

Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

_____ Едуард ЖАРІКОВ

“ ____ ” _____ 2025 р.

**ГРА У ЖАНРІ ROLE-PLAYING-GAME(RPG) НА UNREAL ENGINE 3
ВИКОРИСТАННЯМ ШІ**

Технічне завдання

КПШ.ІІ-1107.045440.01.91

“ПОГОДЖЕНО”

Керівник проєкту:

_____Максим ГОЛОВЧЕНКО

Нормоконтроль:

_____Максим ГОЛОВЧЕНКО

Виконавець:

_____Олександр ГОЛОВНЯ

Київ – 2025

ЗМІСТ

1	НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ	4
2	ПІДСТАВА ДЛЯ РОЗРОБКИ.....	5
3	ПРИЗНАЧЕННЯ РОЗРОБКИ	6
4	ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	7
4.1	Вимоги до функціональних характеристик	7
4.1.1	Користувацького інтерфейсу	7
4.1.2	Забезпечення функціонування головного меню гри:.....	9
4.1.3	Забезпечення функціонування ігрових механік:	10
4.1.4	Забезпечення функціонування ігрового інтерфейсу(компоненти HUD):	12
4.1.5	Забезпечення функціонування взаємодії з інвентарем:	13
4.1.6	Забезпечення функціонування штучного інтелекту неігрових персонажів гри:	14
4.1.7	Забезпечення взаємодії з колізією об'єктів:	15
4.2	Вимоги до надійності.....	17
4.3	Умови експлуатації	17
4.3.1	Вид обслуговування.....	18
4.3.2	Обслуговуючий персонал.....	18
4.4	Вимоги до складу і параметрів технічних засобів.....	18
4.5	Вимоги до інформаційної та програмної сумісності.....	19
4.5.1	Вимоги до вхідних даних	19
4.5.2	Вимоги до вихідних даних	19
4.5.3	Вимоги до мови розробки	20
4.5.4	Вимоги до середовища розробки	20
4.5.5	Вимоги до представленню вихідних кодів	20
4.6	Вимоги до маркування та пакування	20
4.7	Вимоги до транспортування та зберігання	20
4.8	Спеціальні вимоги	20
5	ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ	21

5.1	Попередній склад програмної документації	21
5.2	Спеціальні вимоги до програмної документації	21
6	СТАДІЇ І ЕТАПИ РОЗРОБКИ	22
7	ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ	23

1 НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ

Назва розробки: Гра у жанрі Role-Playing-Game(RPG) на Unreal Engine з використанням ШІ.

Галузь застосування: Індустрія відеоігор

Наведене технічне завдання поширюється на розробку програмного забезпечення «RPG-гра з використанням ШІ» [RPG-AI-001], котра використовується для інтерактивного рольового ігрового середовища, що забезпечує унікальний досвід користувача завдяки динамічній генерації контенту, сценаріїв і поведінки персонажів на основі алгоритмів штучного інтелекту та призначена для галузі цифрових розваг, а саме ігрової індустрії. Потенційними користувачами є гравці, гейм/левел дизайнери, новачки у сфері ШІ та розробники відеоігор.

2 ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки RPG-гри з використанням ШІ на рушії UE5 є завдання на дипломне проектування, затверджене кафедрою інформатики та програмної інженерії Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

3 ПРИЗНАЧЕННЯ РОЗРОБКИ

Розробка призначена для забезпечення функціонування ігрових механік, що дозволяють реалізацію цікавого штучного інтелекту гри у жанрі RPG і використовується у галузі розваг для широкого кола користувачів.

Метою розробки є підвищення якості ігрового досвіду користувачів шляхом реалізації цікавого штучного інтелекту для неігрових персонажів.

4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Вимоги до функціональних характеристик

Програмне забезпечення повинно забезпечувати виконання наступних основних функцій:

4.1.1 Користувацького інтерфейсу

Меню головного екрану, можливість початку нової гри або продовження поточної, доступ до налаштувань гри, вихід з гри(Рис. 4.1).

