

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 в жанрі Fantasy FPS.  
Level design та інтелект противників  
(тема)

Виконав  
студент 4 курсу, групи ПЗП-20-5  
Кісельгова М.Є.

(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 старший викладач Новіков Ю.С.  
(посада, прізвище, ініціали)

Допускається до захисту  
Зав кафедри

\_\_\_\_\_ Дудар З.В.  
(підпис) (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 \_\_\_\_\_ Кісельговій Маргариті Євгенівні \_\_\_\_\_  
 (прізвище, ім'я, по батькові)

1. Тема роботи Ігровий програмний застосунок у жанрі Fantasy FPS. Level design та інтелект противників

Затверджена наказом по університету від 20 травня 2024 р. № 471Ст

2. Термін подання студентом роботи до екзаменаційної комісії 07.06.2024

3. Вихідні дані до роботи Розробити ігровий програмний застосунок в жанрі Fantasy шутер від третьої особи, а саме левел-дизайн, штучний інтелект супротивників, за допомогою ігрового рушію Unreal Engine 5 та мови програмування Blueprints.

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, стор.112, рис.93, джерел.11, табл. 2.

ІГРОВИЙ ПРОГРАМНИЙ ЗАСТОСУНОК, КОМП'ЮТЕРНІ ІГРИ, ТЕСТУВАННЯ, ШТУЧНИЙ ІНТЕЛЕКТ, ШУТЕР, BEHAVIOR TREE, BLUEPRINT, FPS, LEVEL DESIGN, UNREAL ENGINE 5.

Об'єкт розробки – штучний інтелект монстрів (гоблінів) та розробка дизайну рівнів, які є частиною комплексної роботи з розробки ігрового програмного застосунку у жанрі Fantasy FPS.

Мета розробки – створення ігрового додатка, здатного забезпечити високий рівень залучення користувача та ефективну взаємодію з ігровим середовищем.

Метод рішення – система візуального програмування Blueprint, ігровий рушій – Unreal Engine 5. Система контролю версій Git.

У результаті розробки створено систему штучного інтелекту гоблінів із використанням дерев поведінки, також втілений дизайн декількох рівнів на карті. Усе розроблене є частиною комплексної роботи з розробки ігрового програмного застосунку у жанрі Fantasy FPS.

ARTIFICIAL INTELLIGENCE, BEHAVIOR TREE, BLUEPRINT, COMPUTER GAMES, FPS, GAME SOFTWARE APPLICATION, LEVEL DESIGN, SHOOTER, TESTING, UNREAL ENGINE 5.

The development object is the artificial intelligence of monsters (goblins) and the development of level design, which are part of the comprehensive work on the development of a game software application in the Fantasy FPS genre.

The purpose of the development is to create a game application that can provide a

high level of user involvement and effective interaction with the game environment.

The solution method is the Blueprint visual programming system, the game engine is Unreal Engine 5. Git version control system.

As a result of the development, a goblin artificial intelligence system using behavioral trees was created, and the design of several levels on the map was also embodied. Everything developed is part of the complex work on the development of a game software application in the Fantasy FPS genre.

Я, Кісельгова Маргарита Євгенівна, студентка гр. ПЗП-20-5, здобувач вищої освіти на першому (бакалаврському) рівні кафедри «Програмна інженерія», заявляю: моя кваліфікаційна робота на тему «Ігровий програмний застосунок у жанрі Fantasy FPS», 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 ElAr KhNURE. Усі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомена 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.....	8
1 Аналіз предметноїгалузі .....	10
1.1 Аналіз предметної галузі .....	10
1.2 Виявлення проблем та актуалізація рішень.....	11
1.3 Постановка задачі .....	21
2 Формування вимог до програмної системи.....	23
2.1 Функціональні вимоги до ігрового застосунку .....	23
2.2 Нефункціональні вимоги до ігрового застосунку .....	25
3 Архітектура та проектування програмного забезпечення .....	26
3.1 UML проектування ПЗ .....	26
3.2 Вибір архітектури та рушія .....	29
3.3 Розгляд ігрового циклу .....	30
3.4 Розгляд найцікавіших алгоритмів.....	32
3.5 Створення UI/UX.....	34
4 Опис прийнятих програмних рішень .....	40
4.1 Розробка дизайну рівнів.....	40
4.2 Розробка штучного інтелекту ворогів .....	45
5 Тестування програмного забезпечення.....	50
5.1 Розробка мапи думок тестування.....	50
Висновки .....	53
Перелік джерел посилання .....	55
ДОДАТОК А Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ.....	57
ДОДАТОК Б Слайди презентації .....	58
ДОДАТОК В Геймдизайн-документ .....	67
ДОДАТОК Г Тест-план .....	86
ДОДАТОК Ґ Тези доповіді для науково-практичної інтернет-конференції.....	102
ДОДАТОК Д Тези доповіді для науково-практичної інтернет-виставки .....	107

ДОДАТОК Е Приклад програмного коду.....	109
---	-----

## ВСТУП

Індустрія відеоігор продовжує демонструвати стрімке зростання та впровадження інновацій, що відкривають безмежні горизонти для розвитку інтерактивного розважального контенту. Цей сектор є взірцем швидкоплинної динаміки, що підживлюється безперервним впровадженням новітніх технологій, зростанням кількості користувачів та збільшенням інвестицій у розробку ігор на різних платформах. Жанр шутерів від першої особи (FPS) посідає одне з провідних місць у індустрії відеоігор завдяки своїй здатності забезпечувати інтенсивний та захоплюючий ігровий досвід. Ігри цього жанру пропонують динамічний геймплей, який стимулює як фізичну реакцію, так і стратегічне мислення гравців. Візуальна перспектива від першої особи посилює відчуття присутності у віртуальному світі, роблячи кожен бій більш напруженим та реалістичним. Ця особливість жанру робить його особливо популярним серед тих, хто шукає активну взаємодію та повне занурення в ігрове середовище.

Мета даної дипломної роботи полягає в створенні ігрового застосунку, який би інтегрував ігрові механіки з сюжетною лінією, забезпечуючи гравцям новий рівень взаємодії з грою. Робота включає аналіз сучасних трендів у розробці відеоігор, проектування архітектури проекту, програмування ігрових механік, а також їх тестування та оптимізацію.

Однією з ключових характеристик проекту є розробка системи інтерактивності, що дозволяє гравцям використовувати магичні елементи для впливу на оточення та вирішення різноманітних задач, які ставить перед ними гра. Це включає в себе створення механізмів для імплементації стихійних атак, оборонних дій, а також різних комбінацій елементів для досягнення специфічних цілей в ігровому світі.

Областю та гілкою використання результату є індустрія цифрових розваг, зокрема розробка відеоігор для персональних комп'ютерів та ігрових консолей.

Для розробки практичної складової проекту було обрано систему візуального

програмування Blueprint, ігровий рушій – Unreal Engine 5. Система контролю версій – Git.

Результатом виконання цієї дипломної роботи стане не просто готовий до запуску ігровий додаток, але й платформа для подальшого розвитку та оновлення, з потенціалом для введення нових ігрових механік та розширення ігрового світу.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

## 1.1 Аналіз предметної галузі

Індустрія відеоігор невинно розвивається, постійно впроваджуючи нові технології, які змінюють сприйняття та можливості геймплея. Жанр шутерів від першої особи (FPS) відіграє визначальну роль у світі сучасних відеоігор, поєднуючи в собі інтенсивність бойових дій з глибокою стратегією та тактикою. Цей жанр забезпечує гравцям унікальний досвід, ставлячи їх безпосередньо у центр подій, де кожен крок та рішення можуть мати непередбачувані наслідки. Одним із ключових елементів, що сприяють успіху шутерів від першої особи, є їх здатність інтегрувати новітні технології для створення все більш реалістичного візуального та ігрового середовища.

Штучний інтелект (ШІ) противників у FPS іграх постійно розвивається, надаючи персонажам, керованим комп'ютером, здатність діяти обдуманно та тактично. Розробники вдосконалюють алгоритми ШІ, щоб противники могли адаптуватися до дій гравців, використовувати середовище для прикриття, обирати оптимальні позиції для атаки або відступу, та координувати свої дії з іншими персонажами. Важливим аспектом є також здатність ШІ аналізувати ефективність своїх дій та коригувати стратегію в реальному часі, що робить ігровий процес непередбачуваним і захоплюючим.

Дизайн рівнів у FPS вимагає глибокого розуміння того, як гравці взаємодіють із ігровим світом. Ефективний дизайн рівнів повинен балансувати між красою і функціональністю, забезпечуючи як візуально привабливі сцени, так і необхідні тактичні можливості для різноманітних стилів гри. Це означає створення множини маршрутів та просторів, які можуть підтримувати як агресивний штурм, так і обережне просування, інтеграцію відкритих просторів для стрільби на далекі відстані та замкнених приміщень для ближнього бою.

Жанр шутерів від першої особи (FPS) став популярним у ранні 90-ті роки з

появою таких ігор, як "Wolfenstein 3D" та "Doom". Ці ігри вперше дозволили гравцям відчувати себе безпосередніми учасниками дії, пропонуючи перспективу від першої особи та значно підвищуючи рівень іммерсії.

Протягом 90-х років та на початку 2000-х жанр швидко розвивався, вносячи нововведення у графіку, ігрові механіки та штучний інтелект противників. Розвиток технологій дозволив створювати все більш складні та реалістичні ігрові світи, що вимагали від гравців стратегічного мислення та швидкої реакції. Ігри, такі як "Half-Life" та "Call of Duty", перенесли акцент з фантастичних сценаріїв на реалістичні військові конфлікти, підвищуючи іммерсивність та рівень деталізації графіки.

Сучасні технології віртуальної реальності відкрили нові можливості для жанру, забезпечуючи ще більше занурення. Ігри, такі як "Half-Life: Alyx", демонструють потенціал VR у шутерах від першої особи, перевершуючи традиційні ігрові платформи за рівнем інтерактивності та іммерсивності.

Ці еволюції жанру відображають не лише технічний прогрес, але й зміну у вимогах та очікуваннях гравців, які тепер шукають більш глибокий, емоційно заряджений ігровий досвід.

## 1.2 Виявлення проблем та актуалізація рішень

Після проведення аналізу предметної галузі, можемо сказати, що існує ряд проблем, з яким зазвичай зустрічаються подібні проекти. Отже, розглянемо декілька ігрових додатків, що є конкурентними для нашого. Їх варто розглядати з погляду обраної теми у межах комплексної дипломної роботи, тому розглядатися будуть не лише ігри близькі за жанром до розроблюваному продукту (Fantasy FPS.), а й інші, наприклад FPS інших піджанрів. .

