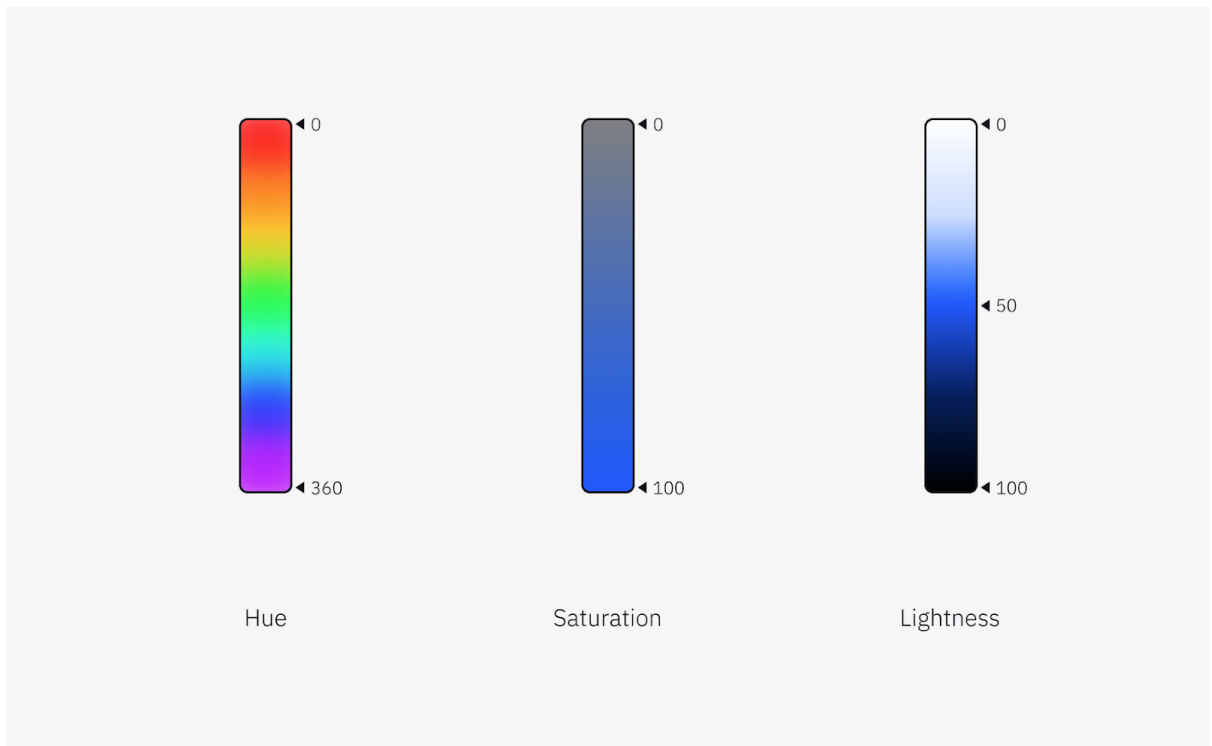


Рис. 2.2 Hue, Saturation, Lightness (HSL) кольорові шкали.

Як ви бачите на зображенні, відтінок є градацією простих (первинних) кольорів. Це те, що ми розуміємо під назвою кожного кольору: червоний, синій, жовтий та інші.

Насиченість - це визначення глибини кольору.

Світло - це рівень темного і світлого кольору.

Застосування моделі HSL має декілька переваг:

1. Шістнадцятковий код не дає нам жодної інформації про знаходження кольорів у системі координат. Використовуючи HSL, ми можемо контролювати кожен параметр за допомогою чисел.

Зауваження: шістнадцятковий код все ще зручніший для передачі інформації про кольори у текстовому форматі, оскільки його легше скопіювати і вставити одне число, а не три, як у інших моделях.

1. У порівнянні з моделлю RGB, HSL більш інтуїтивно зрозумілий і схожий на наше сприйняття кольорів.[6]

2. HSL дозволяє легко побудувати монохромні кольорові схеми.

Вибір певної моделі залежить від цілей і вподобань дизайну. У прикладах ми

продемонструємо кольори за допомогою моделей Hex і HSL.

## Крок 2: Побудувати основні групи кольорів

Кожен окремий товар має свої визначені кольори, а саме: фірмові кольори, умовні кольори. І це те, що потрібно для побудови первинної системи. Нижче наведено один з варіантів створення системи - чорно-білий, основний колір, кольори стану.

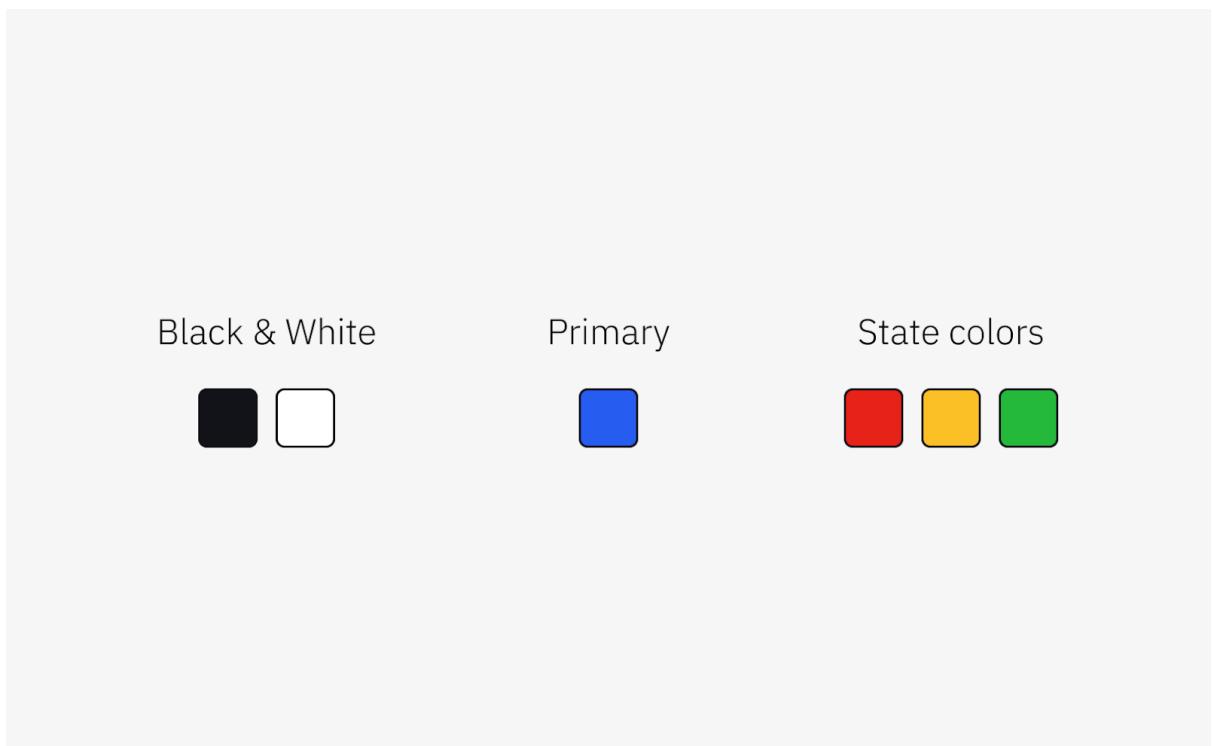


Рис. 2.3 Побудова базових груп кольорів

Обираючи чорно-білі кольори для системи кольорів, ми вважаємо за краще дотримуватися таких критеріїв:

- Не обирати такий простий чорний як #000000. Застосуйте дуже темний знебарвлений відтінок синього або червоного. Це надасть особливого відтінку фінальному продукту. Зважаючи на специфіку чорного кольору (його особливість поглинати світло), користувач не побачить відмінностей з класичним чорним.[7]

Однак, це не варто сприймати як вимогу. Все ж це рекомендація, яку можна використовувати, залежно від обставин та доцільності.