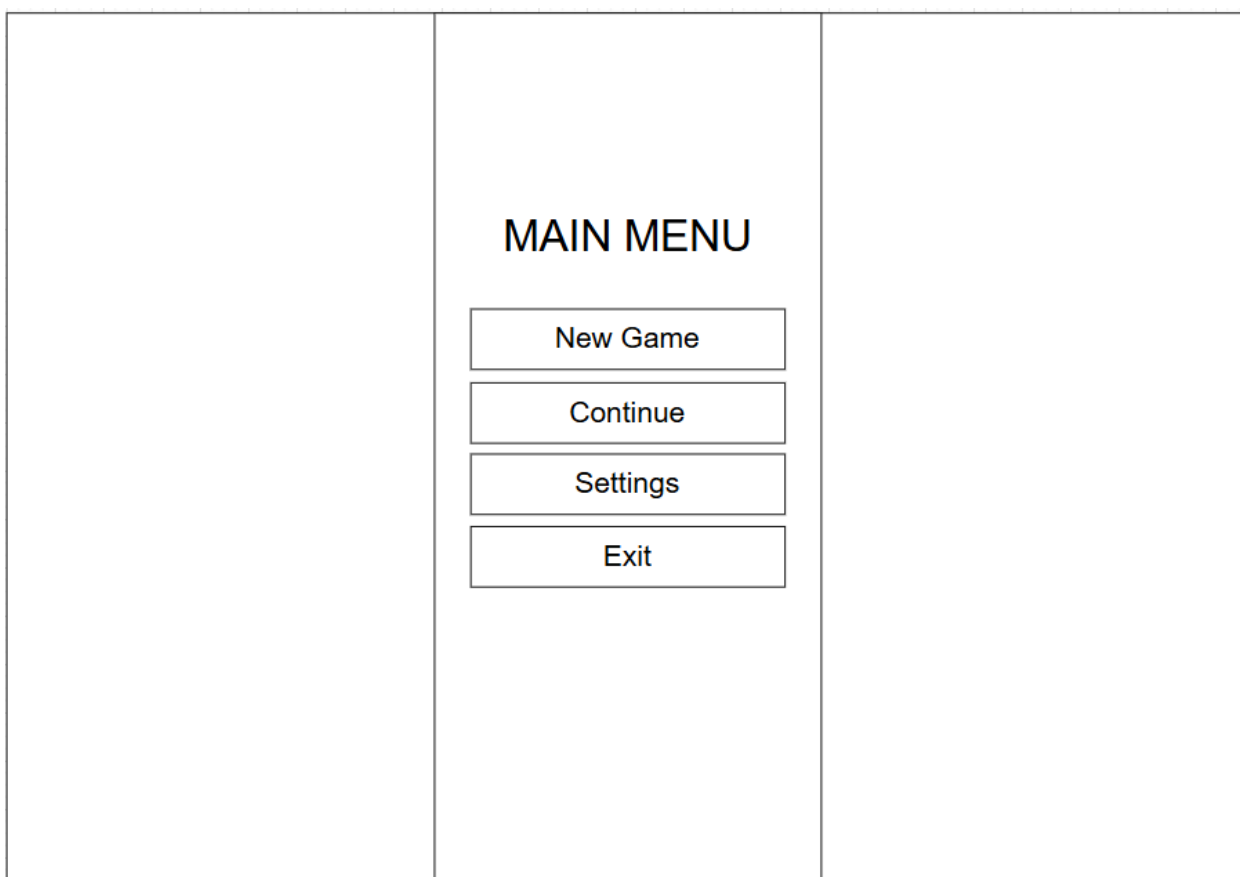


Рисунок 4.1 – Головна сторінка

Інтерфейс управління персонажем, показ основних характеристик персонажа: здоров'я, витривалість, досвід (Рис. 4.2).