Першим конкурентом для розгляду буде Metro Exodus – це шутер від першої

особи з елементами survival horror та стелс, розроблений українською студією 4A Games і виданий Deep Silver у лютому 2019 року для платформ PlayStation 4, Xbox One, Microsoft Windows та Stadia. Гра є продовженням серії "Metro", заснованої на романах Дмитра Глуховського, і занурює гравців у пост апокаліптичну Росію, де головний герой Артем разом з групою побратимів та тих, хто вижив, намагається знайти новий дім поза радіоактивною Москвою. Відмінною особливістю "Metro Exodus" є її відкриті локації, що надають гравцям більше свободи для дослідження порівняно з попередніми іграми серії. Система крафтингу дозволяє створювати унікальну зброю та інші корисні предмети із знайдених ресурсів, а сюжет гри адаптується до виборів гравця, впливаючи на кінцевий результат історії. Гра використовує удосконалений двигун 4A Engine, який забезпечує вражаючу графіку, деталізовані погодні ефекти та реалістичне освітлення, що допомагає створити переконливий, моторошний світ, повний небезпек.

Розглянемо та розберемо локації гри.

Метро. Початкова локація гри, метро, є темним і похмурым місцем, повним небезпек та мутантів. (див. рис. 1.1). Гравці знайомляться з основними механіками гри, досліджуючи вузькі тунелі та занедбані станції.



Рисунок 1.1 – Локація метро

До переваг можна віднести:

- інтенсивне введення в сюжет гри, що забезпечує глибоке занурення у похмуру атмосферу світу Metro;
- примушення гравців швидко адаптуватися до виживання в екстремальних умовах.

Основні недоліки:

- обмежений простір може викликати відчуття стиснутості та обмеженості дій;
- високий рівень загрози від мутантів може бути складним для новачків.

Волга. Волга представляє собою величезну річкову дельту з багатьма водними просторами та заболоченими територіями, де доводиться користуватися човном для переміщення між островами. Ця локація вражає своєю відкритістю та можливостями для дослідження (див. рис. 1.2).

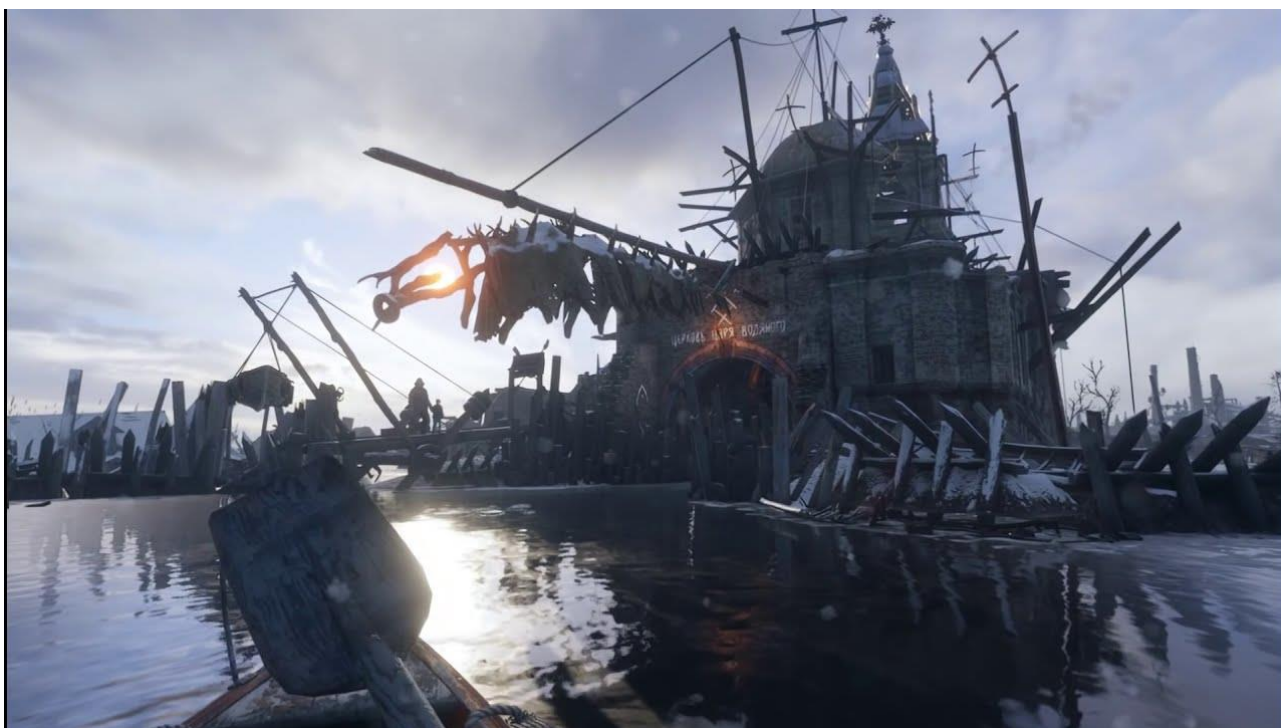


Рисунок 1.2 – Волга

До переваг можна віднести:

- введення нової механіки веслування човном;
- відкритість локації сприяє дослідженню та знаходженню різних предметів і ресурсів.

Основні недоліки:

- велика кількість води може ускладнити переміщення та зробити деякі частини локації менш доступними.

Ямантау. Ямантау – це гірський бункер, що став домівкою для групи канібалів (див. рис. 1.3). Локація складається з закритих підземних комплексів і коридорів, що створює напружену та моторошну атмосферу. Через обмеженість локації, переваги та недоліки майже такі самі, як на локації Московського метро.



Рисунок 1.3 – Ямантау

Каспій. Каспій – це пустельна локація з відкритими просторами та різноманітними зовнішніми умовами (див. рис. 1.4). Ця арена надає гравцям більше свободи для дослідження, пропонуючи широкий спектр тактик і стратегій. Велика

кількість відкритого простору однозначно є перевагою, а також, візуальна унікальність локації, роблять її дуже бажаною для дослідження.



Рисунок 1.4 – Каспій

Тайга. Тайга – це лісиста локація з густими лісами та великими водоймами (див. рис. 1.5). Локація пропонує гравцям чудову візуальну красу та можливості для прихованого підходу в боях.



Рисунок 1.5 – Тайга

Новосибірськ. Новосибірськ, мертво місто, є фінальною локацією (див. рис. 1.6). Місто покрите снігом і льодом, що додає унікальний характер ігровому процесу.

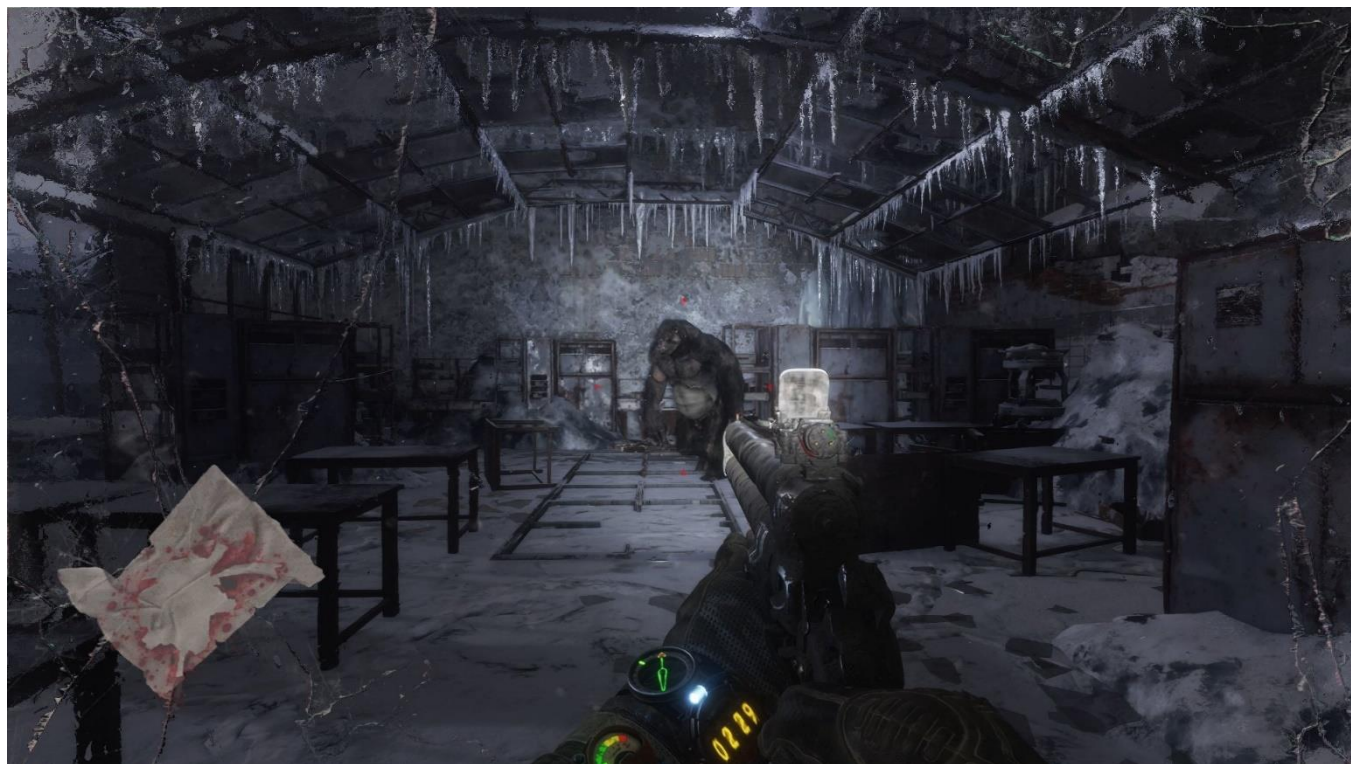


Рисунок 1.6 – Новосибірськ

Локації в "Metro Exodus" вирізняються глибокою атмосферністю та візуальною привабливістю, що сприяє зануренню в пост апокаліптичний світ. Великі відкриті простори забезпечують відмінні можливості для дослідження і тактичного маневрування, але також ставлять перед гравцями складні завдання з виживання через постійні загрози від мутантів та складні погодні умови. Обмежені закриті простори створюють напружену атмосферу і спонукають до обережності та стратегічного підходу, проте можуть викликати відчуття стиснутості та обмеженості дій, що іноді призводить до фрустрації серед гравців.

Графіка та візуальні ефекти в Metro Exodus.

Гра Metro Exodus вирізняється своєю візуальною складністю та деталізацією, завдяки чому вона вважається однією з найкрасивіших ігор постапокаліптичної

тематики. Розробники використали оновлений графічний двигун 4A Engine, що дозволило досягнути вражаючої якості зображення, освітлення, та реалістичних погодних ефектів.

До переваг можна віднести:

- деталізація світу: локації в грі виконані з високою деталізацією, від текстур ґрунту до відтворення зруйнованих будівель, що додає реалістичності загальній атмосфері;
- реалістичне освітлення: освітлення в "Metro Exodus" забезпечує природні переходи від дня до ночі, а також ефектні візуальні ефекти, які відіграють ключову роль у створенні настрою та напруженості гри;
- погодні ефекти: динамічна погода змінює не тільки візуальну картину, але й ігрові умови, від сильних бур до снігопадів, що впливає на поведінку персонажів і мутантів.

Основні недоліки:

- вимоги до апаратного забезпечення: висока деталізація та якість графіки вимагають значних ресурсів від системи, що може стати проблемою для гравців з менш потужними ПК або консолями;
- падіння кадрів на секунду (fps): У деяких випадках, особливо у великих відкритих локаціях з багатьма елементами на екрані, може спостерігатися зниження швидкості кадрів, що негативно впливає на загальне враження від гри.

