

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ І. І. МЕЧНИКОВА

(повне найменування закладу вищої освіти)

Факультет математики, фізики та інформаційних технологій

(повне найменування факультету)

Кафедра інформаційних технологій

(повна назва кафедри)

Кваліфікаційна робота

на здобуття ступеня вищої освіти «Бакалавр»

**«Розробка ігрового застосунку з реалізацією механік
маніпуляції часом»**

(тема кваліфікаційної роботи українською мовою)

**«Development of a game application with the
implementation of time manipulation mechanics»**

(тема кваліфікаційної роботи англійською мовою)

Виконав: здобувач заочної форми навчання
спеціальності 122 Комп'ютерні науки

(код, назва спеціальності)

Освітня програма Комп'ютерні науки

(назва)

ШАМРАЙ Максим Данилович

(прізвище, ім'я, по-батькові здобувача)

Керівник старш. викл. Вохменцева Т.Б.

(науковий ступінь, вчене звання, прізвище, ініціали)

(підпис)

Рецензент Департамент інформації та цифрових рішень Одеської
міської ради, головний спеціаліст відділу впровадження
інформаційних технологій, Клепатська В.В.

(науковий ступінь, вчене звання, прізвище, ініціали)

Рекомендовано до захисту:
Протокол засідання кафедри
Інформаційних технологій

№ від 2025 р.

Завідувачка кафедри

Захищено на засіданні ЕК №
протокол № від 2025 р.

Оцінка / /
(за національною шкалою/шкалою ECTS/ бали)

Голова ЕК

КАЗАКОВА Надія

(підпис)

(прізвище, ім'я)

КОПИЧЕНКО Іван

(підпис)

(прізвище, ім'я)

Одеса 2025

ЗМІСТ

	Стор.
ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ	6
ВСТУП	7
1 ТЕОРЕТИЧНІ ЗАСАДИ РЕАЛІЗАЦІЇ МЕХАНІК ПАРАЛЕЛЬНИХ СВІТІВ У ВІДЕОІГРАХ	9
1.1 Аналіз категорій маніпуляцій паралельних світів	9
1.2 Математичні моделі та алгоритми реалізації систем паралельних світів	15
1.3 Проблематика збереження та відтворення стану ігрового світу	18
2 ОГЛЯД І АНАЛІЗ ІСНУЮЧИХ ЗАСТОСУНКІВ З АНАЛОГІЧНОЮ ІДЕЄЮ	21
2.1 Огляд ігор для мобільних та десктопних платформ із механіками паралельних світів	21
2.2 Порівняння підходів до реалізації механік переміщення між світами	23
2.3 Визначення недоліків і переваг існуючих рішень	25
3 ОБҐРУНТУВАННЯ АРХІТЕКТУРИ ТА ВИБОРУ ПРОГРАМНИХ ЗАСОБІВ	27
3.1 Вибір цільової платформи та рушія	27
3.2 Проектування архітектури ігрового застосунку	29
3.3 Обґрунтування підходів до UI/UX з урахуванням механік паралельних світів	32
4 РЕАЛІЗАЦІЯ ІГРОВОГО ПРОТОТИПУ ТА ФУНКЦІОНАЛЬНИХ ЕЛЕМЕНТІВ	34
4.1 Реалізація основної логіки перемикання між світами та ігрового середовища	34
4.2 Тестування продуктивності і стабільності роботи	38
4.3 Оптимізація під ресурсні обмеження цільових платформ	40
5 ЕКСПЕРИМЕНТАЛЬНА ПЕРЕВІРКА ТА АНАЛІЗ ЕФЕКТИВНОСТІ	43

	5
5.1 Юзабіліті-тестування з реальними користувачами	43
5.2 Аналіз зворотного зв'язку користувачів та сприйняття основних механік	45
5.3 Пропозиції щодо вдосконалення та адаптації під інші жанри	47
ВИСНОВКИ	49
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	50
ДОДАТОК А	52

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ

Об'єктний пулінг – це патерн програмування, який використовується для оптимізації роботи з ресурсозатратними об'єктами. Основна ідея полягає в тому, щоб уникнути постійного створення та знищення об'єктів, замість цього повторно використовувати їх зі спеціального "пулу".

Процедурна генерація – це метод створення контенту в програмуванні, особливо в іграх, за допомогою алгоритмів, а не вручну. Він дозволяє автоматично генерувати великі обсяги унікальних даних (наприклад, рівні, ландшафти, текстури, сюжетні лінії) на основі математичних правил і випадкових чисел.

Юзабіліті-тестування – це метод оцінки зручності використання продукту (сайту, програми, гри, пристрою) на реальних користувачах. Мета — виявити проблеми в інтерфейсі, які ускладнюють взаємодію, і покращити загальний досвід.

Low Performance – це стан системи, програми або пристрою, коли вона працює повільно, з затримками або використовує занадто багато ресурсів.

ПК – персональний комп'ютер

AAA – Triple-A

CPU – central processing unit

FPS – frames per second

LOD – levels of details

NPC – non-player character

RAM – random access memory

UE – Unreal Engine

UI – User Interface

UX – User Experience

VR – Virtual Reality

ВСТУП

Сучасна індустрія відеоігор активно розвивається, демонструючи постійний пошук нових ідей та підходів до організації ігрового процесу. Одним із перспективних напрямів у геймдизайні є використання концепції паралельних світів, що відкриває нові можливості для створення оригінального геймплею, наративу та взаємодії з ігровим середовищем. Така механіка дає змогу гравцеві взаємодіяти з кількома вимірами одночасно, змінюючи умови проходження гри й формуючи унікальний користувацький досвід.

Актуальність обраної теми зумовлена необхідністю пошуку нових способів залучення аудиторії через впровадження інноваційних механік. Ідеї паралельних світів активно досліджуються як у науковому, так і в прикладному контексті, адже вони поєднують технічну складність реалізації з високим потенціалом для творчих рішень.

Метою кваліфікаційної роботи є розробка ігрового застосунку з реалізацією механіки перемикавання між паралельними світами. Основна увага приділяється створенню цілісної ігрової системи, яка поєднує ідеї геймдизайну, технічну реалізацію та користувацьку зручність.

Для досягнення поставленої мети у роботі вирішуються такі завдання:

- аналіз особливостей ігрових механік, пов'язаних із паралельними світами;
- дослідження прикладів реалізації таких рішень у сучасних відеоіграх;
- вибір технологічної бази для реалізації прототипу;
- створення та тестування функціонального ігрового застосунку.

Об'єктом дослідження є інноваційні ігрові механіки, предметом – процес реалізації ігрової логіки з використанням концепції паралельних світів. У роботі використано теоретичні та практичні методи, зокрема аналіз джерел, проєктування, програмну реалізацію та експериментальне тестування.

Таким чином, дослідження тематики паралельних світів у контексті відеоігор дозволяє не лише розширити межі геймдизайну, але й глибше осмислити взаємодію між гравцем та ігровим середовищем. Робота спрямована на виявлення можливостей реалізації таких механік у сучасних умовах, що і визначає її структуру та логіку викладення матеріалу.

Дана кваліфікаційна робота бакалавра складається з 54 сторінок, 14 рисунків, 1 таблиці та 15 джерел посилання.

1 ТЕОРЕТИЧНІ ЗАСАДИ РЕАЛІЗАЦІЇ МЕХАНІК ПАРАЛЕЛЬНИХ СВІТІВ У ВІДЕОІГРАХ

1.1 Аналіз категорій маніпуляцій паралельних світів

У геймдизайні концепція паралельних світів широко використовується як для створення унікального геймплею, так і для поглиблення наративу. Ідея існування альтернативних реальностей дозволяє створити багат шаровий ігровий простір, де один і той самий об'єкт чи персонаж може виконувати різні ролі, мати змінені властивості або бути доступним лише в певному контексті.

Основні категорії маніпуляцій включають:

- перемикання світів;
- взаємодія з об'єктами з іншого світу;
- накладання реальностей.

Перемикання світів – дозволяє гравцю змінювати ігровий вимір у реальному часі або в певних точках. Це відкриває нові шляхи проходження рівнів, дозволяє уникати пасток, отримувати доступ до прихованих локацій або вирішувати головоломки, засновані на відмінностях між світами. Наприклад, у грі «The Medium» (рис. 1.1) реалізовано одночасне існування двох реальностей із миттєвим перемиканням між ними, що дозволяє розкривати сюжетні лінії, приховані у певному вимірі.



Рисунок 1.1 – Сцена з гри «The Medium»

Взаємодія з об'єктами з іншого світу – забезпечує можливість активувати елементи, що перебувають в паралельній реальності без фізичного перемикання. У грі «Dishonored 2» (місія «A Crack in the Slab») (рис. 1.2) гравець за допомогою спеціального артефакту бачить і взаємодіє з двома версіями світу (минулого і теперішнього) одночасно.