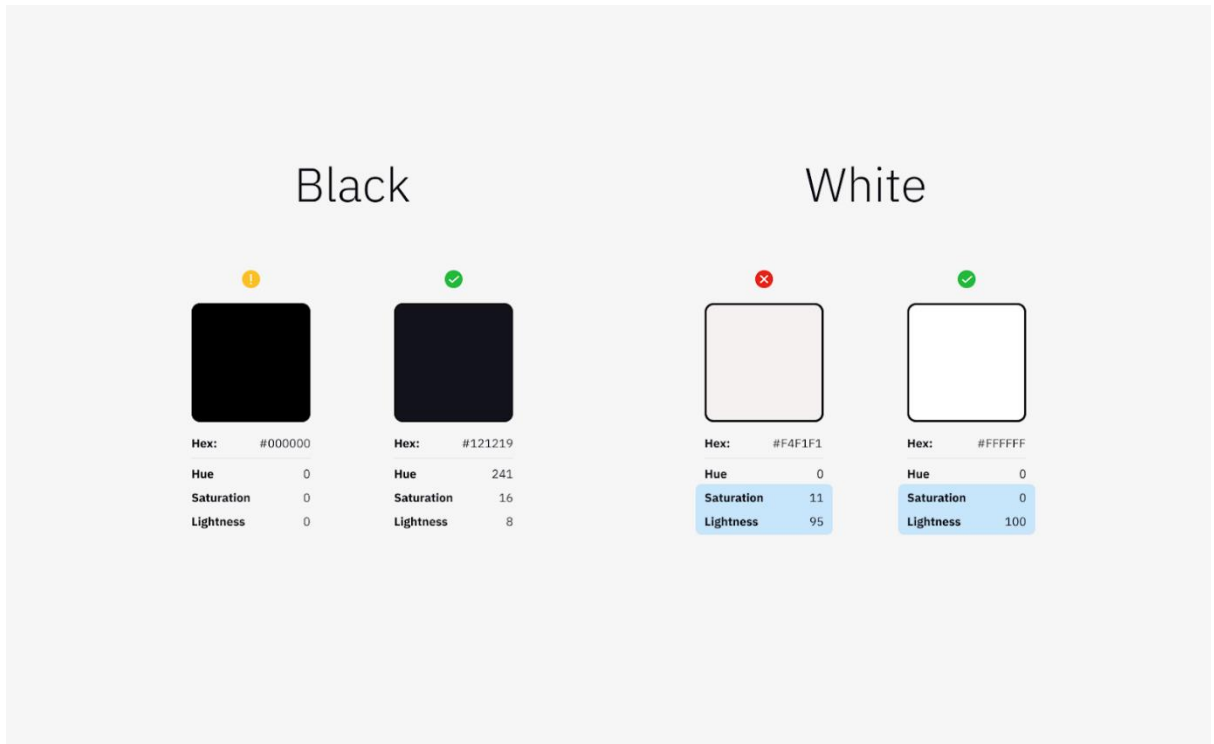


Рис. 2.4 Різниця кольорів

- Це зовсім інше з білим. Класичний, який варто застосовувати без винятку #FFFFFF. Як найкраще відхилення шоу в сидінні і яскраво відрізняється від очікуваного Колір стає брудним і впливає на інші кольори і може.

#### Основний колір

Виберіть колір з палітри бренду. Він використовується для акцентів і активних елементів інтерфейсу.

Продукт може мати кілька кольорів, які йдуть до другорядного, але вони мають один або два з цих кольорів.



Рис. 2.5 Основні кольори

У деяких ситуаціях навіть краще використовувати кольори логотипу компанії, щоб слідувати базовому настрою. Цей колір використовується в елементах ТВТ (виклик до дії), тому він повинен торкатися і нагадувати бренд на всіх етапах цифрового продукту клієнта.[8]

### Кольори держави

Це функціональні кольори, які інформують користувача про помилки, попередження або успішне завершення дії.

Традиційно, зелений використовується для позначення успішної операції, жовтий для попередження, а червоний для помилки. З цими кольорами ми вже встановили зв'язки та асоціації. Якщо ви оберете червоний колір для позначення успішної операції, деякі користувачі можуть неправильно інтерпретувати цей сигнал і думати, що щось пішло не так.





Рис. 2.6 Кольори станів

Це стосується і двох інших кольорів держави. [9] За певних обставин, ролі цих кольорів можуть бути зміни; однак продукти, розроблені для широкого спектру користувачів, краще стають більш традиційними.

Створюйте додаткові відтінки кольорів[10]

Ймовірно, що базових кольорів буде недостатньо для створення цілого продукту і врахування всіх можливих випадків у процесі створення прототипу інтерфейсу. Наступний крок - розширити спектр кольорів за допомогою монохромних варіацій.

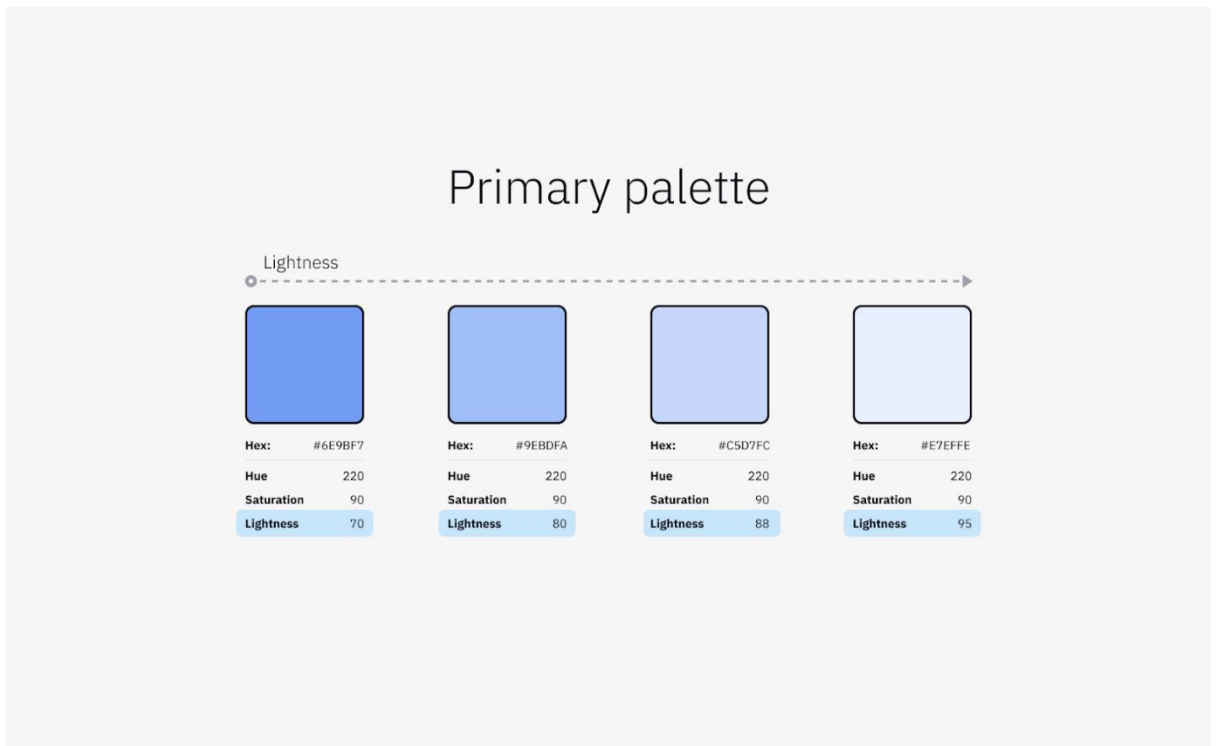


Рис. 2.7 Зміни відповідно до налаштувань світлих і темних відтінків

Наведений вище приклад демонструє, як легко створювати нові кольори, використовуючи наявні кольори в схемі. Усе, що вам потрібно зробити, - це змінити базовий колір на світліший. Навіть незначні зміни кількох параметрів сприймаються як зовсім інші кольори.[11]

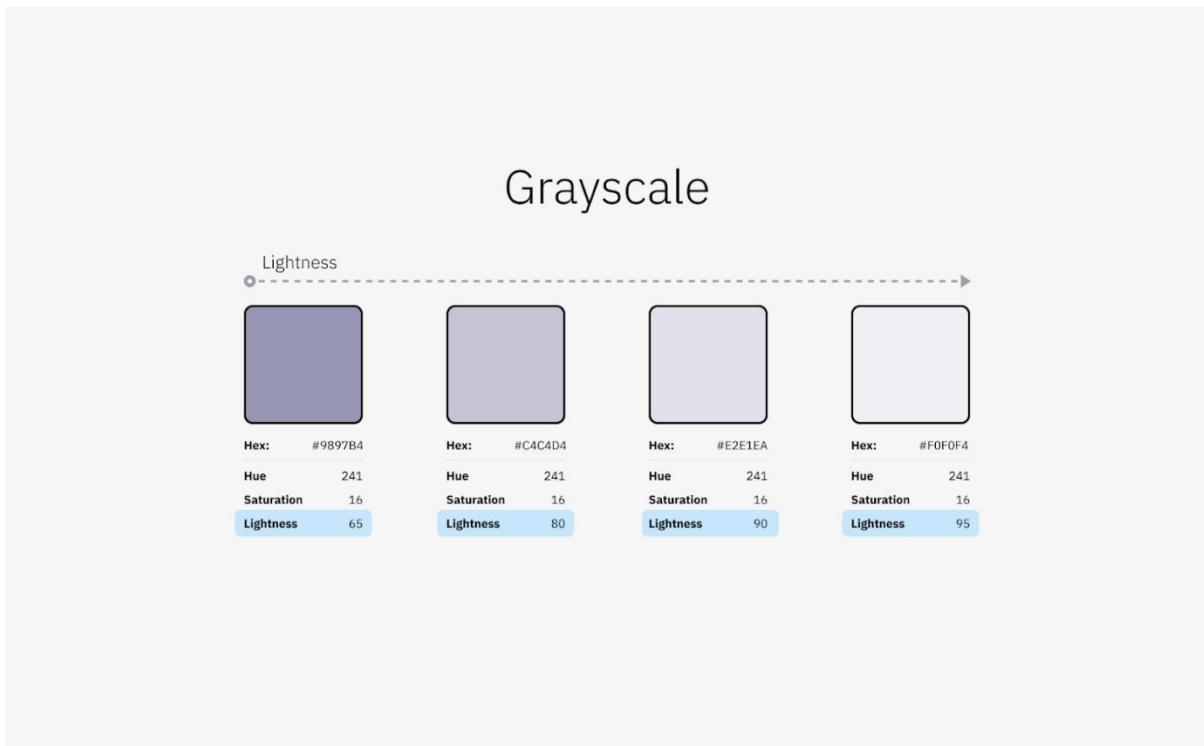


Рис. 2.8 Зміни відповідно до налаштувань світлих і темних відтінків

Те ж саме відбувається з чорним кольором - в результаті ми отримуємо всі варіації відтінків сірого.

