

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет комп'ютерних наук  
(повна назва)

Кафедра програмної інженерії  
(повна назва)

## КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

Ігровий програмний застосунок у жанрі 3D Action-Adventure  
з елементами RPG. Механіки, ШІ, HUD  
(тема)

Виконав:

студент 4 курсу, групи ПЗП-20-7

Крупчак Є. І.  
(прізвище, ініціали)

Спеціальність 121 – Інженерія програмного  
забезпечення  
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Програмна інженерія  
(повна назва освітньої програми)

Керівник ст. викл. кафедри ПІ Новіков Ю. С.  
(посада, прізвище, ініціали)

Допускається до захисту  
Зав. кафедри

\_\_\_\_\_  
(підпис)

З. В. Дудар  
(прізвище, ініціали)

2024 р.

## Харківський національний університет радіоелектроніки

Факультет \_\_\_\_\_ комп'ютерних наук  
 Кафедра \_\_\_\_\_ програмної інженерії  
 Рівень вищої освіти \_\_\_\_\_ перший (бакалаврський)  
 Спеціальність \_\_\_\_\_ 121 – Інженерія програмного забезпечення  
 Тип програми \_\_\_\_\_ Освітньо-професійна  
 Освітня програма \_\_\_\_\_ Програмна Інженерія  
 (шифр і назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_

(підпис)

« \_\_\_\_ » \_\_\_\_\_ 2024 р.

### ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові \_\_\_\_\_ Крупчаку Євгенію Ігоровичу \_\_\_\_\_

(прізвище, ім'я, по батькові)

1. Тема роботи Ігровий програмний застосунок у жанрі 3D Action-Adventure з елементами RPG. Механіки, ШІ, HUD

Затверджена наказом по університету від 20.05.2024 р. № 471 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 06.06.2024

3. Вихідні дані до роботи Розробити механіки, штучний інтелект та інтерфейс користувача для ігрового застосунку в жанрі 3D Action-Adventure з елементами RPG мовами програмування C++ та Blueprint, використовуючи ігровий двигун Unreal Engine 5.

4. Перелік питань, що потрібно опрацювати в роботі

Вступ, аналіз предметної галузі, формування вимог до програмної системи, архітектура та проектування програмного забезпечення, опис прийнятих програмних рішень, тестування розробленого програмного забезпечення, висновки, додатки.



**РЕФЕРАТ / ABSTRACT**

Пояснювальна записка до кваліфікаційної роботи бакалавра, 94 стор., 70 рис., 10 табл., 10 джерел.

ІГРОВИЙ ПРОГРАМНИЙ ЗАСТОСУНОК, ACTION-ADVENTURE, BLUEPRINT, C++, RIDER, UNREAL ENGINE 5

Об'єкт розробки – Ігровий програмний застосунок у жанрі 3D Action-Adventure з елементами RPG.

Мета розробки – розробити демонстраційну версію ігрового застосунку у жанрі 3D Action-Adventure з елементами RPG та в естетиці стімпанк для операційної системи Windows.

Метод рішення – середовище розробки Unreal Engine 5 та JetBrains Rider, мови програмування C++ та Blueprint.

У результаті було створено демонстраційну версію ігрового застосунку у жанрі 3D Action-Adventure з елементами RPG та в естетиці стімпанк для операційної системи Windows.

GAME SOFTWARE, ACTION-ADVENTURE, BLUEPRINT, C++, RIDER, UNREAL ENGINE 5

Object of development is a 3D Action-Adventure game application with RPG elements.

Development goal is to develop a demo version of a 3D Action-Adventure game application with RPG elements and steampunk aesthetics for the Windows operating system.

The solution method was the Unreal Engine 5 and JetBrains Rider development environment; C++ and Blueprint programming languages.

As a result, a demo version of a 3D Action-Adventure game application with RPG elements and steampunk aesthetics for the Windows operating system was created.

Я, Крупчак Євгеній Ігорович, студент гр. ПЗП-20-7, здобувач вищої освіти на першому (бакалаврському) рівні кафедри «Програмна інженерія», заявляю: моя кваліфікаційна робота на тему «Ігровий програмний застосунок у жанрі 3D Action-Adventure з елементами RPG.», що буде представлена до екзаменаційної комісії для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу EIAg KhNURE. Усі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови до допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

## ЗМІСТ

Вступ.....	9
1 Аналіз предметної галузі.....	10
1.1 Аналіз предметної галузі.....	10
1.2 Жанрові характеристики.....	10
1.3 Аналіз механік.....	11
1.4 Аналіз конкурентів.....	12
1.5 Виявлення та вирішення проблем.....	13
1.6 Постановка задачі.....	14
2 Формування вимог до програмної системи.....	15
2.1 Постановка мети.....	15
2.2 Загальний опис.....	15
2.3 Загальні обмеження.....	16
2.4 Припущення та залежності.....	16
3 Архітектура та проектування програмного забезпечення.....	17
3.1 UML проектування ПЗ.....	17
3.2 Проектування архітектури ПЗ.....	17
3.3 Приклади найцікавіших алгоритмів та методів.....	19
3.4 Проектування UI / UX.....	21
4 Опис прийнятих програмних рішень.....	26
4.1 Рух снарядів для зброї дальнього бою.....	26
4.2 Штучний інтелект противників.....	27
4.3 Розробка інтерфейсів інвентарів.....	30
5 Тестування програмного забезпечення.....	34
5.1 Тестування ігрового застосунку.....	34
5.2 Виявлені помилки.....	35
6 Впровадження програмного забезпечення.....	36
6.1 Наукове впровадження проекту.....	36
6.2 Практичне впровадження проекту.....	36
6.3 Соціальне впровадження проекту.....	36

	7
Висновки .....	38
Перелік джерел посилання .....	39
Додаток А. Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ .....	40
Додаток Б. Слайди презентації .....	41
Додаток В. Концепт-документ до гри .....	41
Додаток Г. План тестування.....	71
Додаток Д. Баг-репорти .....	82
Додаток Е. Тези за темою кваліфікаційної роботи.....	87
Додаток Ж. Фрагмент каталогу виставки технічної творчості молоді.....	92

**ПЕРЕЛІК СКОРОЧЕНЬ**

3D – 3-Dimensional

RPG – Role Play Game

UI – User Interface

UX – User Experience

HUD – Head-Up Display

NPC – Not Playable Character

## ВСТУП

Темою кваліфікаційної роботи є ігровий програмний застосунок у жанрі 3D Action-Adventure з елементами RPG. Основним завданням ігрового програмного застосунку в жанрі Action-Adventure з елементами RPG є створення захоплюючого світу, де гравець може відчувати себе частиною захоплюючої історії.

Action-Adventure з елементами RPG – це жанр комп'ютерних ігор, який поєднує у собі низку характеристик та механік з різних жанрів, створюючи унікальний ігровий досвід для гравців. Основними особливостями цього жанру є:

- а) Екшн-орієнтований геймплей: Гравцеві пропонуються екшн-орієнтовані завдання та виклики, такі як битви з ворогами, розв'язування головоломок та дослідження великих світів.
- б) Пригодницька складова: Гра пропонує велику кількість пригодницьких елементів, таких як дослідження різних локацій, знаходження секретів та виконання місій.
- в) Рольова система: Гравці мають можливість налаштовувати та розвивати свого персонажа, вибираючи його навички, характеристики та екіпірування. Це дозволяє створити унікального героя, який відповідає ігровому стилю гравця.
- г) Сюжетна лінія: Часто в цьому жанрі присутня глибока сюжетна лінія, що розгортається в міру проходження гри.

Жанр Action-Adventure з елементами RPG є важливим сегментом ігрової індустрії, що пропонує глибокий та захоплюючий ігровий досвід. Цей жанр ігор відкриває перед гравцями низку можливостей для занурення в атмосферу віртуального світу. Завдяки вдалому поєднанню бойових елементів, пригод та системи розвитку персонажа, він забезпечує глибокий та різноманітний геймплей, що привертає увагу широкого кола гравців.

Метою роботи є розробка програмної системи, що складається з десктопного додатку. Для розробки продукту було використано ігровий рушій Unreal Engine 5 та мови програмування C++ та Blueprint. Середовища розробки – Unreal Editor та JetBrains Rider.

## 1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

### 1.1 Аналіз предметної галузі

Жанр Action-Adventure є одним з найпопулярніших і впливових сегментів ігрової індустрії. Передові розробники постійно досліджують цей жанр, створюючи новий та захоплюючий ігровий досвід для гравців.

Жанр Action-Adventure має велику популярність серед гравців усіх вікових категорій та географічних регіонів. Ігри такого типу часто стають бестселерами та здобувають високі оцінки від критиків [1]: Metal Gear Solid (94/100), God of War: Ragnarök (94/100), The Witcher 3: Wild Hunt (93/100), Dishonored (91/100).

Ринок ігор у жанрі Action-Adventure є досить конкурентним. Розробники постійно змагаються за увагу гравців, шукаючи нові способи відзначитися серед конкурентів.

### 1.2 Жанрові характеристики

Основні риси жанру Action-Adventure [2]:

- Екшен: Головною характеристикою жанру є наявність інтенсивних екшен-сцен, включаючи бойові сцени, швидкі перегони, а також інші захоплюючі події, що створюють динамічний та захоплюючий геймплей.
- Пригоди: Цей жанр часто передбачає дослідження великих та різноманітних світів, виконання завдань, пошук секретів та розв'язування головоломок, що додає глибину занурення та відчуття пригоди.
- Історія та персонажі: Як правило, в іграх жанру Action-Adventure великий акцент робиться на наявності захоплюючої історії та харизматичних персонажів, які ведуть гравця через вражаючі пригоди та інтриги.

Роль RPG-елементів [2]:

- Система розвитку персонажа: Багато героїв у жанрі Action-Adventure можуть пройти розвиток, підвищуючи свої навички, отримуючи нові здібності та збільшуючи статистику своїх характеристик.
- Прокачка навичок та обладнання: Гравці можуть вдосконалювати навички

персонажа, отримувати нове обладнання та вдосконалювати існуюче, щоб стати більш ефективними у боротьбі та дослідження світу.

Action-Adventure жанр приваблює гравців інтенсивним екшеном, пригодами та захоплюючою історією, в той час як RPG-елементи додають глибину геймплею та розвиток персонажів.

### 1.3 Аналіз механік

Типові механіки геймплею:

- Бойова система: Розгляд можливих варіантів бойової системи, таких як реальний час, поштовхи та пари, або система ходів.
- Дослідження світу: Вивчення можливостей для створення великих та різноманітних світів, що запрошують до дослідження.
- Розв'язування головоломок: Аналіз можливих головоломок, їх складності та інтеграції в загальний геймплей.

Елементи екшену:

- Стріляння та ближній бій: Дослідження можливих варіантів для реалізації бойових сцен, включаючи використання зброї та навички ближнього бою.
- Уникання ворогів та стратегія: Розгляд можливостей для впровадження системи уникнення або розумних ворогів, які вимагають від гравця використовувати стратегію.

RPG-механіки:

- Система розвитку персонажа: Визначення способів прокачки персонажа та розвитку його навичок.
- Збирання ресурсів та виготовлення: Дослідження можливостей для впровадження системи збирання ресурсів та створення нових предметів.
- Вибір варіантів розвитку персонажа: Розгляд можливостей для впливу гравця на характеристики свого героя та його розвиток.

Аналіз цих аспектів геймплею та механік допоможе створити цікавий та різноманітний геймплей, який буде задовольняти потреби різних типів гравців у жанрі Action-Adventure з елементами RPG.

## 1.4 Аналіз конкурентів

Під час аналізу попередніх досягнень та конкуренції у жанрі Action-Adventure з елементами RPG важливо визначити, які елементи здобули успіх та які можливості для вдосконалення можуть бути виявлені.

Приклади успішних ігор:

- а) The Legend of Zelda: Breath of the Wild [3]: Ця гра відома своїм великим відкритим світом, де гравці можуть вільно досліджувати та взаємодіяти з оточуючим середовищем. Вона також має захоплюючі головоломки та бойові сцени, які додають глибину геймплею.
- б) The Witcher 3: Wild Hunt [4]: Ця гра славиться своєю дивовижною глибиною історії та комплексними персонажами. Вона також має великий відкритий світ і вражаючу систему прокачки персонажа.

Розглянемо їхні сильні та слабкі сторони.

Сильні сторони:

- а) The Legend of Zelda: Breath of the Wild:
  - 1) Великий відкритий світ: Гра пропонує вражаюче великий світ, де гравці можуть вільно досліджувати різноманітні локації та взаємодіяти з ними;
  - 2) Інтерактивність та відкритість: Велика кількість можливостей для взаємодії з оточуючим середовищем дозволяє гравцям розвивати свої власні стратегії та подорожувати по світу відповідно до їхніх уподобань;
  - 3) Захоплюючі головоломки та бойові сцени: Гра пропонує різноманітні виклики, включаючи цікаві головоломки та захоплюючі бойові сцени, які роблять геймплей цікавим та різноманітним;
- б) The Witcher 3: Wild Hunt:
  - 1) Глибина історії та персонажів: Гра вражає своєю складною історією, яка захоплює гравця з першого ж моменту та дозволяє відчувати себе частиною унікального світу;
  - 2) Великий відкритий світ: Світ гри просто вражає своїм розмаїттям та

реалістичністю, дозволяючи гравцям вільно подорожувати та досліджувати його;

- 3) Вражаюча система прокачки персонажа: Гравці можуть вдосконалювати навички та обладнання свого героя, роблячи гру більш глибокою та цікавою.

Слабкі сторони:

а) The Legend of Zelda: Breath of the Wild:

- 1) Нестабільність технічної реалізації: Деякі гравці зіткнулися з проблемами технічної оптимізації та випадками низької частоти кадрів на деяких платформах;
- 2) Недоліки історії: Для деяких гравців історія може здатися не настільки захопливою або складною, порівняно з іншими аспектами гри;

б) The Witcher 3: Wild Hunt:

- 1) Складність для новачків: Деякі гравці можуть виявити гру складною для вивчення та розуміння, особливо якщо вони не знайомі з усією серією The Witcher;
- 2) Технічні проблеми: На деяких платформах можуть виникати проблеми з технічною оптимізацією та відтворенням, такі як довгі завантаження або випадки затримок у геймплеї;

Цей аналіз дозволяє виявити ключові успішні аспекти, а також слабкі сторони конкурентів, що надає можливість для подальшого вдосконалення і розвитку нової гри в жанрі Action-Adventure з елементами RPG.

### 1.5 Виявлення та вирішення проблем

Існують кілька потенційних проблем, з якими можуть виникнути у грі у жанрі Action-Adventure з елементами RPG.

Баланс геймплею. Створення гармонійного балансу між екшен-сценами та RPG-асpekтами може бути складною задачею. Нерівновага може призвести до того, що гравці відчуватимуть, що гра надто фокусується на одному з аспектів, або ж надмірно складне чи недостатньо цікаве завдання.

Технічні проблеми. Реалізація великих та деталізованих віртуальних світів може створювати технічні виклики, такі як проблеми з оптимізацією продуктивності, завантаженням ресурсів та інші проблеми, які можуть вплинути на ігровий досвід.

Недостатня інноваційність. У насиченому ринку ігор інновації відіграють ключову роль у приверненні уваги гравців. Недостатність новаторських ідей може призвести до того, що гра буде сприйматися як стандартна та неприваблива для аудиторії.

Незадовільна історія та персонажі. Глибока та захоплююча історія разом з цікавими персонажами є ключовими складовими успішної гри в жанрі Action-Adventure. Нездатність розробників створити ці елементи може призвести до втрати інтересу гравців.

Монетизація та мікротранзакції. Несправедлива або надмірна система монетизації може викликати негативні реакції гравців і підірвати загальний досвід від гри.

Уникнення цих проблем потребує детального планування, вивчення вимог аудиторії та постійного вдосконалення ігрового процесу.

## 1.6 Постановка задачі

Для вирішення потреб користувачів, потрібно розробити ігровий програмний застосунок в жанрі Action-Adventure з елементами RPG, який буде захоплюючим досвідом для гравців. Основними цілями розробки є створення гармонійного балансу між екшен- та RPG-аспектами геймплею, створення великого та деталізованого віртуального світу, розробка цікавої історії та персонажів, оптимізація продуктивності та вирішення технічних проблем, таких як завантаження ресурсів. Результатом проекту має бути гра, яка буде конкурентоспроможною на ринку ігор, отримає позитивні відгуки від гравців та критиків і забезпечить задоволення як для новачків, так і для досвідчених гравців жанру.

## 2 ФОРМУВАННЯ ВИМОГ ДО ПРОГРАМНОЇ СИСТЕМИ

### 2.1 Постановка мети

Метою виконання роботи є створення ігрового програмного застосунку на рушії Unreal Engine 5.

У грі повинні бути реалізовані базові механіки жанру, включаючи механіки, пов'язані з гравцем, супротивниками та оточенням. Для супротивників має бути реалізований штучний інтелект, що імітує реалістичну поведінку. Також має бути реалізований інтерфейс користувача, що включає у себе HUD та різноманітні вікна.

### 2.2 Загальний опис

У грі мають бути реалізовані такі механіки:

- а) Переміщення персонажів:
  - 1) Якщо це головний герой – то за допомогою миші/клавіатури;
  - 2) Якщо це не головний герой – то за допомогою ШІ;
- б) Система інвентарю;
- в) Бойова система ближнього та дальнього бою;
- г) Різні види зброї як здібності;
- д) Система покращень зброї;
- е) Система квестів;
- ж) Налаштування аудіо та відео.

Інтерфейс користувача має мати наступні елементи:

- а) Головне меню;
- б) Налаштування;
- в) Пауза;
- г) Інвентар;
- д) Спорядження;
- е) Індикатори здоров'я та ресурсів;
- ж) Панель навичок;
- и) Панель діалогу;

к) Панель завдань.

У грі має бути реалізовано 3 типи противників: паровий павук, бандит і кіборг.

### 2.3 Загальні обмеження

Ігровий програмний застосунок має бути реалізований на ігровому рушії Unreal Engine 5.2 з використанням мов програмування C++ версії 20 та Blueprint. Інтерфейс користувача має бути реалізований за допомогою технології Unreal Motion Graphics.

Мінімальні системні вимоги:

- Процесор: Intel Core i5-8400 або AMD Ryzen 3 3300X.
- ОЗП: 12 GB RAM.
- Відеокарта: AMD Radeon™ RX 5700 / NVIDIA® GeForce® GTX 1080.
- DirectX: Version 12
- Місце на диску: 50 GB на SSD

Ігровий програмний застосунок має працювати на операційній системі Windows 10/11 x64.

### 2.4 Припущення та залежності

Під час розробки розглядаються наступні припущення та залежності:

- Ігровий програмний застосунок може бути запущено на версії Windows, що нижча за 10, наприклад 7.
- Ігровий програмний застосунок може бути запущено на комп'ютері, що не відповідає мінімальним системним вимогам.

У першому випадку достатньо складно визначити, які наслідки будуть, адже гра покладається на DirectX 12, який не має підтримки версій Windows нижчих за 10. Але, наразі кількість гравців, що використовують такі версії дуже мала, тому це не має бути великою проблемою. Щодо другого випадку, у грі передбачаються налаштування відео, що можуть допомогти з лагами у грі на слабкому залізі.

### 3 АРХІТЕКТУРА ТА ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

#### 3.1 UML проєктування ПЗ

Діаграма прецедентів є засобом візуалізації, що використовується для опису функціональних можливостей системи з точки зору користувача.

Для розроблюваного ігрового застосунку було розроблено діаграму прецедентів, що відповідають діям користувача у грі (див. рис. 3.1).