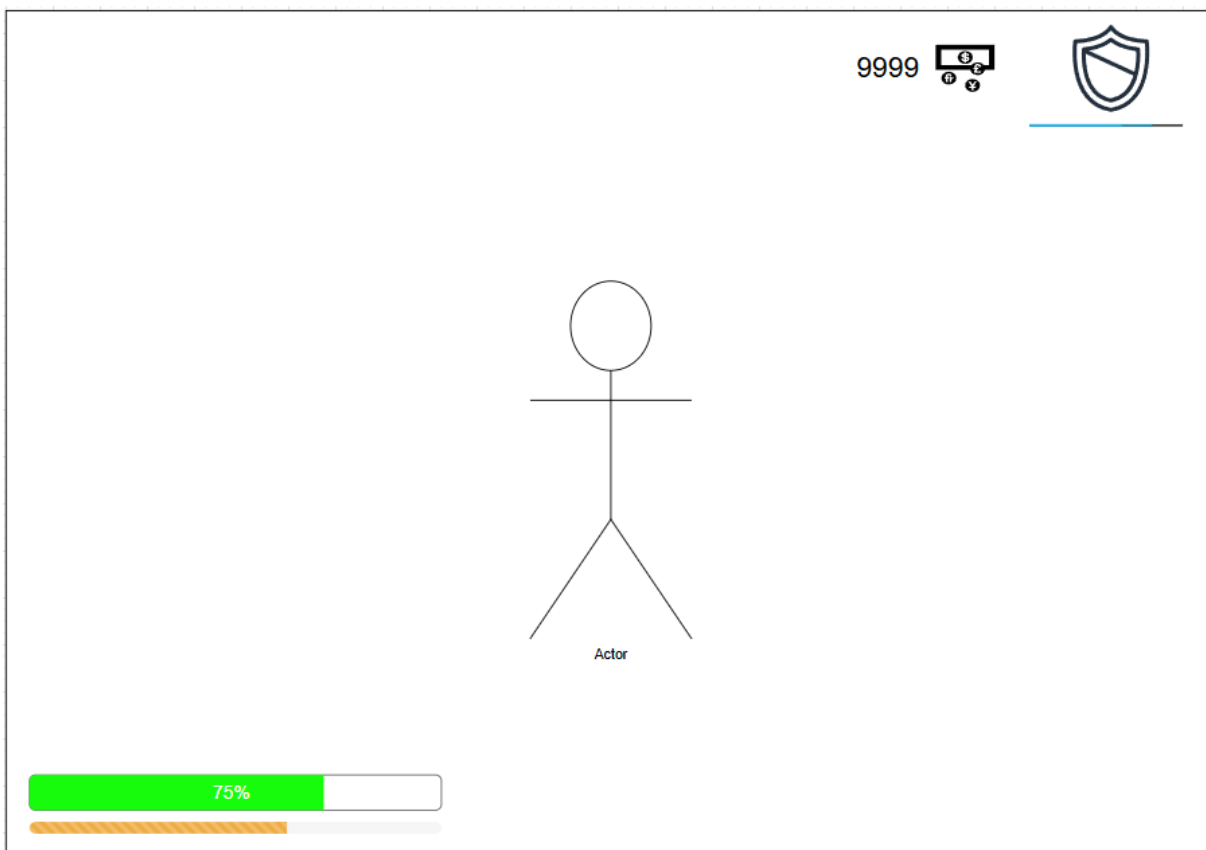


Рисунок 4.2 – Сторінка інтерфейсу управління персонажа

Інвентар персонажа, можливість перегляду та управління предметами в інвентарі, операції з предметами: equip, use, drop. Фільтрація предметів за категоріями: зброя, броня, ліки (Рис. 4.3).

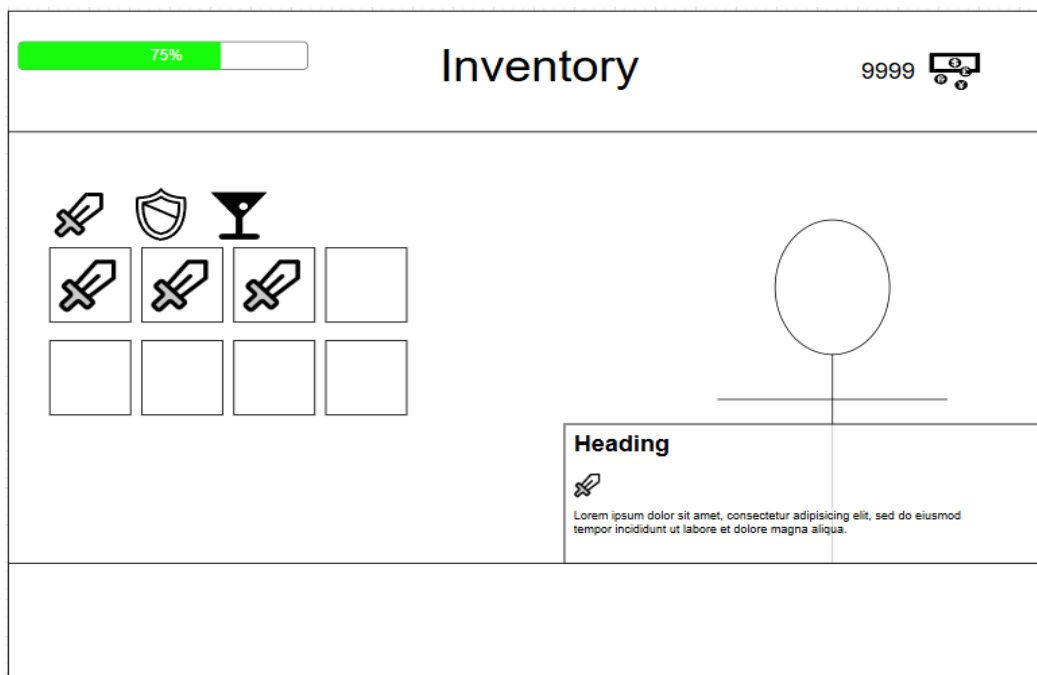


Рисунок 4.3 – Сторінка управління інвентарем гравця

Навігація та мінікарта, відображення локальної локації, позначення важливих об'єктів та NPC - неігрові персонажі(Рис. 4.4)