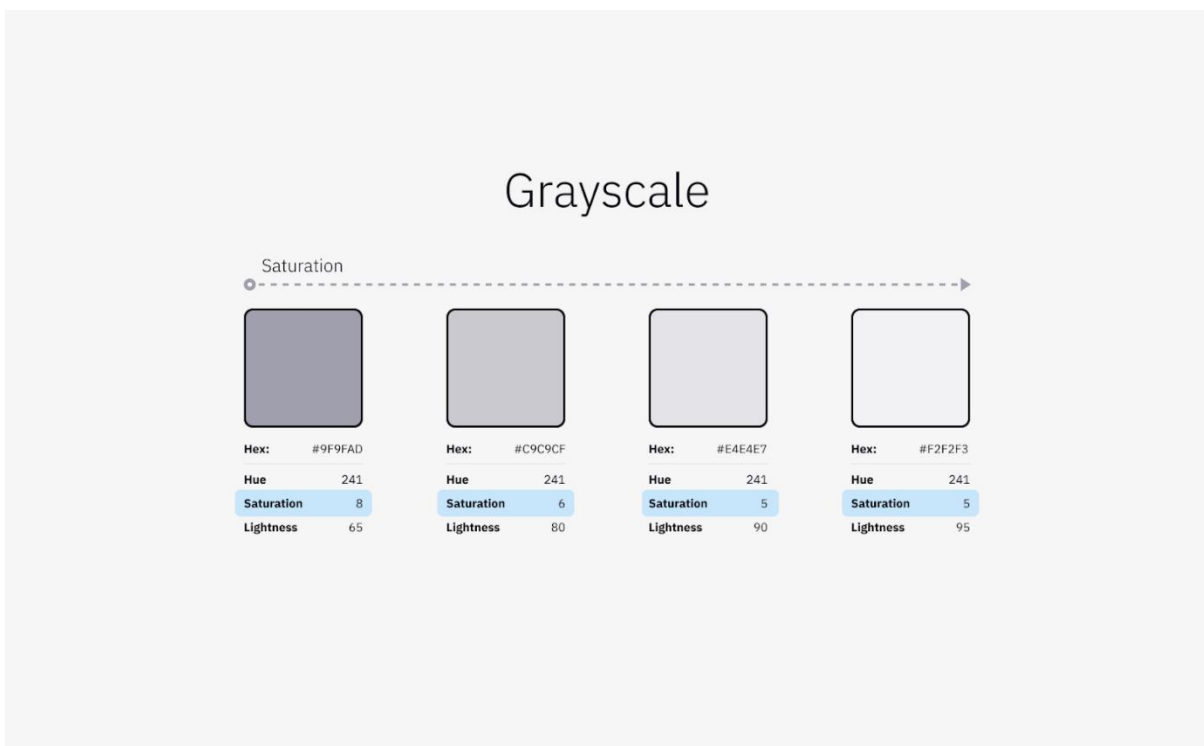


Рис. 2.9 Варіація сірих відтінків

Примітка: якщо застосувати попередню техніку до не дуже правильного чорного, але відтінку будь-якого іншого кольору, то її світліший відтінок буде занадто насиченим для сірого. У цьому випадку ми можемо зменшити насиченість, щоб отримати більш нейтральний сірий колір.

Зміни відповідно до параметрів відтінку[12]

На цьому етапі, здебільшого, ми вже маємо систему кольорів, побудовану для нашого проекту. Однак, завжди можна зробити графічну палітру більш живою, змінюючи тон кольорів і співвідношення яскравості кольорів.

## **2.4. Функціональні характеристики Adobe XD.**

Adobe XD значно спрощує процес створення макетів, візуального дизайну, взаємодій, прототипування, тестування та обміну, надаючи всі необхідні інструменти для проектування інтерфейсів в одному рішенні. Adobe XD розроблявся як всебічний інструмент для дизайну інтерфейсів, у співпраці з професійними UX-дизайнерами. Дизайнери зосередилися не лише на графічній оболонці програми, але й на оптимізації та покращенні взаємодії користувачів з продуктом. Понад 5000 дизайнерів зробили свій внесок у розробку Adobe XD в рамках передрелізної програми. Adobe XD доступний на платформах macOS та Windows 10, що дозволяє дизайнерам працювати без необхідності освоювати нову операційну систему. Програма для веб-дизайну має ряд унікальних функцій, які роблять її особливою та потужнішою, ніж Sketch або InVision Studio.[13]

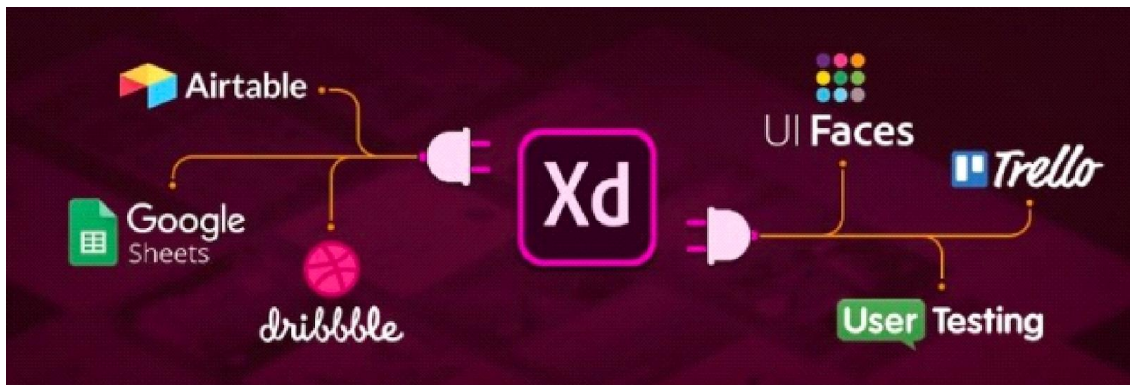


Рис. 2.10 Плагіни

В Adobe XD плагіни доступні як частина нових функцій, і багато з них, що були розроблені для Sketch, вже перенесені в XD. Ці плагіни додають нові можливості до Adobe XD. Стратегія Adobe полягає в тому, щоб дозволити великій спільноті розробників створювати власні плагіни, використовуючи популярні та добре відомі мови програмування, такі як JavaScript, HTML і CSS.[13,14]

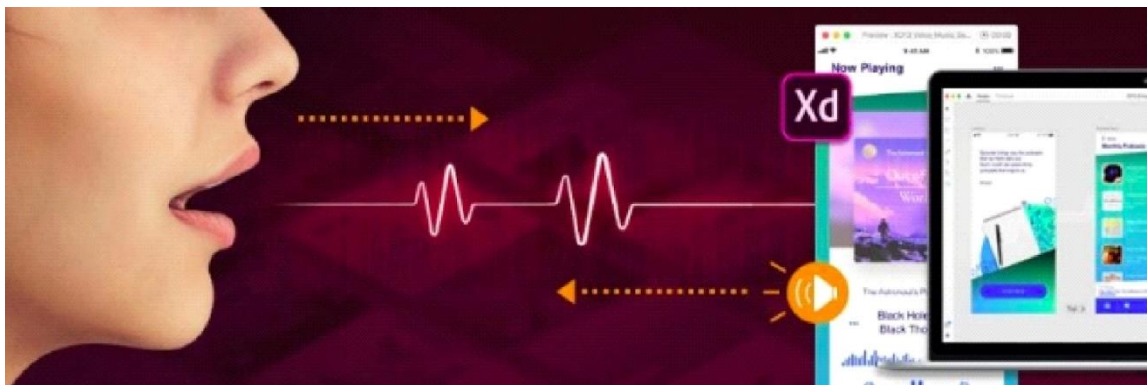


Рис. 2.11 Голосові тригери й мова

Adobe XD також надає можливість користувачам взаємодіяти з прототипами на новому рівні. Голосові тригери та функція відтворення мови дозволяють користувачам виходити за рамки традиційного екранного інтерфейсу, забезпечуючи можливість створення прототипів за допомогою голосових команд. Це унікальна функція, яка відрізняє Adobe XD від інших інструментів для прототипування.