Вороги. У грі присутні багато різновидів супротивників, такі як: люди, стражі (мутанти), павуки, Кикимори, звіролюди, Демони, креветки (мутанти), вовки, носачі, та декілька босів – мутантів (див. рис. 1.7). Можна виділити певний мінус в їх системі – нерівномірна градація сил супротивників. Від креветки можна відбитися веслом човна, а на звіролюдину доведеться витратити не одну обойму, хоча на звичайну людину достатньо декілька патронів. Такий різкий перепад сил ворогів може відбити у новачків бажання продовжити гру.



Рисунок 1.7 – Креветка (мутант)

Сюжет. Однією з ключових особливостей сюжету є його здатність впливати на гравця через рішення, які приймаються під час гри. Ці рішення визначають долі персонажів та розвиток подій, роблячи кожен ігрову сесію унікальною. Взаємодія з іншими персонажами є значною частиною гри, оскільки вони мають свої історії, мотивації, та часто ставлять перед гравцем моральні дилеми. Це додає сюжету глибини та змушує гравців замислюватись над своїми виборами і їхніми наслідками.

Однак, попри можливість вибору, залишається відчуття лінійності в розгалуженні сюжету та деякі моменти, які можуть здатися затягнутими або надмірно складними. Також, іноді гра може змусити гравців відчувати, що їхні вибори не мають значного впливу на кінцівку історії.

Наступною грою розглянемо Destiny 2. Destiny 2 — це масштабна багатокористувацька гра у жанрі шутера від першої особи, розроблена американською компанією Bungie. Вона є продовженням гри Destiny, що вийшла у 2014 році. Гра була випущена у вересні 2017 року та доступна на платформах PlayStation 4, Xbox One, та

ПК.

Гра занурює гравців у футуристичний, пост апокаліптичний світ, де вони виступають у ролі Гвардійців — захисників останнього міста людства. Гравці борються з ворожими прибульцями, здійснюють різноманітні місії та квести, а також змагаються у різних PvP (гравець проти гравця) та PvE (гравець проти середовища) режимах.

Destiny 2 відома своєю вражаючою графікою, складними багатокористувацькими режимами, та динамічним ігровим процесом. Гра регулярно оновлюється, включаючи нові розширення та сезонні події, що постійно розширюють всесвіт та можливості геймплею.

Локації (див. рис. 1.8). Для аналізу локацій в грі "Destiny 2", можна врахувати їхню візуальну різноманітність, атмосферність і взаємодію, яку вони пропонують гравцям:



Рисунок 1.8 – площа Вежі

Локації в грі "Destiny 2" вирізняються візуальною різноманітністю та деталізацією, кожна з яких має свою унікальну атмосферу і естетику, що додає глибини світові гри та забезпечує незабутні враження. Світ гри реагує на дії гравців,

змінюючи деякі аспекти локацій або активуючи події, що робить ігровий процес динамічним та непередбачуваним. Однак, деякі локації можуть бути складними для навігації через їхню розлогість або складну топографію, що іноді може призводити до фрустрації гравців. Також через велику кількість подібних завдань або ворогів у деяких локаціях, гравці можуть відчувати повторюваність.

Вороги. У грі "Destiny 2" гравці зустрічаються з широким спектром ворогів, кожен з яких має свої унікальні характеристики та тактики поведінки, від інвазійних військ Кабал до містичних Прийнятих, що вимагають від гравців різних підходів та роблять бої цікавими та викликаючими; вороги володіють спеціалізованими здібностями, такими як щити або спецатаки, що покращує загальну атмосферу гри, однак іноді гравці можуть стикатися з надто складними супротивниками, що може викликати фрустрацію, особливо для новачків, а постійні бої з подібними ворогами можуть призвести до відчуття повторення; іноді штучний інтелект ворогів може демонструвати нелогічну поведінку, що знижує реалістичність боїв, хоча загалом вороги у "Destiny 2" забезпечують багатий і динамічний ігровий процес, де кожна зустріч вимагає стратегічного мислення та швидкої реакції.

На основі аналізу двох популярних ігор жанру FPS, "Metro Exodus" та "Destiny 2", можна зробити декілька висновків про особливості сюжету, ворогів та локацій у цьому жанрі.

Сюжети у "Metro Exodus" та "Destiny 2" залучають глибиною та емоційністю, але можуть бути складними для новачків через велику кількість деталей та подій. Вороги у цих іграх відрізняються великою різноманітністю та потребують від гравців адаптації стратегій, але іноді можуть спричинити відчуття монотонності. Локації в обох іграх мають високу деталізацію та атмосферність, забезпечуючи глибоке занурення в ігровий світ, однак складність навігації може створювати додаткові виклики.

### 1.3 Постановка задачі

Нижче наведено механіки та задачі, які повинні бути в ігровому застосунку, у відповідній частині, що виконується в якості дипломного проекту.

Штучний інтелект ворогів:

- Патрулювання ворогів: система, яка дозволяє ворогам виконувати рутинні маршрути патрулювання, які змінюються в залежності від контексту ігрової ситуації; вороги мають реагувати на зміни в ігровому світі, такі як звуки або візуальний контакт з гравцем, перериваючи патрулювання для перевірки джерела збурення.
- Системи пошуку гравця ворогом: система, що дозволяє ворогам виявляти гравця, бігти за ним, при цьому наносячи йому урон, у разі зникнення гравця – бігати від останнього його місця знаходження до рандомних місць на карті 20 с, потім повертатися на патрулювання.
- Поведінки ворогів в різних випадках: деталізована система поведінки для ворогів, що включає різні реакції на атаки, пошкодження, виявлення гравця та інші ігрові події. Використання станових машин або системи поведінкових дерев для управління реакціями ворогів на різноманітні стимули. Наприклад, перехід від стану патрулювання до агресивної атаки при виявленні гравця, або втечі при отриманні важких пошкоджень.

Дизайн рівнів:

- Цілі та структура рівня: створення дизайн-документа для кожного рівня з основними цілями, розташуванням ключових елементів і зон дії, планування має сприяти логічному потоку гри і підтримувати плавний ігровий процес.
- Інтерактивність рівня: розробка механізмів для інтеграції гравця з

рівнем, які можуть включати використання об'єктів, зміни у середовищі та вплив на динаміку гри через реакції на дії гравця.

- Візуальна складова: розробка естетично привабливих текстур, освітлення і звукового дизайну, що гармонійно інтегруються з загальною атмосферою гри, підсилюючи іммерсивність та занурення гравця у гру.

У цьому пункті було сформульовано конкретні задачі для розробки ігрового застосунку. Визначено критичні аспекти ігрових механік, які необхідно реалізувати для досягнення цілей проекту. Задачі охоплюють відтворення рухів персонажа, бігу, присідання, взаємодію з предметами, а також розробку складної поведінки супротивників, яка передбачає уникнення відкритої конфронтації і підкреслює стелс-компонент гри.

Ці задачі відображають розуміння сучасних трендів у розробці ігрових систем та інтерактивності в іграх жанру Fantasy FPS, де глибина поведінкових паттернів та реалістичність взаємодій мають велике значення. Визначені завдання демонструють здатність інтегрувати складні аспекти ігрового дизайну для створення захоплюючого та емоційно-впливового ігрового досвіду.

## 2 ФОРМУВАННЯ ВИМОГ ДО ПРОГРАМНОЇ СИСТЕМИ

### 2.1 Функціональні вимоги до ігрового застосунку

Формування вимог до програмної системи є критичним етапом у розробці ігрового застосунку, оскільки воно визначає ключові функціональні та нефункціональні параметри, які повинні бути реалізовані для досягнення заданих цілей проекту. Цей процес включає детальне визначення того, що саме повинна робити програма, як вона повинна взаємодіяти з користувачами, які технологічні рішення будуть застосовані, та які обмеження мають бути враховані.

У цій частині дипломного проекту будуть розглянуті основні функціональні вимоги до штучного інтелекту ворогів, дизайну рівнів, а також забезпечення необхідної взаємодії користувача з програмною системою. Окрім того, будуть визначені ключові технічні та операційні характеристики, які повинні бути втілені в ігровому застосунку для досягнення високої якості геймплею та користувацького досвіду.

Розглянемо детальніше функціональні вимоги для забезпечення необхідної взаємодії гравця з ігровим застосунком. До них входять такі функції:

- стрілянина, а саме: завдання шкоди, фізичне відображення ефектів попадання стихійних снарядів;
- поповнення здоров'я через їжу;
- пересування персонажа (біг, стрибки, прискорення, навпочіпки);
- керування предметами;
- нанесення урону гравцю самим гравцем (падіння з вікна);
- підняття предметів;
- пересування предметів на рівні.

Система штучного інтелекту ворогів передбачає такі функції:

### 1. Патрулювання ворогів:

- Маршрутизація: Розробка системи маршрутів патрулювання, які задаються через серію точок на карті. Ці маршрути повинні бути налаштовані на зміни в залежності від ситуацій у грі.
- Реакція на подразники: Імплементация механізму, за яким вороги тимчасово припиняють патрулювання і переходять до розслідування джерела подразнення, такого як звуки або візуальний контакт із гравцем.

### 2. Системи пошуку гравця ворогом:

- Виявлення гравця: Розробка алгоритмів для визначення положення гравця в полі зору ворогів або при збуреннях, наприклад, від шумів або руху.
- Переслідування: Створення логіки переслідування, де вороги активно намагаються наздогнати гравця і завдати урону, з можливістю втрати цілі та повернення до випадкових пошуків або базового стану патрулювання.

### 3. Поведінка ворогів в різних випадках:

- Станові машини або поведінкові дерева: Розробка деталізованої системи управління поведінкою, яка дозволить ворогам адаптуватися до різних ситуацій на полі бою, наприклад, зміна тактики з агресивної атаки на відступ при отриманні серйозних пошкоджень.
- Реакції на ігрові події: Імплементация логіки реакцій на специфічні ігрові події, такі як виявлення гравця, отримання урону або підтримка від інших ворогів.

Під час розробки даного ігрового додатку я буду виконувати ці конкретні завдання.

## 2.2 Нефункціональні вимоги до ігрового застосунку

Нефункціональні вимоги також важливі для задоволення очікувань користувачів та забезпечення їхньої задоволеності користувацьким досвідом. Вони допомагають переконатися, що програмна система не тільки виконує потрібні дії, але й робить це ефективно, безпечно і зручно. У цій частині проекту будуть розглянуті ключові нефункціональні вимоги, які необхідно врахувати для забезпечення надійності та ефективності ігрового застосунку.

До головних нефункціональних вимог ігрового програмного забезпечення, що розробляється, можна віднести:

- анімації переходів між елементами інтерфейсу;
- переходи за допомогою клавіатури поміж елементами інтерфейсу;
- відображення попадання стихійних снарядів в певні предмети на локації;
- гравець має можливість призупинити гру, активувавши режим паузи;
- гравець може відновити гру після паузи;
- взаємодія гравця з інтерфейсними кнопками;
- гравець може розпочати нову гру з головного меню.