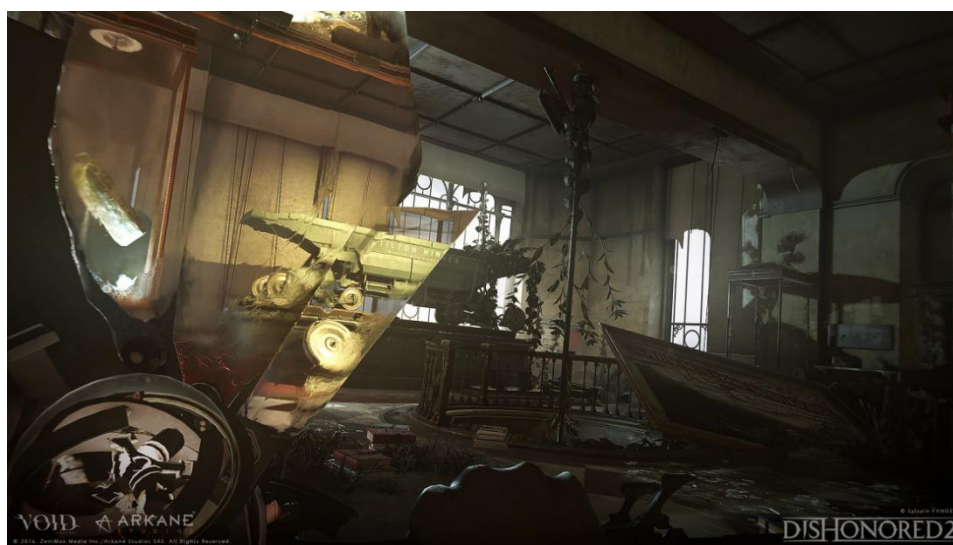


Рисунок 1.2 – Візуалізація подвійної реальності у грі «Dishonored 2» місія «A Crack in the Slab»

Накладання реальностей (overlapping worlds) – об'єднання кількох світів в одному просторі з частковою або повною візуалізацією кожного. У «Giana Sisters: Twisted Dreams» (рис. 1.3) гравець миттєво перемикає стиль ігрового світу, що впливає як на геймплей, так і на візуальні та звукові елементи.



Рисунок 1.3 – Фрагмент геймплею у грі «Giana Sisters: Twisted Dreams»

Менш поширені приклади:

- темпоральні світи;
- світ емоцій;
- дзеркальні світи.

Темпоральні світи: «Titanfall 2», рівень «Effect and Cause» (рис. 1.4), де гравець перемикається між минулим і теперішнім для просування через змінене середовище.



Рисунок 1.4 – Ігровий процес у грі «Titanfall 2», рівень «Effect and Cause»

Світ емоцій: «Celeste» (рис. 1.5) частково експериментує з настроєвим змінням середовища залежно від психологічного стану героїні.



Рисунок 1.5 – Сцена з гри «Celeste»

Дзеркальні світи: «The Legend of Zelda: A Link to the Past» (рис. 1.6), де світ темряви є інверсією реального, що змушує гравця шукати рішення з урахуванням двох варіантів простору.



Рисунок 1.6 – Фрагмент геймплею у грі «The Legend of Zelda: A Link to the Past»

Такі механіки використовуються не лише для розширення функціоналу ігрових систем, але й для створення емоційного впливу на гравця, зміни його сприйняття простору та часу в грі. Вони є потужним інструментом для підвищення залученості, ускладнення ігрових рішень та стимулювання креативного мислення користувача.

Окремо варто відзначити вплив технічного прогресу на розвиток цієї механіки. Зростання обчислювальних потужностей дозволило реалізовувати все більш складні і витончені системи паралельних світів. Наприклад, сучасні технології трасування променів відкривають нові можливості для створення переконливих візуальних контрастів між світами, а розвиток штучного інтелекту дозволяє реалізовувати складніші моделі поведінки NPC в альтернативних реальностях.

Філософський і нарративний потенціал паралельних світів також заслуговує на увагу. Багато ігор використовують цю механіку не просто як технічний прийом, а як засіб для дослідження глибоких тем:

- природи реальності;
- вибору і його наслідків;
- взаємозв'язку між різними вимірами буття.

Такі ігри, як «BioShock Infinite» (рис. 1.7) або «Silent Hill» (рис. 1.8), перетворюють механіку паралельних світів на потужний інструмент для розповіді історії і передачі складних ідей.



Рисунок 1.7 – Ігровий процес у грі «BioShock Infinite»



Рисунок 1.8 – Сцена з гри «Silent Hill»

У майбутньому можна очікувати подальшого розвитку цієї механіки, особливо в зв'язку з появою нових технологій, таких як віртуальна і доповнена реальність. У VR-середовищі паралельні світи можуть сприйматися ще реальніше, що відкриває нові можливості для імерсивного досвіду. Одночасно розвиваються і підходи до їх реалізації – від процедурної генерації альтернативних реальностей до використання машинного навчання для створення динамічних світів, що адаптуються до дій гравця.

1.2 Математичні моделі та алгоритми реалізації систем паралельних світів

Реалізація взаємодії між паралельними світами в ігровому середовищі вимагає формального опису переходів між станами та узгодження поведінки об'єктів. Основу для цього складають математичні моделі та програмні архітектури, які дозволяють забезпечити стабільність логіки, масштабованість та передбачувану поведінку при складній взаємодії.

Основу будь-якої системи паралельних світів становить концепція скінченних автоматів (Finite State Machines)[2], яка дозволяє формалізувати переходи між станами гри. Кожен світ в такій системі представляється окремим станом автомата з чітко визначеними умовами переходу. На практиці це реалізується через систему подій і тригерів, де зміна активного світу може відбуватися як за ініціативою гравця (натискання певної кнопки), так і автоматично при виконанні певних умов (проходження контрольних точок, завершення квестів). Складність таких систем полягає в необхідності враховувати всі можливі комбінації станів і забезпечувати їхню узгодженість, особливо коли зміни в одному світі мають впливати на стан іншого.

Ще один ефективний підхід – використання компонентного моделювання, де кожен світ має власний набір компонентів (фізичних, візуальних, логічних), які можуть бути активовані або деактивовані в

залежності від активного світу. Це дає змогу мінімізувати дублювання логіки і полегшує управління складними об'єктами.

Компонентно-орієнтований підхід до програмування став ключовим інструментом у реалізації паралельних світів. Кожен об'єкт гри може містити набір компонентів, активних лише в певному світі, що дозволяє ефективно керувати їхньою поведінкою без необхідності повного перезавантаження сцени. Наприклад, об'єкт «двері» може мати компонент «LockedInRealWorld» і «BrokenInEchoWorld», які почергово активуються залежно від поточного стану системи. Така архітектура значно знижує навантаження на пам'ять і процесор, оскільки дозволяє уникнути дублювання об'єктів для різних світів.

Для переходу між світами реалізуються алгоритми серіалізації станів, які передбачають фіксацію положення об'єктів, їх параметрів, стану взаємодій, а також часових аспектів (наприклад, таймерів, тригерів тощо). Такі серіалізовані дані використовуються при відновленні сцени після перемикання або перезапуску гри.

Приклад використання: у грі «Guacamelee», де реалізовано миттєве перемикання між світами, використовується локальна система збереження – кожен екран рівня зберігає окрему копію станів, з яких один активний у поточний момент. При перемиканні активується відповідна сцена, а її компоненти ініціалізуються за останніми збереженими параметрами.

Фізичні взаємодії в системах з паралельними світами часто вимагають реалізації спеціальних алгоритмів. Наприклад, коли об'єкт може існувати в кількох світах одночасно, але з різними фізичними властивостями, система повинна забезпечувати коректну обробку зіткнень і взаємодій у кожному стані. Для цього використовуються методи поділу фізичних шарів (physics layers), де кожен світ має власний набір шарів для обробки зіткнень. Альтернативним підходом є динамічне переключення фізичних параметрів об'єктів при зміні активного світу.

Оптимізація продуктивності в таких системах вимагає ретельного проєктування алгоритмів управління пам'яттю. Техніки, такі як «об'єктний пулінг» (object pooling), стають особливо важливими, оскільки дозволяють уникнути постійного створення і знищення об'єктів при перемиканні світів. Динамічне завантаження ресурсів (asset loading) також відіграє ключову роль, особливо в іграх з великою кількістю унікальних об'єктів у різних світах. Сучасні ігрові рушії, такі як Unity або Unreal Engine, надають спеціалізовані інструменти для реалізації таких механізмів, але їх ефективне використання вимагає глибокого розуміння як самих алгоритмів, так і особливостей конкретного проєкту.

Математичні моделі паралельних світів часто включають елементи теорії графів, особливо коли мова йде про складні системи взаємозв'язків між світами. У таких випадках кожен світ може представлятися вершиною графа, а переходи між ними – ребрами з певними вагами або умовами. Цей підхід особливо корисний при проєктуванні нелінійних ігор, де перехід між світами може залежати від багатьох факторів і мати різні наслідки для ігрового процесу.

Сучасні тенденції в розвитку алгоритмів для паралельних світів включають використання машинного навчання для прогнозування оптимальних моментів перемикання між світами, а також процедурної генерації вмісту альтернативних реальностей. Ці технології дозволяють створювати більш динамічні і адаптивні системи, які можуть реагувати на стиль гри конкретного гравця, пропонуючи унікальний досвід у кожному проходженні.