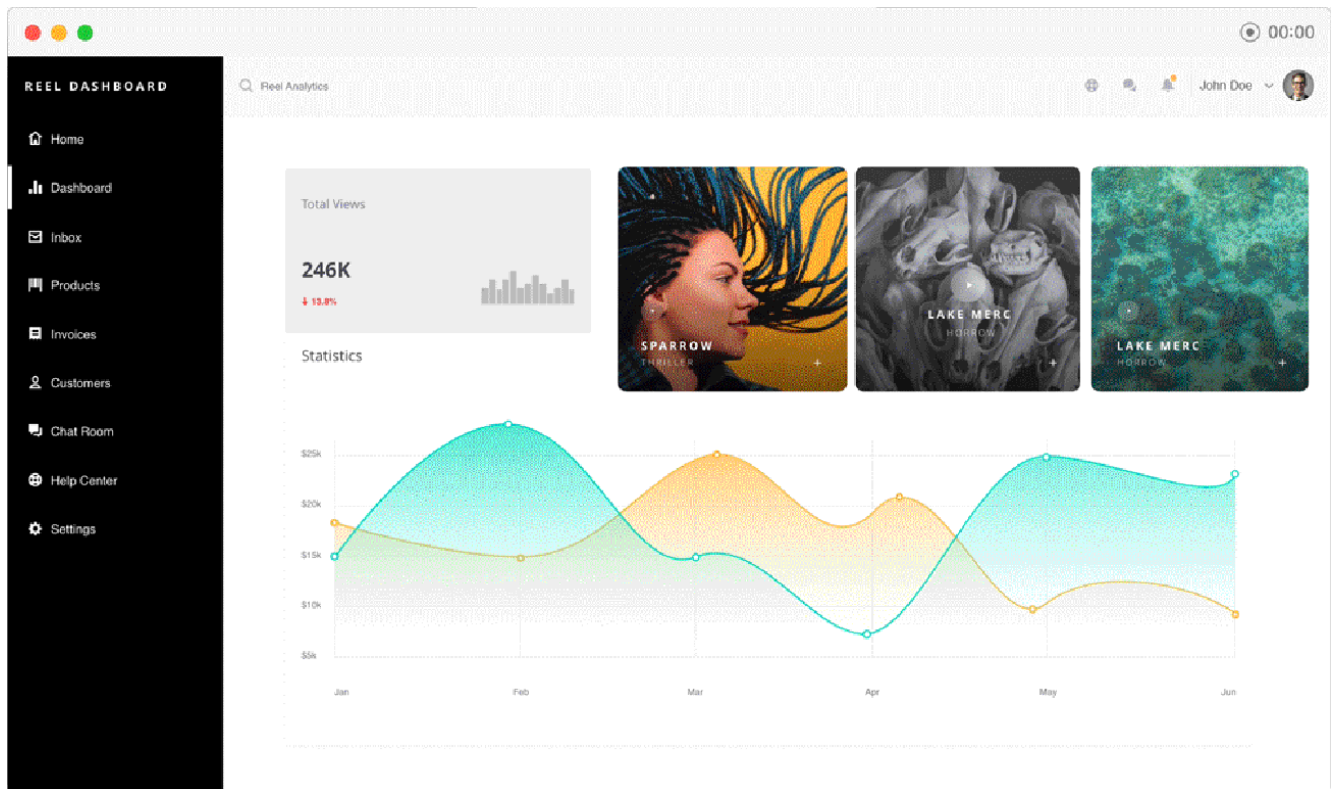


Рис. 2.12 Auto-animate

Adobe XD перевершує InVision Studio завдяки цій функції. Хоча Adobe XD поки що не має такої точної можливості управління, як у InVision Studio, висока швидкість роботи та простота налаштування Adobe XD є тими аспектами, над якими InVision Studio потрібно старанно працювати, щоб залишатися конкурентоспроможним.[15]

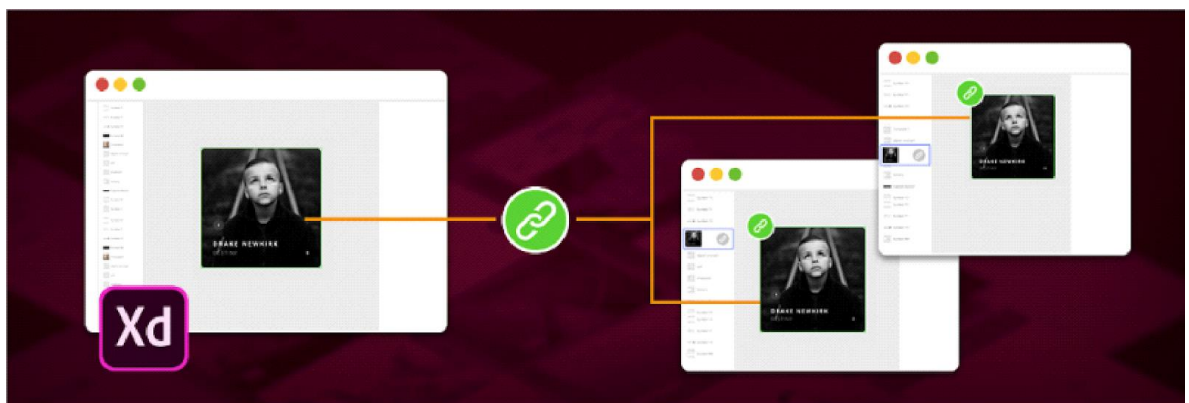


Рис. 2.13 Пов'язані символи

Adobe XD розширив можливості роботи зі символами. Тепер значно простіше відстежувати зміни, оскільки символи, скопійовані з одного документа і вставлені в прототипи, залишаються пов'язаними. Якщо в оригінальному файлі вносяться зміни, ви зможете застосувати ці оновлення і в іншому файлі.[16] Хоча створення дизайн-систем у Sketch завжди було дуже ефективним, завдяки пов'язаним символам в XD Adobe відкрила нові горизонти в управлінні дизайн-системами.

Прототипування жестів перетягування стало важливим додатком в арсеналі XD, що стало значним оновленням для користувачів XD.

Також в XD є кілька інших функцій, які варто відзначити:

1. Користувачі Adobe XD можуть відкривати файли Photoshop і Illustrator безпосередньо в XD - дизайнери, які використовують ці програми, тепер можуть продовжувати роботу над своїми проектами в XD.

2. Файли Sketch також можна відкривати в XD - тепер у користувачів Sketch немає причин не спробувати XD.

3. Можливість відкриття файлів XD в After Effects - складні анімації можна створювати в After Effects, експортувавши проект з XD в AE, що дозволяє точніше контролювати анімації (якщо можливостей Adobe XD недостатньо).

4. Передача розробникам - функція «share for development» дозволяє розробникам переглядати файли дизайну через веб-посилання і завантажувати ресурси, позначені для експорту, навіть без відкриття або наявності копії Adobe XD. Розробники також можуть залишати коментарі та питання.

5. Коментарі - за допомогою функції «Share for review» можна прикріплювати коментарі до проектів, що прискорює співпрацю із зацікавленими сторонами.

6. Інтеграція додатків - Adobe XD тепер інтегрується з багатьма популярними програмами для спільної роботи та підвищення продуктивності, включаючи Slack, JIRA та Microsoft Teams. У ході розвитку клієнт-серверної технології змінювалися, як і змінювалися методи їх реалізації. Далі розглянутьсся способи організації доступу до даних та особливості в контексті клієнт-серверної технології.[18,17]

Використання Adobe XD дозволяє об'єднати прототипування, проектування та анімацію в одному місці, що значно спрощує тестування та ітерації дизайну. Процес проектування стає легшим, оскільки взаємодії та анімації відбуваються в контексті один одного, перетворюючи їх на невід'ємну частину процесу проектування, а не окремі елементи. Це забезпечує високу ефективність і дозволяє здійснювати проектування швидше. Можливість переходу від варфрейму до високоточних прототипів і передачі розробникам в одній програмі демонструє, наскільки Adobe приділяє увагу оптимізації робочих процесів. Робочі процеси стають простішими та зручнішими, [19] вирішуючи основні проблеми розробників, що підкреслює переваги екосистеми Adobe.

Adobe також має значну перевагу в індустрії дизайну завдяки пакету додатків Creative Cloud, який використовує більшість дизайнерів. Багато з них мають ліцензію на Creative Cloud. Adobe XD також доступний безкоштовно, хоча безкоштовна версія обмежує кількість проектів для коментування та обміну. Низький бар'єр для входу робить Adobe XD доступним для широкого кола користувачів.

У майбутніх планах Adobe згадується про створення та управління системами дизайну за допомогою хмарних документів та Creative Cloud Libraries. Це надає всі необхідні інструменти для створення та управління надійними системами дизайну в команді або організації.