При розробці програмного застосунку, особливу увагу потрібно приділити середовищу розгортання, оскільки воно безпосередньо впливає на ефективність, стабільність та доступність програми для кінцевих користувачів. Для розробки дипломного проекту було обрано ігровий рушій Unreal Engine 5, систему візуального програмування Blueprint та систему контролю версій – Git.

### 3 АРХІТЕКТУРА ТА ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

#### 3.1 UML проєктування ПЗ

У процесі розробки ігрового застосунку, була побудована загальна Use Case діаграма [1] для всього проєкту (див. рис. 3.1).

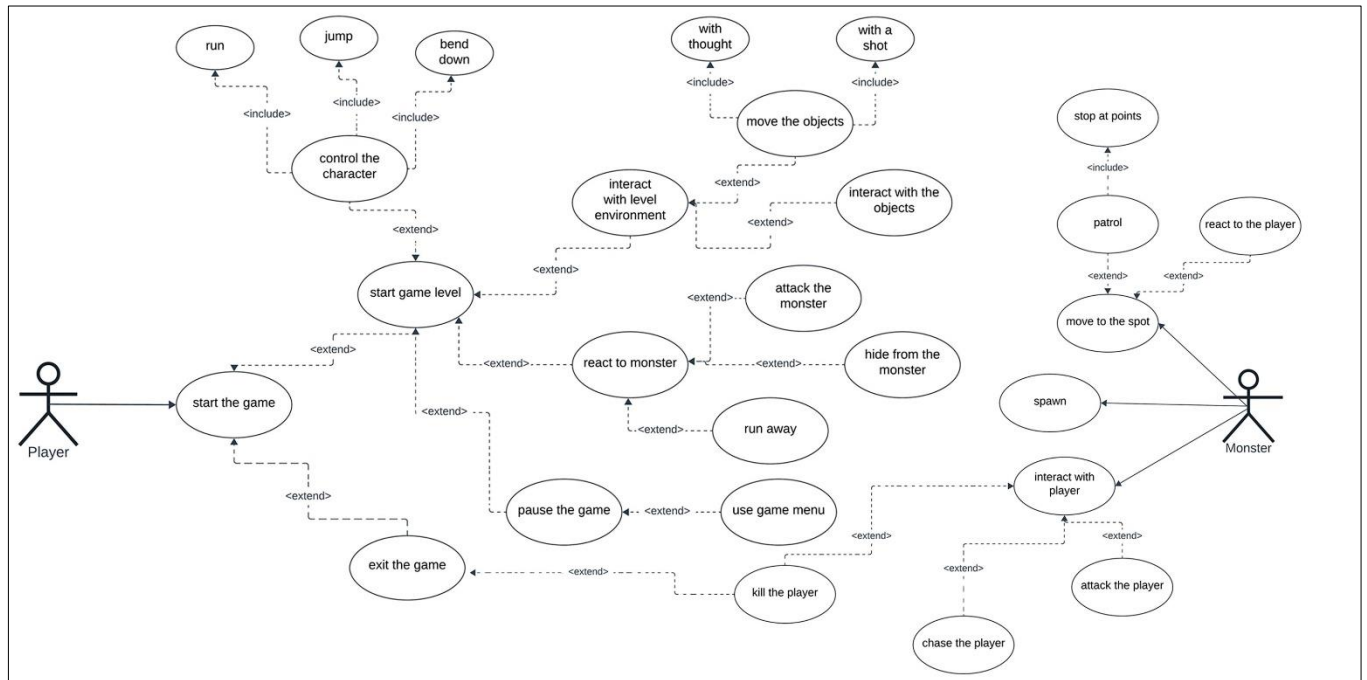


Рисунок 3.1 – Use Case діаграма ігрового застосунку

Розглянувши загальну діаграму системи ігрового застосунку, можна визначити основних акторів.

"Player" — це основний актор в ігровій системі, який представляє користувача, що взаємодіє з ігровим світом. Гравець має можливість керувати головним персонажем гри, виконувати різні дії, такі як рух, використання магії, боротьба з ворогами, взаємодія з ігровим середовищем, розв'язання завдань на кшталт пошуку предметів чи розгадування таємниць. Гравець також відповідає за стратегічні рішення, які впливають на розвиток сюжету та взаємодію з іншими персонажами гри.

"Monster" — це ворожі істоти або противники, з якими стикається гравець

протягом гри. Монстри є ключовою частиною ігрових викликів та перешкод. Вони можуть мати різноманітні властивості та поведінку: від простих патрулів на певних територіях до складних бойових зіткнень, де вони використовують спеціальні атаки чи магічні здібності.

Далі розглянемо основні дії гравця:

#### 1. Start the Game (Розпочати гру)

- Гравець ініціює початок нової гри або продовження збереженої гри.

#### 2. Control the Character (Керування персонажем)

- Основний процес, що включає управління персонажем. Воно дозволяє:
  - Run (Бігти): Швидке переміщення персонажа по ігровому світу.
  - Jump (Стрибати): Подолання перешкод або переміщення через них.
  - Bend Down (Нагнути): Ухиляння або доступ до об'єктів нижче звичайного рівня.

#### 3. Interact with Level Environment (Взаємодія з середовищем рівня)

- Гравець може взаємодіяти з об'єктами або елементами ігрового середовища, що може включати:
  - Move the Objects (Рухати об'єкти): Переміщення об'єктів у рамках ігрового світу.
  - With Thought (За допомогою думки): Механіка взаємодії зі складними або інтерактивними об'єктами, що візуально представлена, як телекінез.
  - With a Shot (Зі стрілом): Взаємодія, що вимагає використання зброї для зміщення предметів.

#### 4. React to Monster (Реагування на монстра)

- Відповідь на дії або присутність монстра, що може включати:
  - Run Away (Втекти): Відступ від зустрічі з монстром.
  - Attack the Monster (Атакувати монстра).
  - Hide from the Monster (Сховатися від монстра): Заховатися для

уникнення виявлення або атаки з боку монстра.

5. Pause the Game (Пауза в грі)

- Тимчасове призупинення ігрового процесу.

6. Use Game Menu (Використання ігрового меню)

- Взаємодія з меню гри для зміни налаштувань, збереження гри або виходу з неї.

7. Exit the Game (Вихід з гри)

- Закінчення ігрового сеансу і вихід з гри.

Далі розглянемо основні дії монстра:

1. Spawn (З'явлення)

- Монстр з'являється в грі в певному місці. Це початковий стан монстра при його вході в ігровий світ.

2. Patrol (Патрулювання)

- Монстр переміщується за заданим маршрутом усередині своєї території. Ця поведінка включає зупинки в певних точках, що представлено дією "stop at points".

3. React to the Player (Реагувати на гравця)

- Монстр реагує на дії або присутність гравця. Це може включати переміщення до місця, де перебуває гравець ("move to the spot"), або ініціацію атаки, якщо гравець знаходиться в полі зору або слуху монстра.

4. Attack the Player (Атакувати гравця)

- Монстр нападає на гравця. Це основна взаємодія монстра з гравцем, де монстр намагається завдати шкоди або іншим чином вплинути на стан гравця.

5. Chase the Player (Переслідувати гравця)

- Якщо гравець намагається втекти або знаходиться на відстані від монстра, монстр починає переслідування. Ця дія пов'язана з спробами монстра

скоротити відстань до гравця, щоб продовжити атаку або перехоплення.

У цій частині було розглянуто основні дії гравця та монстра ігрового застосунку. Було наведено Use Case діаграму для загальної частини проекту.

### 3.2 Вибір архітектури та рушія

На основі технологічних вимог до проекту було вирішено використовувати сучасні інструменти та платформи для розробки ігрового застосунку. Вибір платформи та інструментів впливає на ефективність розробки, можливості імплементації задуманих ігрових механік, а також на якість кінцевого продукту.

Unreal Engine 5 було обрано як ігровий рушій через його передові можливості у створенні високо деталізованих і візуально імпресивних ігрових світів. Цей рушій вирізняється підтримкою складних анімаційних та освітлювальних ефектів, що дозволяє нам створювати реалістичні та атмосферні сцени, які є досить важливими для жанру FPS.

Система Blueprint [2] виявилась найкращою через її здатність забезпечувати швидку та інтуїтивно зрозумілу розробку ігрових механік. Ця система візуального програмування дозволяє дизайнерам і розробникам ефективно співпрацювати, мінімізуючи необхідність глибоких знань у програмуванні, що оптимізує процес розробки та знижує часові витрати на реалізацію складних ігрових функцій [3].

Git був вибраний як система контролю версій через його надійність і широке визнання у галузі. Використання Git дозволяє ефективно управляти змінами в проекті, спрощуючи колаборацію між членами команди і забезпечуючи безпеку коду. Це критично важливо для підтримки стабільності і масштабування проекту, особливо в контексті багатокomпонентної розробки ігрового програмного забезпечення.

### 3.3 Розгляд ігрового циклу

У цьому розділі детально розглядаються ключові компоненти та процеси, що формують основу взаємодії гравця з ігровим світом. Цей аналіз включає опис динамічної послідовності дій і реакцій, які відбуваються від моменту запуску гри до завершення ігрового сеансу. Центральне місце в цьому аналізі займає ігровий цикл — повторюваний процес, який визначає загальний ритм і динаміку гри, впливаючи на ігровий досвід та залученість гравця.

Ігровий цикл додатку, що розробляється, зображений на малюнку 3.2.