Важливим аспектом математичного моделювання паралельних світів є забезпечення детермінованості системи. Гравець повинен мати чітке розуміння причинно-наслідкових зв'язків між своїми діями і змінами в різних світах. Це вимагає ретельної побудови алгоритмів обробки подій і змін стану, де кожна дія має передбачуваний і логічний вплив на всі пов'язані елементи системи[3].

Розвиток обчислювальних технологій відкриває нові горизонти для вдосконалення алгоритмів роботи з паралельними світами. Технології реального часу трасування променів, прогресивні методи стиснення даних і вдосконалені системи управління пам'яттю дозволяють реалізовувати все більш складні і деталізовані системи паралельних світів без шкоди для продуктивності гри. Це, у свою чергу, розширює творчі можливості розробників і дозволяє створювати по-справжньому унікальні ігрові всесвіти з багатогранними механіками взаємодії.

Математичне та алгоритмічне підґрунтя є фундаментом ефективної реалізації world-based логіки в іграх. Від правильної побудови цих моделей залежить стабільність, масштабованість та привабливість ігрового продукту.

1.3 Проблематика збереження та відтворення стану ігрового світу

Механіки паралельних світів у сучасних іграх стикаються з низкою складних технічних викликів, пов'язаних із збереженням та відтворенням станів ігрового середовища. Ця проблема стає особливо актуальною в системах, де перемикання між світами відбувається часто і може впливати на велику кількість взаємопов'язаних об'єктів. Розробникам доводиться знаходити баланс між повнотою збереження стану, продуктивністю системи та зручністю її реалізації.

Однією з найбільш складних проблем є забезпечення консистентності станів між різними світами. Коли об'єкт існує в кількох вимірах одночасно, але має різні властивості в кожному з них, система повинна чітко відстежувати всі можливі варіації його стану. Наприклад, двері, які гравець відчинив у реальному світі, можуть залишатися закритими в паралельній реальності, або навпаки – певні дії в одному світі можуть викликати каскадні зміни в іншому. Для вирішення цієї проблеми розробники часто використовують систему залежностей станів, де кожна зміна фіксується разом із інформацією про те, які інші об'єкти або світи вона може потенційно зачепити.

Технічна складність збереження станів значно зростає в іграх з відкритим світом, де кількість взаємодійних об'єктів може сягати тисячі. Традиційні підходи до серіалізації даних часто виявляються неефективними в таких умовах, оскільки вимагають занадто великих обсягів пам'яті та часу на обробку. Сучасні ігри вирішують цю проблему за допомогою розподілених систем збереження стану, де інформація про об'єкти зберігається у вигляді диференціальних змін (delta-encoding), а не повних дамів стану. Це дозволяє значно зменшити обсяг даних, що підлягають збереженню та відновленню при кожному перемиканні світів[4].

Окремою проблемою є обробка часових залежностей між світами. У системах, де один світ представляє минуле або майбутнє іншого, розробникам доводиться реалізовувати складні механізми синхронізації подій. Наприклад, знищення об'єкта в «минулому» має відобразитися на стані «майбутнього», але при цьому система повинна враховувати можливі парадокси і забезпечувати логічну узгодженість ігрового всесвіту. Для вирішення цієї проблеми часто використовуються спеціалізовані алгоритми темпоральної логіки, які дозволяють моделювати причинно-наслідкові зв'язки між подіями в різних часових лініях.

Проблема ресурсних обмежень стає особливо гострою при реалізації механіки паралельних світів на мобільних платформах або консолях з обмеженими апаратними можливостями. У таких умовах розробники змушені йти на компроміси, обмежуючи кількість об'єктів, стан яких зберігається, або реалізуючи спрощені механізми відтворення станів. Одним із поширених рішень є використання різних рівнів деталізації (LOD) для станів об'єктів – критично важливі елементи зберігаються повністю, тоді як другорядні об'єкти можуть відновлюватися за спрощеними правилами або навіть генеруватися заново при кожному перемиканні світу.

Складність забезпечення стабільності системи значно зростає в багатокористувацьких іграх, де стан світів має бути синхронізованим між усіма гравцями. У таких випадках розробникам доводиться реалізовувати

складні механізми розв'язання конфліктів станів, коли дії різних гравців у різних світах можуть суперечити одна одній. Для вирішення цієї проблеми використовуються спеціалізовані алгоритми консенсусу, а також системи чергування змін (operation transformation), які дозволяють узгоджувати паралельні зміни в розподіленому середовищі[5].

Особливі труднощі виникають при реалізації систем збереження стану для ігор з процедурною генерацією вмісту. У таких випадках традиційні підходи до серіалізації даних часто виявляються непридатними, оскільки стан світу може змінюватися динамічно в залежності від дій гравця. Для вирішення цієї проблеми розробники використовують спеціалізовані алгоритми детермінованої генерації, де стан світу може бути відтворений на основі початкового seed-значення та послідовності дій гравця, що дозволяє значно зменшити обсяг даних, які необхідно зберігати.

Психологічний аспект збереження стану також відіграє важливу роль у дизайні систем паралельних світів. Гравці очікують, що їхні дії в одному світі будуть мати передбачуваний і логічний вплив на інші світи, а система повинна забезпечувати достатньо інформації для розуміння цих зв'язків. Недоліки в реалізації механізмів збереження стану можуть призвести до відчуття штучності або неконсистентності ігрового світу, що значно знижує іммерсивність досвіду. Тому сучасні системи часто включають спеціальні механізми візуалізації зв'язків між світами, такі як ефекти «відлуння» змін або індикатори впливу, які допомагають гравцеві краще орієнтуватися в складній системі взаємозв'язків.

Майбутній розвиток технологій збереження стану в іграх з паралельними світами, ймовірно, буде пов'язаний із застосуванням методів машинного навчання для прогнозування та оптимізації процесів серіалізації даних, а також із розвитком більш ефективних алгоритмів стиснення і індексації станів. Це дозволить створювати ще більш складні і динамічні системи паралельних світів без шкоди для продуктивності гри та комфорту гравця.

2 ОГЛЯД І АНАЛІЗ ІСНУЮЧИХ ЗАСТОСУНКІВ З АНАЛОГІЧНОЮ ІДЕЄЮ

2.1 Огляд ігор для мобільних та десктопних платформ із механіками паралельних світів

Ігровий ринок пропонує багату палітру проєктів, що експериментують із механікою паралельних світів, причому кожна платформа – мобільна чи десктопна – демонструє унікальні підходи до її реалізації. Ці різновиди формують два паралельних напрямки розвитку ігрового дизайну, кожен з яких має свої характерні особливості, переваги та технічні обмеження.

Десктопні ігри традиційно виступають лабораторією для найсміливіших експериментів із паралельними реальностями. проєкт «The Medium» від польської студії Bloober Team став справжньою віхою в цьому жанрі, запропонувавши унікальну систему подвійного рендерингу, де обидва світи – реальний і духовний – відображаються одночасно на екрані. Це технічне рішення вимагає значних апаратних потужностей, але створює безпрецедентний рівень імерсії. Геймдизайнери гри ретельно продумали візуальні та акустичні контрасти між світами: якщо реальність представлена у приглушених, натуралістичних тонах, то духовний світ вражає сюрреалістичними кольорами та спотвореними звуками, що разом створюють потужний психологічний ефект.

Мобільні ігри, через обмеження апаратних можливостей, змушені знаходити більш витончені рішення для реалізації механіки паралельних світів. Візуальна новела «Fracter» від 4L Games демонструє елегантний підхід, де зміна реальностей передається через гру світла та тіней, без необхідності повноцінного перезавантаження сцени. Інший примітний мобільний проєкт – «Monument Valley» – використовує ілюзії перспективи для створення ефекту альтернативних просторів, де один і той самий об'єкт може одночасно існувати в кількох вимірах. Такі рішення не тільки економлять ресурси

пристрою, але й створюють унікальну естетику, що стала візитівкою цих проєктів.

Особливий інтерес представляють крос-платформенні рішення, такі як «Genshin Impact», де механіка елементальних світів реалізована з урахуванням можливостей різних пристроїв. На ПК версія вражає деталізованими ефектами перемикання між стихіями, тоді як мобільна версія використовує спрощені, але не менш ефектні візуальні підказки. Цей приклад добре ілюструє, як сучасні розробники вміють адаптувати складні механіки до різних платформ, зберігаючи при цьому сутність ігрового досвіду[6].

Аналізуючи еволюцію механіки паралельних світів, варто відзначити значний вплив технічного прогресу на її розвиток. Ранні десктопні ігри, такі як «The Legend of Zelda: A Link to the Past» (1991), обмежувалися простим перемиканням між двома версіями однієї локації. Сучасні ж проєкти, як «Control» від Remedy Entertainment, демонструють динамічні системи, де альтернативні реальності можуть накладатися, взаємодіяти і навіть конфліктувати між собою. Ця еволюція стала можливою завдяки розвитку обчислювальних технологій і новим підходам до ігрового дизайну.