Можливості Adobe вражають, і майбутнє Adobe XD виглядає дуже перспективно. Цей інструмент вже завоював серця багатьох користувачів, і наші проекти здебільшого створюються саме в ньому.

Як зазначено в таблиці, яскравість кольорів впливає на їх відтінки. Наприклад, жовтий буде яскравішим за зелений, а зелений - яскравішим за темно-синій. Кольори з високою яскравістю виглядатимуть краще у світлих відтінках, а з низькою - у темних.[20] Якщо розбити наш основний колір за цим методом, отримаємо наступну схему:



Зараз наша схема майже готова для використання, і настав час створювати стилі для зручності її використання. Однак одна з можливих проблем - це присвоєння правильних назв кожному з кольорів системи. Вони повинні бути зручними навіть для дизайнерів, які не знайомі з проектом.

Перша частина назви міститиме назву групи кольорів. Для відтінків всередині групи можна використовувати числове значення яскравості. Якщо розділити частини назви слешем (наприклад, Primary/50), Figma об'єднає ці кольори в групи.[21] Можна також створювати підгрупи всередині груп, використовуючи додатковий слеш.

Таким чином, всі кольори отримують смислові групи з чіткою навігацією серед них. Вони зручні у використанні, особливо коли назви кольорів відповідають їх способам застосування. Навіть дизайнери-початківці, застосовуючи наш гайд, зможуть створити власну систему кольорів для будь-якого цифрового продукту, що значно пришвидшить виконання завдань. Figma надає зручні інструменти, головне – навчитися правильно ними користуватися.

### Від зручності для дизайнерів до зручності для користувачів

Ми описали модель HSL та її основні відмінності від інших моделей, способи створення базових груп кольорів, додаткових відтінків і способи їх структурування. Ці 4 кроки спрощують життя не лише дизайнерам, але й користувачам продукту, адже взаємопов'язані кольори роблять інтерфейс більш гармонійним та зрозумілим.

## **3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ПРОТОТИПУВАННЯ ІНТЕРФЕЙСІВ З VR/AR**

### **3.1 Технічні аспекти реалізації інтерфейсів з підтримкою VR/AR**

У цьому розділі моєї дипломної роботи я зосереджуюся на практичній реалізації програмного забезпечення для прототипування інтерфейсів, які використовують технології віртуальної (VR) та доповненої реальності (AR). [23] Основна мета полягає у створенні зразкового програмного додатка, який дозволить розробникам тестувати і взаємодіяти з інтерфейсами у віртуальному середовищі. Програмний продукт реалізовано на платформі Unity, оскільки вона надає широкі можливості для роботи з VR/AR контентом, включаючи вбудовану підтримку багатьох VR обладнань і стандартних протоколів взаємодії.

Технічні аспекти реалізації інтерфейсів з підтримкою VR та AR мають велике значення для створення іммерсивних та ефективних користувацьких досвідів. Перш за все, важливо зосередитися на розробці 3D моделей та сцен, що вимагають використання спеціалізованих інструментів, таких як Unity або Unreal Engine. [24] Ці платформи забезпечують потужні можливості для створення реалістичних та високодеталізованих віртуальних середовищ (Milgram & Kishino, 1994). [22] Важливу роль відіграє також інтеграція сенсорних датчиків, що дозволяє користувачам взаємодіяти з віртуальним світом природним чином. Сенсори руху, такі як Oculus Touch або HTC Vive Controllers, використовуються для точного відстеження рухів рук, що є критичним для іммерсивної взаємодії (Sherman & Craig, 2003).

Крім того, оптимізація продуктивності є ключовою, оскільки високий рівень FPS (кадрів за секунду) є необхідним для зниження відчуття запаморочення та нудоти в середовищі VR. Використання потужних графічних процесорів та оптимізація рендеринга допомагають підтримувати плавність і реалістичність віртуального середовища (Jacobson & Höllerer, 2008). Для AR важливою є інтеграція з реальним світом, що забезпечується за допомогою камер та алгоритмів

комп'ютерного зору для коректного відображення цифрових об'єктів в реальному часі на основі навколишнього середовища (Azuma, 1997).

Загалом, реалізація інтерфейсів VR та AR вимагає глибокого розуміння як апаратних, так і програмних аспектів, що включають все від архітектури додатків до забезпечення користувацького досвіду, що відповідає високим вимогам сучасних технологій віртуальної та доповненої реальності.

### **3.2 Логічна структура програмного забезпечення для прототипування інтерфейсів з VR/AR**

У розділі, присвяченому логічній структурі програмного забезпечення для прототипування інтерфейсів з VR/AR, важливо розглянути компоненти та їх взаємодії, що забезпечують ефективне створення і використання віртуальних та доповнених реальностей. Сучасні системи прототипування вимагають глибокого інтегрування апаратних та програмних компонентів, що дозволяють розробникам створювати інтерактивні, занурювальні середовища.[25]

Основу логічної структури становить ядро системи, яке обробляє дані вхідних пристроїв — таких як VR-шоломи, трекери руху, і контролери — та координує роботу всіх інших модулів. Ядро включає в себе менеджер сцен, який відповідає за управління об'єктами у віртуальному просторі, та менеджер ресурсів, який оптимізує використання пам'яті та обчислювальних ресурсів.

Модуль рендерингу є критично важливим, оскільки він забезпечує візуалізацію віртуального світу, що має бути високоякісним і реалістичним. Рендеринг має підтримувати різні технології відображення, включаючи стереоскопічні дисплеї та проєкційні системи.

Інтерфейсний модуль включає інструменти для взаємодії користувача з VR/AR середовищем, такі як віртуальні меню, інструменти для маніпуляції об'єктами, та індивідуально адаптовані контрольні панелі. Цей модуль має тісно інтегруватися з сенсорними технологіями, що дозволяють користувачам ефективно взаємодіяти з цифровим контентом.

Важливо також враховувати модуль безпеки, який захищає дані користувачів і взаємодію в середовищах VR/AR від несанкціонованого доступу та забезпечує конфіденційність.[26]

Для забезпечення гнучкості розробки та сумісності з різними платформами, система має підтримувати модульність, що дозволяє легко додавати, видаляти чи замінювати окремі компоненти в залежності від потреб проекту. Ці принципи відображені в науковій літературі та технічних документаціях, які слугують основою для розробки сучасних систем VR/AR, наприклад, як описано у Unity Documentation та Unreal Engine Documentation.

### 3.3 Код програми

Повний код для гри "Star Hunter - VR Archery App"

#### 1. Створення сцени в Unity

1. Відкрийте Unity і створіть новий проект.
2. Імпортуйте XR Interaction Toolkit та необхідний плагін для вашої VR гарнітури (наприклад, **\*\*Oculus XR Plugin\*\***).
3. Додайте XR Rig до сцени (GameObject -> XR -> XR Rig).
4. Додайте 3D-моделі лука та стріли.

#### 2. Скрипт для взаємодії зі стрілами (BowAndArrow.cs)

```
csharp
using UnityEngine;
using UnityEngine.XR.Interaction.Toolkit;

public class BowAndArrow : MonoBehaviour
```

```

{
    public XRController leftController;
    public XRController rightController;
    public Transform arrowPrefab;
    public Transform bowTransform;

    private Transform currentArrow;
    private bool isArrowAttached = false;

    void Update()
    {
        if
(leftController.inputDevice.TryGetFeatureValue(CommonUsages.triggerButton, out
bool leftTriggerValue) && leftTriggerValue)
        {
            AttachArrow();
        }

        if
(rightController.inputDevice.TryGetFeatureValue(CommonUsages.triggerButton, out
bool rightTriggerValue) && rightTriggerValue && isArrowAttached)
        {
            ShootArrow();
        }
    }

    private void AttachArrow()
    {
        if (!isArrowAttached)
        {

```

```

        currentArrow = Instantiate(arrowPrefab, bowTransform.position,
bowTransform.rotation);
        currentArrow.SetParent(bowTransform);
        isArrowAttached = true;
    }
}

private void ShootArrow()
{
    currentArrow.SetParent(null);
    Rigidbody rb = currentArrow.GetComponent<Rigidbody>();
    rb.isKinematic = false;
    rb.velocity = bowTransform.forward * 20f;
    isArrowAttached = false;
}
}

```

### #### 3. Скрипт для телепортації (TeleportationController.cs)

```

csharp
using UnityEngine;
using UnityEngine.XR.Interaction.Toolkit;

public class TeleportationController : MonoBehaviour
{
    public XRController teleportationController;
    public GameObject reticle;
    public LayerMask teleportationLayer;
    public float maxTeleportationDistance = 10f;
}