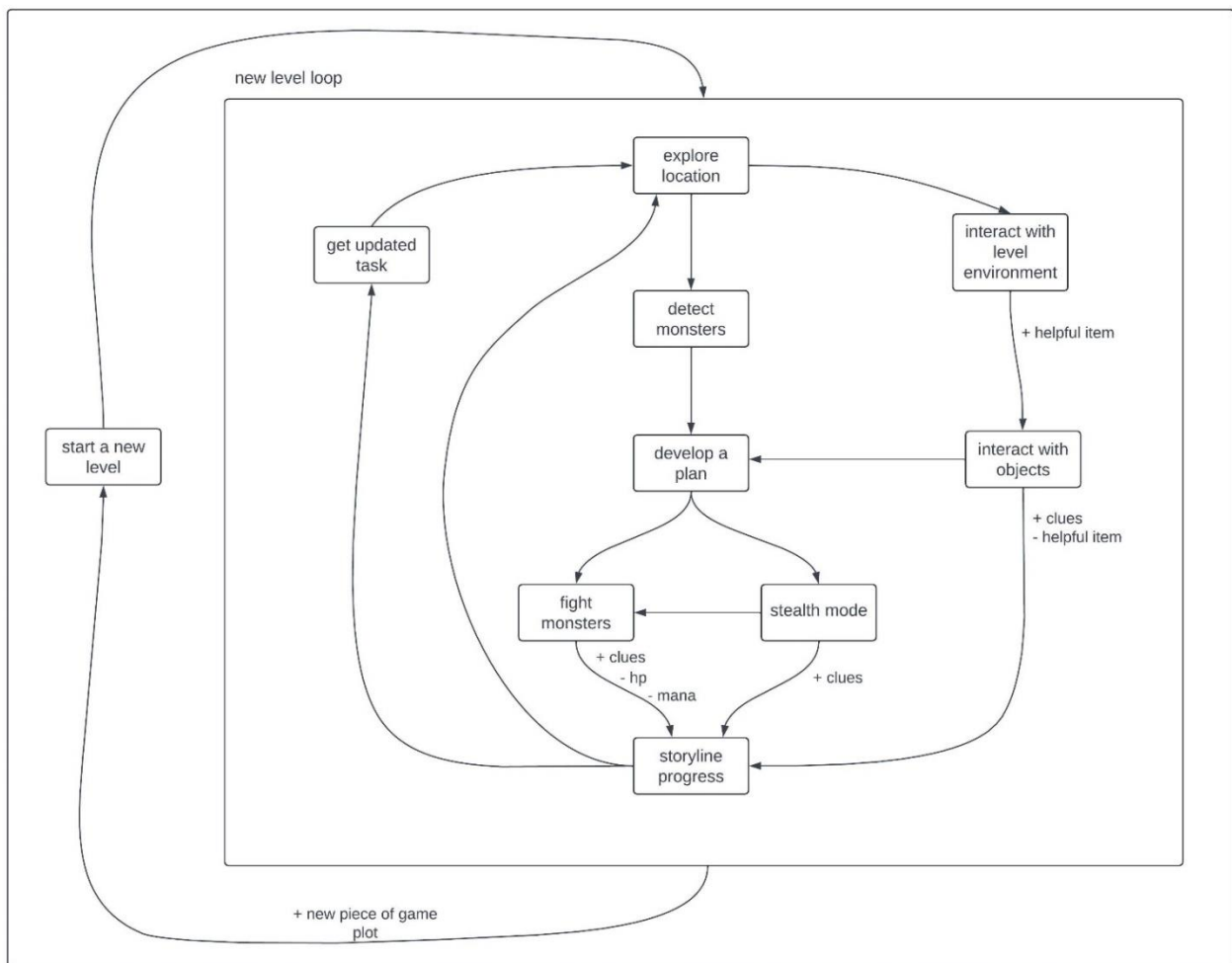


Рисунок 3.2 – Ігровий цикл програмного застосування

Розглянувши зображення ігрового циклу, можна виділити такі пункти:

1. Початок нового рівня:
  - Гравець розпочинає новий рівень, що є входом у новий ігровий цикл.
2. Отримання оновленого завдання:
  - На початку рівня гравець отримує оновлені завдання або місії.
3. Дослідження локації:
  - Гравець досліджує нове оточення, переміщається по картах та взаємодіє з різними елементами ігрового світу.
4. Виявлення монстрів:
  - Під час дослідження гравець може зустріти монстрів, з якими потрібно взаємодіяти або яких потрібно уникати.
5. Розробка плану:
  - гравець розробляє стратегію або план дій.
6. Взаємодія з об'єктами:
  - Гравець взаємодіє з предметами у локації, які можуть дати підказки або корисні предмети для подальшого просування в грі.
7. Бій з монстрами:
  - У разі вибору стратегії бою, гравець атакує монстрів, використовуючи доступні засоби і навички.
8. Стелс режим:
  - Як альтернативу бою, гравець може використовувати стелс для обходу монстрів, що також може принести підказки для завдань.
9. Прогрес сюжету:
  - Залежно від вибору дій і результатів взаємодій на рівні, сюжет гри просувається вперед, відкриваючи нові сегменти історії або завдання.

Розуміння представленого циклу є важливим для оптимізації ігрової механіки та створення глибоких, захоплюючих ігрових сценаріїв, які спонукають гравців повертатися до гри знову і знову.

### 3.4 Розгляд найцікавіших алгоритмів

Одним з найцікавіших алгоритмів ігрового застосунку можна виділити алгоритм штучного інтелекту гоблінів (див. рис. 3.3). Цей алгоритм було обрано, як найбільш повноцінний та оптимальний з частини проекту, що розробляється в якості дипломної роботи.

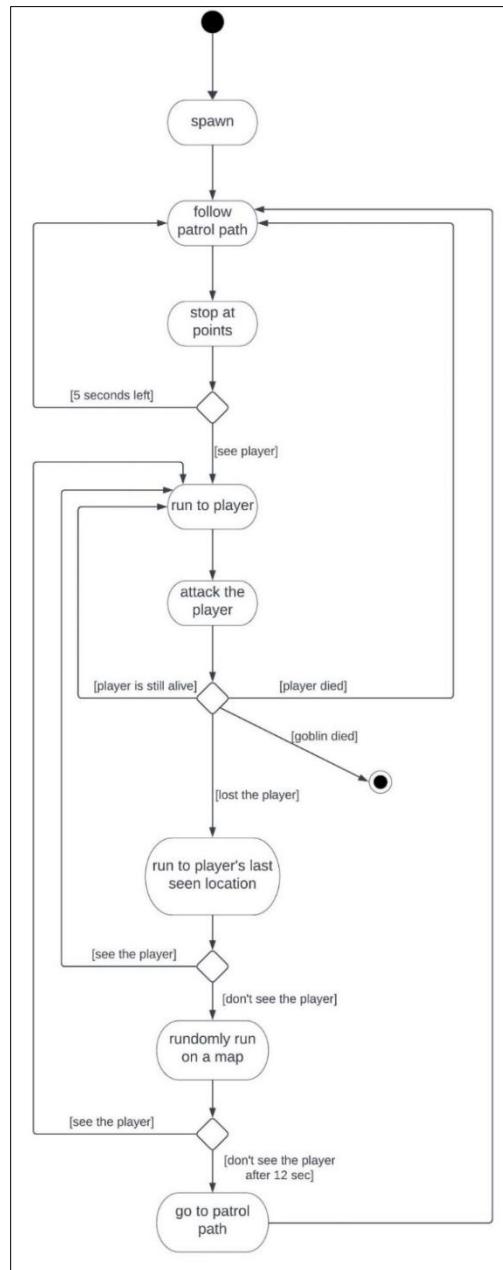


Рисунок 3.3 – Алгоритм штучного інтелекту гоблінів

Детально розглянувши наведений алгоритм, можна виділити такі важливі його етапи:

1. Spawn (З'явлення):
  - Гоблін з'являється в грі на певній локації.
2. Follow Patrol Path (Слідувати шляху патрулювання):
  - Після спавну гоблін починає патрулювати визначену територію, слідуючи встановленому маршруту.
3. Stop at Points (Зупинки у пунктах):
  - На патрульному маршруті є специфічні точки, де гоблін зупиняється на певний час (5 секунд).
4. Фаза виявлення:
  - Якщо гоблін побачить гравця під час зупинки або патрулювання, він переходить до наступної фази.
5. Run to Player (Бігти до гравця):
  - Гоблін біжить до гравця для атаки.
6. Attack the Player (Атакувати гравця):
  - При досягненні гравця, гоблін атакує.
7. Результат атаки:
  - Якщо гравець ще живий після атаки, гоблін продовжує атакувати.
  - Якщо гравець помер, гоблін повертається до патрулювання.
  - Якщо гоблін помирає, алгоритм завершується.
  - Якщо гоблін втрачає гравця з виду, він біжить до останнього місця, де гравець був помічений.
8. Фаза пошуку:
  - Гоблін діє відповідно до ситуації:
    - Якщо він бачить гравця знову, повертається до атаки.
    - Якщо не бачить гравця протягом 12 секунд, гоблін починає бігати випадково по карті.

## 9. Повернення до патрулювання:

- Якщо гоблін не знаходить гравця після випадкового бігу картою, він повертається до свого патрульного маршруту і продовжує патрулювання до наступної зустрічі з гравцем.

Розглянутий алгоритм штучного інтелекту ворогів, є найбільш цікавим з точки зору технічної складової цієї частини ігрового застосунку, що розробляється в рамках дипломного проекту.

## 3.5 Створення UI/UX

Створення прототипів UI/UX є важливим етапом у розробці ігрового застосунку, оскільки це дозволяє на ранніх стадіях проектування візуалізувати та оцінити інтерфейс і користувацький досвід.

Розглянемо розроблені прототипи геймплею та не тільки, до дипломного проекту.

Екран запуску гри (див. рис. 3.4), включає опцію для початку нової гри. Меню має візуально привабливий дизайн, що відображає естетику світу гри.



Рисунок 3.4 – Екран запуску гри

Екран загрузки рівня (див. рис. 3.5). Показує процес завантаження поточного рівня з анімованою індикацією прогресу та корисними порадами або цікавими фактами про світ гри, що можуть з'являтися під час завантаження.



Рисунок 3.5 – Екран загрузки рівня

Геймплей під час битви (див. рис. 3.6). Інтерфейс під час битв включає в себе елементи, такі як індикатор здоров'я (червоний), шкала мани (синя). Шкала здоров'я – після отримання поранень, кількість очків здоров'я знижується і це відображається на шкалі. При досягненні 0 – настає моментальна смерть. Поступово, лише поза битвою, очки здоров'я відновлюються. Шкала мани – потрібна для стрільби із магичної зброї. Різна зброя використовує різну кількість очків мани. Поступово відновлюється під час битви, але поза неї – набагато швидше. Також присутні міні-карта для орієнтації на місцевості та сповіщення про поточні завдання або цілі.

