

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук  
(повна назва)

Кафедра Інформаційних управляючих систем  
(повна назва)

## КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження методів та моделей ІТ-проєкту розробки ігрового додатку  
(тема)

Виконав:  
студент 2 курсу, групи УПГІТм-22-3

Трофімець Ігор Михайлович  
(прізвище, ім'я, по батькові)

Спеціальність 122 Комп'ютерні науки  
(код і повна назва спеціальності)


Тип програми освітньо-наукова  
(освітньо-професійна або освітньо-наукова)

Освітня програма Управління проєктами в галузі інформаційних технологій  
(повна назва освітньої програми)

Керівник професор кафедри Віктор ЛЕВИКІН  
(посада, власне ім'я, прізвище)

Допускається до захисту

Зав. кафедри  
(підпис)

  
(власне ім'я, прізвище)


Костянтин ПЕТРОВ

2024 р.

## Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук  
 Кафедра Інформаційних управляючих систем  
 Рівень вищої освіти другий (магістерський)  
 Спеціальність 122 Комп'ютерні науки  
 (код і повна назва)  
 Тип програми освітньо-наукова  
 (освітньо-професійна або освітньо-наукова)  
 Освітня програма Управління проектами в галузі інформаційних технологій  
 (повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри   
 (підпис)  
 « 01 » квітня 20 24 р.

### ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Трофімцю Ігорю Михайловичу  
 (прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів та моделей ІТ-проекта розробки ігрового додатку

затверджена наказом університету від 01 квітня 2024 р. № 258 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 08 06 2024 р.


3. Вихідні дані до роботи Організаційна, функціональна структура компанії, вимоги до додатку


4. Перелік питань, що потрібно опрацювати в роботі   
- Аналіз існуючих методів та моделей управління ІТ-проектами, зокрема тих, що використовуються в розробці ігрових додатків.  
- Вибір оптимальних методів для розробки ігрового додатку з урахуванням специфічних вимог та особливостей галузі.  
- Розробка детальної моделі управління ІТ-проектом, яка буде адаптована для створення ігрового додатку.  
- Запропонування напрямків подальших досліджень

## КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Вибір теми кваліфікаційної роботи	01.04.2024	Виконано
2	Затвердження плану і завдання кваліфікаційної роботи	01.04.2024	Виконано
3	Аналіз завдання, пошук та аналіз літературних джерел за темою роботи	06.04.2024- 13.04.2024	Виконано
4	Виконання кваліфікаційної роботи	14.04.2024-01.05.2024	Виконано
5	Оформлення пояснювальної записки	02.05.2024-19.05.2024	Виконано
6	Здача на перевірку та підпис кваліфікаційної роботи керівнику	20.05.24	Виконано
7	Захист кваліфікаційної роботи в екзаменаційній	10.06.24	Виконано

Дата видачі завдання 01 квітня 2024 р.

Студент  \_\_\_\_\_  
(підпис)

Керівник роботи  \_\_\_\_\_ д.т.н., проф. каф. ІУС Віктор ЛЕВИКІН.  
(підпис) (посада, власне ім'я, прізвище)

## РЕФЕРАТ

Пояснювальна записка до роботи містить: 70 стор., 6 рис., 8 табл. 20 джерел, 2 додатка

ГІБРИДНІ МЕТОДОЛОГІЇ, ІГРОВИЙ ДОДАТОК, AGILE, SSPM, SCRUM.

Об'єктом дослідження є методи та моделі розробки ІТ-проекта. Предмет дослідження – існуючі методи та моделі розробки ігрових додатків. Мета роботи - дослідження існуючих методів та моделей управління ІТ-проектами, їх адаптації та застосуванні для розробки ігрового додатку.

У роботі виконано:

- формування проблем і задач розробки ігрових додатків;
- аналіз існуючих методів та моделей управління ІТ-проектами, зокрема тих, що використовуються в розробці ігрових додатків;
- запропонування напрямків подальших досліджень.

Наукова новизна цієї роботи полягає в комплексному аналізі існуючих методів розробки ігрових додатків у цій сфері. На основі цього аналізу пропонуються методики та стратегії, які можуть підвищити ефективність та якість розробки ігрових додатків.

Практичне значення отриманих результатів цієї роботи – можуть бути корисними для розробників ігор, менеджерів проектів та інших професіоналів у галузі. Таким чином, результати цього дослідження сприяють вдосконаленню процесу розробки ігрових додатків та можуть стати основою для подальших інновацій у цій сфері.

## ABSTRACT

The explanatory note to the work contains: 58 pages, 6 figures, 8 tables, and 20 sources, 2 attachments.

**HYBRID METHODOLOGIES, GAME APPLICATION, AGILE, CCPM, SCRUM.**

The object of the research is the methods and models of IT project development. The subject of the research is the existing methods and models of game application development. The aim of the work is to study the existing methods and models of IT project management, their adaptation, and application for the development of a game application.

- Formulation of problems and tasks in game application development;
- Analysis of existing methods and models of IT project management, particularly those used in game application development;
- Proposing directions for further research.

The scientific novelty of this work lies in the comprehensive analysis of existing game application development methods in this field. Based on this analysis, methodologies and strategies are proposed that can enhance the efficiency and quality of game application development.

The practical significance of the obtained results can be useful for game developers, project managers, and other professionals in the field. Thus, the results of this research contribute to the improvement of the game application development process and may serve as a foundation for further innovations in this area.

## ЗМІСТ

Скорочення та умовні позначки .....	8
Вступ.....	9
1 Формування проблем та задач розробки ігрового додатку .....	11
1.1 Формування проблем розробки ігрових додатків та існуючі інструменти їх вирішення.....	11
1.2 Формування завдання щодо вдосконалення процесів розробки ігрового додатку.....	13
2 Аналіз існуючих методів і моделей розробки ігрового додатку .....	16
2.1 Огляд сучасних методів планування, розробки та ведення проєктів в контексті ігрової індустрії.....	16
2.1.1 Методології Agile .....	16
2.1.2 Водоспадна модель (Waterfall).....	17
2.1.3 Гібридні методології .....	18
2.1.4 Методологія критичного ланцюга .....	18
2.1.5 Опис існуючих інструментів для управління проєктами в ігровій індустрії.....	19
2.2 Розробка прототипів каркасів та системи ідентифікації та оцінки ризиків .....	20
2.3 Розробка системи ідентифікації та оцінки ризиків .....	21
3 Вибір методології розробки ігрового додатку .....	24
3.1 Збір вимог до ігрового додатку .....	24

3.1.1 Створення діаграм користувацьких сценаріїв .....	27
3.1.2 Розробка каркасів (Wireframes).....	28
3.2 Опис проєкту сформований на основі вимог .....	33
3.3 Обґрунтування вибору Agile для основної розробки.....	35
3.4 Планування проєкту за допомогою методології Agile.....	36
3.5 Обґрунтування вибору методу критичного ланцюга для портування гри .....	40
3.6 Планування за методом критичного ланцюга .....	42
3.7 Управління ризиками .....	44
3.8 Управління ризиками за допомогою методу критичного ланцюга .....	50
4 Аналіз отриманих результатів та перспективи подальших досліджень.....	52
4.1 Аналіз результатів дослідження.....	52
4.2 Перспективи подальших досліджень.....	53
Висновки .....	55
Перелік джерел посилання .....	57
Додаток А Загальна характеристика роботи .....	59
Додаток Б Графічні матеріали .....	61

## **СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ**

CCPM – Critical Chain Project Management

PS5 – PlayStation 5

## ВСТУП

Актуальність теми дослідження полягає в зростаючій популярності ігрових додатків у сучасному світі, що створює потребу в ефективних методах та моделях розробки таких проєктів. Ринок ігрових додатків постійно розширюється, стимулюючи інновації та розвиток нових підходів до управління ІТ-проєктами. У зв'язку з цим, дослідження методів та моделей, що сприяють ефективній розробці ігрових додатків, є надзвичайно актуальним та важливим для забезпечення конкурентоспроможності на ринку[21].

Мета даної кваліфікаційної роботи полягає в дослідженні існуючих методів та моделей управління ІТ-проєктами, їх адаптації та застосуванні для розробки ігрового додатку. В роботі будуть розглянуті як теоретичні аспекти, так і практичні рішення, що дозволять підвищити ефективність і якість розробки. Основною метою є створення комплексної моделі управління ІТ-проєктом, яка буде орієнтована на специфіку розробки ігрових додатків.

Задачі дослідження включають:

- Аналіз існуючих методів та моделей управління ІТ-проєктами, зокрема тих, що використовуються в розробці ігрових додатків;
- Вибір оптимальних методів для розробки ігрового додатку з урахуванням специфічних вимог та особливостей галузі;
- Розробка детальної моделі управління ІТ-проєктом, яка буде адаптована для створення ігрового додатку;
- Запропонування напрямків подальших досліджень.

Оцінка сучасного стану об'єкта дослідження включає аналіз існуючих підходів до управління ІТ-проєктами в галузі розробки ігрових додатків. В роботі буде здійснено огляд основних напрямів теоретичних та прикладних робіт у цій сфері, що дозволить визначити основні тенденції та проблеми, з якими стикаються розробники ігрових додатків[21].

Світові тенденції розв'язання поставлених проблем включають впровадження нових технологій та інноваційних підходів до управління проєктами, що забезпечують скорочення термінів розробки та підвищення якості кінцевого продукту. Важливим аспектом є також інтеграція сучасних інструментів та методик, що сприяють автоматизації процесів та покращенню комунікації в команді розробників[21].

Необхідність проведення цього дослідження обумовлена стрімким розвитком індустрії ігрових додатків та високою конкуренцією на ринку, що вимагає ефективного управління проєктами для досягнення успіху. Враховуючи швидкі зміни в технологічному середовищі, важливо розробляти та впроваджувати новітні підходи до управління ІТ-проєктами, які відповідають сучасним вимогам та тенденціям[21].

В результаті виконання роботи очікується процес планування, та методика формування чіткого бачення кінцевого ігрового додатку яка буде адаптована для специфічних вимог розробки ігрових додатків та сприятиме підвищенню ефективності процесу розробки. Ці вимоги будуть враховувати всі етапи життєвого циклу проєкту, від початкового планування до фінальної реалізації та підтримки продукту[21].

# 1 ФОРМУВАННЯ ПРОБЛЕМ ТА ЗАДАЧ РОЗРОБКИ ІГРОВОГО ДОДАТКУ

## 1.1 Формування проблем розробки ігрових додатків та існуючі інструменти їх вирішення

Розробка ігрових додатків є складним і багатогранним процесом, що включає етапи планування, розробки та ведення проєктів. Успішне виконання цих етапів вимагає застосування новітніх технологій, які здатні вдосконалити об'єкти та процеси, забезпечуючи високу якість кінцевого продукту та ефективність роботи команди розробників. У цьому розділі буде розглянуто основні проблеми, що виникають на кожному з етапів розробки ігрових додатків[21].