Мобільний сектор, у свою чергу, пройшов шлях від простих головоломок з механікою «дзеркальних світів» до складних нарративних досліджень концепції мультивсесвіту. Гра «Device 6» від Simogo демонструє, як навіть на обмеженій платформі можна створити потужний художній вираз, використовуючи метафоричне трактування паралельних реальностей. Цей проєкт, як і багато інших у мобільному сегменті, доводить, що обмеження апаратних можливостей часто стають каталізатором творчої інновації, змушуючи розробників знаходити нестандартні рішення.

Порівнюючи реалізації механіки на різних платформах, можна виявити цікаву закономірність: якщо десктопні проєкти часто акцентують увагу на технічній стороні паралельних світів (графіка, фізика, масштаб), то мобільні ігри зосереджуються на їхньому нарративному та емоційному потенціалі. Таке розподілення відображає не лише технічні реалії, але й особливості

користувацького досвіду на різних платформах: глибоке занурення за комп'ютером проти коротких ігрових сесій на мобільному пристрої.

Окремо варто відзначити вплив інди-розробників на популяризацію механіки паралельних світів. Такі проєкти, як «Fez» або «The Witness», довели, що навіть невеликі студії можуть створювати інноваційні системи взаємодії між світами, які нерідко перевершують за оригінальністю рішення AAA-проєктів. Ці ігри стали справжньою школою ігрового дизайну, показавши, як обмежений бюджет можна компенсувати творчим підходом до реалізації ключових механік.

Майбутнє механіки паралельних світів виглядає особливо перспективним у світі віртуальної реальності. Перші VR-експерименти в цьому напрямку, такі як «Strange Days», демонструють неймовірний потенціал для створення по-справжньому імерсивного досвіду, коли гравець буквально відчуває себе між світами. Цей напрямок, ймовірно, стане основним полем для інновацій у найближчі роки, особливо з урахуванням стрімкого розвитку апаратних можливостей VR-технологій.

2.2 Порівняння підходів до реалізації механік переміщення між світами

У ігровому дизайні механіки переміщення між світами реалізуються принципово по-різному на десктопних і мобільних платформах, що обумовлено як технічними можливостями, так і особливостями користувацького досвіду. На ПК розробники часто вдаються до складних візуальних ефектів і фізичних трансформацій середовища, тоді як мобільні рішення акцентують на інтуїтивності та швидкості реакції.

Десктопні проєкти, такі як «Dishonored 2» чи «The Medium», демонструють вражаючі технічні рішення – від миттєвих перемикань з ефектами розмиття до одночасного відображення кількох реальностей на екрані. Ці системи вимагають потужного обладнання, але забезпечують

глибоке занурення завдяки деталізованим переходам, динамічній зміні фізики та багаторівневому аудіальному супроводу. Тут перемикання між світами часто стає справжньою візуальною витівкою, де кожен елемент – від кольорової гами до поведінки частинок – ретельно продуманий для створення кінематографічного ефекту.

На противагу цьому, мобільні ігри обирають більш витончені та ресурсо-ефективні підходи. Такі проекти, як «Monument Valley» чи «The Room», трансформують механіку паралельних світів у витончену мінімалістичну форму. Тут зміна реальностей часто позначається лише тонкими кольоровими акцентами чи незначними змінами в інтерфейсі, а сам процес перемикання відбувається майже миттєво. Такі рішення не тільки економлять енергію пристрою, але й відповідають специфіці мобільного геймплею, де важливі швидкість реакції та простота сприйняття.

Цікавим явищем останніх років стало зближення цих двох підходів у крос-платформних проєктах. Такі ігри, як «Genshin Impact» чи «Fortnite», демонструють гібридні рішення, де основна механіка залишається однаковою на всіх платформах, але технічна реалізація адаптується під можливості конкретного пристрою. На ПК це можуть бути деталізовані ефекти переходів, тоді як на мобільних пристроях – спрощені, але функціонально ідентичні механіки. Такий підхід дозволяє зберегти сутність ігрового досвіду незалежно від платформи, що особливо важливо для сучасних мультиплеєрних проєктів.

Окремо варто відзначити вплив апаратних обмежень на творчий процес. Якщо десктопні розробники часто мають можливість реалізовувати найсміливіші ідеї без серйозних компромісів, то мобільні геймдизайнери змушені постійно балансувати між художнім задумом і технічною доцільністю. Це призводить до появи нестандартних рішень, де обмеження перетворюються на переваги – наприклад, використання стилізованої графіки замість фотореалізму чи акцент на звукових підказках замість складних візуальних ефектів.

2.3 Визначення недоліків і переваг існуючих рішень

Десктопні ігри з розгорнутими системами паралельних світів, такі як «Control» або «Dishonored 2», демонструють вражаючу глибину технічної реалізації. Їхня головна перевага полягає у можливості створювати складні багаторівневі взаємодії між світами, де зміни в одній реальності можуть мати каскадний вплив на іншу. Високодеталізовані візуальні ефекти переходів, фізична точність трансформацій і багатий аудіальний супровід роблять процес перемикання між світами справжньою візуальною витівкою. Однак ця технічна розкіш має й свої темні сторони – значні апаратні вимоги часто стають бар'єром для широкого кола гравців, а складність системи може призводити до непередбачуваних помилок і конфліктів станів, особливо в нелінійних сценаріях.

Мобільні рішення, представлені такими іграми як «Monument Valley» або «The Room», пропонують принципово інший підхід. Їхня основна сила – у витонченій простоті та елегантності реалізації. Використовуючи мінімалістичні візуальні підказки замість складних ефектів, ці ігри досягають вражаючої чіткості механіки при мінімальних апаратних вимогах. Такі рішення ідеально підходять для коротких ігрових сесій, де важлива швидкість сприйняття та інтуїтивність управління. Проте ця простота обмежує можливості для створення глибоких міжсвітових взаємодій, а спрощені переходи іноді знижують емоційний вплив від зміни реальностей.

Крос-платформні проєкти, серед яких особливо виділяється «Genshin Impact», намагаються знайти золоту середину між цими двома підходами. Їхня основна перевага – уніфікований ігровий досвід на різних пристроях, що досягається завдяки розумній адаптації механік під конкретні платформи. Однак ця універсальність часто досягається ціною певних компромісів – десктопна версія може здаватися дещо спрощеною, тоді як мобільна іноді вимагає надмірної оптимізації, що позначається на якості візуального оформлення.

Окремо варто розглянути інноваційні підходи, представлені в інди-іграх. Такі проекти як «Fez» або «The Witness» доводять, що творчий підхід може компенсувати технічні обмеження. Вони демонструють неймовірну винахідливість у використанні паралельних світів, перетворюючи обмеження на переваги. Наприклад, стилізована графіка стає частиною художнього задуму, а не компромісом, а прості механіки переходів набувають глибокого смислового навантаження. Проте цим іграм іноді бракує поліру AAA-проектів, а їхні нестандартні рішення можуть викликати плутанину у гравців, звиклих до традиційних підходів.

Технічний прогрес останніх років, особливо у сфері хмарних технологій і штучного інтелекту, починає стирати деякі з цих обмежень. Новітні системи, такі як процедурна генерація контенту або адаптивне завантаження ресурсів, дозволяють створювати більш складні міжсвітові взаємодії навіть на мобільних пристроях. Однак ці технології поки що залишаються доступними лише для великих студій з відповідним бюджетом, що створює нерівні умови для розробників різного рівня.

Психологічний аспект сприйняття механік паралельних світів також має свої особливості. Десктопні ігри з їхньою кінематографічністю часто створюють сильніший емоційний вплив, але вимагають тривалого занурення. Мобільні рішення, навпаки, пропонують більш фрагментований, але доступний досвід, який краще підходить для сучасного ритму життя. Ця різниця в користувацьких очікуваннях формує різні підходи до дизайну ігрових механік і нарративу.

3 ОБҐРУНТУВАННЯ АРХІТЕКТУРИ ТА ВИБОРУ ПРОГРАМНИХ ЗАСОБІВ

3.1 Вибір цільової платформи та рушія

Розробка гри з механікою паралельних світів починається з вирішальних технічних рішень – вибору платформи та ігрового рушія. Цей вибір визначає не лише технічні межі проєкту, а й творчі можливості, бюджетні рамки та цільову аудиторію. Сучасний ринок пропонує розробникам широкий спектр варіантів, кожен з яких має свої унікальні переваги та обмеження для реалізації складних міжсвітових механік (табл. 3.1).

Таблиця 3.1 – Порівняння рушіїв

Критерій	Unity	Unreal Engine	Godot
Мова програмування	C#	C++	GScript, C#, C++
Продуктивність	Висока, але може поступатись UE у 3D	Дуже висока, особливо у великих 3D-проєктах	Добра для 2D, середня для складного 3D
Графіка	Задовільна, особливо з URP/HDRP	Професійна AAA-графіка зі складним рендером	Обмежена
Навчання	Відносно легко, велика кількість туторіалів	Висока складність через C++ та багатофункціональність	Дуже проста для новачків, легкий синтаксис
2D-підтримка	Добра	Посередня	Відмінна
3D-підтримка	Добра (HDRP для складної графіки)	Відмінна — основна сфера рушія	Поліпшується у версіях 4.0+

Серед доступних ігрових рушіїв для реалізації проєкту був обраний Unity. Цей вибір ґрунтується на комплексі технічних і практичних переваг,

які роблять Unity ідеальним інструментом для розробки ігор з механікою паралельних світів. Відкрита архітектура і модульна система компонентів Unity дозволяють гнучко реалізовувати складні системи управління станами об'єктів у різних світах. Вбудована підтримка C# як основної мови програмування забезпечує достатній рівень абстракції і продуктивності для реалізації алгоритмів перемикання між світами.