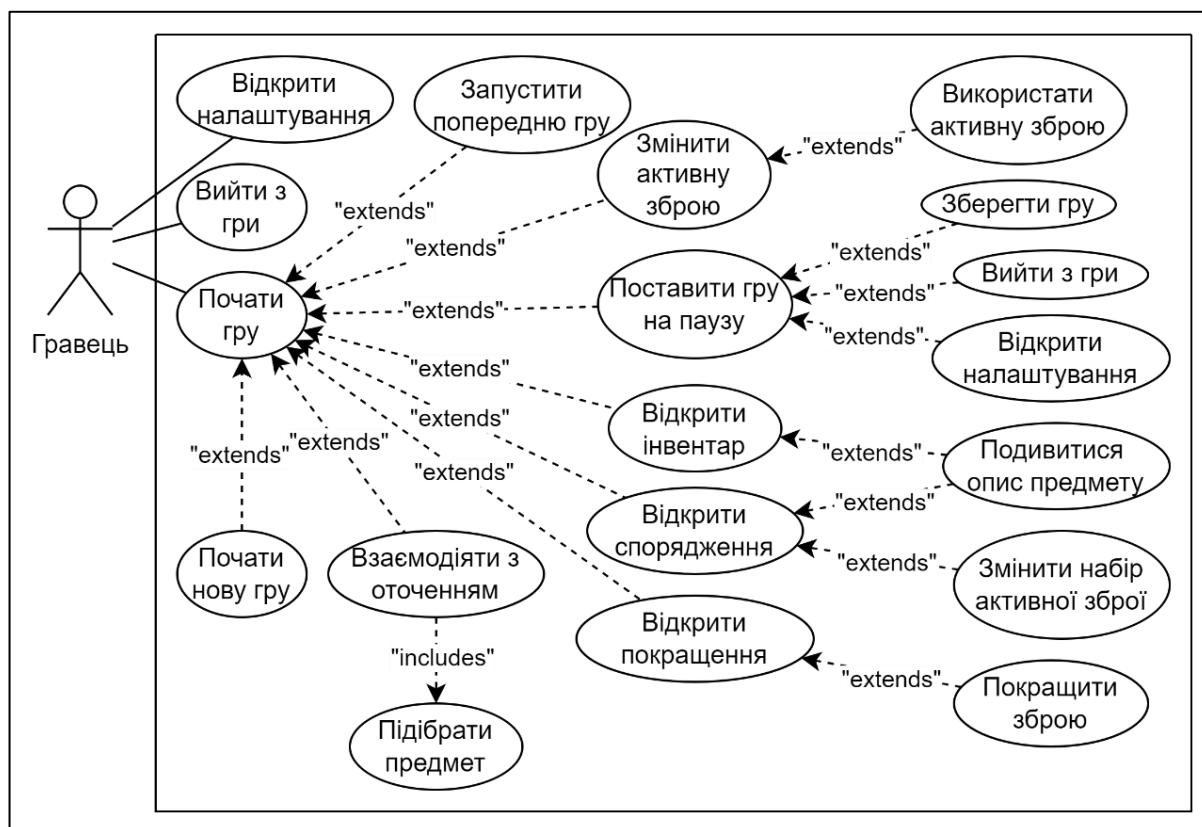


Рисунок 3.1 – Діаграма прецедентів (рисунок виконаний самостійно)

Ця діаграма відображає взаємодію між ігровим застосунком та гравцем через ідентифікацію основних дій, які можуть бути виконані в системі. Цей тип діаграми допомагає зрозуміти потреби користувачів та визначити основні функції системи з їхнього погляду.

#### 3.2 Проєктування архітектури ПЗ

Для ігрового застосунку була обрана компонентна архітектура, так як вона є найбільш підходящою для ігор.

Ключовим компонентом є ігровий режим, який об'єднує у собі компоненти гравця (див. рис. 3.2)

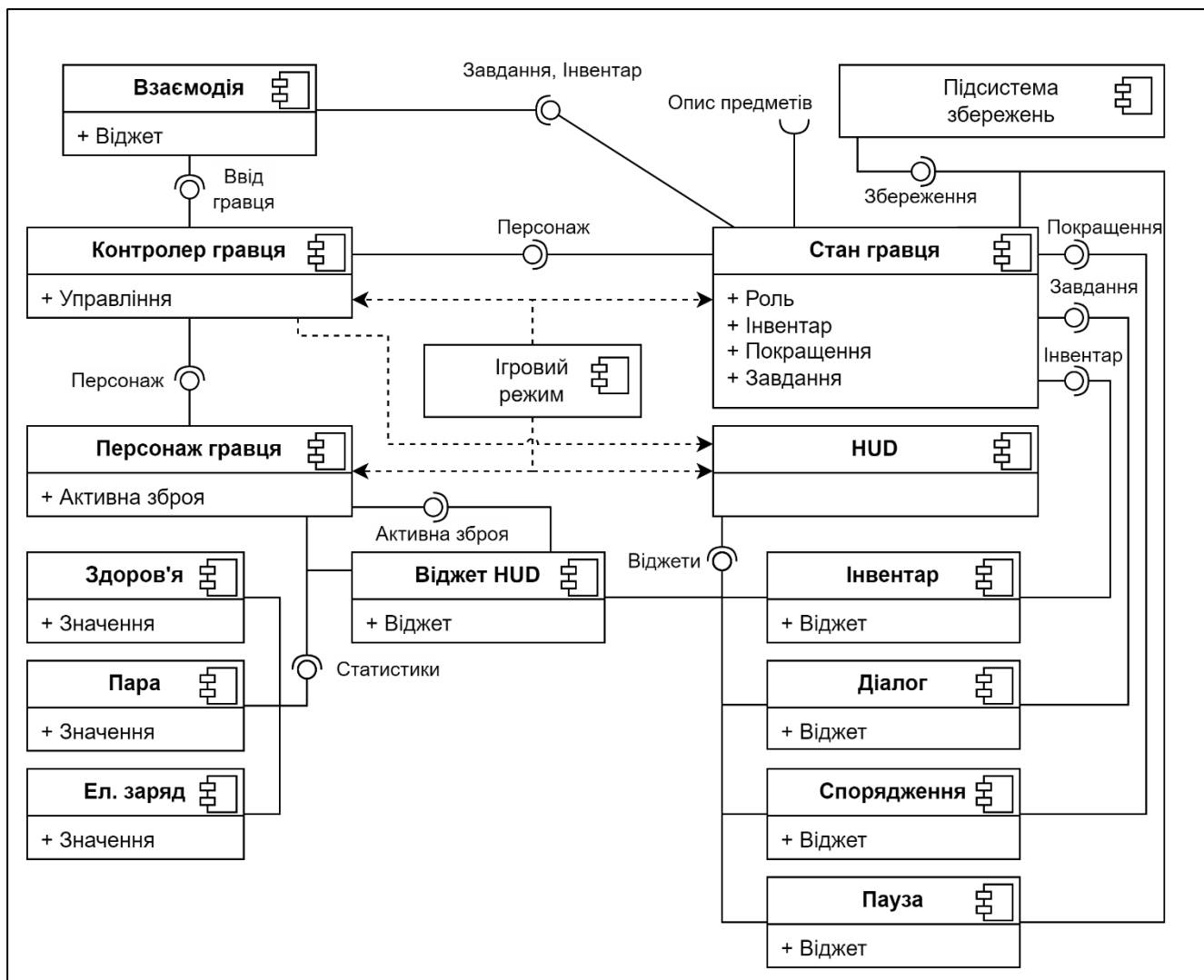


Рисунок 3.2 – Діаграма компонентів, пов'язаних з ігровим режимом (рисунок виконаний самостійно)

Ігровий режим задає 4 головні компоненти гравця:

- Контролер, що є інтерфейсом між гравцем і персонажем, та відповідає за обробку користувацького вводу;
- Стан, що відповідає за такі речі, як інвентар, покращення та завдання гравця;
- Персонаж, що відповідає за усе, що пов'язано з аватаром гравця у грі;
- HUD, що відповідає за інтерфейс користувача.



б) У персонажа було достатньо пари та/або електричного заряду.

Тоді і тільки тоді створюється снаряд і йому надається відповідна швидкість. Також, для поліпшення ігрового досвіду, після пострілу, якщо необхідно, почнеться перезарядка.

Наступним алгоритмом є алгоритм поведінки супротивників у грі (див. рис. 3.4).

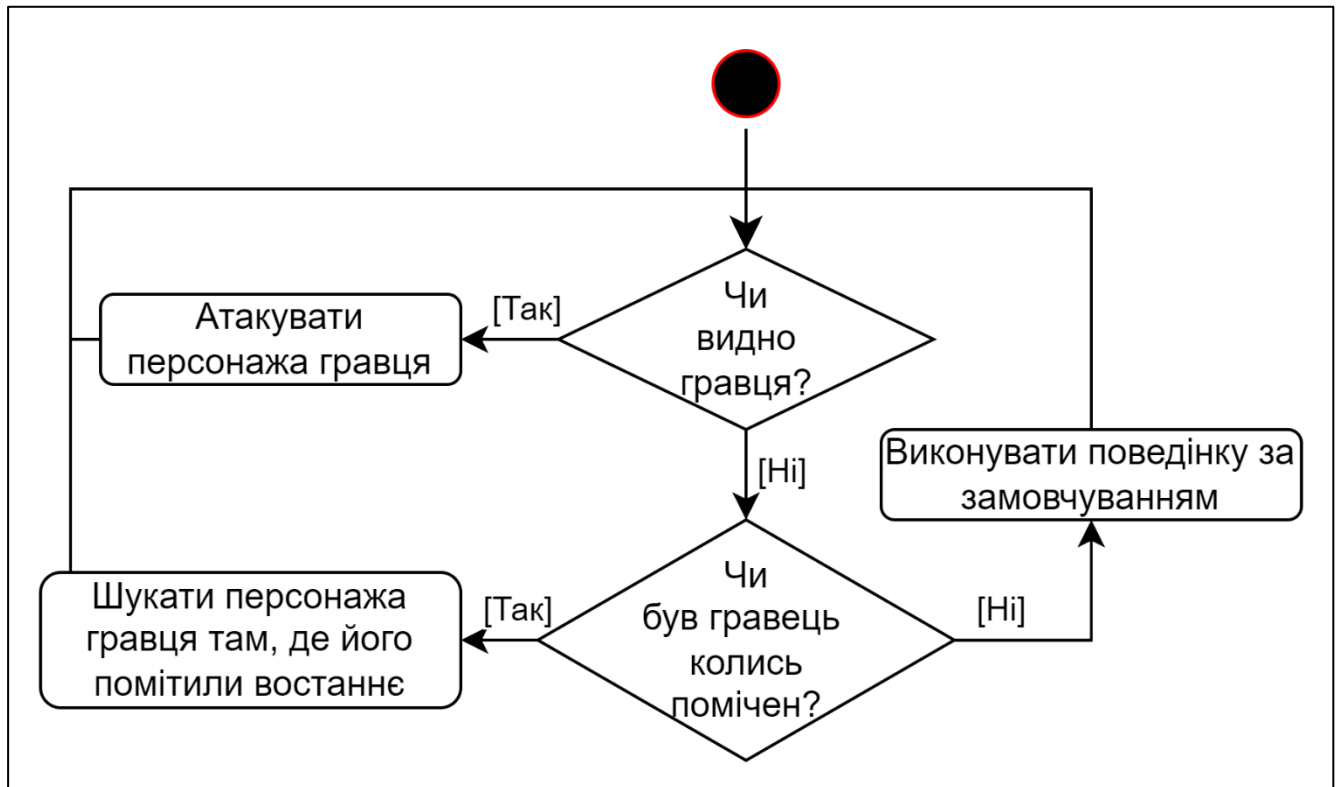


Рисунок 3.4 – Діаграма активності для поведінки супротивників (рисунок виконаний самостійно)

Алгоритм полягає у наступному:

- а) Якщо персонажа гравця видно, то атакувати його;
- б) Якщо персонажа не видно, але його було помічено нещодавно, то шукати його у місці, де його помітили востаннє;
- в) В інших випадках виконувати поведінку за замовчуванням, наприклад озиратися або йти у випадкову точку в радіусі.

Незважаючи на простоту, цей алгоритм з деякими модифікаціями може бути використано для реалізації поведінки майже будь-якого простого супротивника.

### 3.4 Проектування UI / UX

Головне меню (див. рис. 3.5) з'являється при вході.

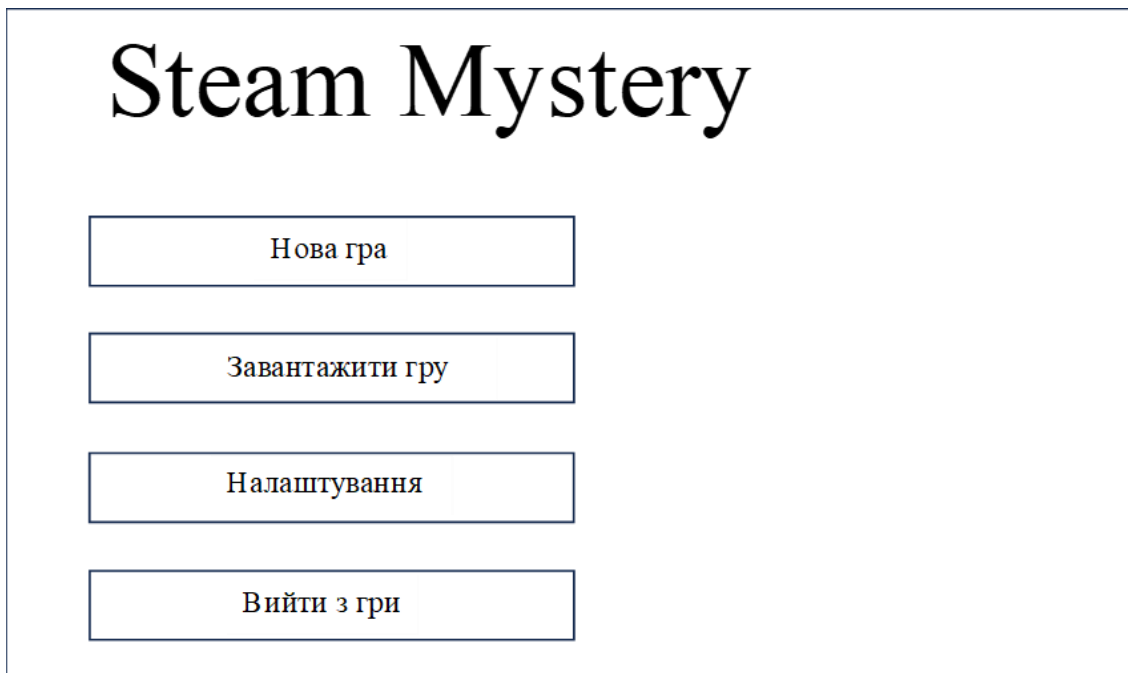


Рисунок 3.5 – Головне меню (рисунок виконаний самостійно)

Тут гравець може здійснити налаштування (див. рис. 3.6), почати нову гру, або завантажити раніше збережену.



Рисунок 3.6 – Вікно налаштувань (рисунок виконаний самостійно)

Коли гравець починає нову гру, або завантажує вже збережену, то він бачить ігровий світ та HUD (див. рис. 3.7).

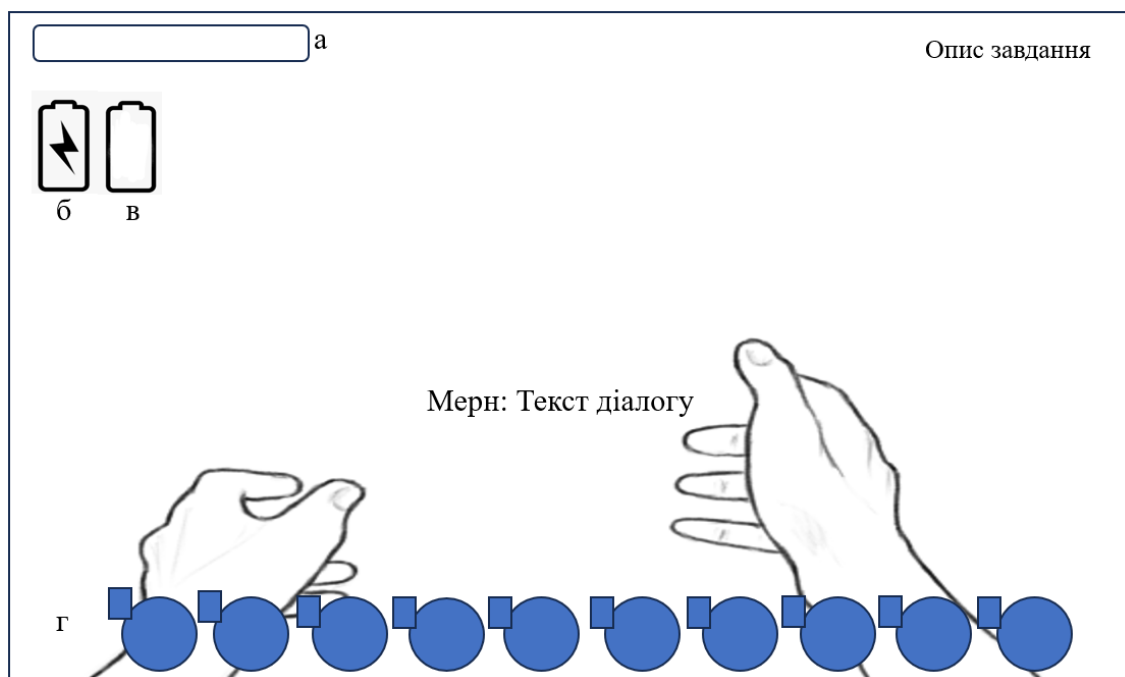


Рисунок 3.7 – HUD гравця (рисунок виконаний самостійно)

На рисунку можна побачити:

- а) Панель здоров'я;
- б) Кількість заряду;
- в) Кількість пари;
- г) Панель швидкого доступу.

Також на рисунку присутні такі елементи, як опис завдання та активний діалог.

Знаходячись у грі, гравець може відкрити інвентар (див. рис. 3.8), натиснувши на клавішу «І».

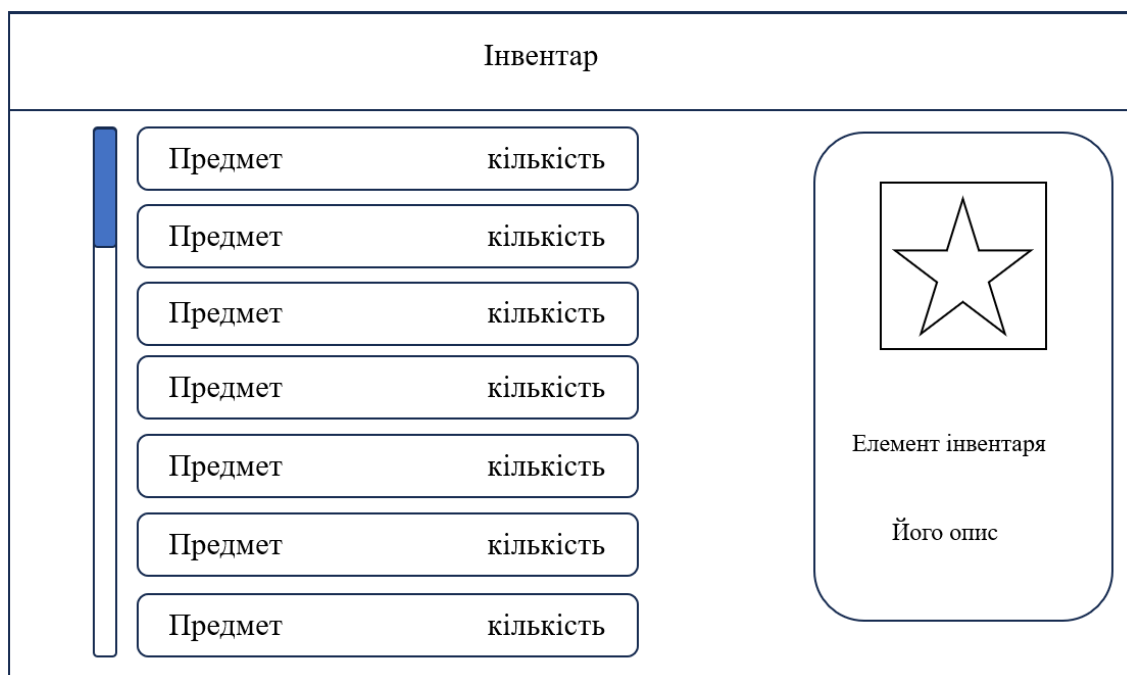


Рисунок 3.8 – Вікно інвентаря (рисунок виконаний самостійно)

Також гравець може відкрити вікно спорядження (див. рис. 3.9), натиснувши на клавішу «О».