Роздивимось недостатню точність оцінки ресурсів. Однією з основних проблем на етапі планування є недостатня точність оцінки ресурсів, необхідних для виконання проєкту. Часто команди розробників недооцінюють або переоцінюють обсяги роботи, що призводить до затримок у виконанні завдань та перевищення бюджету. Недостатня точність оцінки ресурсів також може спричиняти недооцінку часу, необхідного для виконання окремих етапів проєкту, що ускладнює загальне управління проєктом.

Інструменти, такі як Microsoft Project та JIRA, дозволяють аналізувати попередні проєкти та використовувати ці дані для точнішого планування. Застосування методів машинного навчання для аналізу великих обсягів даних і побудови моделей прогнозування також підвищує точність оцінок [4].

Роздивимось відсутність чіткого бачення кінцевого продукту. Часто виникає проблема відсутності чіткого бачення кінцевого продукту, що ускладнює процес планування та призводить до неузгодженостей у вимогах. Це може спричиняти конфлікти між різними зацікавленими сторонами,

затримки у прийнятті рішень та необхідність повторних змін у проєкті на пізніших етапах. Відсутність чіткого бачення також впливає на здатність команди ефективно планувати та пріоритизувати завдання [10].

Існуючі методи вирішення цієї проблеми включають прототипування та візуалізацію. Інструменти для створення прототипів, такі як Figma та Sketch, дозволяють створювати візуальні представлення кінцевого продукту, що допомагає формувати єдине бачення проєкту серед усіх зацікавлених сторін. Проведення воркшопів та брейнстормінг-сесій також сприяє кращому розумінню вимог та очікувань від проєкту [4].

Роздивимось управління командою розробників. Управління командою розробників може бути складним завданням, особливо в умовах віддаленої роботи та великої кількості учасників. Це включає координацію завдань, підтримку комунікації та мотивації, а також управління конфліктами всередині команди. Недостатня комунікація може призводити до непорозумінь та помилок, а погане управління завданнями — до затримок та перевищення термінів виконання проєкту [4].

Існуючі методи вирішення цієї проблеми включають застосування гнучких методологій управління, таких як Agile. Ця методологія дозволяє ефективно планувати, виконувати та контролювати завдання, забезпечуючи постійний зворотний зв'язок та швидку адаптацію до змін. Інструменти для управління проєктами та комунікації, такі як Trello і Slack, покращують координацію та комунікацію між учасниками проєкту [4].

Роздивимось управління ризиками. Управління ризиками є важливою складовою успішного ведення проєкту. Це включає ідентифікацію потенційних ризиків, оцінку їх впливу та розробку планів з їх мінімізації. Недостатня увага до управління ризиками може призводити до серйозних проблем на пізніших етапах проєкту, таких як затримки, перевищення бюджету або навіть невдача проєкту. Важливо регулярно проводити аналіз ризиків та оновлювати плани їх управління [4].

Регулярні аналізи ризиків та оновлення планів управління ризиками допомагають мінімізувати негативні впливи на проєкт та забезпечити його успішну реалізацію [4].

Аналіз існуючих методів вирішення проблем у розробці ігрових додатків показує, що для досягнення успіху необхідно зосередитися на вдосконаленні процесів планування, розробки та ведення проєктів. Враховуючи специфіку розробки ігрових додатків, метою вдосконалення є підвищення точності планування ресурсів, забезпечення високої якості коду, покращення управління командами розробників та ефективне управління ризиками. Основні напрями вдосконалення включають впровадження інструментів для аналізу даних та прогнозування ресурсів, використання методів прототипування та візуалізації для формування чіткого бачення кінцевого продукту, застосування практик безперервної інтеграції та безперервного розгортання для забезпечення якості та продуктивності коду, використання гнучких методологій управління проєктами для покращення координації та комунікації всередині команди, а також впровадження програмного забезпечення для управління ризиками та регулярних аналізів ризиків. Ці напрями вдосконалення дозволять значно покращити процеси розробки ігрових додатків, забезпечуючи їх високу якість, ефективність та конкурентоспроможність на ринку[21].

## 1.2 Формування завдання щодо вдосконалення процесів розробки ігрового додатку

Розробка ігрових додатків є складним і багатогранним процесом, який включає кілька етапів: планування, розробку та ведення проєктів. Для

забезпечення високої якості кінцевого продукту та ефективної роботи команди розробників необхідно вдосконалювати ці процеси за допомогою новітніх технологій. У цьому завданні буде розглянуто процеси вдосконалення розробки ігрових додатків шляхом застосування інноваційних методів та інструментів[21].

Провести аналіз існуючих методів та інструментів, включаючи огляд сучасних методів планування, розробки та ведення проєктів в контексті ігрової індустрії, а також дослідити існуючі інструменти для управління проєктами, зокрема для ігрової індустрії.

Розробити методикку вдосконалення процесів планування, запропонувати підходи для формування чіткого бачення кінцевого продукту за допомогою прототипів та візуалізації.

Розглянути управління ризиками, розробити систему ідентифікації та оцінки ризиків, впровадити методи регулярного аналізу ризиків та оновлення планів їх управління.

Очікувані результати включають формування методів планування ресурсів, впровадження інструментів для аналізу даних та прогнозування ресурсів, що дозволить точніше планувати використання ресурсів та уникати перевитрат. Також розробка методів формування чіткого бачення кінцевого продукту з використанням прототипів та візуалізації допоможе зменшити кількість змін на пізніх етапах розробки та забезпечити узгодженість вимог. Нарешті, розробка та впровадження системи управління ризиками дозволить мінімізувати негативні впливи на проєкт та забезпечити його успішну реалізацію.

Завдання щодо вдосконалення процесів розробки ігрового додатку передбачає аналіз існуючих методів, розробку нових підходів та впровадження новітніх технологій для підвищення ефективності та якості розробки. Виконання цих задач забезпечить успішну реалізацію проєктів

ігрових додатків, підвищить їх конкурентоспроможність та задовольнить вимоги користувачів.

## 2 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ І МОДЕЛЕЙ РОЗРОБКИ ІГРОВОГО ДОДАТКУ

Розробка ігрових додатків є складним процесом, що потребує ефективного планування, управління та виконання. Сучасні методи та інструменти управління проектами дозволяють підвищити ефективність цих процесів та забезпечити високу якість кінцевого продукту. У цьому розділі буде проведено огляд сучасних методів планування, розробки та ведення проектів в контексті ігрової індустрії, а також дослідження існуючих інструментів для управління проектами[21].

### 2.1 Огляд сучасних методів планування, розробки та ведення проектів в контексті ігрової індустрії

#### 2.1.1 Методології Agile

Методології Agile стали одними з найпопулярніших підходів у розробці програмного забезпечення, включаючи ігрові додатки. Agile підходи передбачають ітеративний процес розробки, де проєкт розбивається на короткі цикли (ітерації або спринти), кожен з яких включає планування, виконання, тестування та оцінку результатів. Основні принципи Agile включають гнучкість, адаптивність до змін, тісну співпрацю з замовником та регулярне отримання зворотного зв'язку [1].

Основні методології Agile, що застосовуються в ігровій індустрії, включають Scrum і Kanban. Scrum є однією з найпоширеніших методологій Agile, яка включає регулярні спринти, щоденні зустрічі (Daily Stand-ups),

огляди спринтів (Sprint Reviews) та ретроспективи (Sprint Retrospectives). Ця методологія дозволяє командам швидко адаптуватися до змін та забезпечує високу прозорість процесу розробки. Kanban, в свою чергу, передбачає візуалізацію процесу розробки за допомогою дошки Kanban, на якій відображаються завдання та їх стан. Цей підхід дозволяє контролювати потік роботи та забезпечувати рівномірне навантаження на команду.

### 2.1.2 Водоспадна модель (Waterfall)

Водоспадна модель є традиційним підходом до управління проєктами, що передбачає послідовне виконання фаз проєкту: аналіз вимог, проєктування, розробка, тестування, впровадження та підтримка. Кожна фаза повинна бути завершена перед переходом до наступної[10].

Водоспадна модель має свої основні переваги та недоліки. Переваги включають чітку структуру та зрозумілість процесів, а також легкість у плануванні та контролі. Однак, ця модель також має недоліки, серед яких низька гнучкість, висока ймовірність накопичення помилок до пізніх стадій проєкту, і труднощі з адаптацією до змін.

### 2.1.3 Гібридні методології

Гібридні методології поєднують елементи водоспадної моделі та Agile підходів, дозволяючи зберегти структуру та контроль, притаманний водоспадній моделі, з одночасним впровадженням гнучкості та адаптивності Agile. Це дозволяє адаптувати процес розробки до специфічних вимог проєкту та забезпечити ефективне управління змінами.

### 2.1.4 Методологія критичного ланцюга

Методологія критичного ланцюга (Critical Chain Project Management, CCPM) є підходом до управління проєктами, розробленим Еліяху Голдраттом у рамках теорії обмежень (Theory of Constraints, TOC). CCPM зосереджується на ідентифікації та управлінні критичними ресурсами, які впливають на терміни виконання проєкту, та мінімізації простоїв і збоїв [3].

Основні принципи CCPM полягають у визначенні послідовності завдань, які обмежують загальну тривалість проєкту (ідентифікація критичного ланцюга). Також важливим аспектом є додавання буферів часу для захисту критичного ланцюга від затримок. Існують три типи буферів: проєктний буфер, який захищає кінцевий термін проєкту; буфер живлення, який захищає критичний ланцюг від затримок у не-критичних завданнях; та буфер ресурсів, який захищає від збоїв у роботі ресурсів. Оптимізація використання ресурсів для забезпечення безперервності роботи над критичним ланцюгом також є ключовим принципом CCPM [3].

Основні переваги ССРМ включають зменшення тривалості проєкту, підвищення ефективності використання ресурсів і зниження ймовірності затримок. Однак, цей підхід вимагає ретельного планування та контролю, і може бути складним для впровадження в умовах високої невизначеності.

### 2.1.5 Опис існуючих інструментів для управління проєктами в ігровій індустрії

JIRA є одним з найпоширеніших інструментів для управління проєктами, особливо в Agile-середовищі. Він дозволяє створювати та відстежувати завдання, проводити спринти, керувати беклогом та надавати звіти про прогрес проєкту. JIRA також підтримує інтеграцію з іншими інструментами, такими як Confluence та Bitbucket[9].

Trello є простим та зручним інструментом для управління завданнями за допомогою дошок Kanban. Він дозволяє створювати картки завдань, розміщувати їх на дошці та відстежувати їхній стан. Trello підходить для невеликих команд та проєктів з невеликою кількістю завдань.

Asana є потужним інструментом для управління проєктами, що дозволяє створювати завдання, призначати їх учасникам команди, встановлювати дедлайни та відстежувати прогрес. Asana підтримує різні види візуалізації, включаючи дошки Kanban, списки завдань та календарі.