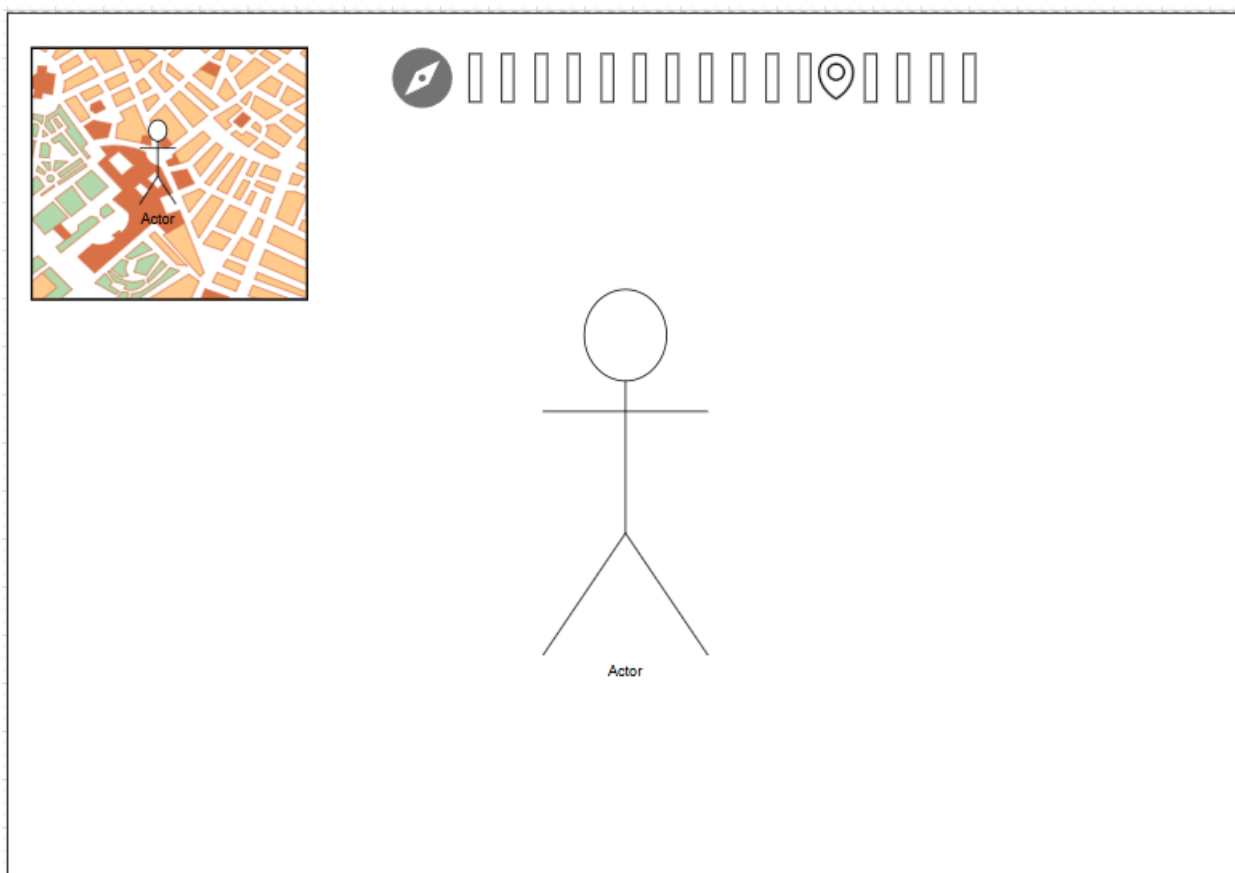


Рисунок 4.4 – Відображення інтерфейсу навігації

4.1.2 Забезпечення функціонування головного меню гри:

- 4.1.2.1 Запуск нової сесії гри.
- 4.1.2.2 Завантаження останнього збереження.
- 4.1.2.3 Доступ до тренувальної арени.
- 4.1.2.4 Налаштування гри.
 - 4.1.2.4.1 Налаштування режиму вікна.
 - 4.1.2.4.2 Налаштування роздільної здатності.
 - 4.1.2.4.3 Якість графіки гри.
 - 4.1.2.4.4 Вертикальна синхронізація.