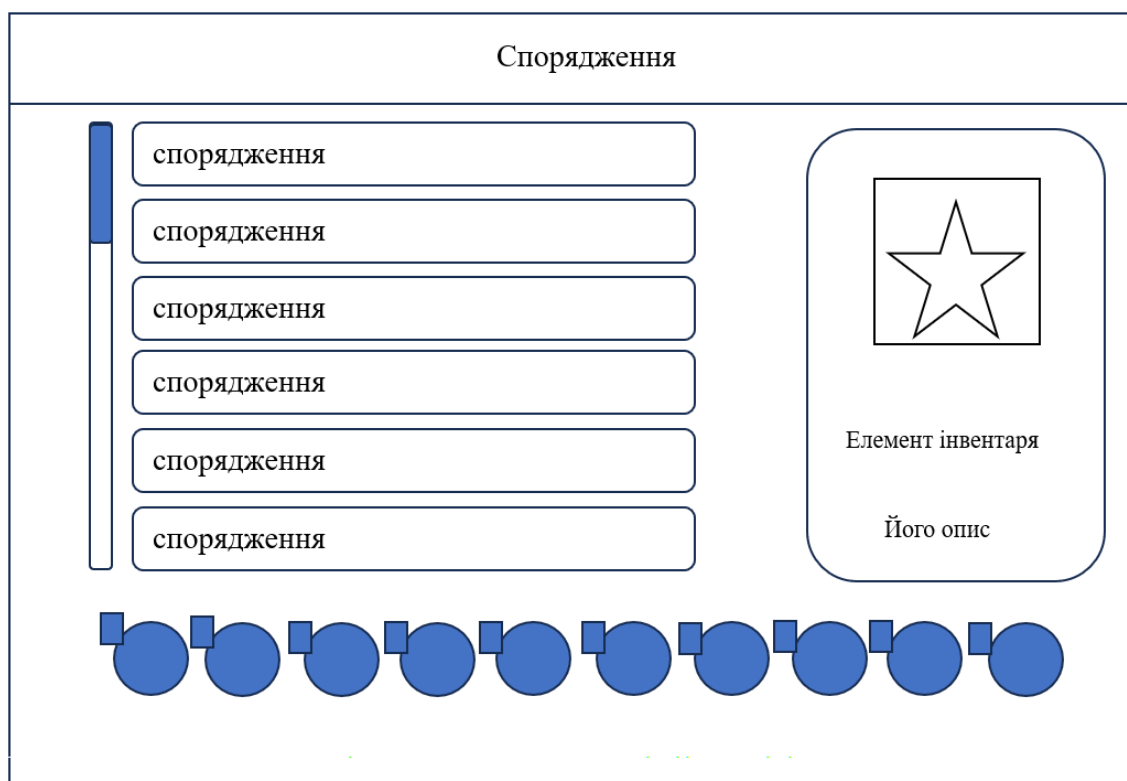


Рисунок 3.9 – Вікно спорядження (рисунок виконаний самостійно)

Також гравець може поставити гру на паузу (див. рис. 3.10), натиснувши

кнопку «Esc», де він може зберегтися, налаштувати гру, або вийти з неї.

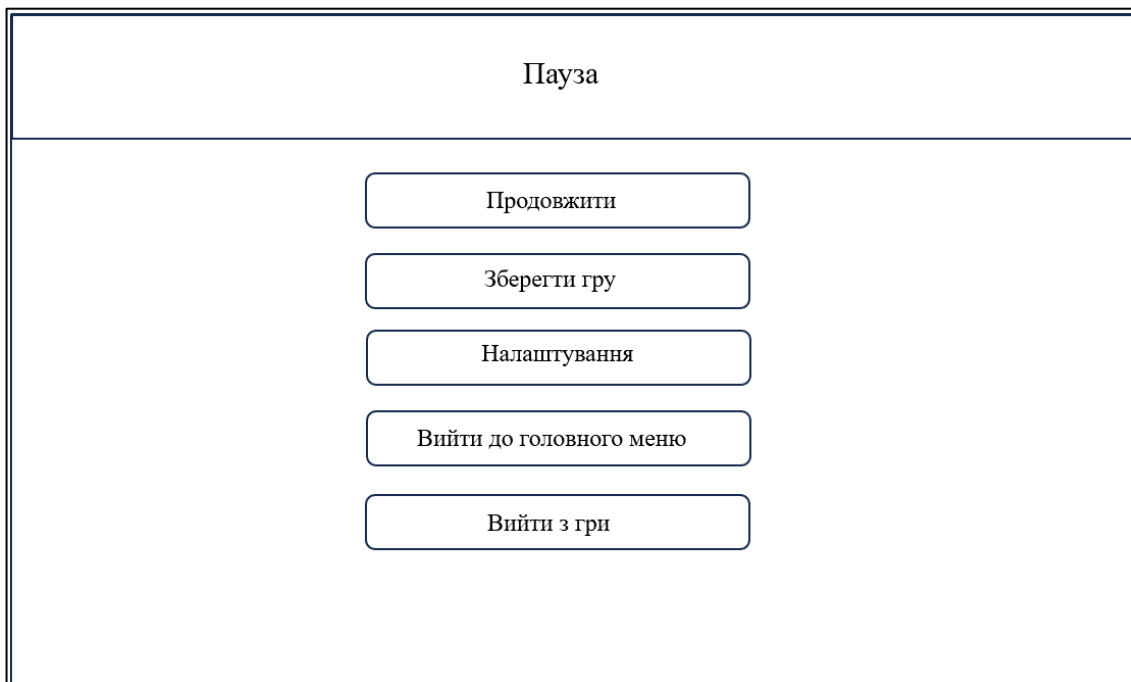


Рисунок 3.10 – Вікно паузи (рисунок виконаний самостійно)

При взаємодії з верстаком, відкривається меню з покращенням зброї (див. рис. 3.11).

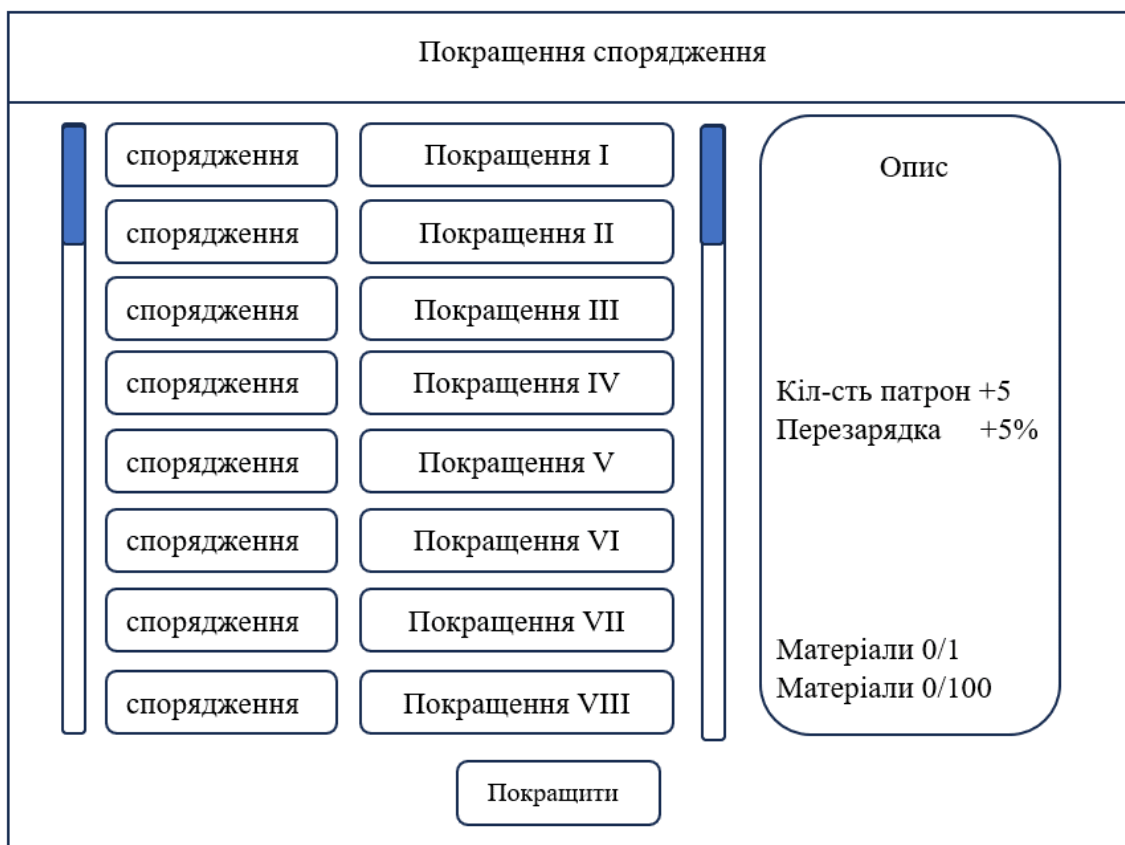


Рисунок 3.11 – Вікно покращення спорядження (рисунок виконаний самостійно)

Гравцю доступне наступне керування у грі:

- Переміщення – кнопки «WASD»;
- Пересування камери – повороти мишею;
- Стрибок – кнопка «Space»;
- Присідання – кнопка «Ctrl»;
- Спринт – кнопка «Shift»;
- Взаємодія – кнопка «E»;
- Відкриття вікна паузи – кнопка «Esc»;
- Відкриття вікна інвентаря – кнопка «I»;
- Відкриття вікна спорядження – кнопка «O»;
- Використання зброї – ліва кнопка миші.

З усіма елементами інтерфейсу, окрім HUD, гравець може взаємодіяти мишею.

## 4 ОПИС ПРИЙНЯТИХ ПРОГРАМНИХ РІШЕНЬ

### 4.1 Рух снарядів для зброї дальнього бою

Для створення рухомих снарядів використовувалось 2 технології: трасування ліній[5] та компонент `UProjectileMovementComponent` [6].

Трасування ліній визначало кінцеву точку, де снаряд мав би опинитися. Математично це можна описати формулою 4.1.

$$M' = \min \begin{cases} M_{impact} \\ M_0 + vd \end{cases} \quad (4.1)$$

де  $M_{impact}$  – точка перетину лінії трасування із будь-яким актором;

$M_0$  – точка погляду гравця або NPC;

$v$  – нормований вектор погляду гравця;

$d$  – максимальна дальність пострілу зброї;

$M'$  – кінцева точка польоту снаряду.

Код функції, що рахує точку перетину:

```
bool AWeapon::Sweep(FHitResult& HitResult,
                   const FVector& TraceStart,
                   const float Range) const
{
    if(const auto OwningCharacter = Cast<APawn>(GetOwner()))
        if(const auto OwnerController = OwningCharacter->GetController())
        {
            FVector Start, Direction;
            FRotator Rotation;
            OwnerController->GetPlayerViewPoint(Start, Rotation);
            if(auto AIController = Cast<AAIController>(OwnerController))
            {
                double Length;
                const auto Vector = AIController->GetFocalPoint() - Start;
                Vector.ToDirectionAndLength(Direction, Length);
            }
            else
                Direction = Rotation.Vector();
            const FVector TraceEnd = Start + Direction * Range;
            auto Params = FCollisionQueryParams::DefaultQueryParam;
            Params.AddIgnoredActor(GetOwner());
            Params.AddIgnoredActor(this);
            HitResult.TraceEnd = TraceEnd;
        }
}
```

```

        return GetWorld()->LineTraceSingleByChannel(
            HitResult, TraceStart, TraceEnd,
            ECC_EngineTraceChannel2, Params);
    }
    return false;
}

```

Тепер, коли ми маємо точки перетину та максимальну точку польоту, ми можемо вирахувати фактичну кінцеву точку польоту снаряду:

```

const auto TraceEnd = Sweep(HitResult, TraceStart, Range)
    ? HitResult.ImpactPoint
    : HitResult.TraceEnd;

```

Далі, все що залишилося – це отримати відповідну швидкість снаряду, скориставшись методом `UGameplayStatics::SuggestProjectileVelocity`:

```

FVector TossVelocity;
if (!UGameplayStatics::SuggestProjectileVelocity(
    GetWorld(), TossVelocity, StartLocation, EndLocation, TossSpeed,
    false, 0, 0,
    ESuggestProjVelocityTraceOption::DoNotTrace,
    FCollisionResponseParams::DefaultResponseParam,
    TArray<AActor*>(), bDrawDebug))

    TossVelocity = UKismetMathLibrary::GetDirectionUnitVector(
        StartLocation, EndLocation) * TossSpeed;

MovementComponent->SetVelocityInLocalSpace(TossVelocity);

```

Таким чином, для зброї, яка використовує повільні снаряди і для якої не можна використовувати лише трасування по лінії, реалізовано рух снарядів. І при цьому, кінцева точка майже співпадає з точкою прицілу гравця, тому не створюється дискомфорт у геймплеї через невизначеність траєкторії польоту.

## 4.2 Штучний інтелект противників

У Unreal Engine 5 найзручнішим та найпотужнішим інструментом для проектування штучного інтелекту є механізм дерев поведінки [7] (див. рис. 4.1).

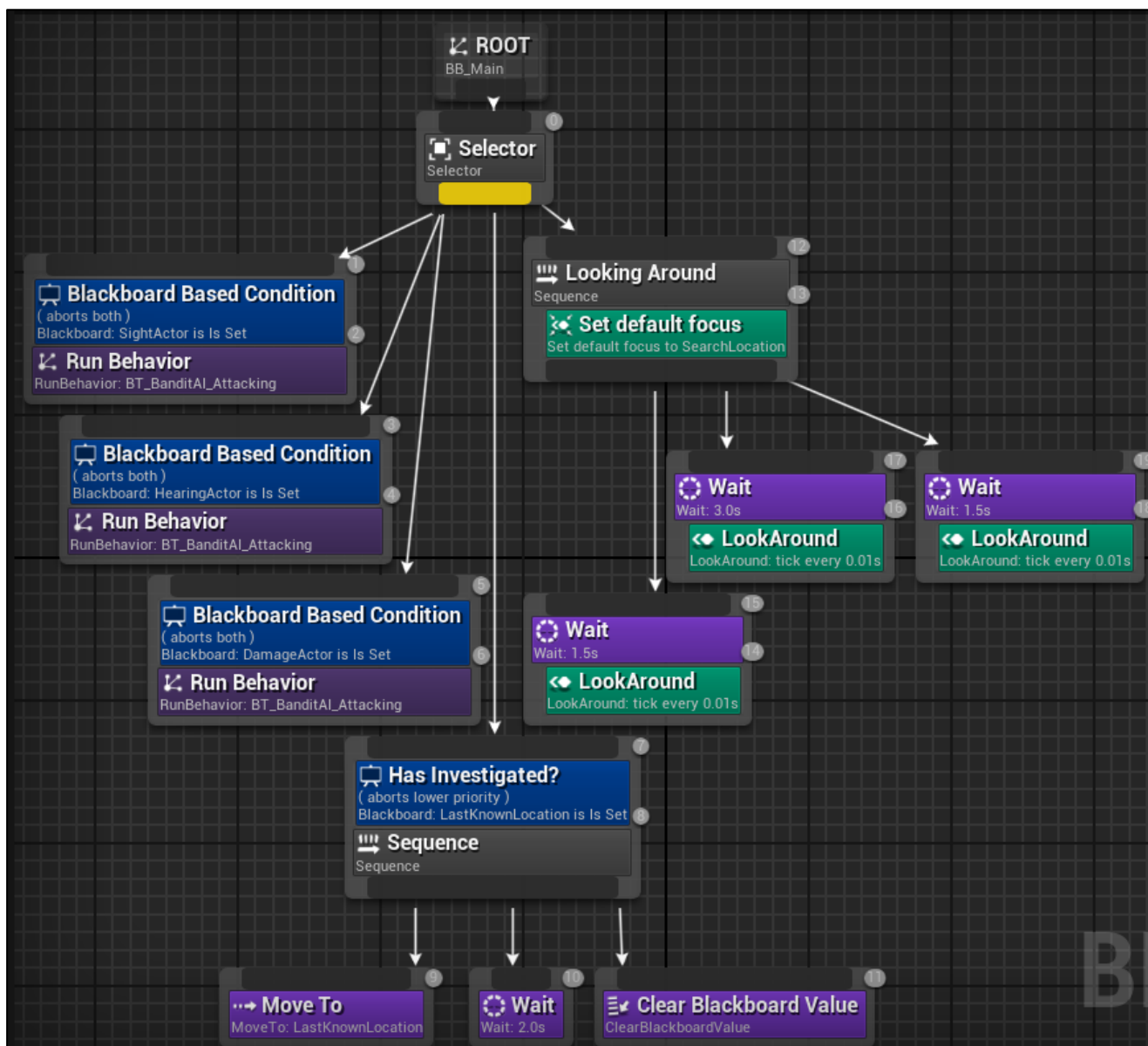


Рисунок 4.1 – Дерево поведінки для бандиту (рисунок виконаний самостійно)

Дерева поведінки для усіх противників схожі, і складаються з 3 частин у порядку пріоритету:

- а) Атакувати гравця (пріоритети 1-6 на рисунку 4.1);
- б) Шукати гравця (пріоритети 7-11 на рисунку 4.1);
- в) Поведінка за замовчуванням (пріоритети 12-19 на рисунку 4.1).

Гілка атаки запускає піддерево атаки, яке унікальне для кожного супротивника і представляє собою виклики доступних атак, зважаючи на час відновлення здібностей або зброї (див. рис. 4.2).

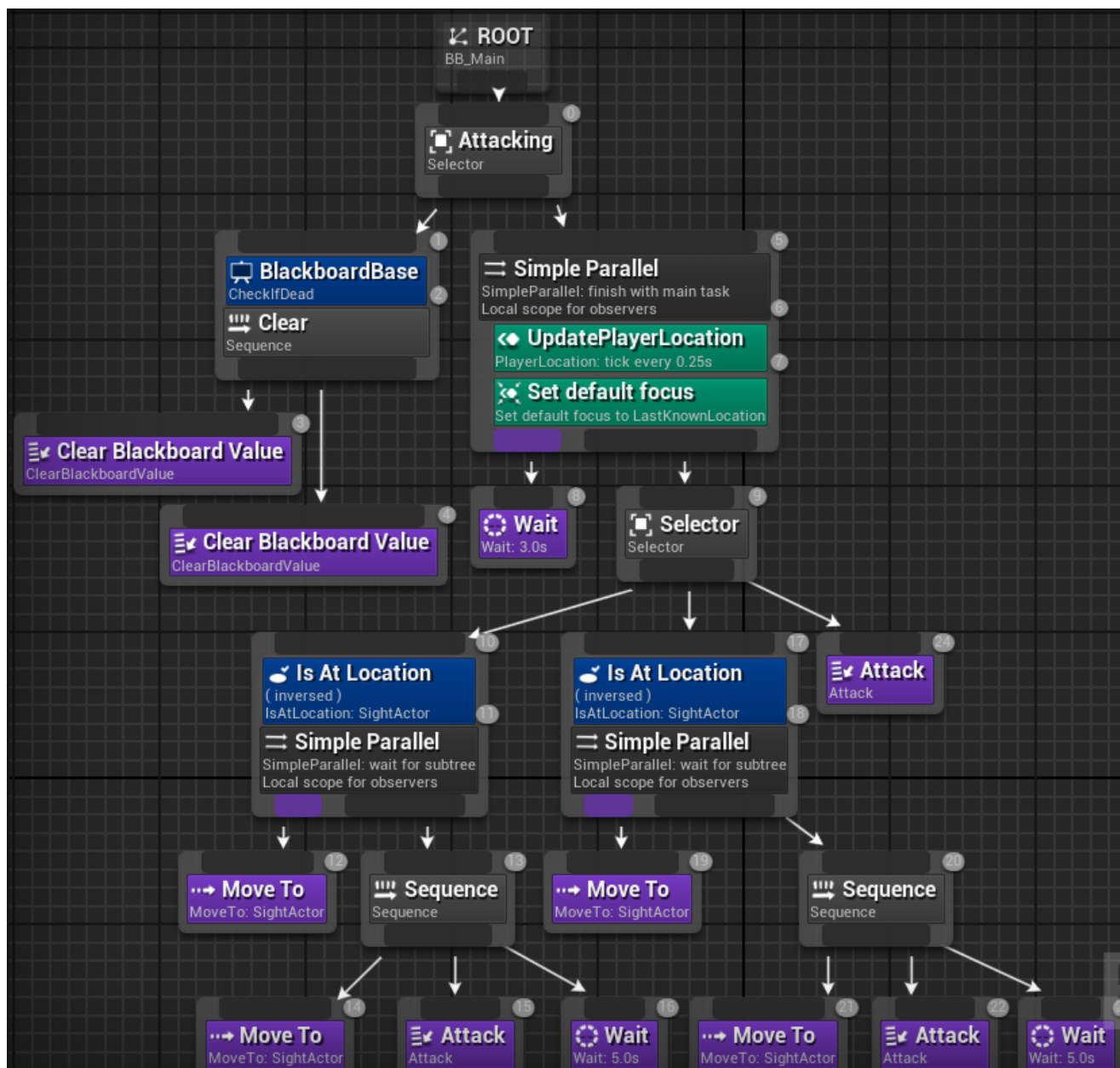


Рисунок 4.2 – Дерево поведінки атаки кіборга (рисунок виконаний самостійно)

По перше йде перевірка на те, чи живий гравець. Якщо ні, то немає сенсу його атакувати. Якщо живий, то кожні 0.25 секунди противник фокусується на гравці і, в залежності від відстані до гравця, використовує одну з трьох здібностей і намагається приблизитися для атаки ближнього бою.