## 2.2 Розробка прототипів каркасів та системи ідентифікації та оцінки ризиків

Розробка прототипів каркасів включає кілька ключових етапів. Спочатку процес починається зі збору вимог за допомогою різних методів, таких як інтерв'ю з зацікавленими сторонами, опитування та воркшопи. Цей етап є важливим, оскільки допомагає визначити основні функціональні та нефункціональні вимоги до ігрового додатку. Після того, як вимоги стають зрозумілими, команда дизайнерів починає створювати початкові ескізи та низькодеталізовані каркаси[21].

Протягом усього процесу команда дизайнерів стежить за тим, щоб каркаси відповідали цілям проєкту, приділяючи особливу увагу зручності використання, функціональності та користувацькому досвіду.

Після кількох ітерацій та доопрацювань каркаси трансформуються у високодеталізовані прототипи. Ці прототипи включають більш детальні елементи дизайну, такі як конкретні компоненти інтерфейсу користувача та інтеракції, і часто створюються за допомогою спеціалізованих інструментів для прототипування.

Створення діаграм користувацьких сценаріїв допомагає візуалізувати шляхи користувачів через продукт, включаючи всі точки взаємодії та можливі варіанти дій. Це дозволяє зрозуміти, як користувачі взаємодіють з різними частинами продукту та виявити можливі проблеми на ранніх етапах. Інструменти, такі як Miro, Lucidchart або Figma, можуть бути використані для створення таких діаграм. Аналіз користувацьких сценаріїв включає виявлення можливих проблем та точок болю (pain points), що дозволяє усунути їх ще до початку розробки[12].

### 2.3 Розробка системи ідентифікації та оцінки ризиків

Розробка системи ідентифікації та оцінки ризиків для проєкту ігрового додатку включає кілька важливих етапів. Спочатку необхідно ідентифікувати потенційні ризики, оцінити їх вплив та ймовірність, а також розробити плани управління цими ризиками[21].

Першим кроком є збір даних. Це включає проведення інтерв'ю з ключовими членами команди, такими як програмісти, тестувальники, дизайнери, менеджери проєктів та інші учасники. Важливо ставити питання, що допоможуть виявити потенційні проблеми, з якими вони стикалися в попередніх проєктах або які бачать у поточному. Аналіз попередніх проєктів дозволяє знайти схожі ситуації, в яких виникали ризики, і використовувати набуті знання для ідентифікації потенційних ризиків у поточному проєкті. Також необхідно проаналізувати технічну та управлінську документацію, а також плани проєкту, щоб виявити можливі ризики, пов'язані з невідповідностями у вимогах, розподілом ресурсів та часовими рамками.

Після збору даних важливо категоризувати ризики, щоб зрозуміти їхню природу та можливі наслідки. Технічні ризики можуть включати проблеми з платформою, на якій працює гра, або технічні обмеження обладнання. Управлінські ризики часто стосуються питань планування та бюджету. Операційні ризики охоплюють проблеми з розподілом ресурсів і кадрові питання. Зовнішні ризики можуть виникнути через зміни в законодавстві або ринкові умови.

Наступним кроком є оцінка ймовірності виникнення кожного ризику, що дозволить пріоритизувати їх та зосередити зусилля на найважливіших. Це можна зробити за допомогою шкали ймовірності, що складається з трьох рівнів: низький, середній та високий. Наприклад, ризик з низькою

ймовірністю має дуже малу ймовірність виникнення, однак його наслідки можуть бути значними. Ризик з середньою ймовірністю може виникнути, але його наслідки можуть бути помірними. Ризик з високою ймовірністю майже напевно виникне, і його наслідки можуть бути суттєвими.

Після оцінки ймовірності виникнення ризику важливо оцінити його вплив на проєкт. Це також можна зробити за допомогою шкали, що складається з трьох рівнів: низький, середній та високий. Наприклад, ризик з низьким впливом має незначний вплив на проєкт, і його наслідки можна легко усунути. Ризик із середнім впливом може суттєво вплинути на певні аспекти проєкту, але його наслідки можуть бути керованими. Ризик з високим впливом може мати серйозні наслідки для проєкту, що можуть призвести до значних затримок або додаткових витрат. Для полегшення оцінки створюється матриця ризиків, де кожен ризик розміщується за ймовірністю та впливом, що дозволяє візуально оцінити, які ризики є найбільш критичними та потребують негайної уваги.

Після оцінки ризиків необхідно розробити стратегії управління. Це включає кілька ключових підходів, таких як уникнення, пом'якшення, прийняття та передача. Уникнення полягає у зміні плану проєкту таким чином, щоб виключити ризик, наприклад, змінивши технологію або процес, що викликає ризик. Пом'якшення передбачає зменшення ймовірності або впливу ризику, наприклад, покращення процесів тестування або залучення додаткових ресурсів. Прийняття означає, що в деяких випадках ризик неможливо уникнути або пом'якшити, і його потрібно прийняти, готуючи план дій на випадок, якщо ризик виникне. Передача полягає у передачі ризику іншій стороні, наприклад, через страхування або аутсорсинг.

Для кожного ризику розробляються конкретні плани дій, що включають визначення конкретних заходів та призначення відповідальних осіб за їх виконання, що дозволяє забезпечити своєчасне та ефективне реагування на виникнення ризиків.

Для ефективного управління ризиками здійснюється постійний моніторинг, що включає регулярні зустрічі для перегляду та оновлення статусу ризиків, а також використання інструментів управління проєктами, таких як Jira, для відстеження ризиків. Плани управління ризиками повинні бути гнучкими та адаптивними, що дозволяє коригувати їх на основі нових даних та змін у проєкті, оновлюючи стратегії та плани дій, що забезпечує ефективне реагування на зміни та мінімізацію негативних наслідків для проєкту.

Ця система ідентифікації та оцінки ризиків дозволяє не лише виявляти та оцінювати ризики, але й розробляти ефективні стратегії управління ними, забезпечуючи успішне завершення проєкту.

### 3 ВИБІР МЕТОДОЛОГІЇ РОЗРОБКИ ІГРОВОГО ДОДАТКУ

#### 3.1 Збір вимог до ігрового додатку

Збір вимог є першим етапом у створенні прототипів і включає: визначення основних функціональних та нефункціональних вимог до гри. Для цього використовуються різні методи збору інформації.

Інтерв'ю з зацікавленими сторонами: Проведення інтерв'ю з користувачами, замовниками, гейм дизайнерами та іншими зацікавленими сторонами для визначення їхніх потреб і очікувань.

Ціль інтерв'ю: Визначити ключові ігрові механіки та особливості, які повинні бути включені в гру.

Питання для обговорення:

- які основні елементи геймплею ви бачите в цій грі;
- які нові функції або механіки можуть привернути увагу гравців;
- які аспекти дизайну повинні бути найбільш детально пророблені.

На основі відповідей гейм дизайнера було визначено, що гра повинна мати динамічні погодні умови, що впливають на геймплей, та систему модифікацій зброї, яка дозволяє гравцям налаштовувати свою зброю під свій стиль гри. Результати наведені у таблиці 3.1.

Таблиця 3.1 - Збір вимог до ігрового додатку

Питання	Відповідь
1	2
Основні елементи геймплею	Масштабні багатокористувацькі битви, динамічні погодні умови, руйнування об'єктів
Нові функції або механіки	Система модифікацій зброї, інтерактивні погодні умови, футуристичні транспортні засоби

Кінець таблиці 3.1

1	2
Детально пророблені аспекти	Графічне виконання динамічних погодних умов, баланс зброї та техніки, інтерфейс користувача

Опитування: Використання анкет та опитувань для збору вимог від великої кількості гравців.

Ціль опитування: Збір інформації про вподобання та очікування гравців щодо нових ігрових функцій.

Анкета:

- які типи зброї ви найчастіше використовуєте в іграх;
- які типи ігрових режимів вам цікаві;
- наскільки важливо для вас мати можливість налаштовувати персонажа та зброю.

Згідно з результатами опитування, більшість гравців віддає перевагу футуристичним видам зброї та режимам командної гри. Також велика кількість респондентів висловила бажання мати гнучкі можливості для налаштування персонажів і зброї. Результати наведені у таблиці 3.2

Таблиця 3.2 - Результати опитування

Питання	Відповіді (відсоток респондентів)
1	2
Типи зброї	Футуристична (70%), Класична (20%), Інші (10%)
Типи ігрових режимів	Командні битви (50%), Завоювання територій (30%), Інші (20%)
Важливість налаштування персонажа та зброї	Дуже важливо (60%), Важливо (30%), Неважливо (10%)

Воркшопи: Організація воркшопів з різними зацікавленими сторонами для колективного обговорення та узгодження вимог.

Ціль воркшопу: Колективне обговорення та узгодження вимог до основних ігрових механік та функцій.

Учасники: Геймдизайнери, розробники, представники маркетингу, потенційні гравці.

Тематика обговорення:

- основні ігрові механіки та їх реалізація;
- баланс між реалістичністю та футуристичними елементами;
- використання технологій для створення динамічних середовищ.

В результаті воркшопу було визначено, що гра повинна включати інтерактивні погодні умови, які змінюють тактичні переваги на полі бою, а також систему модифікацій зброї, яка дозволяє гравцям адаптувати своє обладнання під конкретні умови бою. Результати наведені у таблиці 3.3.

Таблиця 3.3 - Результати воркшопу

Тематика обговорення	Вирішення
1	2
Основні ігрові механіки	Динамічні погодні умови, масштабні багатокористувацькі битви
Баланс між реалістичністю та футуристичними елементами	Використання реалістичної фізики з футуристичними елементами дизайну та зброї
Використання технологій для створення динамічних середовищ	Впровадження системи руйнування об'єктів та змінюваних погодних умов

Ці приклади демонструють, як різні методи збору вимог можуть бути ефективно використані для визначення ключових аспектів гри та задоволення очікувань зацікавлених сторін.

### 3.1.1 Створення діаграм користувацьких сценаріїв

Діаграми користувацьких сценаріїв допомагають візуалізувати шляхи гравців через гру, включаючи всі точки взаємодії та можливі варіанти дій. Це дозволяє зрозуміти, як гравці взаємодіють з різними частинами гри та виявити можливі проблеми на ранніх етапах. Можна використати інструменти, такі як Miro, Lucidchart або Figma для створення діаграм користувацьких сценаріїв.

Для створення діаграми користувацьких сценаріїв для трьох основних екранів, почнемо з головного меню. Користувач заходить у гру і потрапляє на головне меню, де йому пропонується кілька опцій. Вибравши "Грати", він переходить на екран вибору персонажа. Інші опції включають: "Налаштування", "Мультиплеєр", "Профіль" та "Вихід", кожна з яких веде до відповідних екранів або функцій.

Після натискання "Грати" користувач потрапляє на екран вибору персонажа. Тут він може вибрати персонажа зі списку доступних, що відображаються з візуальними прев'ю та характеристиками. Вибравши персонажа, користувач переходить на екран налаштування зброї. Якщо він вирішить повернутися, то може натиснути кнопку для повернення до головного меню.

На екрані налаштування зброї користувач бачить детальну 3D-модель зброї з можливістю змінювати її частини, такі як приціл, приклад та ствол. Після внесення змін він може застосувати модифікації і перейти до початку гри або повернутися до вибору персонажа для подальших налаштувань. Таким чином, користувачеві надається гнучкість у налаштуванні гри під свої потреби та стиль гри.