4.1.2.5 Завершення гри та повернення на робочий стіл.

4.1.3 Забезпечення функціонування ігрових механік:

4.1.3.1 Керування камерою.

4.1.3.1.1 Вільне пересування камерою XY 2D-axis.

4.1.3.1.2 Захоплення цілі.

4.1.3.2 Пересування гравця.

4.1.3.2.1 Ходьба: вліво/право, вперед/назад.

4.1.3.2.2 Біг: вліво/право, вперед/назад.

4.1.3.2.3 Стрибок.

4.1.3.2.4 Присідання.

4.1.3.2.5 Перестрибування.

4.1.3.2.6 Скелелазіння.

4.1.3.2.7 Ухиляння.

4.1.3.3 Можливості персонажа.

4.1.3.3.1 Створення звуку для відволікання.

4.1.3.3.2 Вибір бойового стану.

4.1.3.3.3 Телепорт гравця.

4.1.3.3.4 Взаємодія з об'єктами.

4.1.3.4 Бойова система персонажа.

4.1.3.4.1 Атака магічними ударами.

4.1.3.4.2 Атака ближнього бою.

4.1.3.4.3 Проведення комбо-ударів.

4.1.3.4.4 Блокування.

4.1.3.4.5 Парирування.

4.1.3.5 Система характеристик персонажа.

4.1.3.5.1 Здоров'я.

4.1.3.5.2 Атака.

4.1.3.5.3 Досвід та рівень.

4.1.3.5.4 Витривалість.

- 4.1.3.5.5 Золото.
- 4.1.3.6 Квестова система.
 - 4.1.3.6.1 Відкриття діалогу при взаємодії з NPC.
 - 4.1.3.6.2 Отримання завдання.
 - 4.1.3.6.2.1 Квест має назву, ціль, опис.
 - 4.1.3.6.3 Відстеження прогресу.
 - 4.1.3.6.4 Повідомлення про завершення.
 - 4.1.3.6.5 Отримання винагороди.
- 4.1.3.7 Інтерактивне середовище.
 - 4.1.3.7.1 Взаємодія з об'єктами.
 - 4.1.3.7.1.1 Об'єкти, які можна досліджувати: скрині, двері, механізми.
 - 4.1.3.7.1.2 Взаємодія через натискання клавіші E в радіусі досяжності.
 - 4.1.3.7.2 Фізичні об'єкти (переміщення, піднімання).
 - 4.1.3.7.2.1 Піднімання предметів (зброї, ресурсів, предметів квесту).
 - 4.1.3.7.2.2 Можливість перетягувати об'єкти (наприклад, ящики для вирішення головоломок).
- 4.1.3.8 Навігація та міні-карта.
 - 4.1.3.8.1 Міні-карта.
 - 4.1.3.8.1.1 Відображає поточне розташування гравця.
 - 4.1.3.8.1.2 Динамічно обертається відносно напрямку камери.
 - 4.1.3.8.1.3 Використання Scene Capture 2D для стилізованого вигляду.
 - 4.1.3.8.2 Компас.
 - 4.1.3.8.2.1 Відображення основних напрямків (N, E, S, W).
 - 4.1.3.8.2.2 Динамічна адаптація до орієнтації камери.

4.1.4 Забезпечення функціонування ігрового інтерфейсу(компоненти HUD):

4.1.4.1 WB_HUD.

4.1.4.1.1 Рівень та досвід гравця.

4.1.4.1.2 Ігрова валюта.

4.1.4.1.3 Події/повідомлення.

4.1.4.2 WBP_HealthBar.

4.1.4.2.1 Динамічна шкала здоров'я.

4.1.4.2.2 Стилiстичний підрахунок здоров'я у вигляді сердець.

4.1.4.3 WBP_Compass.

4.1.4.3.1 Відображення напрямків (N, E, S, W).

4.1.4.3.2 Динамічне оновлення залежно від напрямку погляду.

4.1.4.4 WBP_Minimap.

4.1.4.4.1 Відображення навколишнього середовища.

4.1.4.4.2 Динамічне обертання відносно напрямку камери.

4.1.4.5 W_BossHealthBar.

4.1.4.5.1 Окрема панель HP, що з'являється при початку бою з босом.

4.1.4.5.2 Назва боса, індикатор рівня.

4.1.4.6 WB_QuestGiverDialogue.

4.1.4.6.1 Виведення реплік персонажа.

4.1.4.6.2 WB_QuestGiver.

4.1.4.6.2.1 Отримання квесту.

4.1.4.6.3 WB_QuestObjective.

4.1.4.6.3.1 Відображення активних квестів.