Якщо ж противник загубив з виду гравця, він йде до місця, де бачив його востаннє. Якщо й там він не знайде гравця, то тоді буде поведінка за замовчуванням, наприклад стояти та озиратися (бандит), чи йти у довільну точку у радіусі (павук).

### 4.3 Розробка інтерфейсів інвентарів

Усі інтерфейси у грі розроблені за допомогою технології Unreal Motion Graphics [8]. Головним класом у UMG є UUserWidget, від якого спадкуються усі інші класи-віджети.

Логіка багатьох віджетів передбачає доступ до спеціального класу APlayerHUD, що є нащадком AHUD, який відповідає за створення та доступ до віджетів інтерфейсу користувача у грі. Щоб дотримуватися принципу Don't Repeat Yourself, було вирішено винести логіку отримання посилання на APlayerHUD у окремий клас:

```
UCLASS()
class UBaseHUDWidget : public UUserWidget
{
    GENERATED_BODY()

protected:
    virtual void NativeOnInitialized() override;

    UPROPERTY(BlueprintReadOnly)
    class APlayerHUD* HUD;
};

void UBaseHUDWidget::NativeOnInitialized()
{
    HUD = Cast<APlayerHUD>(GetOwningPlayer()->GetHUD());
    Super::NativeOnInitialized();
}
```

У грі є три віджети, що взаємодіють з інвентарем:

- а) Інвентар;
- б) Спорядження;
- в) Покращення.

Усі вони мають спільну ознаку: вони відображають список предметів, що є у гравця у вигляді списку. Отже було вирішено винести це у окремий клас UItemCollectionWidget:

```
UCLASS()
class STEAMMYSTERY_API UItemCollectionWidget : public UBaseHUDWidget
{
```

```

GENERATED_BODY()

protected:
    virtual void NativeOnInitialized() override;
    virtual void Sync();

    UPROPERTY()
    UBaseGameInstance* GameInstance;

    void FillItems(
        UPanelWidget* PanelWidget,
        const UDataTable* InDataTable,
        TMap<FName, int32> InItems,
        const TSubclassOf<UBaseItemWidget> ItemWidgetClass
    ) const;
};

void UItemCollectionWidget::NativeOnInitialized()
{
    Super::NativeOnInitialized();
    GameInstance =
Cast<UBaseGameInstance>(UGameplayStatics::GetGameInstance(GetWorld()))
);
}

void UItemCollectionWidget::Sync()
{
}

void UItemCollectionWidget::FillItems(
    UPanelWidget* PanelWidget,
    const UDataTable* InDataTable,
    TMap<FName, int32> InItems,
    const TSubclassOf<UBaseItemWidget> ItemWidgetClass
) const
{
    PanelWidget->ClearChildren();
    if (ItemWidgetClass && InDataTable)
        for (const auto Element : InItems)
            if(const auto Widget =
CreateWidget<UBaseItemWidget>(GetOwningPlayer(), ItemWidgetClass))
                {
                    FDataTableRowHandle RowHandle;
                    RowHandle.DataTable = InDataTable;
                    RowHandle.RowName = Element.Key;
                    Widget->SetItem(RowHandle, Element.Value);
                    PanelWidget->AddChild(Widget);
                }
}

```

Пуста функція Sync використовується як абстрактна функція, що використовуватиметься для синхронізації предметів у класах-нащадках.

Функція FillItems отримує віджет-контейнер, таблицю даних про предмети, кількість предметів у вигляді словнику, де ключ – це назва предмету, а значення –

його кількість, та тип віджету, що використовуватиметься для відображення предмету. Це функція проходить по словнику предметів, та створює відповідні віджети, заповнюючи їх даними.

Далі, на основі цього класу створюються класи-нащадки для інвентаря, спорядження та покращень.

```

UCLASS()
class STEAMMYSTERY_API UInventoryWidget : public
UItemCollectionWidget
{
    GENERATED_BODY()

protected:
    virtual void NativeConstruct() override;
    virtual void Sync() override;

    UPROPERTY(meta=(BindWidget))
    UPanelWidget* Items;

    UPROPERTY(meta=(BindWidget))
    class UItemDescriptionWidget* DescriptionContainer;

    UPROPERTY(EditAnywhere)
    TSubclassOf<UItemWidget> ItemWidgetClass;
};

void UInventoryWidget::NativeConstruct()
{
    Super::NativeConstruct();
    if (const auto Controller = GetOwningPlayer())
        if (const auto Inventory =
            Controller->GetPlayerState<AMainPlayerState>())
            FillItems(Items, GameInstance->GetItemsDataTable(),
                Inventory->GetItems(), ItemWidgetClass);
    Sync();
}

void UInventoryWidget::Sync()
{
    Super::Sync();
    for (const auto Element : Items->GetAllChildren())
        if (const auto ItemWidget = Cast<UItemWidget>(Element))
            ItemWidget->OnClicked.AddUniqueDynamic(
                DescriptionContainer,
                &UItemDescriptionWidget::SetItem
            );
}

```

Кожний з нащадків реалізує свій метод Sync та викликає у методі NativeConstruct метод FillItems.

Візуальна частина створюється вже безпосередньо у UnrealEditor (див. рис 4.3).



Рисунок 4.3 – Візуальна частина вікна інвентаря у UnrealEditor (рисунок виконаний самостійно)

У подібній стилістиці також виконані й усі інші вікна та елементи інтерфейсу. Вони усі складені із багаторазових модулів, таких як:

- WBP\_Window – модуль рамки та фону вікна;
- WBP\_Item – модуль для відображення предмету;
- WBP\_Description – модуль для відображення опису предмету тощо.

Використання багаторазових модулів допомагає зберегти час на розробку інтерфейсів, досягти єдиного стилю для усіх інтерфейсів та, за необхідністю, централізовано змінювати стилістику компонентів.

## 5 ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 5.1 Тестування ігрового застосунку

Ігровий програмний застосунок був протестований відповідно до заздалегідь розробленого тест-плану, який наведено у додатку Д. Для створення цього тест-плану було використано Mind Map, або «карту думок» (див. рис. 5.1). Тестування проводилося за методологією Blackbox, тобто методом «чорної скриньки».

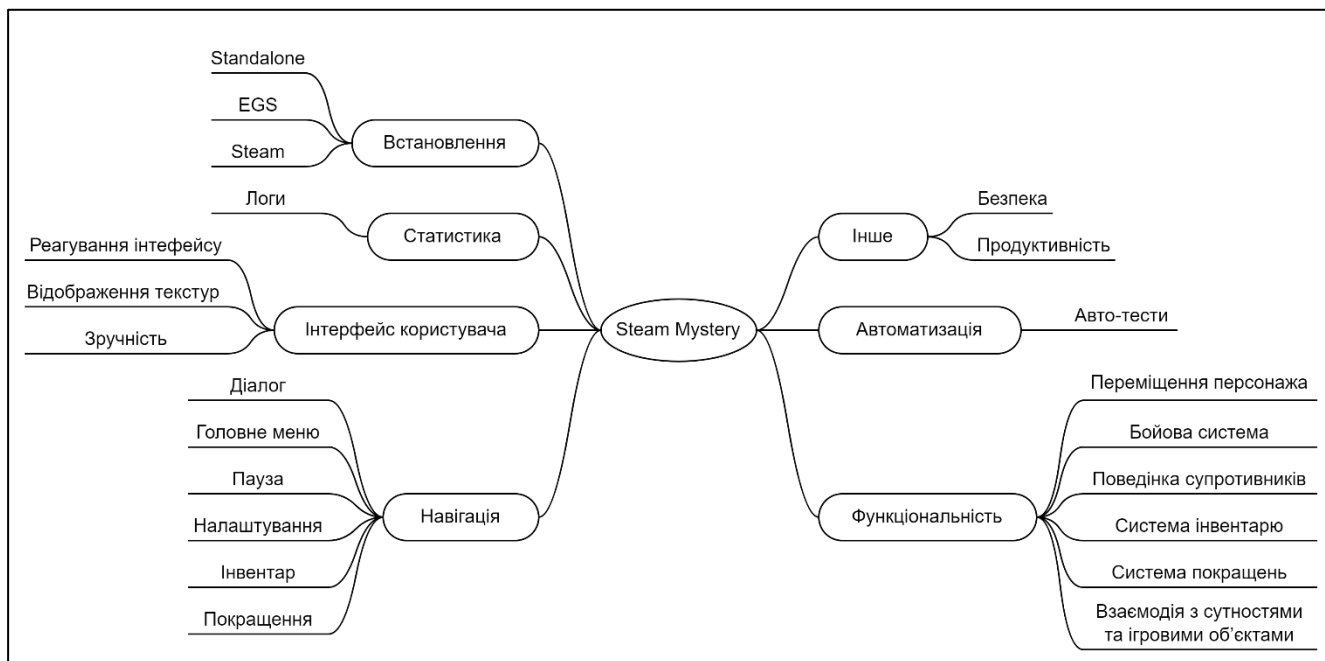


Рисунок 5.1 – Mind Map для гри «Steam Mystery» (рисунок виконаний самостійно)

Наведена карта дозволяє нам розділити тестування проекту на окремі частини:

- а) Функціональність;
- б) Навігація;
- в) Інтерфейс користувача;
- г) Статистика;
- д) Встановлення;
- е) Автоматизація;
- ж) Інше.

В процесі розробки було проведено тестування функціональності, навігації та інтерфейсу користувача.

## 5.2 Виявлені помилки

В процесі тестування було виявлено 10 помилок. Ці помилки було задокументовано у вигляді баг-репортів, що мають таку структуру:

- а) Назва помилки;
- б) Короткий опис проблеми, що явно вказує на причину і тип помилкової ситуації;
- в) Компонент програми, що тестується;
- г) Серйозність:
  - 1) S1 Блокуючий;
  - 2) S2 Критичний;
  - 3) S3 Значний;
  - 4) S4 Незначний;
  - 5) S5 Тривіальний.
- д) Пріоритет:
  - 1) P1 Високий;
  - 2) P2 Середній;
  - 3) P3 Низький.
- е) Кроки відтворення, якими можна легко відтворити ситуацію, що призвела до помилки;
- ж) Фактичний результат;
- и) Очікуваний результат;
- к) Прикріплений файл, який може допомогти прояснити причину помилки або вказати на спосіб вирішення проблеми.

Усі баг-репорти наведено у додатку Е.

Тестування проводилося у процесі написання програмного коду. Усі баги з серйозністю «S2 Критична» та вище були виправлені.

## 6 ВПРОВАДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 6.1 Наукове впровадження проекту

За тематикою кваліфікаційної роботи було написано наукову роботу у вигляді тез для V Міжнародної студентської наукової конференції «Тренди та перспективи розвитку мультидисциплінарних досліджень», яка відбулася 26 квітня 2024 року у місті Одеса, Україна.

Представлену роботу було опубліковано у відповідному збірнику конференції під заголовком «Usage of ProjectileMovementComponent and LineTrace for projectile simulation in Unreal Engine 5» [9]. У роботі описуються проблеми симуляції снарядів у іграх від третьої особи та спосіб їх вирішення за допомогою трасування ліній та використання компонента ProjectileMovementComponent. Текст тез наведено у додатку Ж.

### 6.2 Практичне впровадження проекту

За матеріалами кваліфікаційної роботи було прийнято участь у виставці технічної творчості молоді у рамках XXVIII Міжнародного молодіжного форуму «Радіоелектроніка та молодь у XXI столітті», що проходила з 16 по 18 квітня 2024 року на базі ХНУРЕ.

Виставку було проведено онлайн, презентовано та захищено роботу за темою «Ігровий програмний застосунок Steam Mystery у жанрі 3D Action-Adventure з елементами RPG» у секції «Ігрові технології». [10].

### 6.3 Соціальне впровадження проекту

У процесі розробки застосунку також велися сторінки у соцмережах для просування проекту. Використовувалися такі соцмережі, як:

- Telegram;
- YouTube;
- Instagram;
- Twitter/X;

– Facebook.

Найкращі показники були отримані у соцмережі Telegram, адже вона є найпопулярнішою серед цільової аудиторії нашого ігрового застосунку.

Другими за ефективністю соцмережами були Instagram та YouTube, адже їх алгоритми дозволили нам залучати нову аудиторію.

Twitter та Facebook були досить неефективними, адже цільова аудиторія нашої гри майже не користується ними.

У подальшому, Telegram, Instagram та YouTube можливо використати для подальшого просування гри. Також, для просування варто було б зробити й сторінки у спеціалізованих ресурсах, таких як Steam Community або Fandom.

## ВИСНОВКИ

У результаті виконання кваліфікаційної роботи бакалавра була розроблена демонстраційна версія ігрового застосунку у жанрі 3D Action-Adventure з елементами RPG та в естетиці стімпанк для операційної системи Windows. Ця гра отримала назву «Steam Mystery».

Під час виконання роботи був проведений ретельний аналіз предметної галузі та існуючих програмних рішень, що дозволило виявити як переваги, так і недоліки цих систем. На основі отриманих даних були визначені напрямки удосконалення існуючих рішень, а також сформульовано основні вимоги та завдання для подальшої розробки нового ігрового застосунку.

Для розробки продукту було використано ігровий рушій Unreal Engine 5 та мови програмування C++ та Blueprint. Середовища розробки – UnrealEditor та JetBrains Rider.

В програмному додатку використано візуальний інтерфейс, розроблений за допомогою «Adobe Photoshop» та взятий з відкритих джерел з ліцензією CC0 для вільного використання. Текстову складову роботи, включаючи всі параметри предметів та спорядження, було створено самостійно. Реалізацію інтерфейсу здійснено за допомогою Unreal Motion Graphics.

Розроблений ігровий застосунок має потенціал для подальшого розвитку та монетизації.

Розроблений ігровий застосунок можливо покращити шляхом впровадження локалізації, системи збережень, додавання нових завдань, ігрових локацій, супротивників, зброї.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Best Action-Adventure Games [Електронний ресурс] – URL: <https://www.metacritic.com/browse/game/all/action-adventure/all-time/metascore> (дата звернення: 21.04.2024)
2. Rollings A., Morris D. Game Architecture and Design: Learn the Best Practices for Game Design and Programming. Coriolis Group Books, 1999. 742 с.
3. The Legend of Zelda: Breath of the Wild [Електронний ресурс] – URL: <https://zelda.nintendo.com/breath-of-the-wild> (дата звернення: 01.04.2024).
4. The Witcher 3: Wild Hunt – Official Website. [Електронний ресурс] – URL: <https://www.thewitcher.com/us/en/witcher3> (дата звернення: 01.04.2024).
5. Янченко О. О. Балістика стрільби. Радіоелектроніка та молодь у XXI столітті. Т. 6 : Конференція «Інформаційні інтелектуальні системи» : матеріали 23-го Міжнар. молодіж. форуму, 16-18 квіт. 2019 р. / М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. – Харків : ХНУРЕ, 2019. – 305 с.
6. UProjectileMovementComponent. Epic Developer Community [Електронний ресурс] – URL: <https://dev.epicgames.com/documentation/unreal-engine/API/Runtime/Engine/GameFramework/UProjectileMovementComponent> (дата звернення: 07.05.2024)
7. Behavior Trees. Epic Developer Community [Електронний ресурс] – URL: <https://dev.epicgames.com/documentation/unreal-engine/behavior-trees-in-unreal-engine> (дата звернення: 08.05.2024)
8. Creating User Interfaces. Epic Developer Community. [Електронний ресурс] – URL: <https://dev.epicgames.com/documentation/unreal-engine/creating-user-interfaces-with-umg-and-slate-in-unreal-engine> (дата звернення: 01.05.2024).
9. Тренди та перспективи розвитку мультидисциплінарних досліджень: матеріали V Міжнародної студентської наукової конференції. Вінниця: ТОВ «УКРЛОГОС Груп», 2024. 326 с.
10. XXVIII Міжнародний молодіжний форум «Радіоелектроніка та молодь у XXI столітті». Каталог виставки технічної творчості молоді. – Харків: ХНУРЕ. 2024. – 27 с.