```

```

private bool isTeleporting = false;

void Update()
{
    if
    (teleportationController.inputDevice.TryGetFeatureValue(CommonUsages.primaryButton, out bool primaryButtonValue) && primaryButtonValue)
    {
        if (!isTeleporting)
        {
            StartTeleport();
        }
    }
    else
    {
        if (isTeleporting)
        {
            EndTeleport();
        }
    }
}

private void StartTeleport()
{
    RaycastHit hit;
    if (Physics.Raycast(teleportationController.transform.position,
    teleportationController.transform.forward, out hit, maxTeleportationDistance,
    teleportationLayer))
    {

```

```

        reticle.SetActive(true);
        reticle.transform.position = hit.point;
        isTeleporting = true;
    }
}

private void EndTeleport()
{
    if (isTeleporting)
    {
        transform.position = reticle.transform.position;
        reticle.SetActive(false);
        isTeleporting = false;
    }
}
}

```

#### #### 4. Скрипт для взаємодії з об'єктами (XRController.cs)

```

csharp
using UnityEngine;
using UnityEngine.XR;

public class XRController : MonoBehaviour
{
    public XRNode controllerNode;
    private InputDevice controller;

    void Start()

```



```

    {
        controller = InputDevices.GetDeviceAtXRNode(controllerNode);
    }

    public bool GetTriggerValue()
    {
        controller.TryGetFeatureValue(CommonUsages.triggerButton, out bool
triggerValue);
        return triggerValue;
    }

    public bool GetPrimaryButtonValue()
    {
        controller.TryGetFeatureValue(CommonUsages.primaryButton, out bool
primaryButtonValue);
        return primaryButtonValue;
    }
}

```

#### ##### 5. Скрипт для стріли (Arrow.cs)

```

```csharp
using UnityEngine;

public class Arrow : MonoBehaviour
{
    private Rigidbody rb;

    void Start()

```

```

{
    rb = GetComponent<Rigidbody>();
    rb.isKinematic = true;
}

```

Diana ☐, [5/31/2024 8:09 PM]

```

void Update()
{
    if (!rb.isKinematic)
    {
        // Додайте логіку для руху стріли, якщо потрібно
    }
}
}

```

## 6. СКРИПТ ДЛЯ РЕТИКУЛУ (Reticle.cs)

```

```csharp
using UnityEngine;

public class Reticle : MonoBehaviour
{
    public void SetActive(bool active)
    {
        gameObject.SetActive(active);
    }

    public void SetPosition(Vector3 position)
    {

```

```
    transform.position = position;  
  }  
}
```

## Налаштування сцени в Unity

### 1. XR Rig:

- Додайте XR Rig до сцени.
- Налаштуйте позиції та ротації контролерів.

### 2. Об'єкти лука та стріли:

- Додайте 3D-моделі лука та стріли до сцени.
- Додайте колайдери та Rigidbody до стріли.

### 3. Телепортація:

- Створіть об'єкт для ретикулу (Reticle) та налаштуйте його як активний/неактивний.

## Запуск проекту

1. Підключіть VR гарнітуру.
2. Запустіть проект у режимі Play.
3. Використовуйте VR-контролери для взаємодії з об'єктами, стрільби з лука та телепортації.

Цей повний код забезпечує базову функціональність для прототипування інтерфейсів з VR/AR, дозволяючи користувачам взаємодіяти з об'єктами у віртуальному середовищі. Ви можете розширити цей приклад, додаючи нові функції та вдосконалюючи взаємодію користувача.

## ВИСНОВКИ

Аналіз існуючих технологій та інструментів для створення інтерфейсів у VR/AR

На початку роботи був проведений детальний аналіз існуючих технологій та інструментів для створення інтерфейсів у VR/AR. Розглядалися найпопулярніші платформи та середовища, такі як Unity, Unreal Engine, а також спеціалізовані інструменти для прототипування та дизайну інтерфейсів, як-от Adobe XD та Figma. Було виявлено, що кожна з цих платформ має свої унікальні переваги та недоліки. Наприклад, Unity забезпечує широкі можливості для створення інтерактивних 3D середовищ, однак його використання потребує глибоких знань у програмуванні. Adobe XD, навпаки, пропонує інтуїтивно зрозумілий інтерфейс для швидкого створення прототипів, але обмежується в можливостях створення складних анімацій.

Дослідження методів прототипування інтерфейсів для VR/AR

Далі в роботі було проведено дослідження методів прототипування інтерфейсів для VR/AR. Зокрема, було розглянуто важливість інтерактивності та реалістичності в створенні користувацького досвіду у віртуальній та доповненій реальностях. Було визначено, що одним із ключових аспектів успішного прототипування є можливість швидкого тестування та внесення змін. Використання спеціалізованих інструментів для прототипування, таких як Marvel чи Axure, дозволяє значно прискорити цей процес, оскільки вони надають можливості для створення інтерактивних прототипів з мінімальними затратами часу.

Розробка моделі базової версії програмного забезпечення

На основі проведеного аналізу та досліджень була розроблена модель базової версії програмного забезпечення для прототипування інтерфейсів з VR/AR. Модель включає в себе основні компоненти, такі як модуль рендерингу, модуль фізики та

модуль взаємодії з користувачем. Було розроблено логічну структуру програмного забезпечення, яка дозволяє легко розширювати функціональність за рахунок додавання нових модулів та компонентів. Розробка була виконана на основі Unity, що дозволило забезпечити високу продуктивність та стабільність роботи програмного забезпечення.

### Рекомендації для прототипування інтерфейсів з VR/AR

На основі проведених досліджень та розробки програмного забезпечення були розроблені рекомендації для прототипування інтерфейсів з VR/AR. Основні рекомендації включають:

- використання спеціалізованих інструментів для швидкого створення інтерактивних прототипів;
- застосування методів тестування та ітераційного покращення дизайну для забезпечення високої якості користувацького досвіду;
- використання новітніх технологій, таких як голосові команди та жести, для розширення можливостей взаємодії користувачів з віртуальними середовищами;
- забезпечення сумісності програмного забезпечення з різними платформами та пристроями для підвищення його універсальності.

### Значення розробленого програмного забезпечення

Розроблене програмне забезпечення для прототипування інтерфейсів з VR/AR має велике значення для подальшого розвитку цієї галузі. Воно дозволяє спростити та прискорити процес створення інтерактивних прототипів, що є важливим етапом у розробці VR/AR застосунків. Використання цього програмного забезпечення сприятиме підвищенню якості кінцевих продуктів та забезпеченню кращого користувацького досвіду.

## Перспективи розвитку та вдосконалення

Подальший розвиток та вдосконалення розробленого програмного забезпечення полягає у розширенні його функціональності та інтеграції з іншими інструментами та платформами. Одним із напрямків розвитку може бути додавання підтримки нових типів взаємодії, таких як тактильний зворотний зв'язок, що дозволить зробити користувацький досвід ще більш реалістичним та занурюючим. Також важливим напрямком є подальша оптимізація продуктивності та стабільності роботи програмного забезпечення, що дозволить забезпечити його використання на ширшому спектрі пристроїв та платформ.

## Висновки

Загалом, проведена робота підтвердила актуальність теми розробки програмного забезпечення для прототипування інтерфейсів з VR/AR. Розроблене програмне забезпечення показало свою ефективність та можливості для подальшого розвитку в цій галузі. Впровадження рекомендацій та подальше вдосконалення програмного забезпечення сприятиме підвищенню якості VR/AR застосунків та покращенню користувацького досвіду.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Аналоги вимог до інтерфейсів <https://www.maxzosim.com/interface-analysis/>
2. Initiative (WAI) <https://reporter.zp.ua/wai-wnq.html>
3. Рекомендації Міністерства цифрової трансформації України до дизайну офіційних веб-сайтів (веб-порталів) (Наказ від 23 червня 2022 року № 57) <https://zakon.rada.gov.ua/laws/show/z0785-22#Text>
4. Carnegie Mellon University's XR/TC Will Drive Research into VR, AR Innovations (www.cs.cmu.edu)
5. IEEE VR 2023 Conference, <https://ieeevr.org/2023/>
6. Future Internet: VR, AR, and 3-D User Interfaces for Measurement and Control, [https://www.mdpi.com/journal/futureinternet/special\\_issues/VR\\_AR\\_3DUIs](https://www.mdpi.com/journal/futureinternet/special_issues/VR_AR_3DUIs)
7. Toptal: New Realities: VR, AR, MR and the Future of Design, <https://www.toptal.com/designers/ui/new-realities-vr-ar-mr>
8. BioMedical Engineering OnLine: Virtual and augmented reality in biomedical engineering, <https://biomedical-engineering-online.biomedcentral.com/articles/10.1186/s12938-020-00788-2>
9. Springer: Inclusive AR/VR: Accessibility Barriers for Immersive Technologies, <https://link.springer.com/article/10.1007/s10209-021-00789-6>
10. Springer: AR/VR for Remote Collaboration, <https://link.springer.com/article/10.1007/s00542-021-06259-0>
11. MDPI: Virtual Prototyping in Product Design, [https://www.mdpi.com/journal/sensors/special\\_issues/VR\\_Product\\_Design](https://www.mdpi.com/journal/sensors/special_issues/VR_Product_Design)
24. С. Лівіцький, М. Еверінгем. Автоматичне розпізнавання слів, написаних відбитками пальців британською мовою жестів [Електронний ресурс]. 2009. - Режим доступу: <https://ieeexplore.ieee.org/document/5204291/> .
12. MIT Reality Hack 2023: The World's VR/AR Hackathon, <https://mit-reality-hack-2023.devpost.com/>
13. ScienceDirect: Analyzing Augmented Reality (AR) and Virtual Reality (VR) Recent Developments, <https://www.sciencedirect.com/science/article/pii/S2666389921000205>