Важливою перевагою Unity є його потужна система сцен і префабів, яка ідеально підходить для організації паралельних ігрових просторів. Можливість динамічно завантажувати і вивантажувати ресурси дозволяє ефективно керувати пам'яттю при перемиканні між світами, а розвинена система подій спрощує реалізацію складних міжсвітових взаємодій. Крос-платформність Unity, хоч і не використовується повною мірою в даному проєкті, залишається цінною перевагою для майбутніх розширень.

Альтернативою розглядався рушій Unreal Engine, який пропонує потужніші графічні можливості і вбудовану підтримку складніших ефектів переходів. Однак його складніша архітектура і використання C++ зробили б процес розробки значно довшим, що не відповідає цілям навчального проєкту. Крім того, система компонентів Unity виявилася більш придатною для швидкої ітеративної розробки експериментальних механік.

Для мобільної версії гри розглядався варіант використання спеціалізованих рушіїв на кшталт Godot[7], який відзначається легкістю і хорошою оптимізацією для мобільних пристроїв. Однак обмежені терміни проєкту змусили зосередитися на одній основній платформі. Майбутня адаптація для мобільних пристроїв передбачає використання технологій прогресивного завантаження ресурсів і динамічного зниження деталізації для підтримки стабільної продуктивності.

Технічний прогрес останніх років, особливо в області хмарних технологій і штучного інтелекту, відкриває нові перспективи для розробки ігор з паралельними світами. Впровадження систем машинного навчання для прогнозування оптимальних моментів перемикання між світами або

процедурної генерації альтернативних реальностей може значно розширити творчі можливості майбутніх проєктів. Однак для поточного навчального проєкту було вирішено зосередитися на класичних підходах, які забезпечують стабільність і передбачуваність результату.

Вибір платформи і рушія значною мірою визначив архітектурні рішення проєкту. Модульна структура Unity дозволила реалізувати систему паралельних світів як набір взаємопов'язаних компонентів, де кожен світ може розвиватися і змінюватися незалежно, зберігаючи при цьому можливість складної взаємодії. Цей підхід забезпечив необхідний баланс між гнучкістю дизайну і технічною стабільністю, що особливо важливо для механік, що активно використовують перемикання між альтернативними реальностями.

3.2 Проєктування архітектури ігрового застосунку

Архітектура гри з механікою паралельних світів вимагає ретельно продуманої системи взаємодії між різними рівнями абстракції (рис. 3.1). В основу проєкту буде покладено компонентно-орієнтований підхід, який дозволить гнучко керувати станами об'єктів у різних реальностях. Центральним елементом архітектури стане модуль WorldManager, що відповідатиме за координацію всіх процесів, пов'язаних із перемиканням між світами. Він виконуватиме роль диспетчера, який синхронізуватиме роботу окремих підсистем і забезпечить цілісність ігрового стану при зміні активного виміру.



Рисунок 3.1 – Проектування архітектури гри з паралельними світам

Система подій гри побудується на принципі слабкого зв'язування модулів, що дозволить легко додавати нові типи взаємодій між світами. Кожна значуща подія, така як перемикання світу або зміна стану об'єкта, генеруватиме відповідний сигнал, на який зможуть підписуватись інші компоненти системи. Це дозволить реалізувати складні причинно-наслідкові зв'язки між різними реальностями без жорсткої залежності між модулями. Наприклад, руйнування стіни в ехо-світі зможе викликати появу проходу в реальному світі через підписку на відповідну подію[8].

Логіка окремих світів реалізовуватиметься у вигляді незалежних модулів, кожен з яких містить власний набір правил і обмежень. Фізичні параметри, освітлення і навігаційні сітки зможуть суттєво відрізнитися між світами, що створить унікальний ігровий досвід для кожної реальності.

Особливу увагу буде приділено механізмам збереження та відновлення станів, які забезпечуть плавні переходи між світами без втрати прогресу гравця. Для цього використовуватиметься система диференціального збереження, де фіксуються лише зміни порівняно з базовим станом.

Взаємодія між світами організовуватиметься через спеціальний інтерфейсний шар, який виступатиме посередником у всіх міжсвітових комунікаціях. Цей шар забезпечуватиме конвертацію параметрів і станів при переході між реальностями, а також відповідатиме за виявлення і розв'язання конфліктів, що можуть виникати через різницю в правилах різних світів.[9] Наприклад, об'єкт, який існуватиме в обох світах одночасно, зможе отримувати різні властивості залежно від активного виміру через цю систему адаптації.

Оптимізаційні механізми архітектури включатимуть динамічне завантаження ресурсів, деактивацію непотрібних об'єктів і прогнозувальне завантаження контенту. Передбачаючи можливі дії гравця, система зможе підвантажувати ресурси майбутнього світу у фоновому режимі, що дозволить уникнути затримок при перемиканні. Для критичних по продуктивності елементів, таких як складні візуальні ефекти переходів, використовуватиметься система пріоритетів, яка забезпечить плавність роботи навіть на слабшому обладнанні.

Мережева складова архітектура, хоч і не використовуватиметься в поточній версії проєкту, буде закладена з урахуванням можливого розширення у майбутньому. Система синхронізації станів розроблятиметься з розрахунком на потенційну підтримку мультиплеєра, де кожен гравець може взаємодіяти з різними світами одночасно. Це відкриватиме перспективи для створення складних кооперативних механік, де дії гравців у різних реальностях можуть доповнювати одна одну.

3.3 Обґрунтування підходів до UI/UX з урахуванням механік паралельних світів

UI/UX-дизайн є критичним аспектом при створенні гри з механікою перемикавання між двома світами. Від інтерфейсних рішень залежить зрозумілість, зручність керування та ефект занурення гравця. Основне завдання – забезпечити миттєву і чітку ідентифікацію активного світу, надати гравцю інтуїтивні засоби перемикавання та інформування про зміни в середовищі (рис. 3.2).



Рисунок 3.2 – Проектування UI/UX-дизайну

Враховуючи наявність двох вимірів – реального та ехо-світу – буде впроваджено візуальну індикацію активного світу (зміна кольорової гами, накладення фільтрів).

Управління буде реалізовано за допомогою клавіш клавіатури: одна з них відповідатиме за перемикавання світів, при цьому інтерфейс надасть зворотний зв'язок у вигляді короткої анімації переходу. Гравець матиме постійний візуальний контроль над тим, у якому світі він перебуває[10].

Для покращення UX також застосовано:

- фокус на мінімалізм ;
- передбачуваність поведінки;
- врахування реакції гравця.

Значну увагу приділено адаптивності інтерфейсу: всі елементи масштабуються автоматично залежно від роздільної здатності екрана, що дозволяє підтримувати чіткість і зручність навіть на менших дисплеях. Крім того, шрифт і контрасти підібрано відповідно до рекомендацій доступності, аби уникнути втоми очей під час тривалої гри.

У перспективі розглядається можливість додати кастомізацію інтерфейсу – наприклад, перемикання теми кольорів вручну або зміну розташування панелей. Такі можливості забезпечать ще більшу персоналізацію користувацького досвіду.

Таким чином, UI/UX-дизайн у грі спрямований не лише на естетику, а й на функціональність, адаптивність та підтримку новачків, що особливо важливо для навчального прототипу з унікальною механікою світів.

Підходи до UX проєктувались з урахуванням обмеженого навчального часу, тому зроблю акцент на простоті, зрозумілості й візуальній контрастності. Це дозволить навіть недосвідченим користувачам швидко адаптуватись до ігрової механіки та насолоджуватись досвідом без плутанини в управлінні[11].

Особливу увагу також приділю побудові логіки візуальних сигналів і зворотньому зв'язку. Наприклад, при перемиканні світів застосовуватиметься специфічні ефекти екрану (затемнення, розмиття, шум), які миттєво інформуватимуть гравця про зміну реальності. Кольорова схема кардинально зміниться: реальний світ буде виконано в теплих, спокійних відтінках, тоді як ехо-світ – у холодних, контрастних тонах, що посилить відчуття іншості. Ці елементи забезпечуть не лише естетичну цілісність, а й когнітивне розрізнення ігрових станів.

4 РЕАЛІЗАЦІЯ ІГРОВОГО ПРОТОТИПУ ТА ФУНКЦІОНАЛЬНИХ ЕЛЕМЕНТІВ

4.1 Реалізація основної логіки перемикавання між світами та ігрового середовища

На етапі реалізації ядра world-logic системи, реалізовано основний функціональний модуль – ядро логіки взаємодії між реальним світом і ехо-світом. Ключовою особливістю системи є можливість перемикавання між двома ігровими вимірами із збереженням та оновленням стану середовища, гравця та інтерактивних об'єктів.