## ДОДАТОК А

## Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ

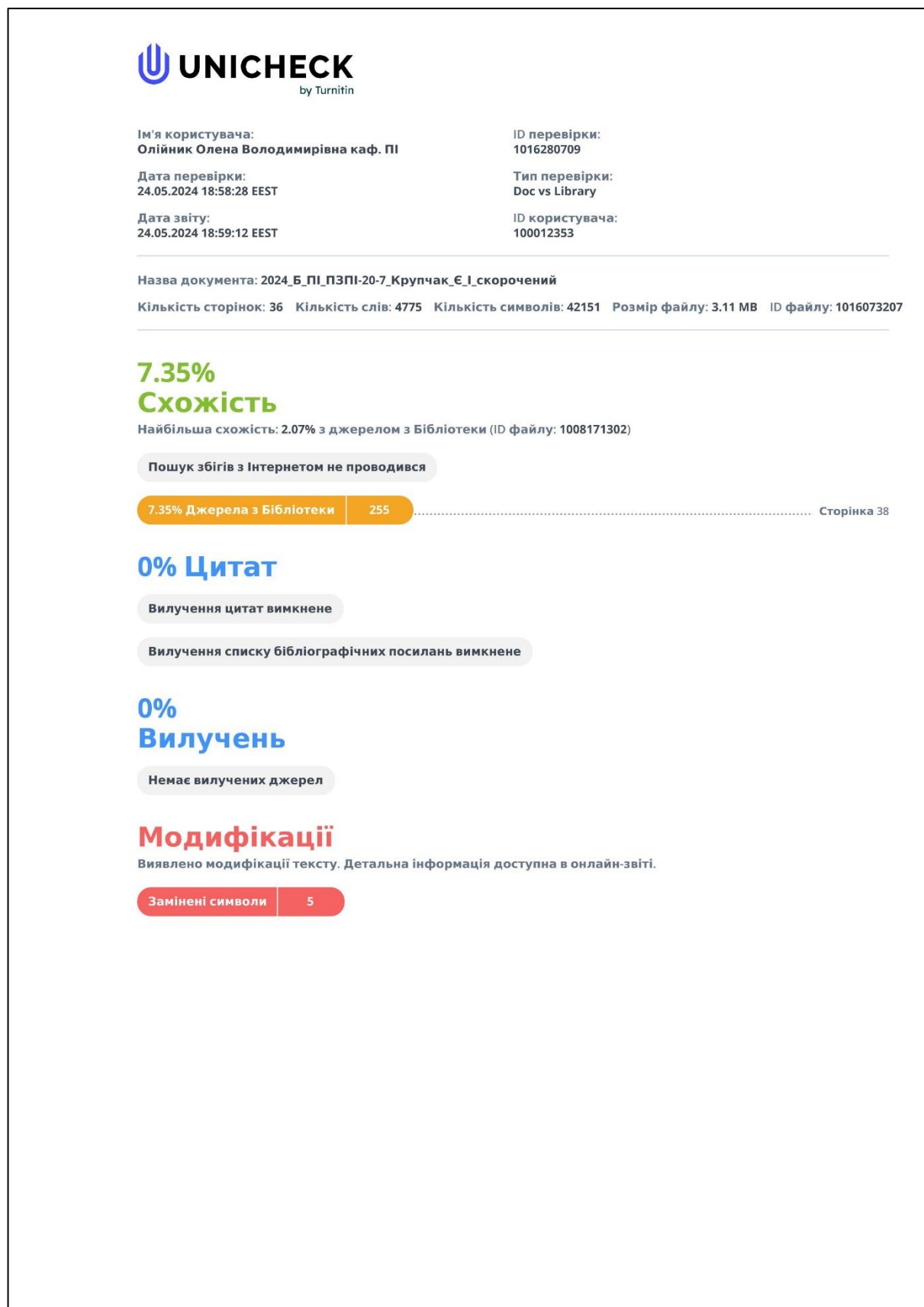


Рисунок А.1 – Звіт Unicheck

ДОДАТОК Б  
Слайди презентації

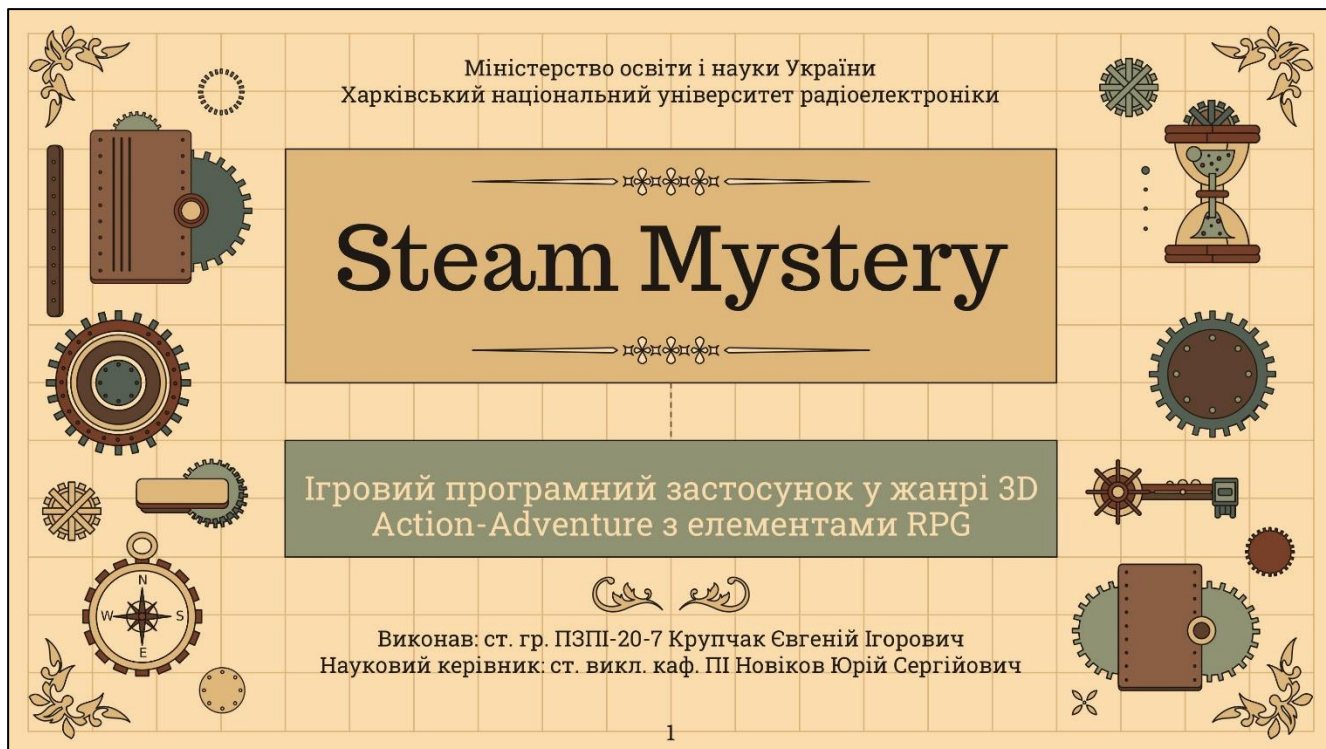


Рисунок Б.1 – Слайд 1



Рисунок Б.2 – Слайд 2

## Аналіз конкурентів

Гра	Переваги	Недоліки
The Legend of Zelda: Breath of the Wild	<ul style="list-style-type: none"> <li>• Великий відкритий світ</li> <li>• Інтерактивність та відкритість</li> <li>• Захоплюючі головоломки та бойові сцени</li> </ul>	<ul style="list-style-type: none"> <li>• Нестабільність технічної реалізації</li> <li>• Недоліки історії</li> </ul>
The Witcher 3: Wild Hunt	<ul style="list-style-type: none"> <li>• Глибина історії та персонажів</li> <li>• Великий відкритий світ</li> <li>• Вражаюча система прокачки персонажа</li> </ul>	<ul style="list-style-type: none"> <li>• Складність для новачків</li> <li>• Технічні проблеми</li> </ul>
Figment: Journey Into the Mind	<ul style="list-style-type: none"> <li>• Плавний, ненав'язливий сюжет.</li> <li>• Різноманіття головоломок</li> <li>• Музика</li> <li>• Велика кількість різних противників</li> </ul>	<ul style="list-style-type: none"> <li>• Не дуже зручне керування.</li> </ul>