14. MDPI: Advances in Tangible and Embodied Interaction, [https://www.mdpi.com/journal/electronics/special\\_issues/advances\\_tangible\\_embodied\\_interaction](https://www.mdpi.com/journal/electronics/special_issues/advances_tangible_embodied_interaction)
15. IEEE Xplore: Prospects and Modern Technologies in the Development of VR/AR, <https://ieeexplore.ieee.org/document/9402431>
16. GUVI: Integrating Augmented Reality (AR) into User Interfaces: Challenges and Opportunities, <https://www.guvi.com/blogs/integrating-augmented-reality-ar-into-user-interfaces>
17. MDPI: Augmented Reality for Urban Planning, [https://www.mdpi.com/journal/sustainability/special\\_issues/Augmented\\_Reality\\_Urban\\_Planning](https://www.mdpi.com/journal/sustainability/special_issues/Augmented_Reality_Urban_Planning)
18. IEEE VR 2023: Modified Egocentric Viewpoint for Softer Seated Experience in Virtual Reality, <https://ieeivr.org/2023/>
19. Springer: The Metaverse and Its Potential for Immersive Experiences, <https://link.springer.com/article/10.1007/s00542-021-06259-0>
20. MDPI: Applications of VR in Education, [https://www.mdpi.com/journal/sensors/special\\_issues/VR\\_Education](https://www.mdpi.com/journal/sensors/special_issues/VR_Education)
21. Springer: Augmented Reality for Industrial Applications, <https://link.springer.com/article/10.1007/s00170-021-06489-3>
22. IEEE Xplore: AI-Powered Metaverse: Analysis, Challenges and Opportunities, <https://ieeexplore.ieee.org/document/9505132>
23. MDPI: VR in Mental Health Treatment, [https://www.mdpi.com/journal/ijerph/special\\_issues/VR\\_Mental\\_Health](https://www.mdpi.com/journal/ijerph/special_issues/VR_Mental_Health)
24. Springer: Mixed Reality in Retail and E-commerce, <https://link.springer.com/article/10.1007/s00542-021-06259-0>
25. ScienceDirect: VR/AR for Training and Simulation, <https://www.sciencedirect.com/science/article/pii/S2666389921000205>
26. IEEE Xplore: Advances in VR Interaction Techniques, <https://ieeexplore.ieee.org/document/9402431>



## ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



### Розробка програмного забезпечення для прототипування інтерфейсів з VR/AR

Виконала студентка 4 курсу  
групи ПД-41  
Сторчак Діана Олександрівна  
Керівник роботи

Д.т.н. професор кафедри ІПЗАС Бондарчук Андрій Петрович  
Київ – 2024

### МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

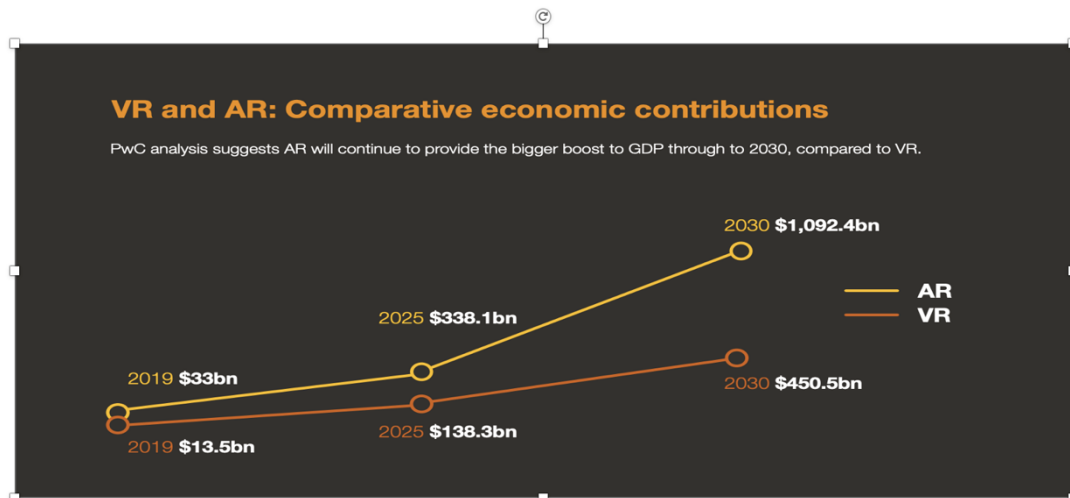
- **Мета роботи** підтримка процесів прототипування інтерфейсів з VR/AR в ігрових застосунках.
- **Об'єкт дослідження** прототипування інтерфейсів з VR/AR для ігрових застосунків.
- **Предмет дослідження** програмне забезпечення для прототипування інтерфейсів в ігрових застосунках.

## ЗАДАЧІ ДИПЛОМНОЇ РОБОТИ

1. Аналіз існуючих технологій та інструментів для створення інтерфейсів у VR/AR.
2. Дослідження методів прототипування інтерфейсів для VR/AR.
3. Розробка моделі базової версії програмного забезпечення.
4. Тестування розробленого програмного забезпечення на різних платформах.
5. Розробка рекомендацій для прототипування інтерфейсів з VR/AR з урахуванням впровадження розробленого програмного забезпечення.

3

### АКТУАЛЬНІСТЬ ТЕМИ



Порівняльний економічний внесок віртуальної реальності (VR) та доповненої реальності (AR) у глобальний ВВП до 2030 року. Графік демонструє прогнозоване зростання впливу AR та VR на економіку з 2019 по 2030 рік, підкреслюючи, що AR матиме більший економічний вплив порівняно з VR.

4

## АНАЛІЗ АНАЛОГІВ

Показник	Unity	Unreal Engine	Figma	Axure	DiProject
Рендеринг	+	+	-	-	+
Врахування фізики об'єктів	+	+	-	-	+
Анімація	+	+	-	-	+
Прототипування	+	+	+	+	+
Інтерактивність VR/AR	+	+	-	-	+
Реалістичність	+	+	-	-	+
Легкість інтеграції	+	-	+	-	+
Кросплатформеність	+	+	+	-	+
Готова бібліотека для виконання ігрових механік	-	-	-	-	+

5

## ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### Функціональні вимоги:

1. Підтримка VR/AR платформ.
2. Можливість захоплення об'єктів.
3. Телепортація користувача.
4. Реалістична фізика стріли.

### Нефункціональні вимоги:

1. Висока продуктивність та стабільність.
2. Підтримка платформи Windows 10 і вище.
3. Вимоги до обладнання: підтримка VR-гарнітур Oculus Rift та Oculus Quest.
4. Інтерактивність і реалістичність.
5. Легкість інтеграції з існуючими системами.

6

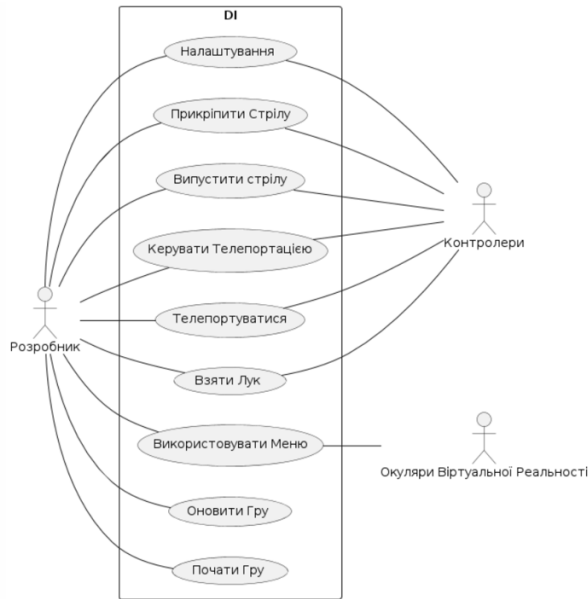
# ПРОГРАМНІ ТА АПАРАТНІ ЗАСОБИ РЕАЛІЗАЦІЇ