Цей вибір елементів забезпечує інтуїтивний і зручний шлях для користувача, дозволяючи йому легко орієнтуватися між основними

функціями гри та налаштуваннями. Кожен елемент екрана має своє призначення і сприяє поліпшенню загального користувацького досвіду.

Діаграма користувацьких сценаріїв зображена на рисунку 3.1

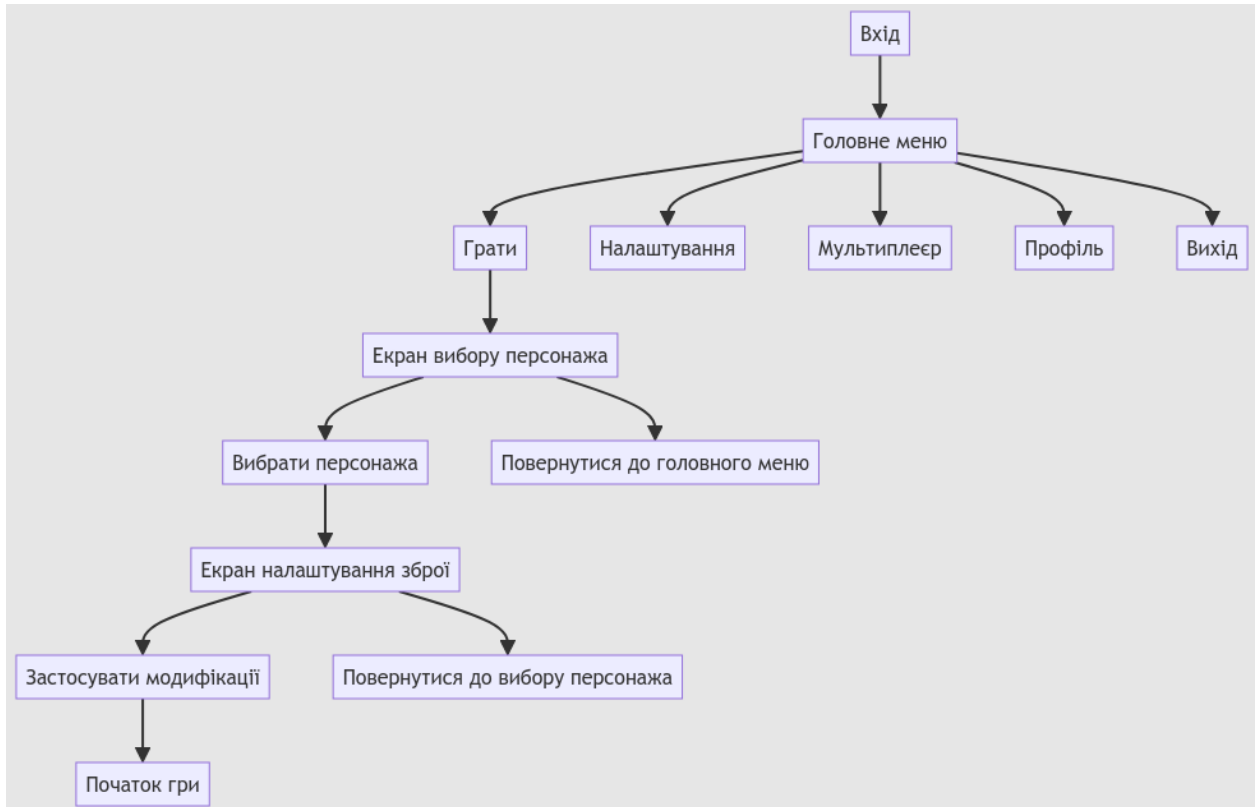


Рисунок 3.1 - Діаграма користувацьких сценаріїв

### 3.1.2 Розробка каркасів (Wireframes)

На основі цих вимог та діаграми користувацьких сценаріїв розробляються каркаси (wireframes), які представляють структуру інтерфейсу гри. Каркаси допомагають визначити розташування основних елементів інтерфейсу та їх взаємодію. Можна використовувати інструменти, такі як

Balsamiq, Figma або Sketch для створення каркасів. На основі попередніх вимог розглянемо декілька екранів.

Розглянемо головне меню:

Фон - футуристичний міський пейзаж або поле бою, що створює атмосферу гри з самого початку, занурюючи гравця у футуристичний світ. Для цього потрібна висока якість графіки та динамічні елементи, як-от рухомі транспортні засоби чи погодні умови.

Кнопка "Грати" є основною, оскільки вона дозволяє розпочати гру. Вона повинна виділятися серед інших кнопок і бути легкою для натискання. Кнопка "Налаштування" надає доступ до налаштувань гри (графіка, звук, управління), що дозволяє гравцям налаштовувати гру під свої потреби для кращого досвіду. Меню повинно бути інтуїтивним і простим.

Кнопка "Мультиплеєр" забезпечує перехід до багатокористувацьких режимів гри, що дозволяє гравцям швидко знаходити та приєднуватися до онлайн-матчів. Важливо забезпечити легкий доступ та видимість поточного статусу сервера або матчів.

Кнопка "Профіль" надає доступ до профілю гравця (досягнення, статистика), що дозволяє переглядати свій прогрес і досягнення. Інтерфейс повинен бути інтуїтивно зрозумілим та забезпечувати легкість навігації.

Кнопка "Вихід" закриває гру, дозволяючи завершити ігрову сесію. Кнопка повинна бути чітко видимою з підтвердженням виходу.

Верхній бар включає логотип гри, налаштування звуку та мови, повідомлення. Це забезпечує постійний доступ до важливих налаштувань і повідомлень. Розташування повинно бути зручним та легко доступним.

Результат реалізації представлений на рисунку 3.2

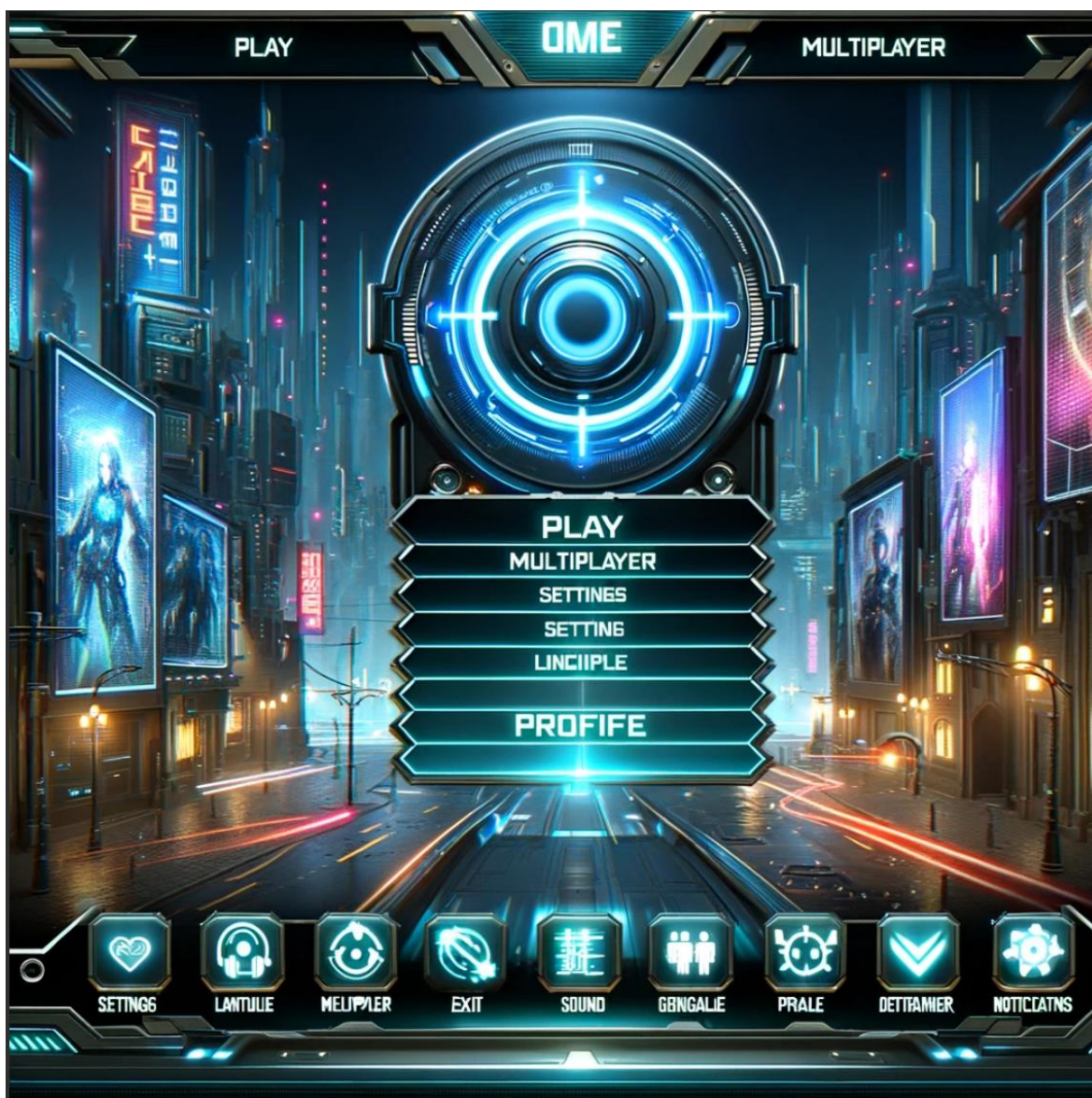


Рисунок 3.2 - Головне меню

Розглянемо екран вибору персонажа

Список доступних персонажів з їх характеристиками та особливими навичками важливий для стратегічного вибору гравців залежно від їх стилю гри. Необхідно забезпечити детальні описи персонажів, візуальні прев'ю та інтуїтивний інтерфейс для вибору.

Кнопка "Вибрати персонажа" повинна дозволяти легко і швидко підтвердити вибір. Вона має бути зрозумілою та видимою. Кнопка "Повернутися до головного меню" забезпечує можливість відміни вибору та

повернення до попереднього екрану без збереження змін. Вона повинна бути легко доступною та видимою.

Фон, який демонструє сцени з різних континентів, де відбуваються битви, створює атмосферу та контекст для персонажів. Необхідна висока якість зображення та відповідність тематиці гри.

Результат реалізації представлений на рисунку 3.3



Рисунок 3.3 - Екран вибору персонажа

Розглянемо екран налаштування зброї

Список доступної зброї з можливістю її модифікації дозволяє гравцям вибирати зброю, яка підходить їхньому стилю гри. Інтерфейс повинен бути інтуїтивним, забезпечуючи легкість навігації.

Опції для зміни прицілу, прикладу, ствола тощо дозволяють гравцям налаштовувати зброю для підвищення ефективності. Необхідно забезпечити детальні описи модифікацій, візуальні прев'ю та легкість застосування змін.