Рисунок Б.3 – Слайд 3



Рисунок Б.4 – Слайд 4



Рисунок Б.5 – Слайд 5

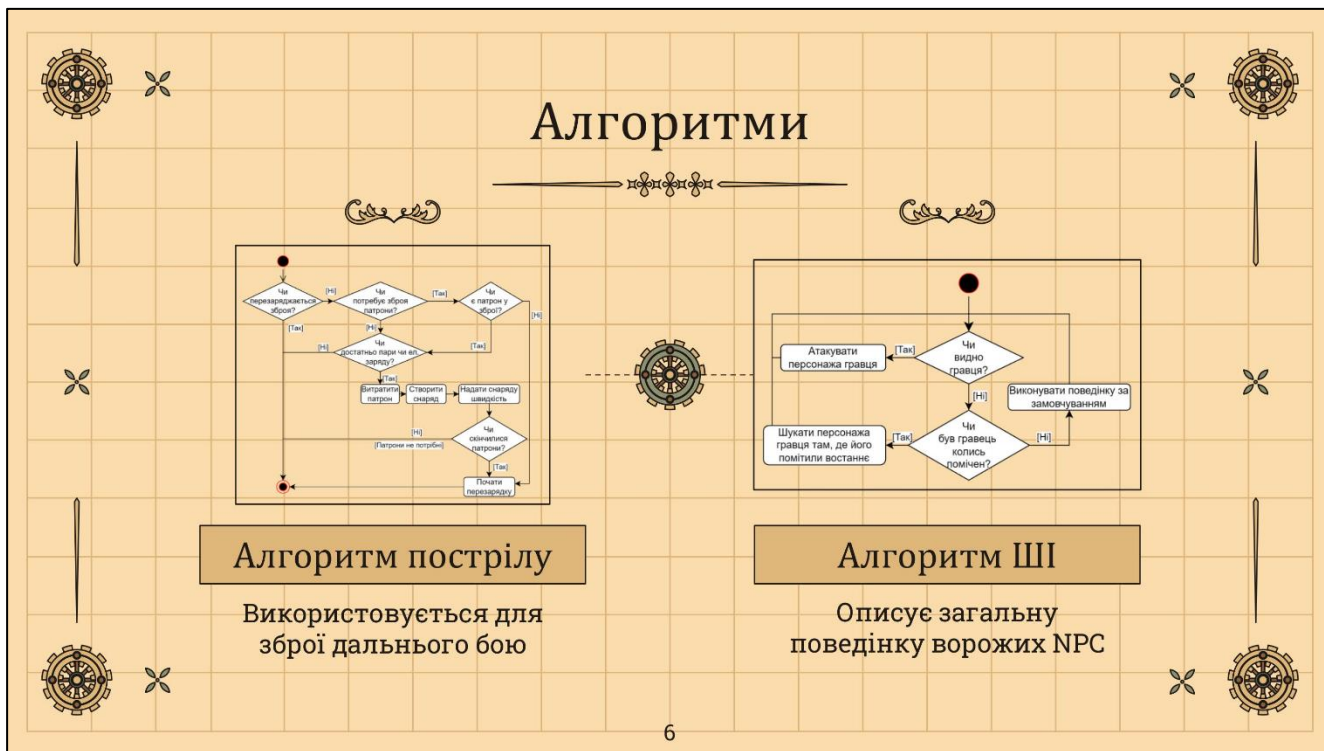


Рисунок Б.6 – Слайд 6

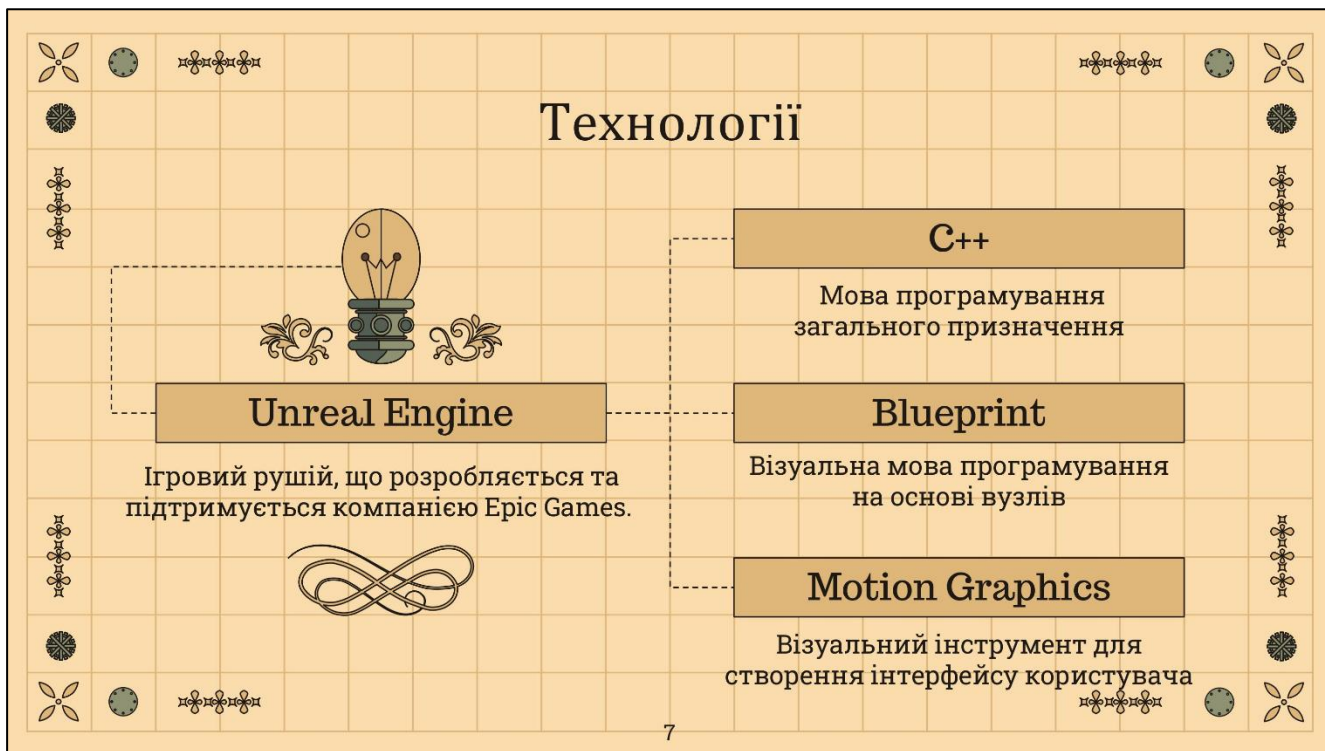


Рисунок Б.7 – Слайд 7

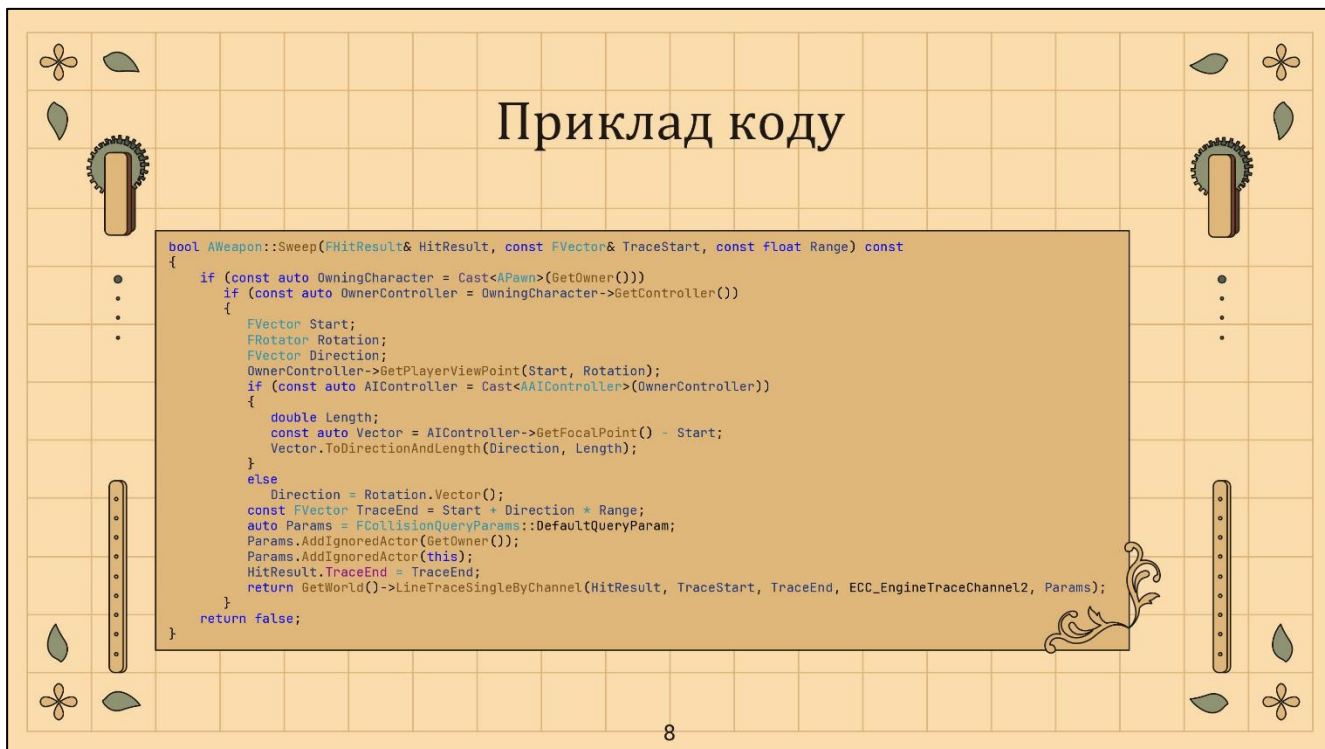


Рисунок Б.8 – Слайд 8

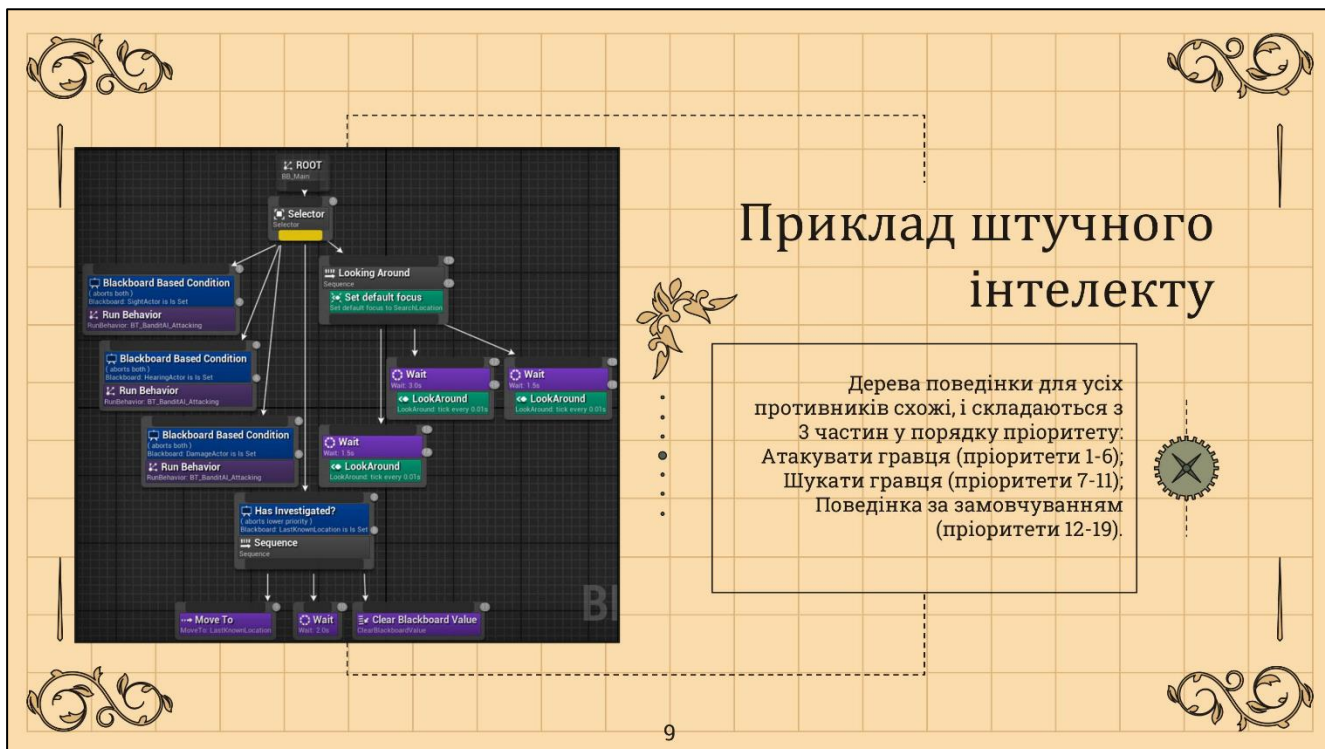


Рисунок Б.9 – Слайд 9

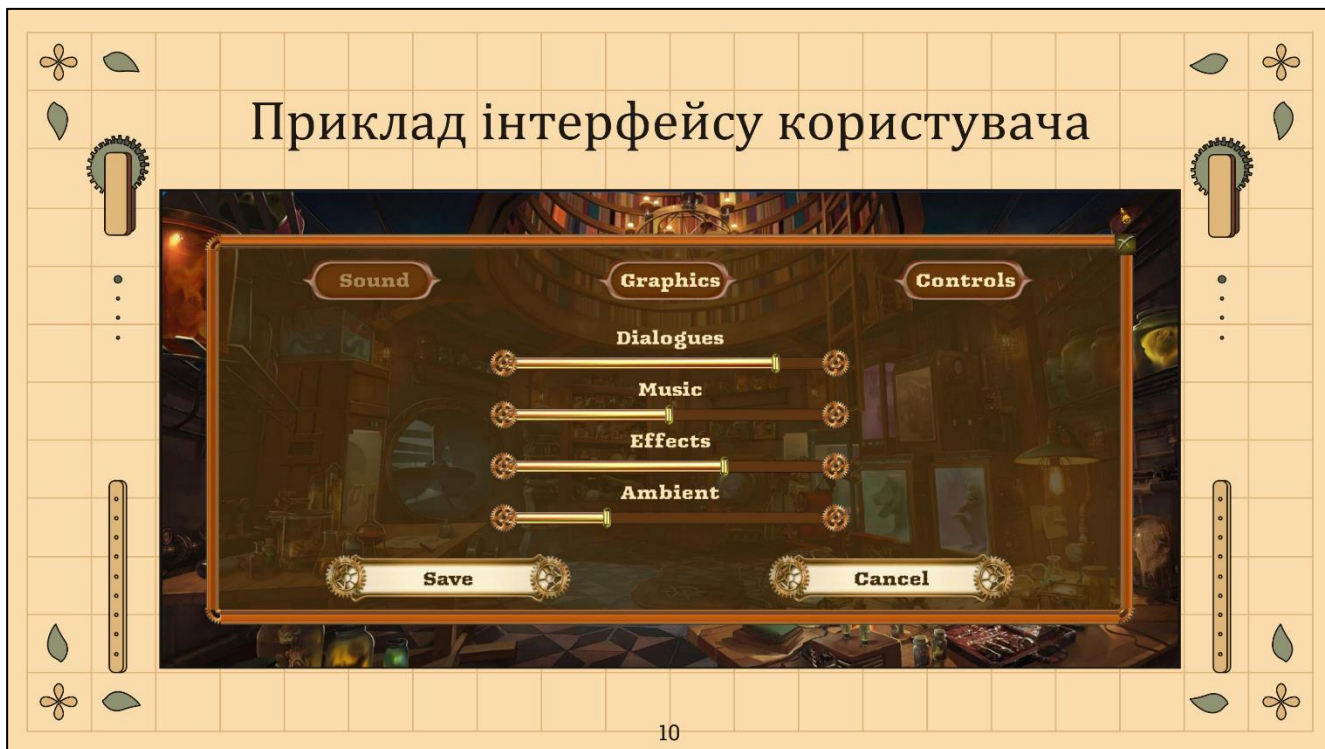


Рисунок Б.10 – Слайд 10

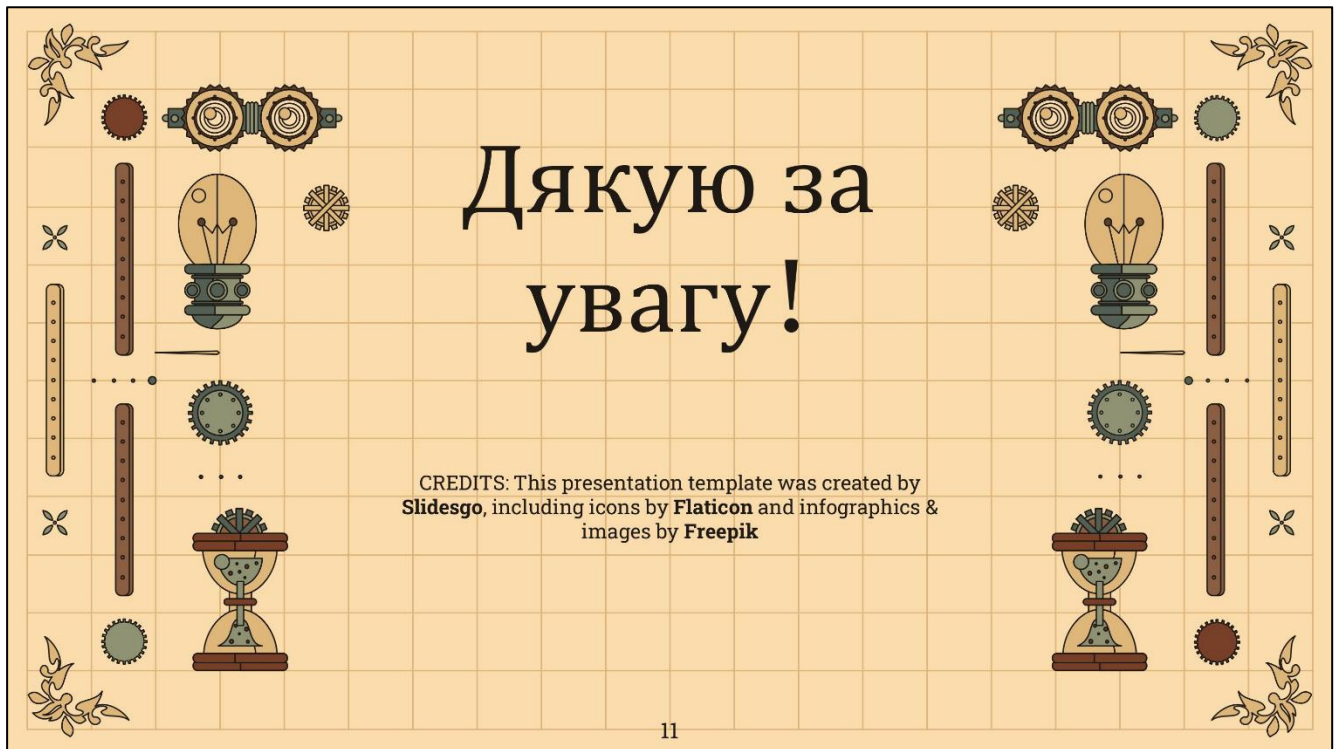


Рисунок Б.11 – Слайд 11

## ДОДАТОК В

Концепт-документ до гри

# Геймдизайн документ

# Steam Mystery

Рисунок В.1 – Сторінка 1 концепт-документу до гри

**ІСТОРИЯ РЕВІЗІЙ**

Версія	Автори	Дата	Коментарі
1.00	Євгеній, Анна	05.08.2023	
1.01	Євгеній, Анна	04.09.2023	
1.02	Євгеній, Анна	12.09.2023	
1.03	Євгеній, Анна	13.09.2023	

Рисунок В.2 – Сторінка 2 концепт-документу до гри

## ЗМІСТ

1 Специфікація.....	5
1.1 Тетра .....	5
1.2 Цільова аудиторія.....	5
1.3 USP.....	5
1.4 Час ігрової сесії .....	5
2 Геймплей .....	6
3 Механіки.....	7
3.1 Правила гри.....	7
3.2 Модель ігрового всесвіту .....	7
3.3 Дії персонажа.....	7
3.4 Потік екранів.....	8
4 Історія та наратив .....	13
4.1 Передісторія.....	13
4.2 Елементи сюжету .....	13
4.3 Прогрес ігрової історії .....	14
5 Ігровий всесвіт.....	15
6 Персонажі та противники.....	16
6.1 Персонажі.....	16
6.2 Противники.....	18
7 Рівні.....	21
7.1 Дім гільдії.....	21
7.2 Західний район міста Толлор .....	21
8 Інтерфейс користувача .....	22
8.1 HUD .....	22
8.2 Система контролю.....	22
8.3 Аудіо ефекти.....	22

Рисунок В.3 – Сторінка 3 концепт-документу до гри

8.4 Налаштування.....	22
9 Розгортання та розробка.....	24
9.1 Порядок встановлення.....	24
9.2 Системні вимоги.....	24
9.3 Розробка .....	24

Рисунок В.4 – Сторінка 4 концепт-документу до гри

## 1 СПЕЦИФІКАЦІЯ

### 1.1 Тетра

Механіка: Action-adventure з елементами RPG

Технологія: ПК

Історія: Головний герой – найманець

Естетика: Steampunk

### 1.2 Цільова аудиторія

Основна цільова аудиторія гри: чоловіки 16-35 років. Цю аудиторію мають привабити: зброя, битви, естетика, захоплююча історія.

### 1.3 USP

- Пориньте в атмосферу Steampunk;
- знищуйте противників та босів, щоб виконати основну місію;
- підвищуйте рівень зброї, щоб стати сильнішим.

### 1.4 Час ігрової сесії

Середній час на виконання 1 місії – 15-60 хвилин, у залежності від складності та навичок гравця.

## 2 ГЕЙМПЛЕЙ

Ціллю гри є проходження ігрової кампанії.

Прогрес гри вимірюється кількістю пройдених сюжетних етапів.

Сюжетний етап представляє собою ігрову місію, де гравець повинен здійснити певну дію або послідовність дій. Місія починається після триггеру чи іншої місії, а також може бути взята в секретарки гільдії або на дошці оголошень. Місія завершується виконанням усіх її цілей.

Випробування представляють собою бої з мобами. Бій починатиметься, коли персонаж знаходитиметься у конусі перед противником або завдасть шкоди противникові.

## 3 МЕХАНІКИ

### 3.1 Правила гри

Персонаж має наступні характеристики:

- здоров'я (якщо воно нижче за 0, персонаж помирає і гра закінчується);
- водяна пара (використовується для зброї. Не відновлюється. Можна лише зарядити пристроєм – капсулою. Варіюється в діапазоні 0-100%);
- електричний заряд (використовується для пристроїв. Самостійно відновлюється на 25%. Можна зарядити батарейкою. Варіюється в діапазоні 0-100%).

### 3.2 Модель ігрового всесвіту

Фізика діє як на персонажів так і на деякі ігрові об'єкти. Наприклад, персонаж може впасти з виступу, деякі елементи оточення можуть бути зламані або переміщені.

У грі присутня наступна валюта: золоті монети. Монети використовуються для торгівлі та покращення зброї. Валюта може бути добыта з противників або підібрана.

### 3.3 Дії персонажа

Персонаж може пересуватися вперед, вправо, вліво, назад; стрибати; присідати та спринтити.

Предмети можна отримати трьома способами: інвентарі (скриня, коробки тощо), мобі, квести. В усіх випадках предмети поміщаються в інвентар автоматично, тобто при луті моба, відкритті інвентаря, проходженні квесту.

Упродовж ігрової кампанії гравець може відчиняти двері, натискати на кнопки, важелі. Також гравець може наступати на пастки.

Здібності гравця представлені різними видами зброї:

Рисунок В.7 – Сторінка 7 концепт-документу до гри

- меч (ближня атака, середній урон, середня швидкість);
- молот (ближня атака, високий урон, низька швидкість);
- пістолети (дальня атака, середній урон, середня швидкість);
- гвинтівка (дальня атака, високий урон, низька швидкість);
- ручний арбалет (дальня атака, низький урон, висока швидкість);

Зброя дальнього бою використовує відповідні боєприпаси й пару/електрику.

Зброю та деякі пристрої можна буде покращувати за допомогою запчастин та монет.

Приклад покращень для пістолетів:

- Швидкість перезарядки I/II. Ціна – 200/400 монет + 2/4 шестерні.
- Точність I/II. Ціна – 300/600 монет + 2/4 трубки.
- Вибухові кулі. Ціна – 450 монет + 3 запалювачі. Не може бути разом з бронебійними кулями.
- Бронебійні кулі. Ціна – 450 монет + 3 металеві пластини. Не може бути разом з вибуховими кулями.
- Розширення магазину I/II/III. Ціна – 200/300/400 монет + трубка/трубка/2 трубки. Повторюване до 3 разів.

#### 3.4 Потік екранів

Головне меню(див. рис. 3.4.1) з'являється при вході. Тут гравець може здійснити налаштування(див. рис. 3.4.2), почати нову гру, або завантажити раніше збережену.

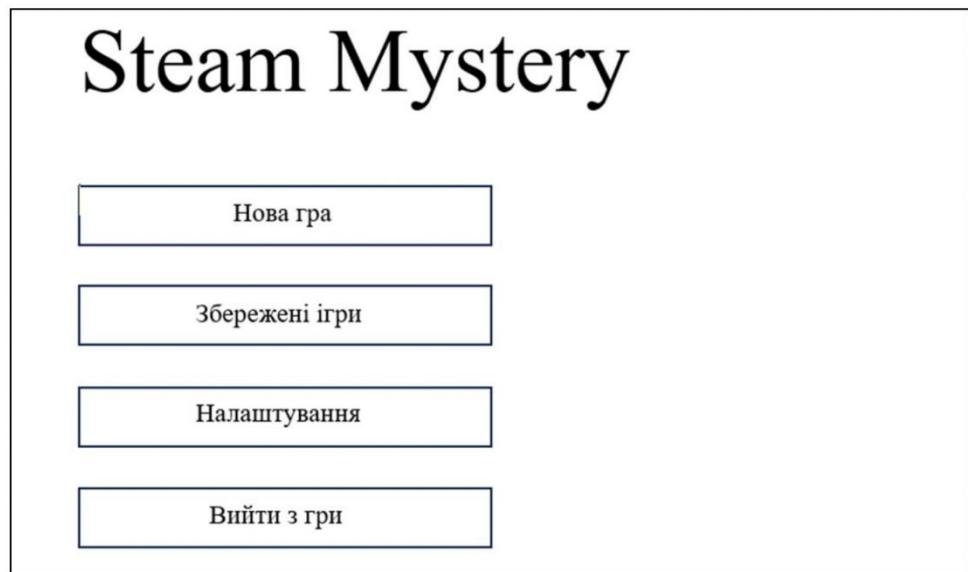


Рисунок 3.4.1 – Головне меню



Рисунок 3.4.2 – Налаштування

При вході в ігрову сесію, гравець бачить ігровий світ, HUD (див. рис. 3.4.3).

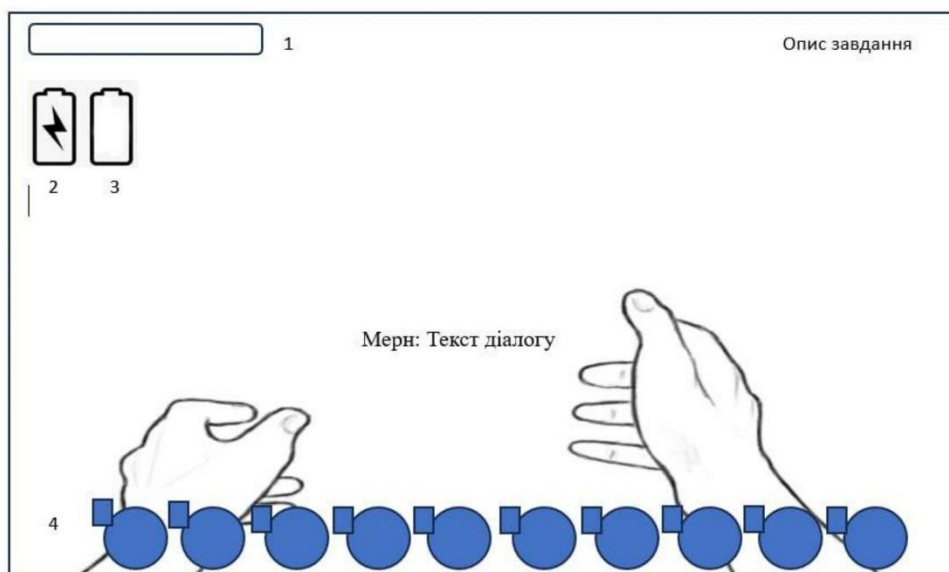


Рисунок 3.4.3 – HUD гравця

1. Панель здоров'я;
2. Кількість заряду;
3. Кількість пари;
4. Панель швидкого доступу.

Тут він може відкрити інвентар(див. рис. 3.4.4) або вікно спорядження(див. рис. 3.4.5).

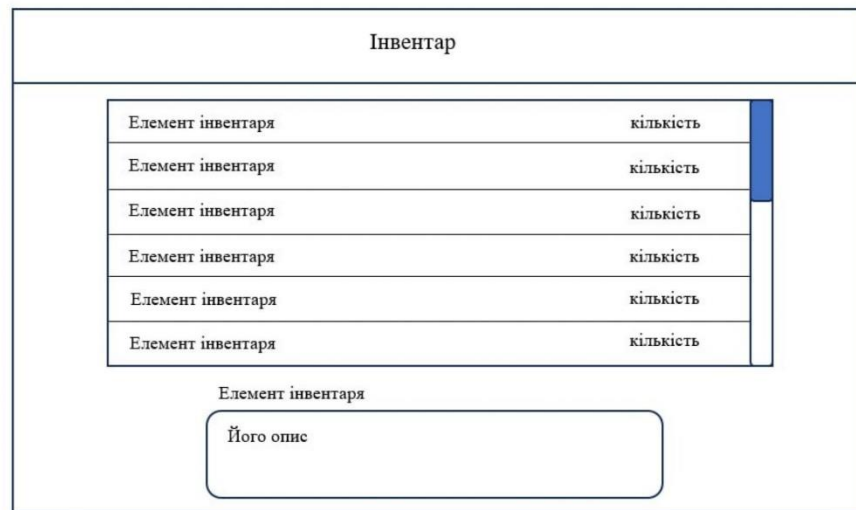


Рисунок 3.4.4 – Інвентар

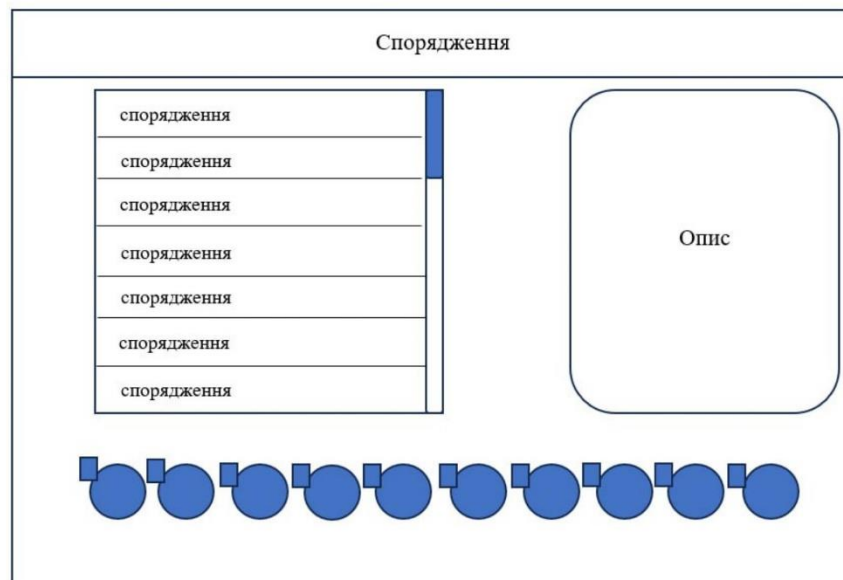


Рисунок 3.4.5 – Спорядження

Також гравець може поставити гру на паузу(див. рис. 3.4.6), де він може зберегтися, налаштувати гру, або вийти з неї.

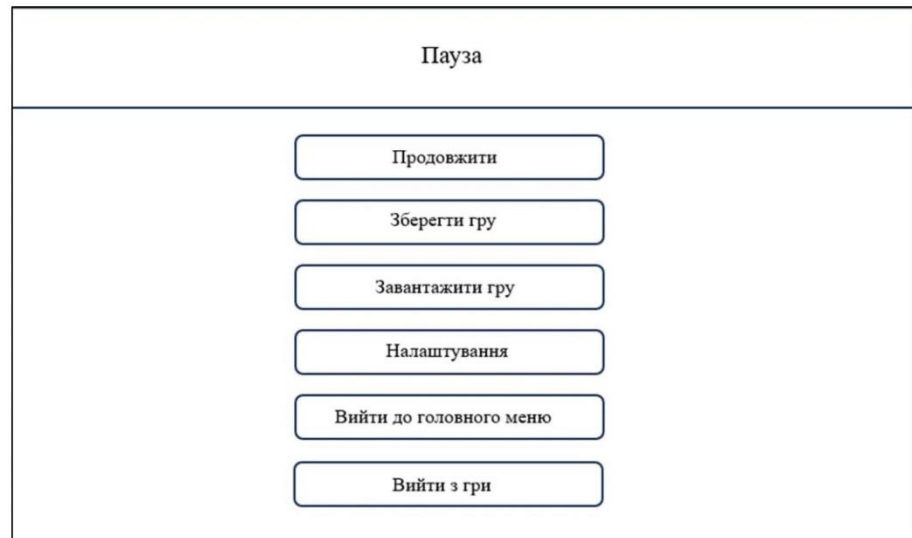


Рисунок 3.4.6 – Пауза

При взаємодії з верстаком, відкривається меню з покращенням зброї(див. рис. 3.4.7).