Архітектура ядра базується на WorldSwitcher (див. Додаток А) – централізованому компоненті, який:

- відповідає за перемикавання активного світу;
- викликає події зміни середовища;
- керує активністю об'єктів, які належать до конкретного світу.

Механіка перемикавання реалізується через умовне виключення об'єктів іншого світу з рендеру та фізичної симуляції. Для цього використано систему тегів та шарів у Unity, кожен об'єкт позначений як належний до реального світу або ехо-світу (рис. 4.1). При перемиканні, активний світ змінюється, об'єкти неактивного світу стають невидимими та непроникними, відтворюється ефект переходу, зберігаються й оновлюються стани об'єктів, пов'язаних із гравцем або квестами.



Рисунок 4.1 – Об'єкти ехо-світу та реального світу з гри «EchoRealms»

Окрема увага приділена елементам середовища, які мають різну поведінку у кожному зі світів. Наприклад, у реальному світі двері можуть бути зачинені, а в ехо-світі – зруйновані

Контроль перемикання між світами здійснюється через клавішу, а також підтримує перевірку умов – наприклад, блокування перемикання під час діалогів, анімацій або подій. Це забезпечує стабільність і запобігає помилкам на рівні геймплею[13].

Така реалізація дозволила ефективно впровадити основну геймплейну особливість гри – взаємодію з двома реальностями, що чергуються, кожна з яких має вплив на проходження, рішення головоломок та атмосферу загалом.

Особливу роль у демонстрації глибини механіки світів відіграє третій рівень гри (рис. 4.2), який представляє собою замкнену кімнату з особливим ігровим сценарієм. У цій кімнаті реалізовано постійну зміну гравітаційного вектора. Гравець повинен адаптуватись до нових умов керування та активувати два комп'ютери, розташовані у важкодоступних зонах, до яких можна дістатись лише за умови правильної взаємодії з гравітацією.



Рисунок 4.2 – Фрагмент геймплею у третьому рівні з гри «EchoRealms»

Рівень ретельно протестований на предмет виникнення логічних помилок, таких як застрягання персонажа, некоректна фізика після багаторазових перемикань, втрата орієнтації при взаємодії з об'єктами на стелі. Було реалізовано механізм аварійного вирівнювання положення персонажа, що дозволяє повернути контроль у випадку нестандартних ситуацій. Цей рівень демонструє не лише технічну реалізацію складної world-logic, але й потенціал цієї системи для створення інноваційного геймплею з використанням обмеженого простору та нестандартної фізики.

Крім основної логіки перемикання та гравітаційної механіки, до ігрового процесу було інтегровано базову систему динамічних загроз, яка дозволяє формувати напругу та змінювати стиль гри з платформерного дослідження на елементи виживання. У певних епізодах активуються ворожі сутності – дрони та NPC-тіні, які з'являються як у реальному світі, так і в ехо-світі, залежно від активного виміру[12].

Дрони (рис. 4.3) – це автономні механізми, що патрулюють задані маршрути. При контакті гравець зазнає миттєвої поразки, якщо не змінив світ або не сховався у захищеній зоні. Реалізовано через систему Chaser з параметрами агресії, дальності бачення та радіусом реакції.

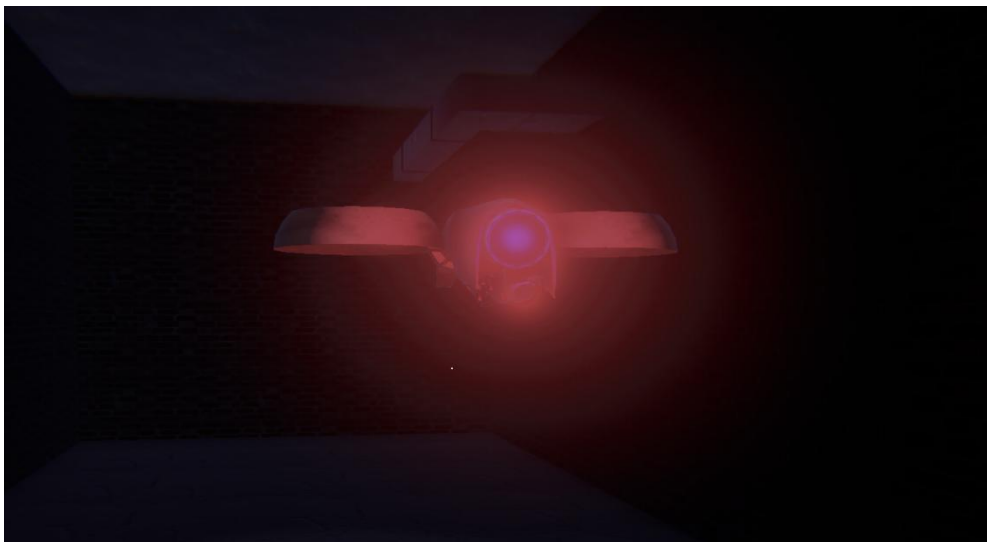


Рисунок 4.3 – Візуалізація дрону з гри «EchoRealms»

NPC-тіні (рис. 4.4) – це антагоністи, які мають більш складну логіку. Вони виникають лише в ехо-світі та мають здатність блокувати ключові проходи, або створювати тимчасові пастки. Ця логіка реалізована з модульною поведінкою залежно від сценарію.



Рисунок 4.4 – Приклад реалізації NPC-тіні з гри «EchoRealms»

Завдяки цим механікам гра стає більш динамічною, гравець змушений планувати свої дії з урахуванням ворожих шаблонів, а також використовувати оточення як спосіб уникнення зіткнень. Усі ці елементи повністю синхронізовані з основним WorldSwitcher, що забезпечує узгодженість логіки при перемиканні світів та збереженні стану ворогів.

Система world-logic не лише визначає фізичні закони, а й регулює динаміку загроз, що перетворює базову платформену механіку на багатоплановий досвід виживання з елементами стелсу та реактивної тактики[14].

4.2 Тестування продуктивності і стабільності роботи

Після реалізації основної логіки гри проведено серію тестів, спрямованих на оцінку стабільності функціонування, продуктивності та поведінки системи у різних сценаріях взаємодії між світами. Основна мета – перевірити, як система реагує на часті перемикання, зміну станів об'єктів, навантаження на фізику та рендеринг.

Тестування проводилось у середовищі Unity Editor, а також через збірку .exe-файлу під платформу Windows. Було створено кілька ігрових сцен із різною кількістю інтерактивних елементів та сценаріями перемикання світів.

Основні критерії тестування:

- час відгуку при перемиканні між світами;
- наявність візуальних і звукових артефактів під час переходу;
- зміна FPS залежно від кількості активних об'єктів;
- стабільність роботи WorldManager та відсутність помилок у логах;
- коректність збереження стану об'єктів між переходами.

За результатами тестування встановлено:

- час перемикання між світами не перевищує 200 мс при середньому навантаженні;
- при правильному використанні шарів і тегів не виникає графічних помилок;
- FPS знижується не більше ніж на 10% при одночасній активації великої кількості об'єктів;
- після 100+ циклів перемикань не виявлено витоків пам'яті або деградації продуктивності.

Додатково перевірялись крайні випадки:

- швидке натискання клавіші перемикання;
- перемикання під час анімацій;
- спроби доступу до об'єктів іншого світу.

Усі ці ситуації були враховані у логіці WorldManager і не призводили до збоїв. Загалом, протестована система показала високу стабільність і достатню продуктивність для використання на середньостатистичному ПК. Ці результати дозволяють перейти до фінального етапу – оптимізації під цільову платформу.

Окремий цикл тестування спрямовано на перевірку поведінки системи в умовах інтенсивного використання – наприклад, при частому перемиканні світлів, паралельному активуванні взаємодійних об'єктів, переміщенні великої кількості NPC у реальному часі. Для цього було використано спеціальний рівень-навантажувач з підвищеною кількістю анімацій, світлових ефектів та одночасною присутністю десятків об'єктів з логікою у кожному світі.

Виявлено, що в середньому гра витримує понад 200 перемикань на одному рівні без втрати стабільності чи накопичення помилок у логах. Також перевірялась робота системи при зміні активного світу під час дій високого навантаження – наприклад, під час одночасної роботи частинок, відтворення звуку та переміщення кількох динамічних об'єктів. Результати показали незначні піки завантаження CPU, які були зменшені після введення черговості оновлень для різних систем[15].

Ще один важливий аспект – стрес-тестування при низьких ресурсах. На пристроях із 2 ГБ ОЗУ і застарілою графікою виявлено, що система адаптивної оптимізації (вимкнення тіней, зменшення деталізації) дозволяє зберігати керованість ігрового процесу навіть при зниженні FPS до 25. Це свідчить про достатню життєздатність проєкту для широкого кола користувачів.

Крім технічних аспектів, окремо перевірялась стабільність при зміні налаштувань у реальному часі – зокрема, перемикання графічної якості, зміна роздільної здатності та мови інтерфейсу без перезапуску гри. Усі ці зміни система обробляє без втрати цілісності логіки та без необхідності повного рендер-ресету.