Кнопка "Застосувати модифікації" дозволяє зберегти зміни та повернутися до вибору персонажа. Вона повинна бути видимою та легкою для натискання. Кнопка "Повернутися до вибору персонажа" дозволяє відмінити зміни та повернутися без збереження змін. Вона повинна бути легко доступною та видимою.

Фон у вигляді футуристичного арсеналу відповідає тематиці гри та створює відповідну атмосферу. Необхідна висока якість графіки та деталізація.

Результат реалізації представлений на рисунку 3.4

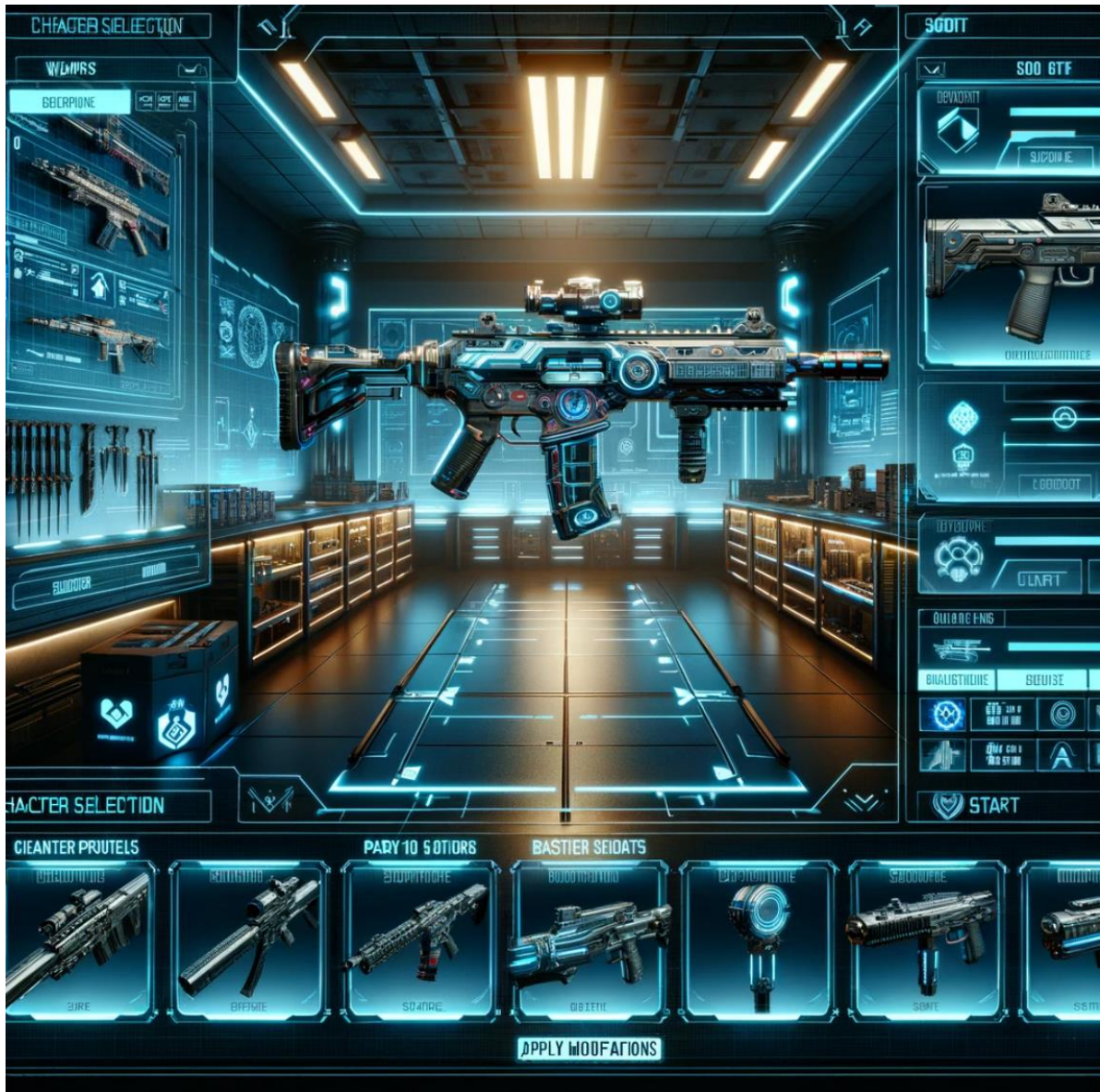


Рисунок 3.4 - Экран налаштування зброї

### 3.2 Опис проекту сформований на основі вимог

Проект полягає в розробці інноваційного шутера від першої особи, створеного компанією. Гра переносить гравців у футуристичний світ, де

технології та війна злилися в одне ціле, створюючи епічні поля битв на різних континентах.

Ігровий процес включає масштабні багатокористувацькі битви до 128 гравців на одній карті, що дозволяє відчувати справжній хаос і напруження сучасної війни. Кожна битва може змінюватися під впливом динамічних погодних умов, таких як бурі, торнадо та піщані бурі. Гравці повинні адаптуватися до мінливих умов на полі бою, використовуючи руйнування об'єктів для створення нових тактичних переваг. У грі представлено широкий арсенал футуристичної зброї та транспортних засобів, включаючи дрони, мехи та гіперзвукові літаки. Гравці можуть модифікувати свою зброю та обладнання, щоб адаптуватися до різних стилів гри. Гра представлена унікальними персонажами, кожен з яких має свої особливі навички та обладнання. Гравці можуть вибирати персонажів, щоб створити ідеальну команду для виконання місії.

Події розгортаються в недалекому майбутньому, де світова стабільність опинилася під загрозою через конфлікти між великими корпораціями та державами. Гравці занурюються у світ геополітичних інтриг і техногенних катастроф, де кожне рішення може змінити хід історії. Гра пропонує різноманітні мультиплеєрні режими, включаючи класичні командні битви, завоювання територій та нові, інноваційні режими гри, що дозволяють гравцям змагатися та співпрацювати у різних умовах.

Технологічний стек та інфраструктура проєкту включають Unreal Engine, який використовується для розробки гри на ПК та консолі PS5. Портування на консоль відбувається паралельно розробці на ПК, що вносить свої складнощі. TeamCity використовується для безперервної інтеграції та доставки. Perforce застосовується для управління версіями, а Swarm інтегрується з ним для перегляду змін. Для управління проєктами використовується Jira, а Confluence служить для ведення проєктної документації.

Взаємодія технологічного стеку забезпечується за допомогою кількох важливих процесів. Unreal Engine використовується для розробки основного ігрового коду та візуальних елементів, а TeamCity забезпечує автоматичну збірку та тестування проєкту, що дозволяє швидко виявляти та виправляти помилки в коді. Після кожної зміни в коді, внесеної через Perforce, TeamCity автоматично запускає процес збірки та тестування, що забезпечує постійну інтеграцію та перевірку змін, мінімізуючи ризики виникнення конфліктів у коді. Perforce керує версіями всіх файлів проєкту, а Swarm інтегрується з ним для проведення код-рев'ю, забезпечуючи високу якість коду та дотримання стандартів. Jira використовується для управління завданнями та проєктами, що дозволяє командам ефективно планувати та відслідковувати прогрес, тоді як Confluence слугує платформою для документації, де зберігаються всі проєктні документи, технічні специфікації та інші важливі матеріали.

### 3.3 Обґрунтування вибору Agile для основної розробки

Методологія Agile, зокрема Scrum, є найкращим вибором для основної розробки гри з наступних причин:

Agile дозволяє швидко адаптуватися до змін у вимогах та умовах ринку, що є критично важливим для індустрії ігор. Ітеративний підхід Agile передбачає регулярні спринти, які дозволяють команді швидко реагувати на зворотний зв'язок від гравців та замовників, вносячи необхідні корективи на кожному етапі розробки[1].

Agile забезпечує тісну співпрацю з замовником та регулярне отримання зворотного зв'язку, що дозволяє вчасно виявляти та виправляти проблеми,

підвищуючи якість кінцевого продукту. Щоденні зустрічі (Daily Stand-ups), огляди спринтів (Sprint Reviews) та ретроспективи (Sprint Retrospectives) сприяють високій прозорості процесу розробки та ефективному вирішенню проблем.

Scrum дозволяє забезпечити високу прозорість процесу розробки через регулярні зустрічі та огляди. Використання Kanban для візуалізації процесу розробки за допомогою дошки Kanban допомагає контролювати потік роботи та забезпечувати рівномірне навантаження на команду[2].

### 3.4 Планування проєкту за допомогою методології Agile

У цьому розділі ми розглянемо приклад планування задач для розробки ігрового додатку. В рамках аналізу та планування визначені вимоги, створений беклог проєкту, проведений аналіз технологій та обраний технологічний стек. Основні механіки гри розроблені протягом шести тижнів, включаючи створення базової механіки управління персонажем, інтеграцію механіки стрільби та розробку механіки руху. Мультиплеєр був розроблений, інтегрований та протестований, після чого були створені і оптимізовані рівні та середовища. Оптимізація продуктивності, налаштування системи CI/CD та автоматичне тестування завершені до фінальних налаштувань та релізу гри. Список задач буде у таблиці 3.4

Таблиця 3.4 - План робіт

	Спринт	Завдання	Відповідальні особи	Тривалість завдання
1	2	3	4	5

Продовження таблиці 3.4

1	2	3	4	5
Аналіз та планування	Спринт 1	Визначення вимог	Product Owner	2 дні
		Створення беклогу проекту	Product Owner, Scrum Master	3 дні
		Аналіз існуючих технологій	Development Team	5 днів
		Вибір технологічного стеку	Development Team	2 дні
		Планування спринтів	Scrum Master, Product Owner	3 дні
Розробка основних механік гри	Спринт 2	Створення базової механіки управління персонажем	Front-End Developer, Back-End Developer	2 тижні
	Спринт 3	Інтеграція механіки стрільби	Front-End Developer, Back-End Developer	2 тижні
	Спринт 4	Розробка механіки руху та взаємодії з об'єктами	Front-End Developer, Back-End Developer	2 тижні
		Тестування та відладка механік	QA Engineer	3 тижні (паралельно з розробкою)
Розробка мультиплеєру	Спринт 5	Розробка архітектури мультиплеєру	Back-End Developer	2 тижні
	Спринт 6	Інтеграція мережеских функцій	Back-End Developer	2 тижні
	Спринт 7	Тестування та оптимізація мережевого коду	QA Engineer	2 тижні