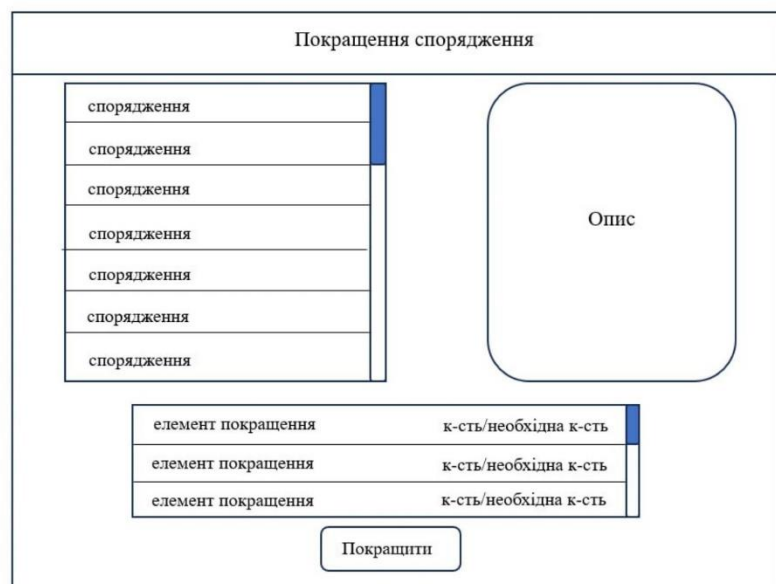


Рисунок 3.4.7 – Вікно покращення спорядження

Рисунок В.12 – Сторінка 12 концепт-документу до гри

## 4 ІСТОРІЯ ТА НАРАТИВ

### 4.1 Передісторія

Сім'я без статків. Батьки багато працюють, щоб забезпечити необхідним своїх дітей. Старший син постійно дорікав відсутністю тих бажаних благ, які є в інших дітей і намагався отримати більше. При чому спосіб, як він це отримував, його мало хвилював. Менший син м'якший, проявляє любов і вдячність до своїх батьків.

Час ішов, брати підросли. Старший досить рано вирішив жити своїм життям. Одного дня о пізній годині він зібрав свої речі й пішов. Тоді була ніч і ніхто одразу не помітив, що він зник.

Молодший син допомагає батькам, має жагу стати найманцем й цим заробляти собі на прожиття, а про його брата нічого не було чути з того часу, як він пішов.

### 4.2 Елементи сюжету

Модель розвитку сюжету представлений схемою Дена Хармона:

1. Персонаж знаходиться у зоні комфорту
2. Але він чогось хоче
3. Він потрапляє у незнайому ситуацію
4. Адаптується до неї
5. Отримує те, що йому потрібне.
6. Платить за це високу ціну
7. Потім повертається до знайомої ситуації
8. Змінюється

Ця схема використовуватиметься для кожної частини сюжету. Наприклад, щодо першої частини. Головний герой знаходиться в гільдії, у зоні комфорту. Він прагне поліпшити свої навички, заробити кошти. Потрапляє у незнайомі ситуації,

бо на шляху трапляються противники чий спосіб бою треба зрозуміти. У решті-решт, він адаптується до ситуації, перемагає противника та отримує винагороду. Платить ціну у вигляді здоров'я, часу. Потім здає квест у гільдію, тобто повертається до знайомої ситуації й у висновку отримує зміни: поліпшені навички у володінні зброєю й таке інше.

#### 4.3 Прогрес ігрової історії

Планується створення артів, які використовуватимуться для ліпшого представлення передісторії гравцеві. Упродовж сюжету історія розгортатиметься з використанням діалогів, підказок, які гравець зустрічатиме на своєму шляху.

## 5 ІГРОВИЙ ВСЕСВІТ

Ігровий всесвіт буде пронизаний естетикою стімпанк, яку можна помітити в стилізації міста, в елементах одягу, зброї, противниках тощо. Також характерною для стімпанку є наявність пари, яка доповнюватиме атмосферу міста та використовуватиметься в зброї.

Дії гри відбуваються у вигаданому всесвіті Літу, на континенті Оттоне. На цьому континенті є країна Вапор, і в її місті Толлор розгортаються події гри (див. рис 5.1).



Рисунок 5.1 – Прототип карти гри

## 6 ПЕРСОНАЖІ ТА ПРОТИВНИКИ

### 6.1 Персонажі

#### Ліз та Берн

Батьки Мерна, головного героя. Не бідні, але з народженням синів часто були в матеріально невідатному становищі. З'являються в передісторії.

#### Арнос

Старший брат Мерн. Вік: 24 роки. Високий, волосся коротке чорного кольору. Має темно-зелені очі, лукавий погляд.

Завжди аналізує, обережний, впевнений собі.

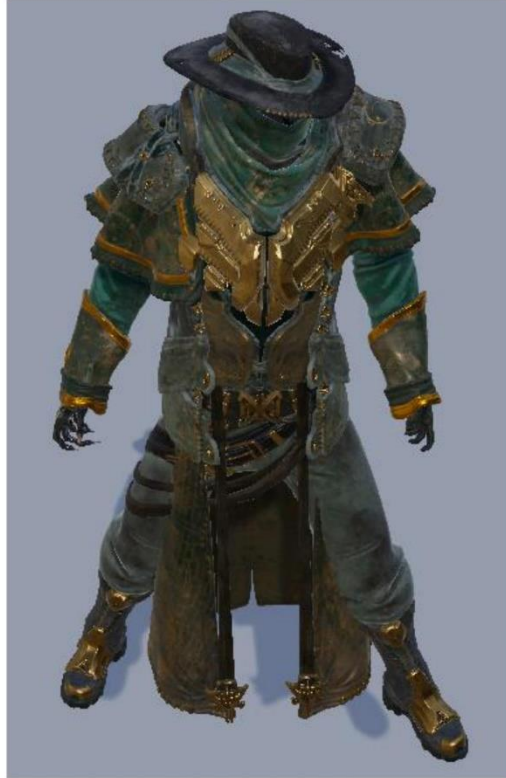


Мерн

Головний герой-найманець. Вік: 20 років. Високий, носить капелюха та шарф.

Спритний, дужий, безстрашний, виділяється добротою, розумінням.

Ігровий персонаж.



Матіль

Секретар гільдії. Вік: 25 років. Висока, струнка, має синьо-зелені очі, темно-руде волосся.

Відповідальна, майже завжди в курсі всіх подій.

Видає завдання в гільдії.



Особливістю зовнішнього вигляду персонажів буде те, що вони матимуть як частину вбрання речі, які пов'язані з естетикою стімпанк. Це можуть бути окуляри, капелюхи, чи такі дрібні деталі як зубці, ланцюжки в кольоровій гамі, яка є характерною для стімпанку.

## 6.2 Противники

### Парові павуки

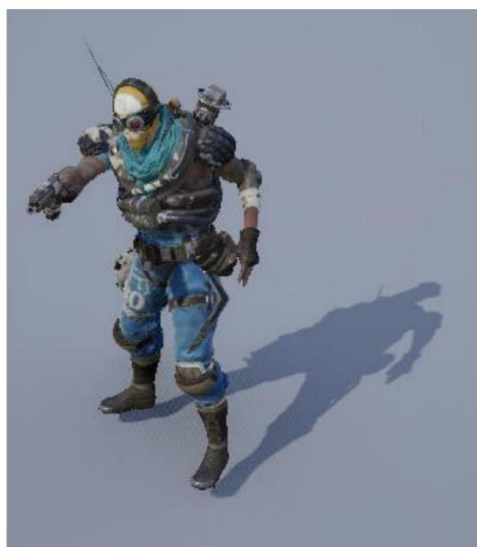
Атакує парою, кусається. Атака середнього та ближнього бою. Урон від пари низький, від укусу середній.

Рисунок В.18 – Сторінка 18 концепт-документу до гри



Розбійники (люди)

Використовують тип зброї, що доступний гравцеві. Урон береться від зброї.



Робот (елітний моб)

Удар рукою. Атака ближнього бою. Урон високий.

Вистріл з гармати. Атака дальнього бою. Урон високий.

Вистріл парою. Атака середнього бою. Урон середній.



Рисунок В.20 – Сторінка 20 концепт-документу до гри

## 7 РІВНІ

### 7.1 Дім гільдії

Перший рівень представляє собою дім гільдії головного персонажа. При першому потраплянні на рівень гравця ознайомлюють зі зброєю та навичками, ключовими стартовими персонажами, такими як секретарка гільдії. Також тут гравець знайомиться із системою поліпшення зброї. Звідси гравець бере завдання та відправляється виконувати їх.

### 7.2 Західний район міста Толлор

Це рівень, на якому виконується перше завдання. На цьому рівні буде навчання ігровому бою. Цей рівень матиме кілька нескладних противників. Деякі з них знаходитимуться на видному місці поруч із гравцем, інші будуть розпорошені по рівневі.

## 8 ІНТЕРФЕЙС КОРИСТУВАЧА

### 8.1 HUD

На екрані гравець бачить:

- своє здоров'я, пару та заряд;
- доступну зброю на панелі швидкого доступу;
- діалог, квести за наявністю;
- модель зброї в руках гравця.

### 8.2 Система контролю

Гравцю доступні наступний контроль:

- переміщення – [W][A][S][D];
- пересування камери – Миша;
- стрибок – [Space];
- присідання – [Ctrl];
- спринт – [Shift];
- взаємодія – [E];
- використання зброї/пристрою – [LMB]/[RMB].

### 8.3 Аудіо ефекти

У грі буде просторовий звук. Він може поширюватися від персонажів у момент діалогів; деяких дій, що передбачають звуковий супровід: атака, отримання урон тощо.

### 8.4 Налаштування

Гравець може налаштувати:

- якість текстур;
- якість відбивань;

- якість освітлення;
- дальність видимості;
- якість пост-обробки;
- якість згладжування;
- якість затінення;
- якість тіней;
- якість візуальних ефектів;
- якість рослинності;
- роздільну здатність;
- повноекранний режим;
- вертикальну синхронізацію;
- гучність звуків.

Рисунок В.23 – Сторінка 23 концепт-документу до гри

## 9 РОЗГОРТАННЯ ТА РОЗРОБКА

### 9.1 Порядок встановлення

Гру можна буде встановити за допомогою інстлятора або через Steam/Epic Games тощо.

Якщо гра встановлюватиметься через інстлятор, користувач зможе обрати у яку директорію встановлювати, чи створювати ярлик на робочому столі та у папці пуск.

### 9.2 Системні вимоги

Операційна система: Windows 10 x64 або вище.

Процесор: Intel Core i5-8400 | AMD Ryzen 3 3300X.

ОЗП: 12 GB RAM.

Відеокарта: AMD Radeon™ RX 5700 / NVIDIA® GeForce® GTX 1080.

DirectX: Version 12

Місце на диску: 50 GB на SSD

### 9.3 Розробка

Гра буде розроблена на рушії Unreal Engine 5.2.1, з використанням технологій Nanite, Megascans, Lumen тощо.

Репозиторій: <https://github.com/Geneirodan/SteamMystery>

ДОДАТОК Г  
План тестування

**Ігровий програмний застосунок у жанрі 3D action-  
adventure з елементами RPG «SteamMystery»**

**План тестування**

**Версія 1.0**

Рисунок Г.1 – Сторінка 1 плану тестування

**ІСТОРІЯ ЗМІН**

<b>Дата</b>	<b>Версія</b>	<b>Опис</b>	<b>Автор</b>
08.04.2024	1.0		Гречка А. О. Крупчак Є. І.

Рисунок Г.2 – Сторінка 2 плану тестування

**ЗМІСТ**

1. Вступ	4
1.1 Мета	4
1.2 Довідкова інформація	4
1.3 Галузь застосування	4
1.4 Визначення проекту	5
2 Вимоги до тестування	6
3 Стратегія тестування	7
3.1 Типи тестування	7
3.1.1 Функціональне тестування	7
3.1.2 Тестування інтерфейсу користувача	7
3.1.3 Тестування встановлення гри	8
3.2 Інструменти	8
4 Ресурси	9
4.1 Ролі	9
4.2 Система	9
5 Етапи проекту	10
6 Кінцевий продукт	11
6.1 Тестова модель	11
6.2 Звіти з дефектів	11

Рисунок Г.3 – Сторінка 3 плану тестування

## 1. ВСТУП

### 1.1 Мета

Наведений тест-план для ігрового програмного застосунку «Steam Mystery» описує вимоги для проведення випробування, рекомендації стосовно опису стратегії тестування, визначення необхідних ресурсів та забезпечення оцінки випробувань, перелік тестових елементів проекту.

Тестовий план використовуватиметься для проведення тестування, що базуватиметься на визначених в тест-плані критеріях якості та типах тестування.

### 1.2 Довідкова інформація

Відповідно до сюжету головний герой Мерн є найманцем і цим заробляє собі на прожиття. Упродовж гри гравець розвиває свого персонажа, виконує квести під час яких перемагає противників, крім того поринає в таємничу історію, яка приведе його до неочікуваної розв'язки.

«SteamMystery» – гра у жанрі 3D action-adventure з елементами RPG, розроблена для операційної системи Windows. Поки наявна лише англійська локалізація.

### 1.3 Галузь застосування

Рисунок Г.4 – Сторінка 4 плану тестування

Тестування є необхідним, щоб досягти певних критеріїв якості, які визначені тестовим планом. Серед основних цілей тестування виділяємо наступні:

- перевірка відповідності до вимог програмного продукту;
- підтвердження якості та надійності програмного забезпечення;
- поліпшення зручності використання графічного інтерфейсу користувача;
- оптимізація програмного продукту.

Для досягнення зазначених цілей, було прийнято рішення обрати наступні типи тестування:

- функціональне тестування;
- тестування інтерфейсу користувача;
- тестування встановлення.

#### 1.4 Визначення проекту

Документ	Створено або доступно	Отримано або перевірено	Автор або ресурс	Примітки
Дизайн- документ гри	Так	Так	Крупчак Є. І. Гречка А. О.	

Рисунок Г.5 – Сторінка 5 плану тестування

## 2 ВИМОГИ ДО ТЕСТУВАННЯ

Функціональне тестування:

- Переміщення персонажа.
- Бойова системи.
- Поведінка супротивників.
- Взаємодія з сутностями та ігровими об'єктами.
- Система інвентарю.
- Система покращень.

Тестування інтерфейсу користувача:

- Перевірка правильності відображення текстур на персонажах, локаціях.
- Перевірка зручності користувацького інтерфейсу.
- Перевірка зміни інтерфейсу при зміні стану гри (наприклад, смерть головного героя).

Тестування встановлення:

- Перевірка наявності гри, що встановлюється.

Рисунок Г.6 – Сторінка 6 плану тестування

### 3 СТРАТЕГІЯ ТЕСТУВАННЯ

#### 3.1 Типи тестування

##### 3.1.1 Функціональне тестування

Мета випробування	Протестувати механіки гри.
Технічний прийом	Протестувати функції: – переміщення персонажа; – бойової системи; – поведінки супротивників; – системи інвентарю; – системи покращень.
Критерії завершення	Усі випробування були проведені.

Тип тестування за якого тестується основний функціонал гри. Крім того функціональне тестування включатиме в себе перевірку колізій.

##### 3.1.2 Тестування інтерфейсу користувача

Мета випробування	Переконалися, що інтерфейс користувача зручний, текстури та анімації правильно відображаються.
Технічний прийом	Перевірити правильність відображення текстур на персонажах, локаціях. Перевірити зручність користувацького інтерфейсу.
Критерії завершення	Усі випробування були проведені.

Тестування інтерфейсу є важливим, бо від правильної роботи інтерфейсу та інтуїтивної зрозумілості залежить користувацький досвід, а також успіх гри.

Рисунок Г.7 – Сторінка 7 плану тестування

### 3.1.3 Тестування встановлення гри

Мета випробування	Переконатися, що гра правильно встановлюється.
Технічний прийом	Ручна або автоматична установка. Перевірка наявності гри, що встановлюється.
Критерії завершення	Усі випробування були проведені.

### 3.2 Інструменти

	Інструмент	Постачальник	Версія
Контроль версій	GitHub	GitHub, Inc.	2024
Управління проектами	GitHub Projects	GitHub, Inc.	2024
Відслідковування дефектів	Jira	Atlassian	2024

Рисунок Г.8 – Сторінка 8 плану тестування

## 4 РЕСУРСИ

### 4.1 Ролі

Людські ресурси		
Працівник	Рекомендований мінімальний обсяг осіб	Конкретні обов'язки або коментарі
Проектувальник тестів	1	Визначення, пріоритетів, і реалізація тестів. Обов'язки: – створення плану тестування, – генерація тестових моделей,
Тестувальник	1	Виконання тестів. Обов'язки: – виконання тестів; – запис результатів.

### 4.2 Система

Системні ресурси	
Ресурс	Ім'я або тип
ПК для тестування Включаючи спеціальні конфігураційні вимоги	Персональний ПК Операційна система: Windows 10 або 11. Процесор: Intel Core i5-9XXX або аналог ОЗП: 16+ Гб Відеокарта: NVIDIA GeForce GTX 1650 або аналог ПЗП: SSD від 25 Гб
Репозиторій тестування Посилання	Geneirodan/SteamMystery <a href="https://github.com/Geneirodan/SteamMystery">https://github.com/Geneirodan/SteamMystery</a>

Рисунок Г.9 – Сторінка 9 плану тестування

**5 ЕТАПИ ПРОЄКТУ**

Таблиця 5.1 - Етапи проєкту

<b>Етап</b>	<b>Обсяг робіт</b>	<b>Дата початку</b>	<b>Дата закінчення</b>
План випробувань	7 годин	10.03.2024	11.03.2024
Тест-дизайн	13 годин	12.03.2024	13.03.2024
Реалізація випробувань	25 годин	14.03.2024	16.03.2024
Виконання тестів	25 годин	17.03.2024	19.03.2024
Оцінка випробувань	8 годин	20.03.2024	22.03.2024

## 6 КІНЦЕВИЙ ПРОДУКТ

### 6.1 Тестова модель

У ході виконання тестування будуть отримані такі елементи:

- план тестування – опис цілей та стратегій тестування, методів реалізації процесу тестування;
- тест-кейси – документи, що відповідають примірникам тестового сценарію. Повинні містити: унікальний номер, опис, кроки відтворення, пріоритет, важливість тестового сценарію та очікуваний результат;
- баг-репорти – звіти про знайдені недоліки в системі із зазначенням рівня серйозності проблеми.

### 6.2 Звіти з дефектів

Звіти з дефектів будуть створені з використанням текстового процесору MS Word, систем баг-трекінгу Jira, а також систем управління проектом GitHub Projects.

Рисунок Г.11 – Сторінка 11 плану тестування

## ДОДАТОК Д

## Баг-репорти

Таблиця Д.1 – Помилка 1

Назва помилки	Світлі цятки перед гравцем при виборі зброї
Короткий опис	При виборі гравцем рушниці, перед ним з'являються кілька світлих цяток. З рухами гравця вони теж рухаються.
Компонент додатку	SKM_Rifle
Важливість	S4 Незначна
Пріоритет	P2 Середній
Кроки відтворення	1. Підібрати рушницю. 2. Обрати рушницю як поточну зброю.
Фактичний результат	З'являються цятки.
Очікуваний результат	Нічого не з'являється зайвого в полі зору.

Таблиця Д.2 – Помилка 2

Назва помилки	Гравець не може підібрати зброю
Короткий опис	При підбиранні пістолету, який лежить на верстаку, замість очікуваного результату, відкривається вікно, яке має відкриватися безпосередньо при взаємодії з верстаком.
Компонент додатку	LV_Guild
Важливість	S4 Незначна
Пріоритет	P2 Середня
Кроки відтворення	1. Підійти до верстака в кімнаті гільдії. 2. Навестися на пістолет 3. Натиснути «Е» для взаємодії
Фактичний результат	Відкривається вікно взаємодії з верстаком.
Очікуваний результат	Гравцем підбирається пістолет

Таблиця Д.3 – Помилка 3

Назва помилки	Ядро вибухає в руках персонажа
Короткий опис	При киданні ядра противником-кіборгом іноді воно вибухає в його ж руках.
Компонент додатку	LV_City
Важливість	S3 Висока
Пріоритет	P1 Високий
Кроки відтворення	Наблизитися в зону противника-кіборга, щоб він атакував
Фактичний результат	Ядро вибухає в руках противника.
Очікуваний результат	Кіборг кидає його в бік гравця.