4.1.4.6.3.2 Статус виконання.

4.1.4.7 WBP_Inventory.

4.1.4.7.1 Відображення інвентарю гравця.

4.1.4.7.2 Можливість переглядати, сортувати, використовувати або екіпірувати предмети.

4.1.4.7.3 Підтримка drag-and-drop функціональності.

4.1.4.7.4 Відображення опису предметів та їх характеристик.

4.1.4.8 WBP_Interact.

4.1.4.8.1 Відображення повідомлення при можливості взаємодії з об'єктом (наприклад, «Натисніть [E], щоб взаємодіяти»).

4.1.4.8.2 Динамічне оновлення залежно від типу об'єкта.

4.1.4.9 WB_MainMenu.

4.1.4.9.1 Головне меню гри з доступом до нової гри, продовження, арені, налаштувань та виходу.

4.1.4.10 WB>LoadingScreen - Завантажувальний екран між переходами сцен.

4.1.4.11 WB_PauseMenu.

4.1.4.11.1 Меню паузи з можливістю продовжити гру, повернення до головного меню, зміни налаштувань.

4.1.4.11.2 Блокує ігрову взаємодію під час паузи.

4.1.5 Забезпечення функціонування взаємодії з інвентарем:

4.1.5.1 Підбір предметів.

4.1.5.1.1 Підбір через взаємодію.

4.1.5.1.2 Надання системою.

4.1.5.2 Огляд власних предметів.

4.1.5.3 Переміщення предметів у інвентарі.

4.1.5.4 Вибір зброї з інвентаря.

4.1.5.5 Вибір щита з інвентаря.

4.1.5.6 Застосування предметів: equip/use/eat.

4.1.5.7 Викидання непотрібних предметів.

4.1.5.8 Перегляд балансу ігрової валюти.

4.1.5.9 Перегляд здоров'я персонажа.

4.1.6 Забезпечення функціонування штучного інтелекту неігрових персонажів гри:

4.1.6.1 Базова поведінка NPC.

4.1.6.1.1 Патрулювання по заданих маршрутах.

4.1.6.1.2 Реакція на гравця.

4.1.6.1.2.1 Виявлення.

4.1.6.1.2.2 Атака.

4.1.6.1.2.3 Втеча.

4.1.6.1.2.4 Реагування на шкоду.

4.1.6.1.3 Idle-анімації та природня поведінка(сидить, ходить, спілкується з іншими NPC).

4.1.6.1.4 Слідування за гравцем.

4.1.6.2 Ворожі NPC.

4.1.6.2.1 Виявлення гравця.

4.1.6.2.2 Перехід у бойовий стан.

4.1.6.2.3 Атакуюча поведінка(ближній/дальній бій).

4.1.6.2.4 Ухилення, блокування.

4.1.6.2.5 Режим переслідування гравця.

4.1.6.2.6 Режим втечі.

4.1.6.2.7 Використання укриттів.

4.1.6.3 Дружні NPC.

4.1.6.3.1 Надання квестів.

4.1.6.3.2 Діалоги.

4.1.6.3.3 Використання NPC(наприклад: вихованець).

4.1.6.4 Навігація та орієнтація в просторі.

4.1.6.4.1 Обхід перешкод.

4.1.6.4.2 Вибір оптимального маршруту.

4.1.6.4.3 Використання NavMesh для руху.

4.1.6.5 Станова машина (Behavior Tree).

4.1.6.5.1 Використання Behavior Tree для логіки прийняття рішень.

4.1.6.5.2 Blackboard для зберігання стану NPC.

4.1.6.6 Стан штучного інтелекту.

4.1.6.6.1 Заморожений.

4.1.6.6.2 Бойовий.

4.1.6.6.3 Слідчий.

4.1.6.6.4 Пасивний або патрулюючий.

4.1.7 Забезпечення взаємодії з колізією об'єктів:

4.1.7.1 Загальні налаштування.

4.1.7.1.1 Визначення типів об'єктів для системи колізій.

4.1.7.1.1.1 Персонажі.

4.1.7.1.1.2 Статичні/динамічні об'єкти.

4.1.7.1.1.3 NPC.

4.1.7.1.1.4 Фізичні об'єкти.

4.1.7.1.1.5 Тригери, невидимі бар'єри.

4.1.7.1.2 Призначення колізійних профілів (Collision Presets) відповідно до ролі об'єкта.

4.1.7.1.2.1 Pawn, WorldStatic, WorldDynamic, PhysicsBody, Trigger тощо.

4.1.7.2 Колізія персонажа.