Продовження таблиці 3.4

1	2	3	4	5
		Впровадження античіт-системи	Back-End Developer	2 тижні (паралельно з розробкою)
		Тестування мультиплеєру	QA Engineer	2 тижні (паралельно з розробкою)
Розробка рівнів та середовищ	Спринт 8	Дизайн рівнів та середовищ	UI/UX Designer, Game Designer	4 тижні
	Спринт 9	Інтеграція рівнів у рушій	Front-End Developer	2 тижні
	Спринт 10	Налаштування та оптимізація середовищ	Front-End Developer, Back-End Developer	2 тижні
		Тестування рівнів	QA Engineer	3 тижні (паралельно з розробкою)
Оптимізація та автоматизація	Спринт 11	Оптимізація продуктивності для різних платформ	DevOps Engineer	4 тижні
	Спринт 12	Налаштування системи CI/CD	DevOps Engineer	2 тижні
	Спринт 13	Автоматичне тестування та виправлення помилок	QA Engineer, DevOps Engineer	4 тижні (паралельно з оптимізацією)
		Ручне тестування та відладка	QA Engineer	3 тижні (паралельно з автоматизацією)
Бета-тестування та вдосконалення	Спринт 14	Проведення бета-тестування	QA Engineer	2 тижні
	Спринт 15	Збір та аналіз відгуків	Product Owner, QA Engineer	1 тиждень
	Спринт 16	Внесення змін та покращень	Development Team	3 тижні

Кінець таблиці 3.4

1	2	3	4	5
	Спринт 17	Повторне тестування оновлених версій	QA Engineer	2 тижні
		Розробка та налаштування версії гри для консолей	Development Team	2 тижні
		Тестування версії для консолей	QA Engineer	2 тижні
	Спринт 18	Фінальна оптимізація та тестування	Development Team, QA Engineer	1 тиждень
		Підготовка документації для релізу	Product Owner, Scrum Master	1 тиждень
		Офіційний реліз гри	Вся команда	1 день

На основі запланованих спринтів та завдань ми створили діаграму Ганта, щоб візуалізувати хід виконання проекту. Кожне завдання з таблиці було відображене як окрема смуга на діаграмі з відповідною тривалістю та відповідальними особами. Це дозволило легко відстежувати прогрес та взаємозалежність між різними етапами розробки, забезпечуючи ефективне управління часом і ресурсами. Діаграма Ганта стала важливим інструментом для контролю та коригування плану роботи команди. Діаграма Ганта наведена на рисунку 3.5

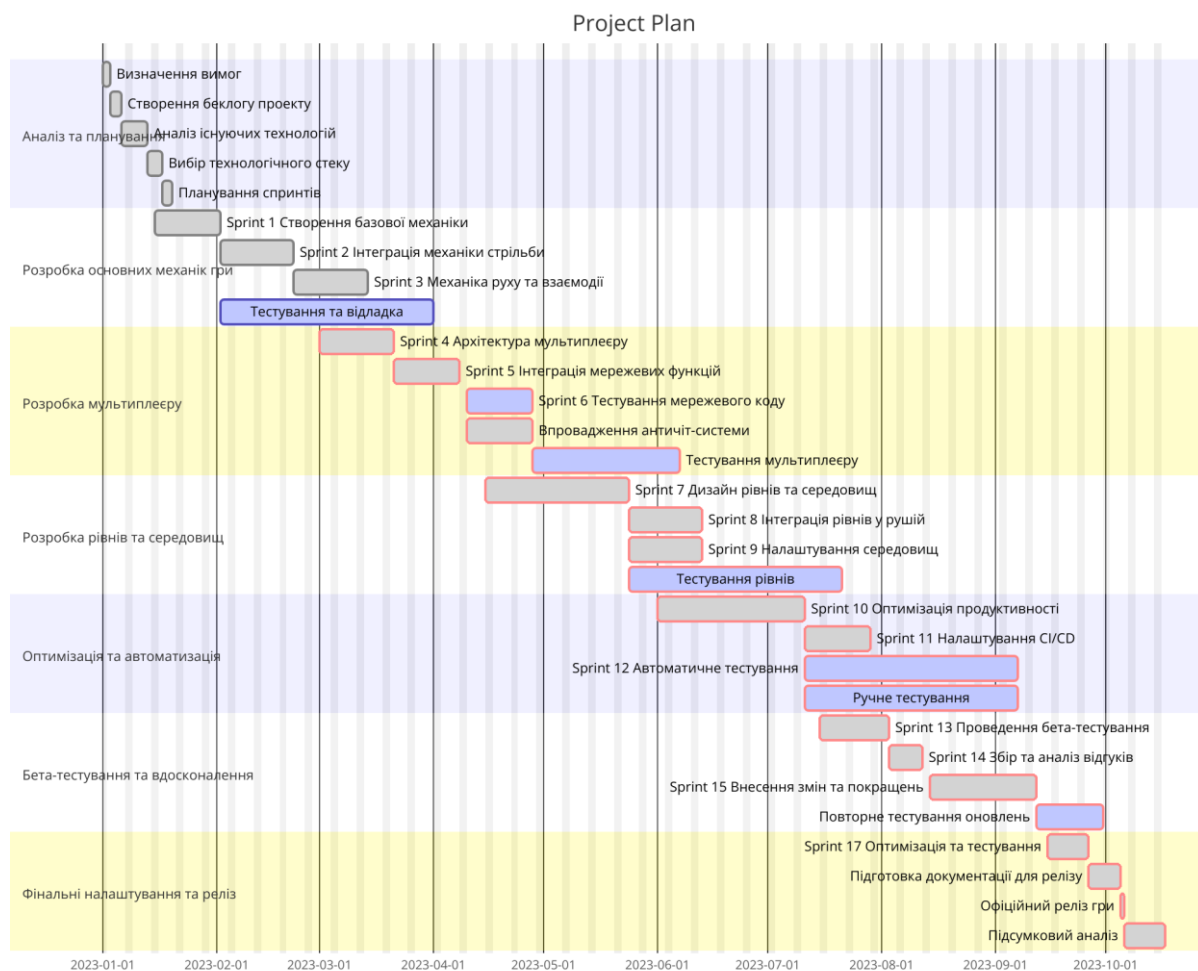


Рисунок 3.5 - Діаграма Ганта

### 3.5 Обґрунтування вибору методу критичного ланцюга для портування гри

Портування гри на інші платформи, такі як PS5, є складним завданням, що потребує ретельного планування та контролю. Метод критичного ланцюга (ССРМ) був обраний для цього етапу з наступних причин:

1) Оптимізація використання ресурсів - ССРМ зосереджується на ідентифікації та управлінні критичними ресурсами, які впливають на терміни виконання проєкту, що є особливо важливим у процесі портування гри. Це дозволяє мінімізувати простої та збої, оптимізуючи використання ресурсів для забезпечення безперервності роботи над критичним ланцюгом.

2) Управління буферами часу - ССРМ передбачає додавання буферів часу для захисту критичного ланцюга від затримок. Використання проєктних буферів, буферів живлення та буферів ресурсів допомагає забезпечити дотримання кінцевих термінів навіть у разі непередбачених затримок. Це особливо важливо для процесу портування, де жорсткі терміни та вимоги сертифікації від Sony можуть впливати на успіх проєкту.

3) Ретельне планування та контроль - ССРМ вимагає ретельного планування та контролю, що допомагає забезпечити високу якість кінцевого продукту. Це включає постійний моніторинг прогресу, оцінку ризиків та своєчасне прийняття рішень для мінімізації можливих проблем.

Вибір гібридного підходу, який включає використання методології Agile для основної розробки та методу критичного ланцюга для портування гри, обґрунтований необхідністю забезпечити гнучкість, адаптивність та ефективне управління ресурсами. Agile підходить для динамічних процесів розробки, де важлива швидка реакція на зміни та зворотний зв'язок, тоді як ССРМ оптимальний для ретельно спланованих завдань з жорсткими термінами, таких як портування гри. Такий комбінований підхід дозволяє максимально ефективно вирішувати задачі проєкту шутера, забезпечуючи високу якість та своєчасне виконання.

### 3.6 Планування за методом критичного ланцюга

Перед створенням таблиці задач було проведено ретельне планування процесу розробки гри для консолей. Спочатку визначені основні етапи робіт, включаючи аналіз вимог, розробку та налаштування версії гри, тестування та фінальні налаштування. Кожен етап розбитий на конкретні задачі з вказаними естимейтами, буфером часу та описом робіт. Це дозволило створити чіткий план дій та забезпечити контроль за виконанням кожного завдання. У таблиці 3.5 наведені задачі.

Таблиця 3.5 – План робіт

Задачі	Оцінка	Буфер	Опис задачі
1	2	3	4
Аналіз вимог консолей	2 дні	1 день	Вивчення специфікацій та вимог для консолей (PS5, Xbox Series X/S). Ідентифікація особливостей апаратної платформи та програмного забезпечення.
Підготовка технічної документації	3 дні	1 день	Розробка детальної технічної документації для портування гри. Опис вимог до оптимізації та адаптації графіки, звуку та ігрового процесу.
Адаптація графічних налаштувань	5 днів	2 дні	Оптимізація графіки для апаратного забезпечення консолей. Налаштування роздільної здатності, текстур, освітлення та інших графічних параметрів.
Адаптація аудіо-налаштувань	3 дні	1 день	Оптимізація звукових ефектів та музики для консолей. Налаштування аудіо параметрів для забезпечення найкращого звукового досвіду.
Оптимізація продуктивності	5 днів	2 дні	Виявлення та усунення вузьких місць у коді. Зниження навантаження на процесор та графічний процесор консолей.
Інтеграція контролерів	3 дні	1 день	Налаштування управління під консольні контролери. Оптимізація інтерфейсу для геймпадів.

Кінець таблиці 3.5

1	2	3	4
Функціональне тестування	5 днів	1 день	Перевірка всіх основних функцій гри на консолях. Тестування ігрового процесу, графіки, звуку та управління.
Тестування продуктивності	3 дні	1 день	Оцінка продуктивності гри на різних моделях консолей. Тестування на стабільність фреймрейту та відсутність лагів.
Тестування сумісності	2 дні	1 день	Перевірка сумісності гри з різними версіями операційних систем консолей. Тестування на різних моделях та ревізіях консолей.
Виправлення знайдених помилок	4 дні	1 день	Усунення всіх виявлених проблем та багів. Проведення додаткового тестування після виправлення помилок.
Підготовка документації для релізу	3 дні	1 день	Оформлення всіх необхідних документів для публікації гри на консолях. Підготовка інструкцій для користувачів та технічної підтримки.
Офіційний реліз гри	1 день	1 день	Завантаження гри на платформи дистрибуції. Анонс релізу для гравців та медіа.