Таблиця Д.4 – Помилка 4

Назва помилки	Можливий вихід на іншу локацію, коли початковий квест не був завершений.
Короткий опис	Гравець може вийти з гільдії до іншої локації не виконавши початкового квесту.
Компонент додатку	LV_Guild
Важливість	S3 Значна
Пріоритет	P2 Середній
Кроки відтворення	1. Опинитися на початковій локації «Гільдія». 2. Вийти до наступної локації «Місто» не виконавши початковий квест.
Фактичний результат	Гравець виходить до іншої локації.
Очікуваний результат	Гравець не може вийти з локації.

Таблиця Д.5 – Помилка 5

Назва помилки	Персонаж рухається, коли це не повинно відбуватися
Короткий опис	Якщо персонаж знаходиться на предметі, який симулює фізику, то персонаж починає дивно рухатися.
Компонент додатку	LV_City
Важливість	S5 Тривіальна
Пріоритет	P3 Низький
Кроки відтворення	Опинитися на предметі, який симулює фізику.
Фактичний результат	Персонаж рухається.
Очікуваний результат	Персонаж або не рухається, або рухається з урахуванням керування гравця.

Таблиця Д.6 – Помилка 6

Назва помилки	Меш зброї проходить крізь персонажа
Короткий опис	Якщо гравець тримає арбалет, то іноді можна помітити як зброя проходить крізь нього.
Компонент додатку	SKM_Crossbow
Важливість	S5 Тривіальна
Пріоритет	P3 Низький
Кроки відтворення	Взяти арбалет як поточну зброю гравця.
Фактичний результат	Прохід мешу крізь персонажа.
Очікуваний результат	Чітке відображення, без накладень мешу.

Таблиця Д.7 – Помилка 7

Назва помилки	Гравцеві надається інформація про виконану перезарядку, хоча вона не мала би бути виконаною.
Короткий опис	Якщо змінити обрану зброю на іншу не дочекавшись її перезарядки, а потім переключитися знову на неї, то гравцеві буде показана інформація про виконання перезарядки.
Компонент додатку	Зброя дальнього бою
Важливість	S3 Значна
Пріоритет	P2 Середня
Кроки відтворення	1. Обрати зброю дальнього бою. 2. Звільнити весь магазин. 3. Не дочекатися завершення перезарядки. 4. Переключитися на іншу зброю. 5. Повернутися до попередньо обраної зброї.
Фактичний результат	Відбулася перезарядка.
Очікуваний результат	Перезарядка не відбулася.

Таблиця Д.8 – Помилка 8

Назва помилки	При пострілі куля летить в інший бік.
Короткий опис	Якщо задана швидкість занадто мала, то при пострілі куля летітиме вгору.
Компонент додатку	BP_Bullet
Важливість	S2 Критична
Пріоритет	P1 Високий
Кроки відтворення	1. Обрати зброю дальнього бою. 2. Зробити постріл.
Фактичний результат	Куля летить по іншій траєкторії.
Очікуваний результат	Куля летить в логічно визначеному напрямку.

Таблиця Д.9 – Помилка 9

Назва помилки	Гравець може перечепитися й упасти.
Короткий опис	Гравець може перечепитися через якийсь предмет, наприклад, камінь та впасти.
Компонент додатку	LV_City
Важливість	S3 Значна
Пріоритет	P2 Середній
Кроки відтворення	Перечепитися через щось.
Фактичний результат	Гравець упав.
Очікуваний результат	Гравець залишається на ногах.

Таблиця Д.10 – Помилка 10

Назва помилки	Прогляд крізь персонажа.
Короткий опис	Якщо гравець подивиться вниз, то він може подивитися крізь персонажа.
Компонент додатку	SKM_Player
Важливість	S5 Тривіальна
Пріоритет	P3 Низький
Кроки відтворення	Нахилитися, подивитися вниз.
Фактичний результат	Прогляд крізь персонажа.
Очікуваний результат	Нормальне відображення.

## ДОДАТОК Е

Тези за темою кваліфікаційної роботи

**МОЛОДІЖНА  
НАУКОВА  
ЛІГА**

МАТЕРІАЛИ V МІЖНАРОДНОЇ  
СТУДЕНТСЬКОЇ НАУКОВОЇ  
**КОНФЕРЕНЦІЇ**

.....

**ТРЕНДИ ТА ПЕРСПЕКТИВИ  
РОЗВИТКУ МУЛЬТИДИСЦИП-  
ЛІНАРНИХ ДОСЛІДЖЕНЬ**

.....

м. Одеса, Україна  
26 квітня 2024 рік

Вінниця, Україна  
**«UKRLOGOS Group»**  
2024

Рисунок Е.1 – Титульна сторінка матеріалів конференції

26 квітня 2024 рік • Одеса, Україна • Молодіжна наукова ліга

## СЕКЦІЯ 12.

### ЕЛЕКТРОНІКА ТА ТЕЛЕКОМУНІКАЦІЇ

ДОСЛІДЖЕННЯ ІМІТАЦІЙНОГО МОДЕЛЮВАННЯ В МЕРЕЖІ З КОМУТАЦІЄЮ ПАКЕТІВ

Панов Є.І., *Науковий керівник: Штих І.А.*..... 154

## СЕКЦІЯ 13.

### КОМП'ЮТЕРНА ТА ПРОГРАМНА ІНЖЕНЕРІЯ

USAGE OF PROJECTILEMOVEMENTCOMPONENT AND LINETRACE FOR PROJECTILE SIMULATION IN UNREAL ENGINE 5

Крупчак Ye., *Scientific supervisor: Novikov Yu.*..... 156

КИБЕРФІЗИЧНА СИСТЕМА СПОСТЕРЕЖЕННЯ ЯК ЧАСТИНА ІНТЕРНЕТУ ТРАНСПОРТНИХ ЗАСОБІВ

Бойко М.С., *Науковий керівник: Яцків В.В.* ..... 159

РОЗРОБКА ВЕБСАЙТУ «ROKSANA.COMPANY»

Єфанова Д.С., *Науковий керівник: Штих І.А.*..... 162

РОЗРОБКА ВЕБСАЙТУ ДЛЯ ІНТЕРНЕТ-МАГАЗИНУ

Роменський М.С., *Науковий керівник: Штих І.А.*..... 164

## СЕКЦІЯ 14.

### СИСТЕМНИЙ АНАЛІЗ, МОДЕЛЮВАННЯ ТА ОПТИМІЗАЦІЯ

ПРОЄКТ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ КОНТРОЛЮ ЛІКІВ ІЗ МОНІТОРИНГОМ ПЕВНИХ ПОКАЗНИКІВ НА ОСНОВІ ІНТЕРНЕТУ РЕЧЕЙ

Камінецький О.Т., *Науковий керівник: Рибчак З.Л.* ..... 166

## СЕКЦІЯ 15.

### ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА СИСТЕМИ

БЕЗПЕКА ОПЕРАЦІЙ З ПЛАТІЖНИМИ ТЕРМІНАЛАМИ

Шитіков В.А., *Науковий керівник: Штих І.А.*..... 168

ВПЛИВ ІНФОРМАЦІЙНИХ СИСТЕМ НА УПРАВЛІННЯ ТРЕНАЖЕРНИМ ЗАЛОМ: ПЕРЕВАГИ ТА ВИКЛИКИ

Каськів В.П., *Науковий керівник: Демків Л.І.*..... 171

ІНТЕЛЕКТУАЛЬНЕ РАНЖУВАННЯ РЕЗУЛЬТАТІВ ОБРОБКИ ПОШУКОВОГО ЗАПИТУ КОРИСТУВАЧА

Савуляк В.О., *Науковий керівник: Глюнін О.О.* ..... 174

ІНФОРМАЦІЙНА СИСТЕМА ДЛЯ УПРАВЛІННЯ ІТ-ПРОЄКТАМИ

Жеплінський В.В., *Науковий керівник: Лучкевич М.М.* ..... 176

### СЕКЦІЯ 13.

## КОМП'ЮТЕРНА ТА ПРОГРАМНА ІНЖЕНЕРІЯ

**Krupchak Yevhenii**, Higher education applicant of the Faculty of Computer Science  
*Kharkiv National University of Radio Electronics, Ukraine*

**Scientific supervisor: Novikov Yurii**, Senior Lecturer  
of the Department of Software Engineering  
*Kharkiv National University of Radio Electronics, Ukraine*

### USAGE OF PROJECTILEMOVEMENTCOMPONENT AND LINETRACE FOR PROJECTILE SIMULATION IN UNREAL ENGINE 5

There is a problem with simulating projectile movement in first-person games. In shooters, this problem is often solved by line trace, but this method is not suitable for low velocity projectiles.

Line Trace[1] is used to determine the intersection points between the line and objects in the game, which allows you to determine whether the projectile will hit the target and handle the corresponding events.

Using UProjectileMovementComponent in addition to line tracing allows you to create realistic projectile behavior and interaction with the surrounding world.

The UProjectileMovementComponent[2] allows you to customize the speed, mass, and other parameters of the projectile's movement, which allows you to achieve the desired behavior of the projectile in the game.

But for first-person games we have a problem with the initial vector of projectile motion, because usually firing point of projectile launcher is not directed to the point the player is looking at. This problem can be solved using line tracing and the UGameplayStatics::SuggestProjectileVelocity[3] method.

SuggestProjectileVelocity calculates a launch velocity for a projectile to hit a specified point.

To implement this behavior, we need to create 2 classes: one for projectile, and one for launcher.

Firstly, the AProjectile class, which represents our projectile. It must have a UProjectileMovementComponent. In addition, our projectile must know its starting point and speed, as well as the point the player is looking at. Finally, our projectile must calculate its velocity, using UGameplayStatics::SuggestProjectileVelocity and set it in UProjectileMovementComponent (Fig. 1).

```
void AProjectile::BeginPlay()
{
    FVector TossVelocity;
    UGameplayStatics::SuggestProjectileVelocity(GetWorld(), TossVelocity, StartLocation, EndLocation,
    TossSpeed, false, 0, 0,
    ESuggestProjectileVelocityTraceOption::DoNotTrace,
    FCollisionResponseParams::DefaultResponseParams, TArray<Actor*>());
    MovementComponent->SetVelocityInLocalSpace(TossVelocity);
}
```

Fig. 1. Velocity calculation

We can then proceed to implement a launcher class, such as ARangedWeapon. It should contain a USceneComponent to reference the projectile's spawn location and a TSubclassOf<AProjectile> to reference the projectile class. Then we need to implement the trace function. For example, it is implemented in the parent ARangedWeapon class called AWeapon (Fig. 2).

```
bool AWeapon::Sweep(FHitResult& HitResult, const FVector& TraceStart, const float Range) const
{
    if (const auto OwningCharacter = Cast<A Pawn>(GetOwner()))
    {
        if (const auto OwnerController = OwningCharacter->GetController())
        {
            FVector Start;
            FRotator Rotation;
            OwnerController->GetPlayerViewPoint(Start, Rotation);
            const FVector End = Start + Rotation.Vector() * Range;
            auto Params = FCollisionQueryParams::DefaultQueryParams;
            Params.AddIgnoredActor(GetOwner());
            Params.AddIgnoredActor(this);
            GetWorld()->LineTraceSingleByChannel(HitResult, TraceStart, End, ECC_EngineTraceChannel2, Params);
            return true;
        }
    }
    return false;
}
```

Fig. 2. Method AWeapon::Sweep for line tracing

Also, we need some sort of usage function that is called when the player or AI wants to use our weapon (Fig. 3).

```
bool ARangedWeapon::Use_Implementation()
{
    ...
    FHitResult HitResult;
    const auto TraceStart = FirePoint->GetComponentLocation();
    const auto TraceEnd = Sweep(HitResult, TraceStart, GetStats().FindRef(EStat::Range) * 100);
    if (HitResult.IsValid() && HitResult.IsValidActor())
    {
        const FTransform Transform(FRotator::ZeroRotator, TraceStart);
        if (const auto Projectile = GetWorld()->SpawnActorDeferred<AProjectile>(ProjectileClass,
                                                                              Transform,
                                                                              GetOwner(),
                                                                              GetInstigator()))
        {
            Projectile->SetDamageAndRadius(GetStats().FindRef(EStat::Damage), GetStats().FindRef(EStat::ExplosionRadius));
            Projectile->SetPoints(TraceStart, TraceEnd);
            Projectile->FinishSpawning(Transform);
        }
    }
    ...
}
```

Fig. 3. Implementation of ARangedWeapon::Use native event

The whole process is as follows: The player uses a ranged weapon; it performs a collision trace along the player's line of sight and finds the desired impact location; it spawns a projectile actor, sets its properties. Finally, the projectile calculates the desired velocity using the start and end points.

Conclusions. In first-person games with slow projectiles, we can use UProjectileMovementComponent in addition to line tracing to achieve a balance between realism and player-friendliness.

#### References:

1. UWorld::LineTraceSingleByChannel. Unreal Engine 5.3 Documentation | Unreal Engine 5.3 Documentation. URL: <https://docs.unrealengine.com/4.27/en->

**Тренди та перспективи розвитку мультидисциплінарних досліджень**

- US/API/Runtime/Engine/Engine/UWorld/LineTraceSingleByChannel/ (дата звернення: 01.03.2024).
- UProjectileMovementComponent. Unreal Engine 5.3 Documentation | Unreal Engine 5.3 Documentation. URL: <https://docs.unrealengine.com/5.3/en-US/API/Runtime/Engine/GameFramework/UProjectileMovementComponent/> (дата звернення: 01.03.2024).
  - SuggestProjectileVelocity. Unreal Engine 5.3 Documentation | Unreal Engine 5.3 Documentation. URL: <https://docs.unrealengine.com/4.26/en-US/BlueprintAPI/Game/Components/ProjectileMovement/SuggestProjectileVelocity/> (дата звернення: 01.03.2024).

Рисунок Е.5 – Третя сторінка тез

## ДОДАТОК Ж

Фрагмент каталогу виставки технічної творчості молоді

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ  
УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

МАТЕРІАЛИ  
XXVIII МІЖНАРОДНОГО МОЛОДІЖНОГО ФОРУМУ

**«РАДІОЕЛЕКТРОНІКА ТА МОЛОДЬ  
У XXI СТОЛІТТІ»**

16 – 18 квітня 2024 р.

**КАТАЛОГ ВИСТАВКИ  
ТЕХНІЧНОЇ ТВОРЧОСТІ МОЛОДІ**

Харків 2024

Рисунок Ж.1 – Обкладинка збірника

### **7. Шутер від першої особи "Tale of the Awakened"**

**Автори:** *Беліков Данило Юрійович, Кісельгова Маргарита Євгенівна*, ст. гр. ПЗПІ-20-5, ХНУРЕ.

**Науковий керівник:** Новіков Юрій Сергійович, старший викладач. каф. ПІ, ХНУРЕ.

У цій роботі описується один рівень з власної розробленої гри. "Tale of the Awakened" — це шутер від першої особи, що втілює в собі змішання стилю меджикпанку та середньовічного фентезі, призначений для ПК.

Ігровий процес поєднує в собі елементи головоломки, де гравцеві належить вибирати між прямим зіткненням та таємним проходженням, використовуючи різноманітні магичні стихії.

У проєкті описані завдання розробників, виконані для досягнення оптимального результату, такі як: можливість обирати власний спосіб проходження рівнів, можливість якісної взаємодії гравця з ігровим середовищем, розробка штучного інтелекту ворогів, комп'ютерні ефекти, що якісно відображають взаємодію стихій у грі.

### **8. Ігровий програмний застосунок у жанрі 3D Rogue-like шутер від третьої особи «Synthetic Supremacy»**

**Автори:** *Фурсов Данило Сергійович, Борисенко Артемій Едуардович*, ст. гр. ПЗПІ-20-7, ХНУРЕ.

**Науковий керівник:** Новіков Юрій Сергійович, старший викладач. каф. ПІ, ХНУРЕ.

Розроблено ігровий програмний застосунок у жанрі 3D Rogue-like шутер від третьої особи під назвою «Synthetic Supremacy».

Дана гра розроблена на ігровому рушії Unreal Engine 5 з використанням технологій оптимізації графіки Nanite та освітлення Lumen.

За сюжетом гри гравець виступає в ролі бойового робота, головною метою якого є пройти всі рівні, подолавши всіх ворогів на шляху. Події гри відбуваються в футуристичному мегаполісі, ігровий світ виконаний у стилістиці кіберпанку. Рівні складаються з кімнат та коридорів, які генеруються автоматично під час кожного проходження.

Основними функціями гри є битва з ворогами, знаходження предметів і зброї та покращення персонажу гравця.

### **9. Ігровий програмний застосунок «Steam Mystery» у жанрі 3D Action-Adventure з елементами RPG**

**Автори:** *Крупчак Євгеній Ігорович*, ст. гр. ПЗПІ-20-7, *Гречка Анна Олександрівна*, ст. гр. ПЗПІ-20-9, ХНУРЕ.

**Науковий керівник:** Новіков Юрій Сергійович, старший викладач. каф. ПІ, ХНУРЕ.

Розроблено демонстраційну версію ігрового застосунку Steam Mystery у жанрі 3D Action-Adventure з елементами RPG та в естетиці стімпанк. Гравців очікує комбінація вікторіанського шарму та парових технологій.

Головним героєм постає хлопець найманець, який виріс у небагатій сім'ї. Перед ним постає завдання перемагати небезпечних противників і впродовж ігрової кампанії розкрити таємницю, яка переслідує його.

Гра дозволяє отримати ігровий досвід у ролі головного героя найманця зі зброєю або дальнього, або ближнього бою.

Ігровий застосунок розроблено для персонального комп'ютеру з операційною системою Windows. Гру розроблено з використанням рушія Unreal Engine 5.

### **10. Ігровий програмний застосунок у жанрі Roguelite RPG. Bounty Chase**

**Автори:** *Донець Дмитро Сергійович*, ст. гр. ПЗПІ-20-2, *Возсова Маргарита Володимирівна*, ст. гр. ПЗПІ-20-5, ХНУРЕ.

**Науковий керівник:** Новіков Юрій Сергійович, старший викладач. каф. ПІ, ХНУРЕ.

Ігровий програмний застосунок у жанрі Roguelite RPG "Bounty Chase" створений для проведення вільного часу.

Основа мета гри – проходження якомога більшої кількості хвиль ворогів, між якими гравець може покращувати свої атрибути та здібності, або відкривати нові для нього. Окрім цього гравець після поразки повертається сильнішим, маючи сильніші здібності і може проходити гру знову за бажанням. Вороги посилюються з рівнем. Декілька гравців можуть проходити разом один і той самий рівень одночасно.

Переваги розробки: кросплатформеність, простий та зрозумілий інтерфейс, легке впровадження дизайнерами нових значень за допомогою таблиць, підтримка мультиплеєрної гри.

### **11. Ігровий програмний застосунок у жанрі 2d action-adventure з елементами RPG "BloodGrace"**

**Автор:** *Кайданюк Герман Сергійович*, ст. гр. ПЗПІ-20-9, ХНУРЕ.

**Науковий керівник:** Новіков Юрій Сергійович, старший викладач. каф. ПІ, ХНУРЕ.

Проект передбачає ігровий застосунок у жанрі 2d action-adventure з елементами RPG, що дозволить гравцеві занирнути в темний ігровий світ та дізнатися усіх таємниць, що його очікують. Метою є пройти всі локації. Основні функції: битви, дослідження, керування віком головного героя. Бойова система орієнтована на комбінації, де більшість дій буде збільшувати вік персонажа, що змусить його більш думати на своїми діями.

З іншого боку, існує можливість зменшити вік завдяки спеціальному захопленню, яке потребує час, тому важливо зберігати баланс. На поточний час реалізована стартова локація та декілька типів ворогів разом зі стандартним набором магії.