4.1.7.2.1 Активне використання Capsule Component з відповідною сферою впливу.

4.1.7.2.2 Обмеження проходження крізь тверді об'єкти (наприклад, стіни, скелі).

4.1.7.2.3 Відключення колізії з об'єктами UI або прив'язаними елементами HUD.

4.1.7.2.4 Налаштування взаємодії з фізичними об'єктами: штовхання, торкання, блокування шляху.

- 4.1.7.2.5 Перешкоди для входу в закриті зони (двері, невидимі бар'єри) до певних умов.
- 4.1.7.3 Колізія бойових механік.
 - 4.1.7.3.1 Виявлення влучань (Hit Detection) за допомогою Line Trace, Sphere Trace, або Box Trace.
 - 4.1.7.3.2 Розрізнення між фізичним ударом і візуальним ефектом.
 - 4.1.7.3.3 Активація колізії зброї тільки під час атаки (наприклад, за допомогою Enable Collision у анімації).
- 4.1.7.4 Колізія для взаємодії (інтеракція).
 - 4.1.7.4.1 Використання Sphere Overlap або Line Trace для визначення наявності об'єкта попереду персонажа.
 - 4.1.7.4.2 Активація відповідного UI/підказки при входженні в зону колізії (наприклад, WBP_Interact).
- 4.1.7.5 Колізія для дверей, плит, тригерів.
 - 4.1.7.5.1 Двері мають Box Collision для виявлення контакту з натискною плитою або ключовим об'єктом.
 - 4.1.7.5.2 Плити активують подію лише за входження в колізію (OnComponentBeginOverlap).
 - 4.1.7.5.3 Можливість обмеження повторного активаційного тригера – уникнення нескінченних подій.
 - 4.1.7.5.4 Система перевірки умов: наявність квесту, ключа, подолання ворогів тощо.
- 4.1.7.6 Оптимізація та продуктивність.
 - 4.1.7.6.1 Вимкнення колізії для декоративних або непотрібних об'єктів (наприклад, дрібні рослини).
 - 4.1.7.6.2 Активація колізії лише за необхідності.
- 4.1.7.7 Основні вимоги:
 - Забезпечити можливість збереження прогресу гравцем.
 - Забезпечити максимально можливу швидкість гри.

- Забезпечити інтерактивність ігрового світу.

4.1.7.8 Додаткові вимоги:

- Забезпечити адекватну поведінку штучного інтелекту.

4.2 Вимоги до надійності

Передбачити контроль введення інформації та захист від некоректних дій користувача. Користувач не може виконати недопустимі дії в інтерфейсі, такі як використання неіснуючих предметів, неприпустимі переміщення, та інше, що непередбачувано грою.

В разі збоїв у роботі гри (наприклад, зависання додатка), час на відновлення системи не повинен перевищувати 30 секунд для відновлення ігрового прогресу.

Гра повинна автоматично створювати контрольні точки в ключових моментах (наприклад, після завершення квесту, в бою, або після збереження). Якщо гравець не завершив гру, прогрес зберігається на найближчій контрольній точці.

Гра повинна забезпечувати загальноприйнятий нижній поріг, при якому гравець може грати без критичних затримок або смикань, а саме 30 FPS (кількість кадрів на секунду) при рекомендованих системних вимогах вказується як цільовий показник стабільної продуктивності. А також гра повинна мінімізувати споживання ресурсів, ігрові ресурси (текстури, моделі, звуки) повинні бути оптимізовані для економії пам'яті

4.3 Умови експлуатації

Умови експлуатації згідно СанПін 2.2.2.542 – 96.

4.3.1 Вид обслуговування

Для даної гри передбачено періодичне оновлення гри для виправлення помилок, оптимізації продуктивності, підтримка і оновлення контенту гри (нові локації, персонажі, предмети, квести).

4.3.2 Обслуговуючий персонал

Вимоги до обслуговуючого персоналу не висуваються

4.4 Вимоги до складу і параметрів технічних засобів

Мінімальна конфігурація технічних засобів:

- операційна система: Windows 10 (64-bit);
- процесор: Intel Core i3-8100 або AMD Ryzen 3 1200;
- оперативна пам'ять: 8 GB RAM;
- відеокарта: NVIDIA GeForce GTX 750 Ti або AMD Radeon R7 360;
- жорсткий диск: 10–15 GB вільного простору;
- directX: Версія 11;
- інше: Клавіатура та миша, доступ до інтернету не потребується;

Рекомендована конфігурація технічних засобів:

- операційна система: Windows 11 (64-bit);
- процесор: AMD Ryzen 7 3750H;
- оперативна пам'ять: 16 GB RAM;
- відеокарта: NVIDIA GeForce RTX 2060;
- жорсткий диск: SSD із мінімум 15 GB вільного простору;
- directX: Версія 12;
- інше: Клавіатура та миша, доступ до інтернету не потребується;
- роздільна здатність дисплею: 1920×1080 або вища (Full HD).