Після визначення та структурування всіх задач у таблиці, ми перейшли до створення діаграми Ганта. Діаграма Ганта наочно відображає часові рамки для виконання кожної задачі, враховуючи естимейти та буфери. Це дозволяє легко відстежувати прогрес проекту, бачити взаємозалежності між задачами та забезпечує ефективне управління ресурсами. За допомогою цієї діаграми команда може швидко ідентифікувати можливі затримки та оперативно вносити корективи в план роботи. Діаграма Ганта наведена на рисунку 3.2

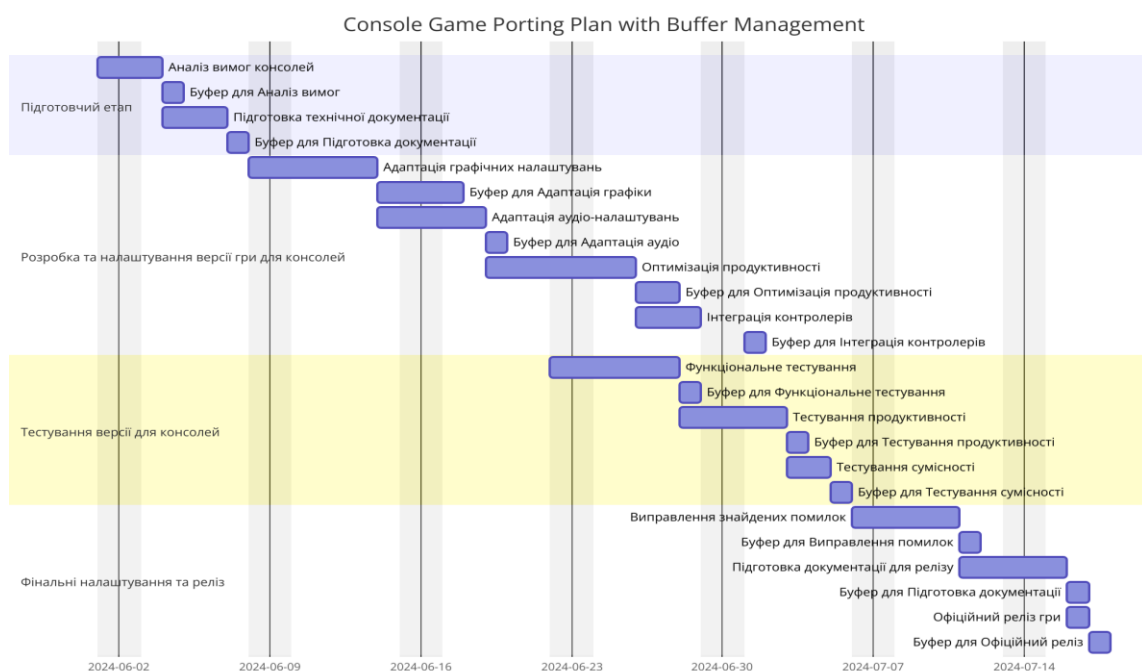


Рисунок 3.2 Діаграма Ганта

### 3.7 Управління ризиками

Реалізація система ідентифікації та оцінки ризиків для проєкту ігрового додатку

#### 1) Збір даних

Інтерв'ю з ключовими членами команди дозволяє зібрати цінну інформацію про можливі ризики з їх точки зору. Результати інтерв'ю наведені у таблиці 3.6

Таблиця 3.6 Результати інтерв'ю

Член команди	Посада	Виявлені ризики	Пропоновані вирішення
1	2	3	4

Кінець таблиці 3.6

1	2	3	4
Олексій	Провідний розробник	Нестабільність рушія Unreal Engine на PS5	Додаткове тестування та оптимізація коду
Марія	Менеджер проєкту	Недостатній бюджет на маркетинг	Перегляд бюджету та залучення додаткового фінансування
Іван	Технічний художник	Проблеми з інтеграцією динамічних погодних умов	Співпраця з експертами з візуальних ефектів

## 2) Аналіз планів проєкту

Проблеми з платформою або технічні обмеження можуть створювати значні перешкоди у розробці проєкту. Приклад технічних проблем та їх вирішення наведено у таблиці 3.7

Таблиця 3.7 технічні проблем та їх вирішення

Ризик	Опис	Вирішення
1	2	3
Проблеми з платформою	Нестабільність гри на PS5	Додаткове тестування, оптимізація коду, залучення експертів
Технічні обмеження	Недостатня потужність обладнання для тестування	Закупівля нового обладнання або оренда потужних серверів
Інтеграція нових технологій	Проблеми з інтеграцією динамічних погодних умов	Залучення спеціалістів з відповідних технологій

### 3) Система ідентифікації та оцінки ризиків для проєкту ігрового додатку

Визначення шкали ймовірності включає оцінку ймовірності виникнення кожного ризику. Це дозволяє пріоритизувати ризики та зосередити зусилля на найважливіших. Шкала ймовірності складається з трьох рівнів: низький, середній та високий.

Нестабільність гри на PS5 має високу ймовірність через технічні складнощі платформи. Проблеми з сумісністю та оптимізацією можуть спричинити часті збої та поганий досвід користувачів.

Недостатній бюджет на маркетинг оцінений як середня ймовірність через можливі зміни в фінансовому плануванні. Можливе недофінансування може виникнути через перерозподіл ресурсів або непередбачені витрати.

Втрата ключових співробітників оцінена як низька ймовірність, оскільки команда мотивована і задоволена умовами праці. Проте завжди існує ризик непередбачених обставин, які можуть вплинути на працівників.

### 4) Оцінка впливу ризику на проєкт

Оцінка впливу ризику на проєкт включає визначення шкали впливу. Це дозволяє зрозуміти, наскільки серйозним може бути вплив кожного ризику на проєкт. Шкала впливу складається з трьох рівнів: низький, середній та високий.

Нестабільність гри на PS5 оцінена як високий вплив через серйозний вплив на якість гри та досвід користувачів. Часті збої та погана оптимізація можуть призвести до негативних відгуків та зниження продажів.

Недостатній бюджет на маркетинг оцінений як середній вплив через помірний вплив на промоцію та охоплення аудиторії. Недостатня реклама може знизити впізнаваність гри та вплинути на обсяги продажів.

Втрата ключових співробітників оцінена як високий вплив через значний вплив на продуктивність команди та терміни виконання. Втрата

досвідчених працівників може призвести до затримок у розробці та додаткових витрат на пошук і навчання нових співробітників.

### 5) Матриця оцінки ризиків

Матриця оцінки ризиків створюється для візуалізації ризиків за ймовірністю та впливом. Це допомагає визначити, які ризики є найбільш критичними та потребують негайної уваги.

Нестабільність гри на PS5 має високу ймовірність та високий вплив, що робить цей ризик критичним. Потрібно зосередити зусилля на тестуванні та оптимізації гри для цієї платформи.

Недостатній бюджет на маркетинг має середню ймовірність та середній вплив, що робить цей ризик помірним. Необхідно ретельно планувати бюджет та шукати додаткові джерела фінансування.

Втрата ключових співробітників має низьку ймовірність, але високий вплив, що робить цей ризик високим. Важливо мати план дій на випадок втрати ключових працівників, зокрема програми мотивації та ретенції. Матриця наведена у таблиці 3.8.

Таблиця 3.8 Матриця оцінки ризиків

Ризик	Ймовірність	Вплив	Критичність
1	2	3	4
Нестабільність гри на PS5	Висока	Високий	Критичний
Недостатній бюджет на маркетинг	Середня	Середній	Помірний
Втрата ключових співробітників	Низька	Високий	Високий

Стратегії управління ризиками включають різні методи для зниження або усунення ризиків. Вони включають уникнення, пом'якшення, прийняття та передачу ризиків. Уникнення (elimination) передбачає зміну плану проєкту для виключення ризику. Наприклад, ризик нестабільності гри на PS5 можна

уникнути, змінивши основну платформу на більш стабільну, таку як ПК або Xbox.

Пом'якшення (mitigation) включає зменшення ймовірності або впливу ризику. Недостатній бюджет на маркетинг можна пом'якшити, розробивши детальний маркетинговий план з пошуком додаткових джерел фінансування, включаючи краудфандинг або партнерство з іншими компаніями.

Прийняття (acceptance) передбачає прийняття ризику та підготовку до його наслідків. Втрата ключових співробітників може бути прийнята, визнавши ризик і розробивши план на випадок відставки, включаючи пошук нових працівників та програми навчання.

Передача (transfer) ризику іншій стороні (наприклад, страхування) може бути застосована до ризику пошкодження обладнання. У цьому випадку можна закупити страхування для обладнання, щоб покрити витрати у випадку його пошкодження.

Розробка планів дій включає визначення конкретних заходів для кожного ризику та призначення відповідальних осіб за їх виконання. Ризик нестабільності гри на PS5 можна вирішити виконанням додаткового тестування гри на PS5 для виявлення та усунення проблем з сумісністю.

Недостатній бюджет на маркетинг можна вирішити створенням команди для пошуку додаткових джерел фінансування та розробкою маркетингової кампанії з обмеженими ресурсами. Втрата ключових співробітників може бути вирішена розробкою програм навчання для нових співробітників та створенням резерву кандидатів на ключові посади.

Пошкодження обладнання можна вирішити закупівлею запасного обладнання та підготовкою плану швидкого відновлення у разі поломки.

Таким чином, для кожного ризику розробляються конкретні стратегії управління та плани дій з відповідальними особами, що забезпечує своєчасне та ефективне реагування на виникнення ризиків.

## б) Моніторинг та контроль ризиків

Постійний моніторинг ризиків включає регулярні зустрічі для перегляду та оновлення статусу ризиків, а також використання інструментів управління проектами для відстеження ризиків. Регулярні зустрічі забезпечують команді можливість обговорювати поточні ризики, оновлювати їх статус та розробляти нові стратегії управління у разі необхідності. Використання інструментів управління проектами, таких як Jira, дозволяє систематично відстежувати ризики, документувати їх та забезпечувати прозорість усього процесу управління ризиками.

Нестабільність гри на PS5 можна вирішити регулярними зустрічами команди для обговорення прогресу у вирішенні проблем сумісності з PS5 та використанням Jira для документування кожного кроку та оновлення статусу ризику. Недостатній бюджет на маркетинг вирішується щотижневими зустрічами для аналізу фінансового стану маркетингових кампаній та використанням Jira для відстеження всіх фінансових транзакцій та оновлень бюджету. Втрата ключових співробітників потребує регулярних зустрічей для оцінки задоволеності працівників та планування кадрових резервів, а також використання Jira для документування всіх кадрових змін та планів дій.

## 7) Адаптація планів

Адаптація планів управління ризиками передбачає коригування планів на основі нових даних та змін у проекті. Це дозволяє оперативно реагувати на нові виклики та забезпечувати ефективне управління ризиками.

Нестабільність гри на PS5 потребує коригування плану дій після отримання нових технічних даних про проблеми сумісності. План дій коригується для включення додаткових тестувань та оптимізації коду. Недостатній бюджет на маркетинг потребує коригування плану маркетингової кампанії після аналізу нових фінансових даних для включення додаткових заходів зі збору коштів та оптимізації витрат. Втрата ключових

співробітників потребує коригування плану кадрових резервів після аналізу нових даних про задоволеність працівників для включення нових програм мотивації та навчання.

Таким чином, моніторинг та контроль ризиків включає регулярні зустрічі, використання інструментів управління проектами та адаптацію планів на основі нових даних, що забезпечує своєчасне та ефективне управління ризиками.

### 3.8 Управління ризиками за допомогою методу критичного ланцюга

Управління ризиками є важливим аспектом успішного ведення проєктів, особливо у складних і великих проєктах, таких як розробка інноваційного шутера від першої особи. Метод критичного ланцюга (Critical Chain Project Management, CCPM) є ефективним підходом до управління ризиками, оскільки він зосереджується на ідентифікації та управлінні критичними ресурсами і часом, необхідним для виконання проєкту. Цей метод допомагає мінімізувати затримки та забезпечити своєчасне виконання проєкту[3].