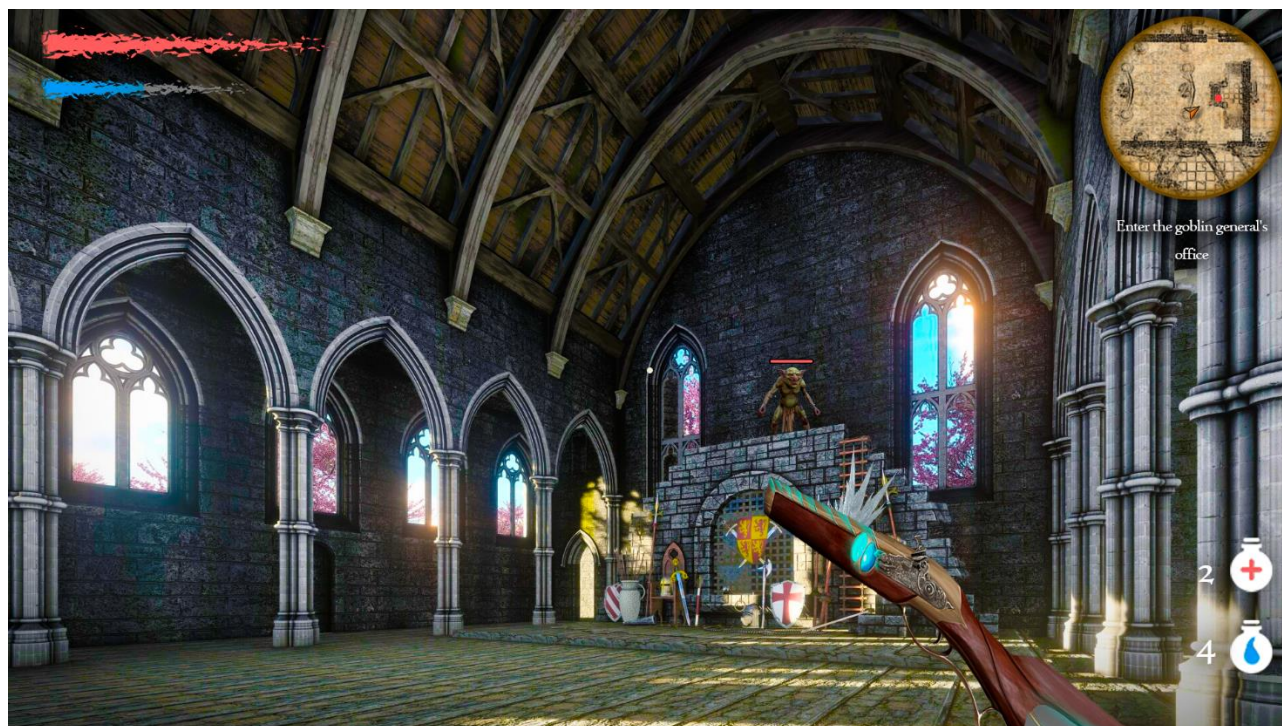


Рисунок 3.6 – Геймплей під час битви

Геймплей поза битвою (див. рис. 3.7). Інтерфейс поза битвами зосереджується на взаємодії зі світом і персонажами. Основні елементи цього інтерфейсу включають управління інвентарем, діалогове вікно та систему квестів. Управління інвентарем дозволяє гравцям переглядати, сортувати та використовувати предмети, які вони зібрали під час своєї подорожі. Діалогове вікно надає можливість спілкуватися з різними персонажами, дізнаватися нову інформацію та отримувати завдання. Система квестів дозволяє гравцям стежити за виконанням поточних завдань та дізнаватися про нові місії. Вся ця інформація про поточні завдання та важливі події відображається в спеціальних спливаючих вікнах, які інформують гравця про важливі моменти гри.

Крім цього, гравець може взаємодіяти з деякими ігровими об'єктами, такими як важелі, двері та інші механізми. Ці взаємодії можуть відкривати нові шляхи, запускати певні події або змінювати стан навколишнього середовища. У грі часто зустрічаються місця, до яких можна потрапити лише після взаємодії з іншими об'єктами, створення шляху або усунення перешкод.

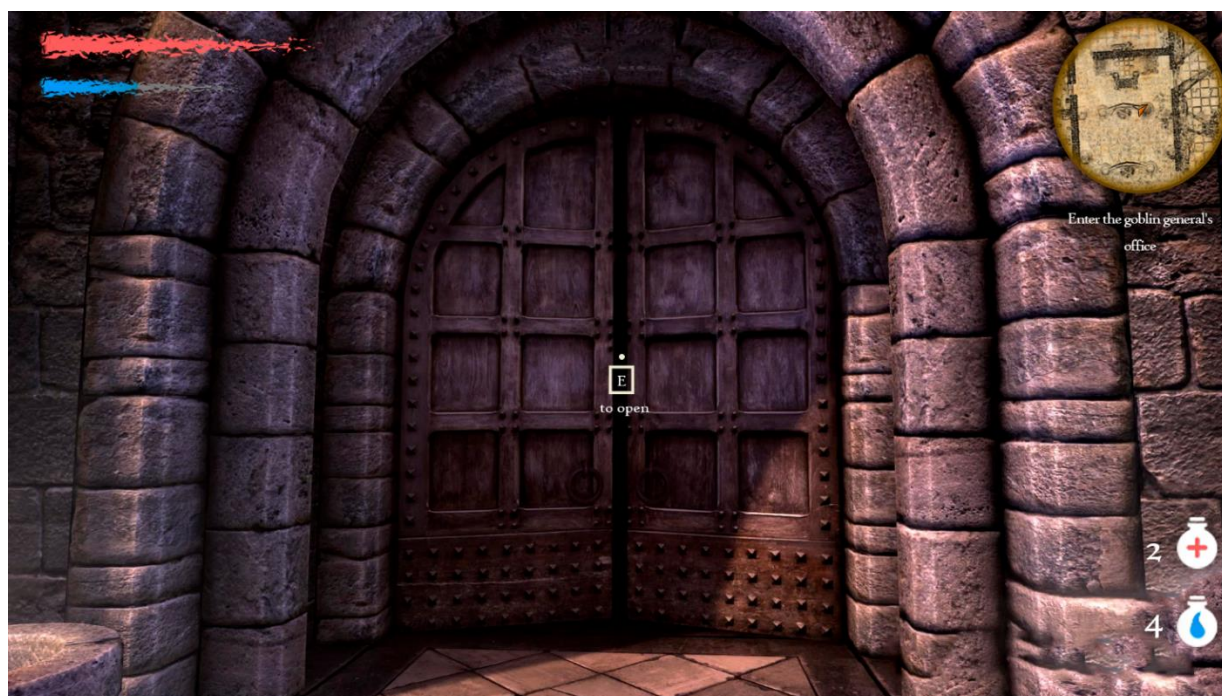


Рисунок 3.7 – Геймплей поза битвою

Інтерфейс інвентаря (див. рис. 3.8). Включає в себе управління предметами, які гравець зібрав або отримав. Також дозволяє вибирати зброю, зілля, та інші корисні предмети, які можуть бути використані в грі.

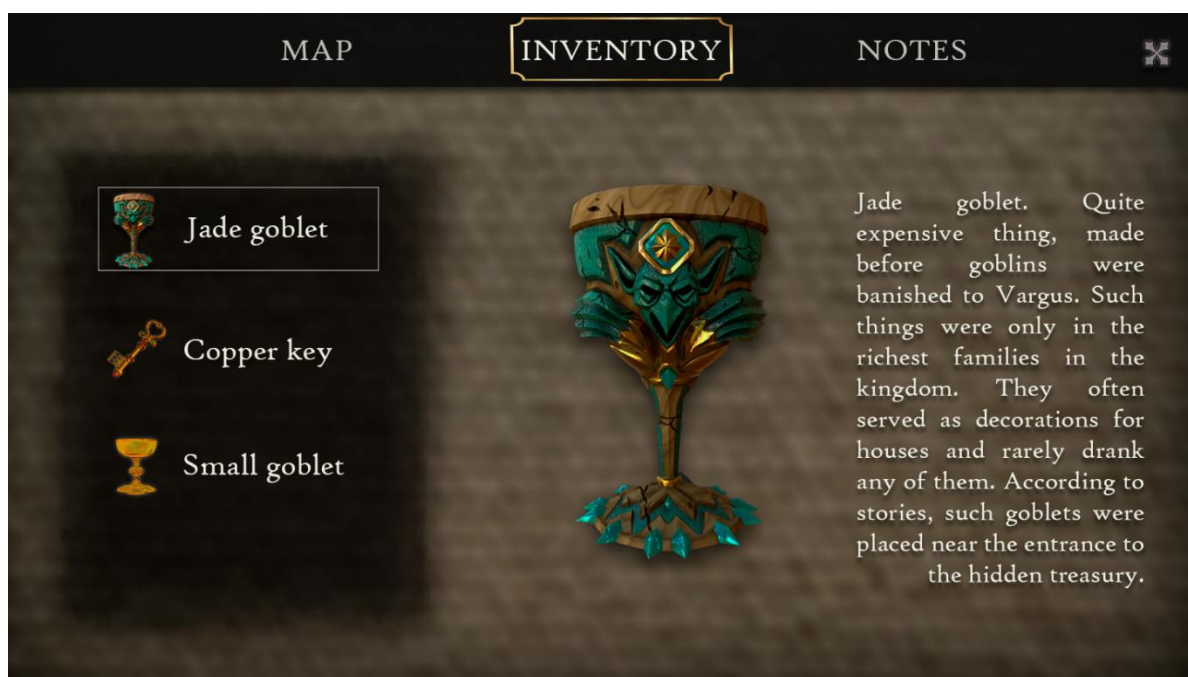


Рисунок 3.8 – Інтерфейс інвентаря

Інтерфейс з мапою (див. рис. 3.9). Повноекранна карта світу, що допомагає гравцю орієнтуватися в грі, показуючи важливі локації, поточне місце розташування героя та об'єкти для дослідження.

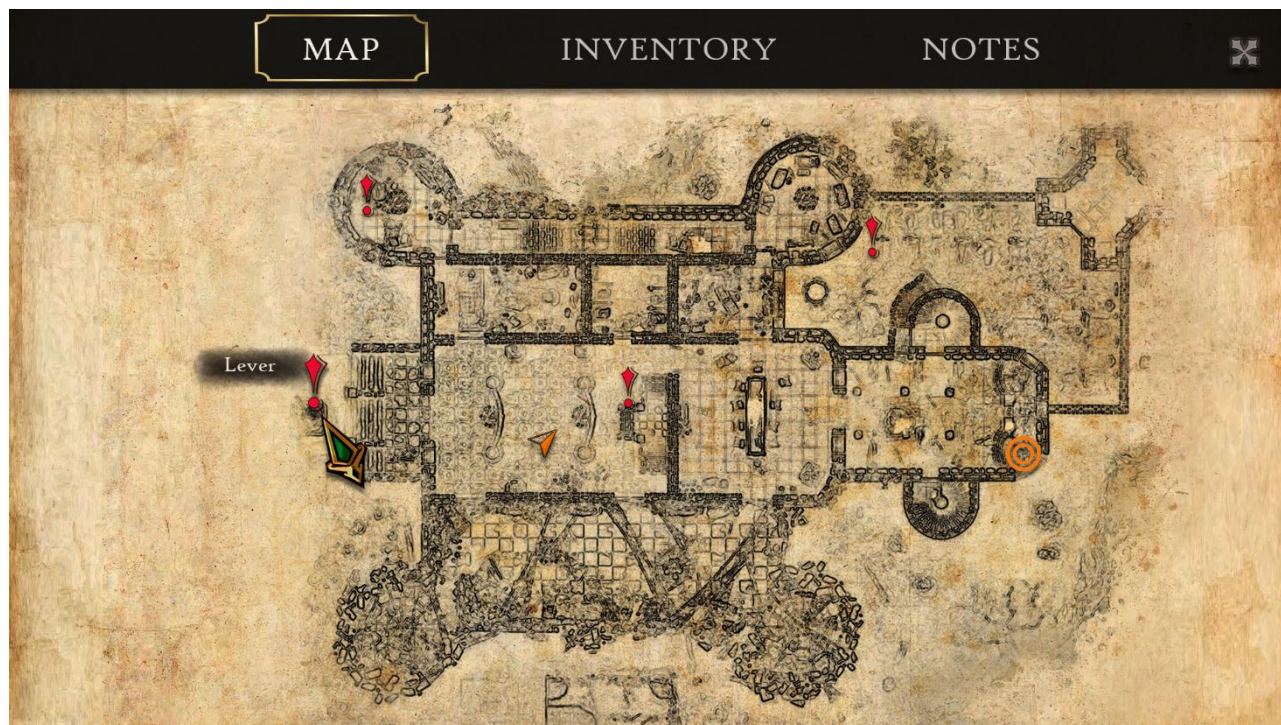


Рисунок 3.9 – Інтерфейс з мапою

Інтерфейс із записками (див. рис. 3.10). Секція для зберігання знайдених документів, листів та записок, які гравець збирає під час гри. Вони можуть містити важливу інформацію для розгадування сюжетних загадок або додаткові історії, які розширюють світогляд гри. Наприклад, у грі часто зустрічаються місця до яких можна потрапити лише якщо провзаємодіяти з іншими об'єктами, або створити собі шлях туди, або прибрати перешкоди. Про це можна дізнатися із записок, що були знайдені на цій локації до цього. Ця система також стимулює гравців до більш детального дослідження ігрових локацій, адже кожен знайдений документ може відкрити частину великої історії або надати важливу інформацію про персонажів та події, які впливають на розвиток сюжету. Взаємодія з об'єктами та оточенням в пошуках цих документів додає глибину геймплею, змушуючи гравців мислити креативно та використовувати

всі доступні ресурси для досягнення своїх цілей.