Oculus quest contollers

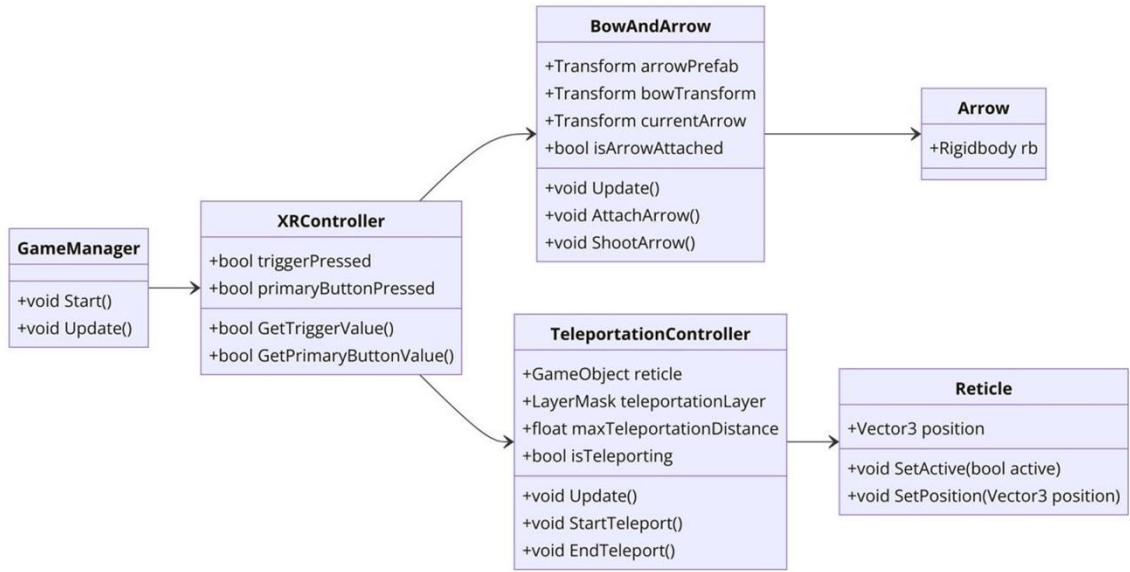
7

## Діаграма варіантів використання

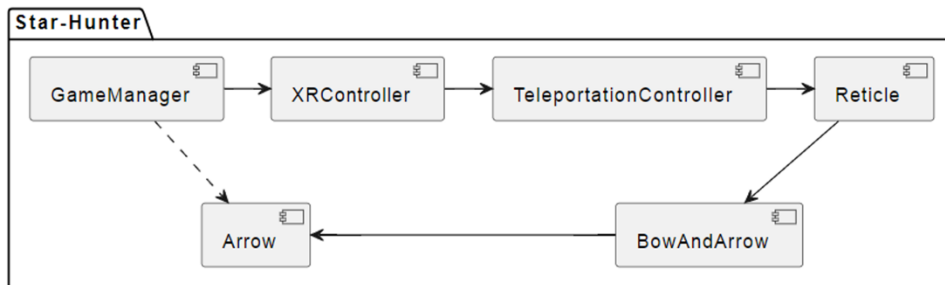


8

## Діаграма пакетів



## Діаграма класів

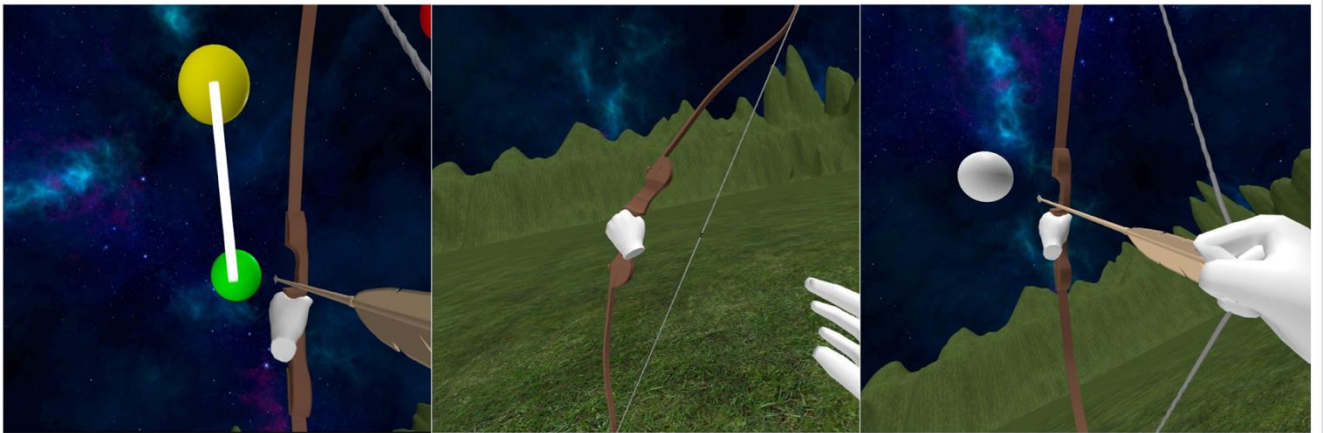


## Діаграма діяльності



11

## ЕКРАННІ ФОРМИ



12

## АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

1. Сторчак Д.О. Розробка програмного забезпечення для прототипування інтерфейсів з VR/AR. Матеріали IV Всеукраїнська науково-практичної конференція «Сучасні інтелектуальні інформаційні технології в науці та освіті». Збірник тез, ДУІКТ, м. Київ - К.: ДУІКТ, 2024. - С. 49.
2. Сторчак Д.О. Інтеграція постачання сонячної енергії на «розумному» розподільному щиті на основі системи IoT / А.В. Заячковський, В.М. Заячковський, К.П. Сторчак // журнал "Зв'язок" №5, 2022, с. 54-57.

13

## ВИСНОВКИ

1. На основі проведеного аналізу існуючих технологій та інструментів для створення інтерфейсів у VR/AR, було виявлено, що Unity та Unreal Engine є найефективнішими платформами.
2. Досліджені методи швидкого прототипування, такі як використання XR Interaction Toolkit та Oculus Integration, дозволили створити базову версію програмного забезпечення з інтерактивними елементами для захоплення та переміщення об'єктів у VR середовищі.
3. Проведено тестування програми на платформах Oculus Rift та Oculus Quest.
4. Розроблені рекомендації включають ефективне використання інструментів прототипування, оптимізацію взаємодії користувача та забезпечення сумісності з різними платформами.

14

## ДОДАТОК Б. ЛІСТИНГИ ПРОГРАМНИХ МОДУЛІВ

### 1. MainScene.unity

```

```csharp
using UnityEngine;
using System.Collections;

public class MainScene : MonoBehaviour
{
    public GameObject arrowPrefab;
    public Transform arrowSpawn;
    public float fireRate = 0.5f;
    private float nextFire = 0.0f;

    void Update()
    {
        if (Input.GetButton("Fire1") && Time.time > nextFire)
        {
            nextFire = Time.time + fireRate;
            Fire();
        }
    }

    void Fire()
    {
        var arrow = (GameObject)Instantiate(arrowPrefab, arrowSpawn.position, arrowSpawn.rotation);
        arrow.GetComponent<Rigidbody>().velocity = arrow.transform.forward * 6;
        Destroy(arrow, 2.0f);
    }
}
```

```

### #### 2. BowController.cs

```

```csharp
using UnityEngine;
using System.Collections;

public class BowController : MonoBehaviour
{
    public Transform bow;
    public Transform stringTransform;
    public Transform arrowRest;
    public GameObject arrowPrefab;

    private GameObject currentArrow;

    void Update()
    {
        if (Input.GetButtonDown("Fire1"))
        {
            PullString();
        }
        else if (Input.GetButtonUp("Fire1"))

```



```

    {
        ReleaseString();
    }
}

void PullString()
{
    currentArrow = Instantiate(arrowPrefab, arrowRest.position, arrowRest.rotation);
    currentArrow.transform.parent = arrowRest;
}

void ReleaseString()
{
    currentArrow.transform.parent = null;
    currentArrow.GetComponent<Rigidbody>().velocity = arrowRest.forward * 30;
    currentArrow = null;
}
}
...

```

#### #### 3. Arrow.cs

```

```csharp
using UnityEngine;
using System.Collections;

public class Arrow : MonoBehaviour
{
    void OnCollisionEnter(Collision collision)
    {
        if (collision.gameObject.tag == "Target")
        {
            Destroy(collision.gameObject);
        }
        Destroy(gameObject);
    }
}
...

```

#### #### 4. Target.cs

```

```csharp
using UnityEngine;
using System.Collections;

public class Target : MonoBehaviour
{
    public int health = 3;

    void OnCollisionEnter(Collision collision)
    {
        if (collision.gameObject.tag == "Arrow")
        {
            health--;
            if (health <= 0)
            {

```

```
        Destroy(gameObject);
    }
}
}
...

```

#### 5. GameController.cs

```
```csharp
using UnityEngine;
using System.Collections;

public class GameController : MonoBehaviour
{
    public GameObject targetPrefab;
    public Transform[] spawnPoints;

    void Start()
    {
        SpawnTargets();
    }

    void SpawnTargets()
    {
        foreach (Transform spawnPoint in spawnPoints)
        {
            Instantiate(targetPrefab, spawnPoint.position, spawnPoint.rotation);
        }
    }
}

```