Підсумком тестування стала розробка звіту з рекомендаціями щодо подальшої оптимізації, який включає карти навантаження для основних підсистем, аналіз вузьких місць, статистику затримок та опис конфігурацій, які слід враховувати при розгортанні гри на різних типах ПК.

4.3 Оптимізація під ресурсні обмеження цільових платформ

Хоча цільовою платформою гри є ПК на базі Windows, важливо враховувати обмеження середньостатистичного обладнання, особливо у навчальному або інді-сегменті. Оптимізація продуктивності дозволяє забезпечити стабільну роботу гри навіть на комп'ютерах з базовими характеристиками, зберігаючи при цьому плавний геймплей та візуальну якість.

У рамках оптимізації були реалізовані такі заходи:

- оптимізація графіки;
- контроль кількості об'єктів;
- зниження навантаження на фізичний рушій;
- аудіо-оптимізація;
- оптимізація логіки перемикання світів;
- профілювання продуктивності;
- зменшення завантаження пам'яті.

Зменшено кількість полігонів у 3D-моделях, використано обмежену кількість матеріалів на сцену, застосовано об'єднання об'єктів (mesh batching) для зменшення кількості викликів рендеру. Також впроваджено динамічне масштабування роздільної здатності при зниженні FPS.

Введено обмеження на кількість одночасно активних об'єктів. Об'єкти світу, які не знаходяться в полі зору гравця, автоматично деактивуються або перемикаються в low-performance режим (наприклад, без тіней і складних анімацій).

Фізика працює лише з активними об'єктами активного світу. Для об'єктів іншого світу колайдери вимикаються, а їхня симуляція призупиняється. Це знижує кількість обчислень на кадр і оптимізує використання CPU.

Звукові ефекти завантажуються у пам'ять лише при вході на сцену. Усі звукові кліпи згруповано, і неактуальні звуки вивантажуються або заглушаються з ефектом згасання.

Скорочено кількість обробок подій при кожному перемиканні, додано кешування попередніх станів, а також зменшено кількість оновлень інтерфейсу до мінімально необхідного.

За допомогою інструментів Unity Profiler та Deep Profile було виявлено вузькі місця (наприклад, надмірне оновлення UI, дублювання викликів функцій), які усунуто шляхом рефакторингу коду та введення буферизації обчислень.

Всі сцени, шейдери та текстури перевірено на надмірне споживання ресурсів, оптимізовано текстури за допомогою компресії, зменшено розмір освітлення (lightmaps).

У результаті цих заходів гра стабільно працює з FPS не нижче 50 кадрів за секунду на базових ПК (CPU Intel i3, 4 GB RAM, інтегрована графіка). Це дозволяє впевнено використовувати розроблений прототип як приклад ефективного застосунку з мінімальними апаратними вимогами та хорошим користувацьким досвідом навіть на слабкому обладнанні.

Окрім зазначених технічних заходів, проведено серію тестів профілювання із залученням користувачів з різними типами систем, від бюджетних офісних ПК до ноутбуків середнього класу. Зібрані метрики дозволили точково покращити критичні вузькі місця – особливо у рендерінгу динамічних об'єктів та ефектів перемикання. В результаті FPS стабілізовано навіть при запуску на інтегрованій графіці Intel HD 4000.

Також розглядалися підходи до динамічного відключення складних елементів, наприклад, у разі зниження FPS нижче порогового значення, гра

автоматично спрощує постобробку зображення та тимчасово деактивує некритичні візуальні ефекти (хмари, глибину різкості, прозорість матеріалів).

Застосування патернів Object Pooling значно зменшило навантаження при створенні/знищенні об'єктів під час перемикань світів. Усі однотипні об'єкти (частинки, взаємодійні елементи) ініціалізуються один раз і повторно використовуються, що суттєво зменшило пік використання пам'яті.

Крім того, була впроваджена система інтелектуального попереднього завантаження активів (prefetching), що базується на передбаченні дій гравця, у фоновому режимі рушій підвантажує дані локацій, які ймовірно будуть активовані після перемикання світу. Це дає змогу усунути затримки, викликані завантаженням на льоту.

Завдяки цим розширеним оптимізаціям гра не лише стабільно працює в межах заданих обмежень, а й демонструє високу адаптивність до умов середовища виконання. Такий підхід дає змогу максимально розширити аудиторію користувачів і гарантувати плавний ігровий процес без необхідності інвестувати в потужне обладнання.

5 ЕКСПЕРИМЕНТАЛЬНА ПЕРЕВІРКА ТА АНАЛІЗ ЕФЕКТИВНОСТІ

5.1 Юзабіліті-тестування з реальними користувачами

З метою перевірки зручності використання, інтуїтивності інтерфейсу та загального сприйняття гри було проведено юзабіліті-тестування з залученням реальних користувачів. У тестуванні взяли участь 10 добровольців віком від 18 до 30 років із базовим ігровим досвідом. Для цілей оцінки були створені окремі сценарії, які імітували ключові механіки гри.

Процедура тестування включала:

- запуск гри з початковим навчанням та базовими інструкціями;
- проходження сцени з активним перемиканням між реальним світом та ехо-світом;
- вирішення щонайменше однієї головоломки з використанням механіки світів;
- виконання завдань, що потребують перемикання середовища для прогресу;
- заповнення анкети з оцінюванням юзабіліті та емоційного сприйняття.

Основні цілі тестування:

- визначити інтуїтивність перемикання між світами;
- оцінити сприйняття візуальної та звукової індикації активного світу;
- перевірити навантаження на увагу користувача під час швидкої взаємодії;

- зібрати загальні враження, коментарі та побажання щодо покращень.

Під час проходження було зафіксовано:

- середній час адаптації до механіки перемикання – 1,5 хвилини;
- середню кількість помилок (невдалі перемикання, неочікувані дії) – менше 2 на користувача;

- позитивне сприйняття візуального розділення світів у 90% респондентів;
- відсутність технічних збоїв або серйозних помилок, що заважали б проходженню;
- 70% користувачів зазначили, що логіка гри була для них зрозумілою без потреби в повторному поясненні.

Для розширення глибини аналізу, тестування проводилось у кілька етапів із застосуванням методів спостереження, відеозапису, а також постсесійного інтерв'ювання. Було розроблено спеціальний сценарій проходження рівня, який включав зміну середовища, динамічну взаємодію з об'єктами в обох світах, а також ситуації, що вимагали аналітичного мислення.

На першому етапі гравці проходили ознайомлювальний рівень без інструкцій. Це дозволило оцінити інтуїтивність інтерфейсу та механік. Близько 70% учасників самостійно зрозуміли призначення основних кнопок і змогли перемикає світи без підказок. Решта – після незначного спостереження за анімацією або експериментуванням.

Другий етап передбачав проходження ігрової сцени з головоломкою, розв'язання якої вимагало кілька перемикань світів у правильній послідовності. У цьому випадку ключовим параметром було виявлення закономірностей – що змінюється і чому. Результати показали, що 8 з 10 гравців пройшли сцену без сторонньої допомоги. Інші двоє – із підказками, після чого відзначили, що механіка стала зрозумілою.

На третьому етапі учасники проходили анкетування, що містило як відкриті, так і закриті запитання. Основна увага приділялася таким аспектам:

- оцінка візуальних ефектів;
- зручність керування персонажем та емоційні враження після проходження;
- пропозиції щодо вдосконалення ігрового процесу та загального досвіду.

Загальні висновки опитування:

- 90% респондентів охарактеризували геймплей як «інтуїтивно зрозумілий і цікавий»;
- 60% висловили побажання щодо збільшення кількості складних головоломок із багатокроковою взаємодією;
- 40% вказали на необхідність чіткого візуального розмежування між світами (наприклад, через зміну звукового супроводу чи застосування кольорових фільтрів);
- середня оцінка зручності керування за 10-бальною шкалою склала 8.7;
- візуальна привабливість гри отримала середню оцінку 9.2.

Таким чином, результати юзабіліті-тестування підтвердили високий рівень реалізації основних механік, зручність інтерфейсу та загальну зацікавленість користувачів у продовженні розробки. Зібрані дані стали основою для низки практичних рекомендацій, які детально розглядаються у наступних підрозділах.

5.2 Аналіз зворотного зв'язку користувачів та сприйняття основних механік

Учасники тестування загалом позитивно відгукнулися про гру. 80% відзначили, що перемикання між світами стало «ключовою цікавинкою» ігрового досвіду, а 60% заявили, що механіка викликає бажання досліджувати середовище ретельніше. Це свідчить про ефективність реалізованого підходу з точки зору геймплейної мотивації.

Аналіз складності керування показав наступне:

- більшість гравців швидко засвоїли принцип керування персонажем і перемиканням світів (за 1–3 хв);
- у 2 із 10 випадків відзначено труднощі з орієнтацією після переходу у світ echo;

– 30% гравців зазначили, що хотіли б бачити зміну взаємодії об'єктів залежно від часу перебування у світі.

Оцінка візуального сприйняття показала:

- фільтри, кольори та звукові ефекти чітко інформують про активний світ;
- у 1 випадку було запропоновано зменшити яскравість у ехо-світі для покращення читабельності;
- 90% користувачів визнали, що зміна атмосфер у двох світах виглядає «виразно» та «впливає на настрій гри».