Рисунок 3.10 – Інтерфейс із записками

Інтерфейси ретельно розроблені, щоб забезпечити гравцям інтуїтивно зрозуміле і зручне управління, а також глибоке занурення у гру. Вони охоплюють всі аспекти геймплею, від навігації та управління боєм до взаємодії з ігровим світом і персонажами. Завдяки цьому гравці можуть безперешкодно переходити між різними видами діяльності, отримуючи повноцінний досвід від гри. Особлива увага була приділена деталям інтерфейсу, які підсилюють загальну атмосферу світу ігрового застосунку, роблячи його більш реалістичним та яскравим на емоції.

## 4 ОПИС ПРИЙНЯТИХ ПРОГРАМНИХ РІШЕНЬ

### 4.1 Розробка дизайну рівнів

Розробка дизайну рівнів є однією з ключових складових у процесі створення ігрового застосунку, оскільки вона безпосередньо впливає на геймплей і загальне сприйняття гри користувачами. При плануванні рівнів важливо використовувати перевірені принципи дизайну, які допомагають створити збалансоване та цікаве ігрове середовище. Один з таких фундаментальних принципів — це застосування золотого перетину, який відіграє ключову роль у створенні естетично привабливих і гармонійних рівнів[4].

Золотий перетин — це математичний принцип, що стверджує, коли пропорції діляться за певним правилом, результатом є приємна для ока гармонія. У контексті дизайну рівнів, це може означати розміщення ключових елементів, таких як важливі об'єкти, зони конфліктів або шляхи проходження в пропорціях, що відповідають цьому правилу (див. рис. 4.1). Це допомагає створити більш організовані та цікаві рівні, які натурально привертають гравців до важливих зон і водночас зберігають їхню увагу і інтерес.

При створенні зарисовок і концептуальних макетів рівнів для дипломного проекту було важливо інтегрувати принцип золотого перетину для досягнення оптимального балансу між естетичною привабливістю та функціональністю. Це включало детальне планування розташування важливих ігрових елементів, оптимізацію шляхів евакуації та конфліктних зон, щоб забезпечити, що кожен рівень є інтуїтивно зрозумілим і захоплюючим для гравця.

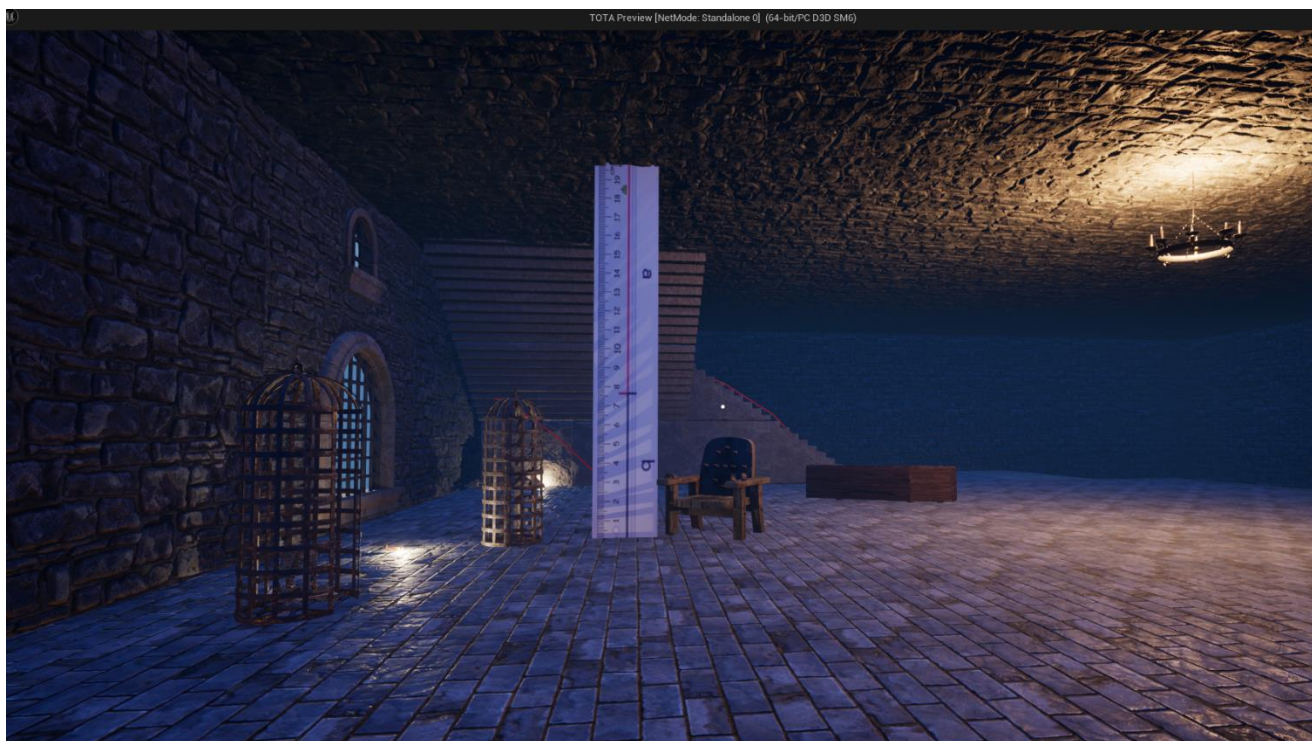


Рисунок 4.1 – Втілення принципу золотого перетину

Під час розробки ігрового застосунку, для втілення принципу золотого перетину, було додано об'єкт лінійки на рівень [5]. На малюнку можна побачити приклад використання цього принципу. Видно, що лінійка розподілена на дві частини згідно принципу золотого перетину. Вона використовувалась для візуального порівняння певних важливих об'єктів у грі. На конкретному малюнку, що вказаний в якості прикладу, показано порівняння клітки та стільця, відносно простору сцени. Бачимо, що клітка знаходиться чітко на межі пропорції, а стілець трішки нижче.

Для правильного та лаконічного дизайну рівнів, необхідним є створення попередніх скетчів. На розробленому рівні планувалися 2 локації: головна будівля (див. рис. 4.2) гоблінів та печера (див. рис. 4.3).



Рисунок 4.2 – Головна будівля

Голубим кольором наведено контури печери.

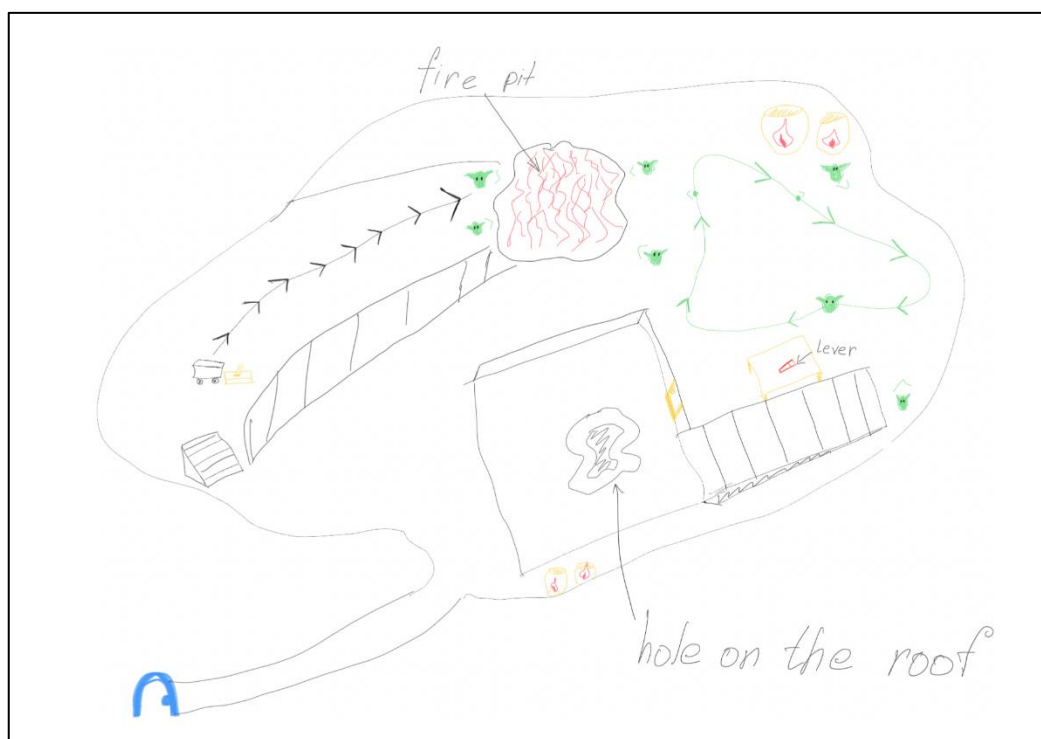


Рисунок 4.3 – Схема печери

Далі будуть наведені замальовки кожного з поверхів головної будівлі, починаючи з першого (див. рис. 4.4 – 4.7). Вороги – гобліни схематично намальовані як зелені голови з вухами.