Метод критичного ланцюга базується на кількох основних принципах. Ідентифікація критичного ланцюга передбачає визначення послідовності завдань, які обмежують загальну тривалість проєкту. Це дозволяє зосередити увагу на найважливіших завданнях та ресурсах, необхідних для їх виконання. Управління буферами включає додавання буферів часу для захисту критичного ланцюга від затримок. Існують три типи буферів: проєктний буфер, буфер живлення та буфер ресурсів. Управління ресурсами спрямоване на оптимізацію використання ресурсів для забезпечення безперервності

роботи над критичним ланцюгом, уникаючи перевантажень та конфліктів між завданнями.

Буфери є ключовим елементом методу критичного ланцюга і служать для захисту проекту від затримок, дозволяючи ефективніше управляти ризиками. Існують три основні типи буферів: проєктний буфер, буфер живлення та буфер ресурсів. Проєктний буфер – це додатковий час, який додається до кінця проєкту, щоб захистити його від затримок у критичному ланцюзі завдань. Він створюється шляхом аналізу тривалості всіх завдань критичного ланцюга та визначення ймовірних ризиків затримок, зазвичай становить від 25% до 50% від загальної тривалості критичного ланцюга. Наприклад, якщо загальна тривалість критичного ланцюга становить 100 днів, то проєктний буфер може становити від 25 до 50 днів[3].

Буфер живлення – це додатковий час, який додається до некритичних завдань, що можуть вплинути на критичний ланцюг. Цей буфер захищає критичний ланцюг від затримок у некритичних завданнях. Він створюється шляхом аналізу некритичних завдань, які можуть вплинути на критичний ланцюг, і тривалість буфера живлення залежить від кількості та складності таких завдань. Як і у випадку з проєктним буфером, тривалість буфера живлення зазвичай становить від 25% до 50% від загальної тривалості відповідних некритичних завдань[3].

Буфер ресурсів – це додатковий час, який додається для врахування можливих затримок у доступності критичних ресурсів, таких як обладнання або ключові співробітники. Він створюється шляхом аналізу ресурсів, які мають вирішальне значення для виконання завдань критичного ланцюга, з урахуванням можливих збоїв або недоступності ресурсів у певні періоди. Тривалість буфера ресурсів визначається на основі оцінки ймовірності таких збоїв та їх впливу на проєкт[3].

## 4 АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

### 4.1 Аналіз результатів дослідження

Результати проведеного дослідження зосереджені на кількох ключових аспектах розробки ігрового додатку, а саме: створення вимог, каркасів, прототипів, планування розробки та оцінки ризиків. Практичне застосування цих результатів дозволило значно покращити процес розробки та забезпечити високу якість кінцевого продукту.

Створення вимог. Проведення інтерв'ю, опитувань та воркшопів із зацікавленими сторонами дозволило точно визначити основні потреби та очікування користувачів. Це забезпечило чітке розуміння функціональних та нефункціональних вимог до ігрового додатку. Отримані вимоги стали основою для подальшого розроблення ігрового продукту.

Створення каркасів (wireframes). Розробка каркасів допомогла візуалізувати структуру інтерфейсу гри, розташування основних елементів та їх взаємодію. Це забезпечило зручність користувацького досвіду та дозволило виявити можливі проблеми на ранніх етапах розробки, що зменшило кількість змін у майбутньому.

Розробка прототипів. Використання інструментів таких як Figma та Sketch дозволило створити високодеталізовані прототипи, які візуалізували кінцевий продукт та забезпечили узгодженість вимог серед усіх зацікавлених сторін. Це сприяло точнішому плануванню та ефективному управлінню ресурсами протягом усього процесу розробки.

Планування розробки. Застосування методології Scrum дозволило організувати процес розробки з регулярними спринтами, що забезпечило гнучкість та швидку адаптацію до змін. Щоденні зустрічі, огляди спринтів та ретроспективи сприяли високій прозорості процесу розробки та ефективному вирішенню проблем.

Оцінка ризиків. Розробка системи ідентифікації та оцінки ризиків дозволила виявляти потенційні проблеми на ранніх етапах та розробляти ефективні стратегії управління ними. Регулярний аналіз ризиків та оновлення планів управління ризиками забезпечило успішну реалізацію проєкту та мінімізацію негативних впливів.

Практичне застосування отриманих результатів дозволило створити фундамент для подальшої розробки ігрового додатку, що відповідає сучасним вимогам та очікуванням користувачів. Це підтверджує ефективність використаних підходів та методологій у процесі розробки.

## 4.2 Перспективи подальших досліджень

Перспективи подальших досліджень зосереджені на удосконаленні процесів створення вимог, каркасів, прототипів, планування розробки та управління ризиками.

Подальші дослідження можуть бути спрямовані на аналіз ефективності гібридних методологій, які поєднують переваги різних підходів до управління проєктами. Впровадження нових інструментів автоматизації допоможе підвищити продуктивність команди розробників та покращити якість кінцевого продукту.

Поглиблене дослідження ризиків дозволить розробити більш детальні моделі ідентифікації та оцінки ризиків, що забезпечить ще більшу ефективність управління проєктами. Розробка інноваційних технологій для створення реалістичних та інтерактивних ігрових середовищ сприятиме підвищенню якості ігрового досвіду для користувачів.

Загалом, отримані результати та їхнє подальше дослідження можуть значно покращити процеси розробки ігрових додатків, забезпечити їхню

конкуренстоспроможність на світовому ринку та сприяти розвитку індустрії комп'ютерних ігор.

## ВИСНОВКИ

У рамках проведеного дослідження було досягнуто важливих результатів у створенні вимог, каркасів, прототипів, планування розробки та оцінки ризиків. Це підтверджує ефективність використання методологій Agile, зокрема Scrum, для управління процесом розробки ігрового додатку. Застосування гнучких методів дозволило адаптувати процес розробки до швидкозмінних вимог ринку та забезпечити високу якість кінцевого продукту.

Практичне застосування отриманих результатів включало розробку чітких вимог до ігрового додатку шляхом проведення інтерв'ю, опитувань та воркшопів з зацікавленими сторонами. Це дозволило виявити основні потреби та очікування користувачів, що стало основою для подальшої розробки. Створення каркасів (wireframes) допомогло визначити структуру інтерфейсу та його елементів, що забезпечило зручність користувацького досвіду та зменшило кількість змін на пізніх етапах розробки.

Розробка високодеталізованих прототипів дозволила візуалізувати кінцевий продукт та забезпечити узгодженість вимог серед усіх зацікавлених сторін. Це сприяло більш точному плануванню та ефективному управлінню ресурсами протягом усього процесу розробки.

Планування розробки з використанням методології Scrum організувало процес з регулярними спринтами, що забезпечило гнучкість та швидку адаптацію до змін. Щоденні зустрічі, огляди спринтів та ретроспективи сприяли високій прозорості процесу розробки та ефективному вирішенню проблем. Метод критичного ланцюга (CCPM) був використаний для оптимізації використання ресурсів та забезпечення дотримання жорстких термінів для різних етапів проекту, що мінімізувало ризики.

Розробка системи ідентифікації та оцінки ризиків дозволила виявляти потенційні проблеми на ранніх етапах та розробляти ефективні стратегії

управління ними. Це включало проведення регулярних аналізів ризиків та оновлення планів їх управління, що забезпечило успішну реалізацію проєкту.

Результати дослідження мають високий потенціал для подальшої реалізації проєкту. На основі отриманих даних можна створити повноцінний ігровий додаток, що відповідатиме сучасним вимогам та очікуванням користувачів. Використання отриманих знань та напрацювань може бути розширене на інші проєкти в галузі розробки комп'ютерних ігор, а також на управління IT-проєктами в різних секторах, де важлива гнучкість та ефективне управління ресурсами.

Доцільність продовження досліджень за відповідною тематикою є високою. Подальші дослідження можуть бути спрямовані на удосконалення процесів створення вимог, каркасів, прототипів, планування розробки та управління ризиками, що дозволить підвищити якість кінцевих продуктів та забезпечити їхню конкурентоспроможність на світовому ринку.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., & Thomas, D. (2001). Manifesto for Agile Software Development. URL: <https://agilemanifesto.org/>
2. Schwaber, K., & Sutherland, J. (2017). The Scrum Guide. Scrum Alliance. URL: <https://scrumguides.org/>
3. Goldratt, E. M. (1997). Critical Chain. The North River Press.
4. Anderson, D. J. (2010). Kanban: Successful Evolutionary Change for Your Technology Business. Blue Hole Press.
5. Ries, E. (2011). The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses. Crown Business.
6. Humble, J., & Farley, D. (2010). Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation. Addison-Wesley.
7. Freeman, S. (2009). Growing Object-Oriented Software, Guided by Tests. Addison-Wesley.
8. North, D. (2006). Introducing BDD. URL: <https://dannorth.net/introducing-bdd/>
9. Kerzner, H. (2017). Project Management: A Systems Approach to Planning, Scheduling, and Controlling. Wiley.
10. Sommerville, I. (2011). Software Engineering (9th ed.). Addison-Wesley.
11. Royce, W. W. (1970). Managing the Development of Large Software Systems. Proceedings of IEEE WESCON.
12. Cohn, M. (2004). User Stories Applied: For Agile Software Development. Addison-Wesley.
13. Poppendieck, M., & Poppendieck, T. (2003). Lean Software Development: An Agile Toolkit. Addison-Wesley.
14. Rubin, K. S. (2012). Essential Scrum: A Practical Guide to the Most Popular Agile Process. Addison-Wesley.
15. Newman, S. (2015). Building Microservices: Designing Fine-Grained Systems. O'Reilly Media.

16. Fitzgerald, B., & Stol, K. (2017). Continuous Software Engineering: A Roadmap and Agenda. *Journal of Systems and Software*, 123, 176-189.
17. Cusumano, M. A., & Selby, R. W. (1995). *Microsoft Secrets: How the World's Most Powerful Software Company Creates Technology, Shapes Markets, and Manages People*. Free Press.
18. Kruchten, P. (2004). *The Rational Unified Process: An Introduction (3rd ed.)*. Addison-Wesley.
19. Rubin, K. S. (2013). *Essential Scrum: A Practical Guide to the Most Popular Agile Process*. Addison-Wesley.
20. Левикін В.М., Діденко Д.О., Альошкін А.О. Метод формування заявок природною мовою на основі вдосконаленої моделі BERT // АСУ та прилади автоматизації. 2024. № 180. С. 55-71.
21. Методичні вказівки щодо розробки та оформлення магістерської атестаційної роботи за спеціальністю 122 Комп'ютерні науки (освітня програма «Управління проектами в галузі інформаційних технологій» освітньо-кваліфікаційного рівня «магістр» / Упоряд.: Петров К.Е., Левикін В.М., Чалий С.Ф., Євланов М.В., Саєнко В.І., Міхнов Д.К., Міхнова А.В., Чала О.В. – Харків: ХНУРЕ, 2019. – 24 с.