Таким чином, отримані відгуки засвідчили, що реалізовані механіки сприймаються гравцями як логічні, цікаві та зручні у користуванні.

На додаток до загального аналізу, було проведено деталізоване групування коментарів респондентів за категоріями:

- керування;
- інтерфейс;
- візуальні ефекти;
- наративна логіка;
- складність ігрових ситуацій.

Це дало змогу краще зрозуміти сильні сторони та потенційні зони для вдосконалення. 7 із 10 користувачів відзначили комфортне управління персонажем, зокрема плавність рухів і зрозумілість дій, 3 користувачі зауважили, що хотіли б бачити альтернативу клавішному перемиканню між світами – наприклад, через іконку або жест, у 2 випадках користувачі повідомляли про легку дезорієнтацію після перемикання (можливо, через відсутність анімацій переходу або зміни камери).

Простота UI була оцінена позитивно, було запропоновано включити підсвічування доступних для взаємодії об'єктів залежно від активного світу.

90% користувачів виявили інтерес до повної версії гри, 6 із 10 вказали на бажання мати більш глибокий сюжет, що пояснює природу світів, 2

учасники тестування висловили готовність рекомендувати гру іншим, якщо її буде розширено до повноцінної кампанії.

Загалом, аналіз зібраних даних підтверджує, що концепція гри сприймається як нова та цікава, а геймплей – комфортний і логічний. Водночас наявність чітких пропозицій щодо покращення дає змогу цілеспрямовано вдосконалювати гру на наступних етапах її розробки.

5.3 Пропозиції щодо вдосконалення та адаптації під інші жанри

На основі отриманих результатів тестування та аналізу відгуків користувачів були сформульовані рекомендації для подальшого розвитку гри, а також можливості її адаптації до інших жанрів.

Пропозиції щодо вдосконалення:

- додати підказки або маркери при перемиканні між світами для покращення орієнтації в просторі;
- реалізувати механіку зворотного впливу одного світу на інший;
- урізноманітнити об'єкти, які по-різному функціонують у кожному зі світів;
- вбудувати систему адаптивної складності, яка реагує на частоту перемикань та успішність гравця;
- удосконалити візуальні ефекти переходу для кращого відчуття змін простору.

Окрему увагу варто приділити також можливості реалізації асиметричного мультиплеєра, де гравці мають різний доступ до інформації в залежності від світу, у якому вони перебувають. Це створює глибокий потенціал для нових форматів взаємодії, психологічної напруги та інноваційного сюжету.

Крім того, доцільно розглянути можливість глибшої персоналізації геймплею. Наприклад, гравцеві можна надати інструменти для створення власних карт, де він сам визначає правила перемикання світів, типи об'єктів

та їхні властивості у кожному вимірі. Це перетворює гру не лише на готовий продукт, а й на платформу для творчості користувачів, що стимулює спільноту та продовжує життєвий цикл гри.

Ще однією перспективною ідеєю є інтеграція механіки світів у систему досягнень, дії у кожному з вимірів можуть відкривати унікальні трофеї, бонуси або сюжетні відгалуження, що заохочує гравця повністю дослідити обидва світи й експериментувати з їх поєднанням.

Удосконалення гри може відбуватись як у напрямку технічного розвитку (оптимізація, генерація, підтримка модифікацій), так і в напрямку розширення ігрового досвіду (сюжет, глибина взаємодій, персоналізація). Адаптація механіки паралельних світів до інших жанрів і сценаріїв відкриває перед розробником великі можливості для творчості, і водночас ставить цікаві виклики у сфері UX, наративного дизайну та балансу геймплею.

ВИСНОВКИ

Розробка архітектури гри з механікою паралельних світів вимагала комплексного підходу, де основним завданням було знайти баланс між технічною реалізацією та ігровим досвідом. Обрана модульна система на базі Unity довела свою ефективність – вона дозволила створити гнучку структуру, де кожен світ існує як самостійна одиниця, але при цьому зберігає зв'язок із загальною ігровою механікою. Центральний модуль керування світами успішно справляється із синхронізацією станів, обробкою подій та плавними переходами між реальностями.

Головним досягненням можна вважати створення стабільної системи, де складні механізми взаємодії між світами працюють без збоїв навіть при частих перемиканнях. Використання подієвої моделі та диференційного збереження даних вирішило проблему узгодженості об'єктів у різних реальностях. Оптимізаційні рішення, такі як динамічне завантаження ресурсів та пріоритезація процесів, забезпечили плавність роботи навіть на середньопотужних пристроях.

Однак у процесі розробки довелося зіткнутися з низкою труднощів. Найбільші проблеми викликала синхронізація фізичних параметрів об'єктів під час переходу між світами – це завдання вимагало перегляду початкового підходу до зберігання станів. Також знадобилася додаткова робота з оптимізації продуктивності, особливо у сценах із великою кількістю динамічних об'єктів.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

Книга:

1. Gamma E., Helm R., Johnson R., Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software. – Boston: Addison-Wesley, 1994. – 395 с.
2. Alexandrescu A. Modern C++ Design: Generic Programming and Design Patterns Applied. – Boston: Addison-Wesley, 2001. – 352 с.
3. Norman D. The Design of Everyday Things. – New York: Basic Books, 2013. – 368 с.
4. ISO/IEC 9126-1:2001. Software engineering – Product quality – Part 1: Quality model. – Женева: ISO, 2001. – 46 с..
5. Schell J. The Art of Game Design: A Book of Lenses. – Boca Raton: CRC Press, 2008. – 512 с.
6. Togelius J., Yannakakis G. N. Artificial Intelligence and Games. – Cham: Springer, 2018. – 385 с.
7. Bates B. Game Design. – Boston: Thomson Course Technology, 2004. – 400 с.
8. Новіков А.М. Методологія наукових досліджень. – М.: Синтез, 2006. – 268 с.
9. Nacke L. Games User Research: A Case Study Approach. – CRC Press, 2021. – 222 с.
10. Salen K., Zimmerman E. Rules of Play: Game Design Fundamentals. – MIT Press, 2003. – 688 с.

Web-сторінки:

11. Unity Technologies. Unity Documentation. – Режим доступу: <https://docs.unity3d.com>
12. Godot Engine. Godot Documentation. – Режим доступу: <https://docs.godotengine.org>
13. Game Developer Magazine. – Режим доступу: <https://www.gamedeveloper.com>

14. Unity Learn. Performance Optimization in Unity. – Режим доступа:
<https://learn.unity.com>
15. NVIDIA. Best Practices for Real-Time Ray Tracing. – Режим доступа:
<https://developer.nvidia.com>

ДОДАТОК А

Реалізація WorldSwitcher

```

using System.Collections.Generic;
using UnityEngine;

public class WorldSwitcher : MonoBehaviour
{
    public GameObject realWorld;
    public GameObject echoWorld;
    public EchoWorldCutscene cutsceneManager;
    private bool cutscenePlayed = false;
    public GameObject echoPostProcessingVolume;
    public GameObject activateBridge;
    public static bool IsInEchoWorld { get; private set; }
    private bool isEchoWorld = false;
    private List<EchoObject> echoObjects = new
List<EchoObject>();
    void Start()
    {

echoObjects.AddRange(echoWorld.GetComponentsInChildren<EchoObject>(true));

        IsInEchoWorld = isEchoWorld;
    }
    public void ForceSwitchToRealWorld()
    {
        isEchoWorld = false;
        IsInEchoWorld = false;
        realWorld.SetActive(true);
        echoWorld.SetActive(false);
        echoPostProcessingVolume.SetActive(false);
        foreach (var echo in echoObjects)
        {

            if (echo.isDiscovered)
            {
                echo.gameObject.SetActive(true);
                echo.MakeReal();

echo.transform.SetParent(realWorld.transform);
            }
            else
            {

```

```

        echo.gameObject.SetActive(false);
    }
}
}
public void ActivateBridge()
{
    activateBridge.SetActive(true);
}
void Update()
{
    if (Input.GetKeyDown(KeyCode.E))
    {
        isEchoWorld = !isEchoWorld;
        IsInEchoWorld = isEchoWorld;
        realWorld.SetActive(!isEchoWorld);
        echoWorld.SetActive(isEchoWorld);
        echoPostProcessingVolume.SetActive(isEchoWorld);
        if (!isEchoWorld)
        {
            foreach (var echo in echoObjects)
            {
                if (echo.isDiscovered)
                {
                    echo.gameObject.SetActive(true);
                    echo.MakeReal();
                }
            }
        }
        echo.transform.SetParent(realWorld.transform);
    }
    else
    {
        foreach (var echo in echoObjects)
        {
            if (!echo.isDiscovered)
            {
                echo.gameObject.SetActive(true);
                echo.SetPhantom();
            }
        }
        echo.transform.SetParent(echoWorld.transform);
    }
}

```

```
        }  
    if (!cutscenePlayed && cutsceneManager != null)  
    {  
        cutsceneManager.TriggerCutscene();  
        cutscenePlayed = true;  
    }  
}  
}  
}
```