4.5 Вимоги до інформаційної та програмної сумісності

Програмне забезпечення повинно працювати під управлінням операційних систем сімейства Windows 11.

4.5.1 Вимоги до вхідних даних

Вхідні дані для цієї гри не є критично важливими, оскільки більшість ігрового процесу зосереджена на внутрішній логіці та механіках, які не залежать від зовнішніх джерел даних.

4.5.2 Вимоги до вихідних даних

Вихідні дані гри чітко структуровані, що представляють з себе збереження поточного стану гри, включаючи всі дані, що дозволяють відновити гру на тому ж етапі. Це може включати позицію гравця, статус квестів, інвентар гравця, параметри персонажа: здоров'я, витривалість, валюту, досвід та рівень.

Приклад файлу збереження у форматі .sav:

```
{
  "PlayerID": "player_01",
  "Position": { "X": 120.5, "Y": 45.2, "Z": 10.0 },
  "Health": 8.5,
  "Stamina": 100.0,
  "Level": 12,
  "Experience": 1050,
  "ExperienceToNextLevel": 1200
  "Inventory": [
    { "ItemID": "sword_01", "Quantity": 1 },
    { "ItemID": "banana_01", "Quantity": 3 } ]
}
```

Логи гри теж входять до вихідних файлів, логування помилки, виключення, повідомлення системи.

Вихідний файл: Лог файл для системи моніторингу, який включає важливі дані для аналізу: ErrorCode, Message, Time.

4.5.3 Вимоги до мови розробки

Розробку виконати з використанням мови програмування C++23 та інструментів візуального скриптування Blueprints, вбудованих в Unreal Engine 5.

4.5.4 Вимоги до середовища розробки

Розробку виконати на платформі Unreal Engine 5 із використанням інтегрованого середовища Unreal Editor, яке забезпечує повний цикл створення, налагодження та тестування ігрових проєктів.

4.5.5 Вимоги до представленню вихідних кодів

Вихідний код програми має бути представлений у вигляді структурованого набору файлів проєкту Unreal Engine 5, що включає C++ класи, файли конфігурацій, ресурси Blueprints, а також необхідні матеріали, шейдери, текстури та інші ассети, згруповані відповідно до структури контенту Unreal Engine.

4.6 Вимоги до маркування та пакування

Вимоги до маркування та пакування не висуваються.

4.7 Вимоги до транспортування та зберігання

Вимоги до транспортування та зберігання не висуваються.

4.8 Спеціальні вимоги

Згенерувати інсталяційну версію програмного забезпечення.

5 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

5.1 Попередній склад програмної документації

У склад супроводжувальної документації повинні входити наступні документи на аркушах формату А4:

- пояснювальна записка;
- технічне завдання;
- текст програми;
- програма та методика тестування;
- керівництво користувача.

Графічна частина повинна бути виконана на аркушах формату А3 та містити наступні документи:

- схема структурна варіантів використання;
- схема структурна класів програмного забезпечення;
- креслення вигляду екранних форм.

5.2 Спеціальні вимоги до програмної документації

Програмні модулі, котрі розробляються, повинні бути задокументовані, тобто тексти програм повинні містити всі необхідні коментарі.

6 СТАДІЇ І ЕТАПИ РОЗРОБКИ

№	Назва етапу	Строк	Звітність
1.	Вивчення літератури за тематикою проєкту	25.02	
2.	Розробка технічного завдання	15.03	Технічне завдання
3.	Аналіз вимог та уточнення специфікацій	19.04	Специфікації програмного забезпечення
4.	Проектування структури програмного забезпечення, проектування компонентів	30.04	Схема структурна програмного забезпечення та специфікація компонентів (діаграма класів, схема алгоритму)
5.	Програмна реалізація програмного забезпечення	05.05	Тексти програмного забезпечення
6.	Тестування програмного забезпечення	10.05	Тести, результати тестування
7.	Розробка матеріалів текстової частини проєкту	14.05	Пояснювальна записка
8.	Розробка матеріалів графічної частини проєкту	20.05	Графічний матеріал проєкту
9.	Оформлення технічної документації проєкту	29.05	Технічна документація

7 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ

Тестування розробленого програмного продукту виконується відповідно до «Програми та методики тестування».