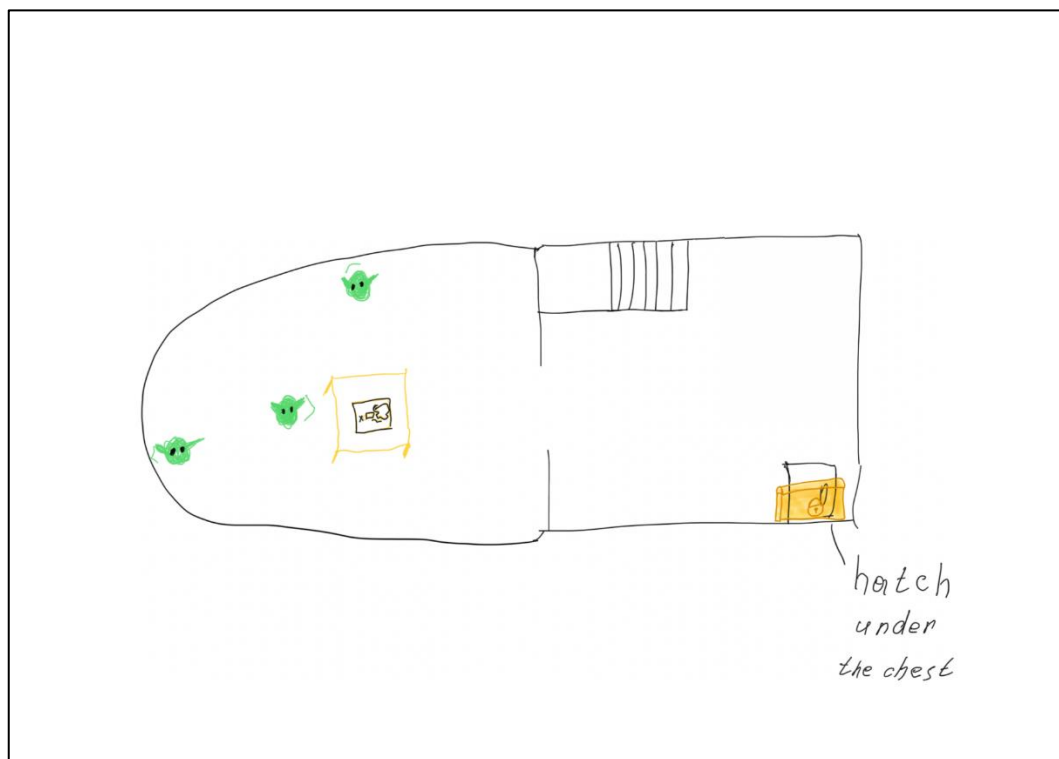


Рисунок 4.4 – Перший поверх головної будівлі

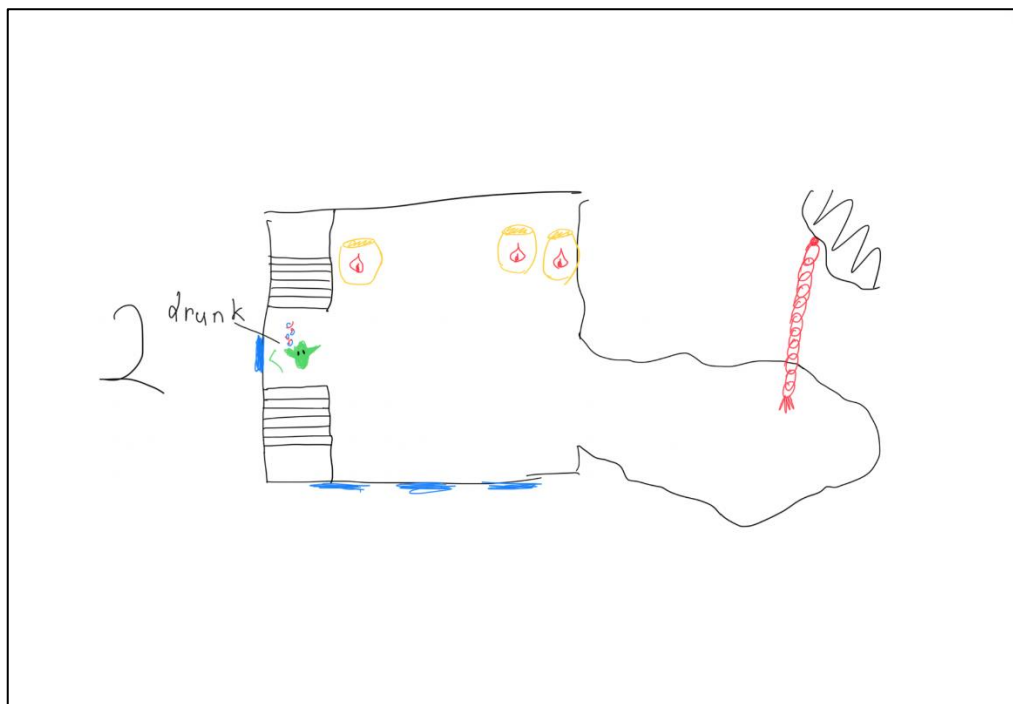


Рисунок 4.5 – Другий поверх головної будівлі



Рисунок 4.6 – Третій поверх головної будівлі

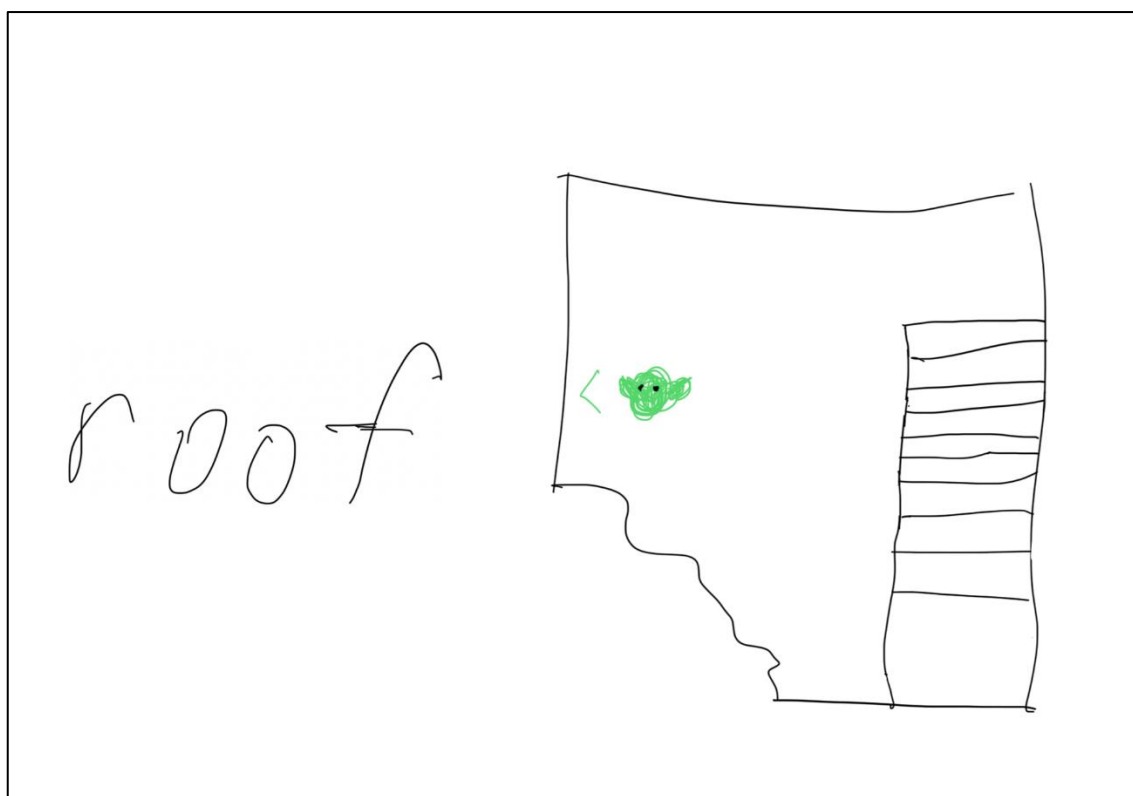


Рисунок 4.7 – Четвертий поверх головної будівлі

Попередня розробка скетчів допомогла правильно сконструювати сходи, розмістити вікна. Також, одразу було продумано куди я в якій кількості будуть розміщені супротивники. Вікна розроблені таким чином, щоб гравець мав змогу вистрибнути з певних отворів на нижні поверхи.

Головний ландшафт рівня було зроблено за допомогою технології Landscape (див. рис. 4.8).

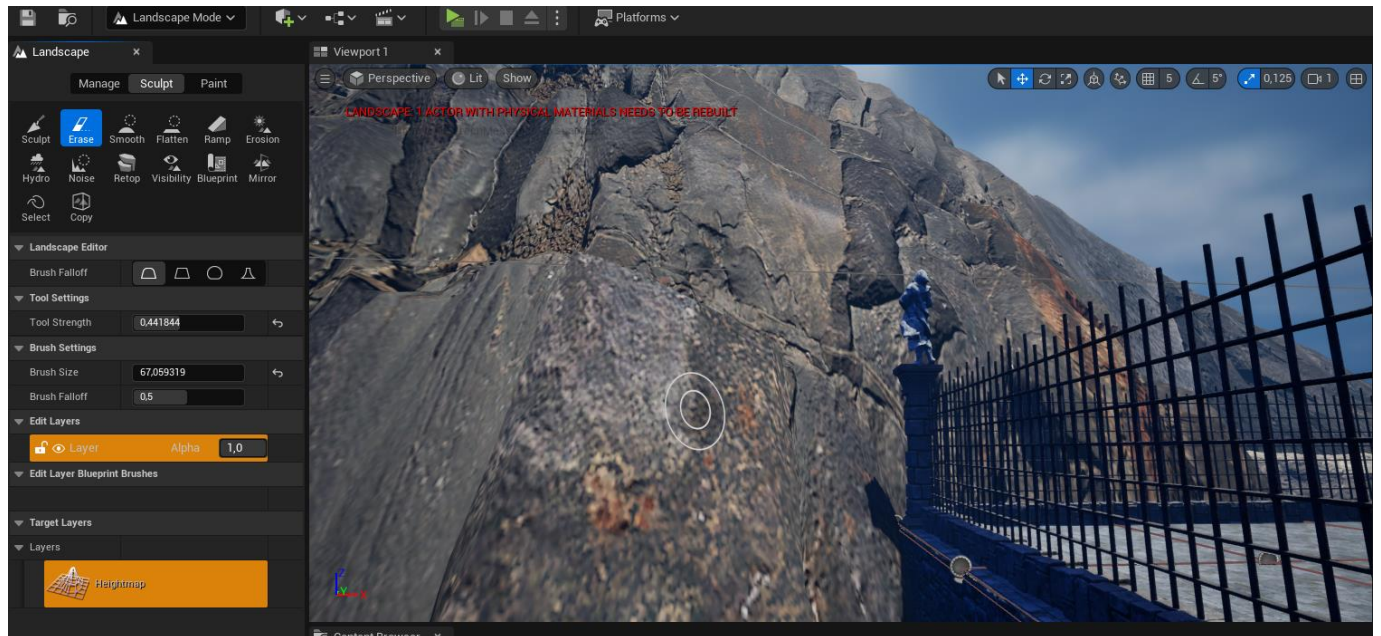


Рисунок 4.8 – Використання технології Landscape

В якості фонового ландшафту рівня було обрано гори. Такий вибір було зроблено спираючись на сюжетне місцезнаходження локації, що розробляється.

## 4.2 Розробка штучного інтелекту ворогів

У розробленому ігровому застосунку гравцеві дається вибір проходження рівня по стелсу або звичайним шляхом через сутички із ворогами. Розглянемо оптимальні

умови виявлення гравця ШІ ворогів. У противника є спеціальні колізії: сферична (відповідає слуху противника) та конічна (відповідає зору) (див. рис. 4.9) [6]. Якщо гравець заходить в них, або робить шумні/підозрілі дії – у противника починає заповнюватись шкала тривоги. Якщо джерело тривоги не щезає – через 3 секунди гравця буде помічено. Також є додаткові колізії – з меншим радіусом для сфери та кутом для конуса. В цих зонах гравця помічають одразу. У режимі стелсу, колізії тривоги (жовті на малюнку) не працюють, якщо не шуміти. Колізії моментального помічання (червоні на малюнку) працюють без змін.

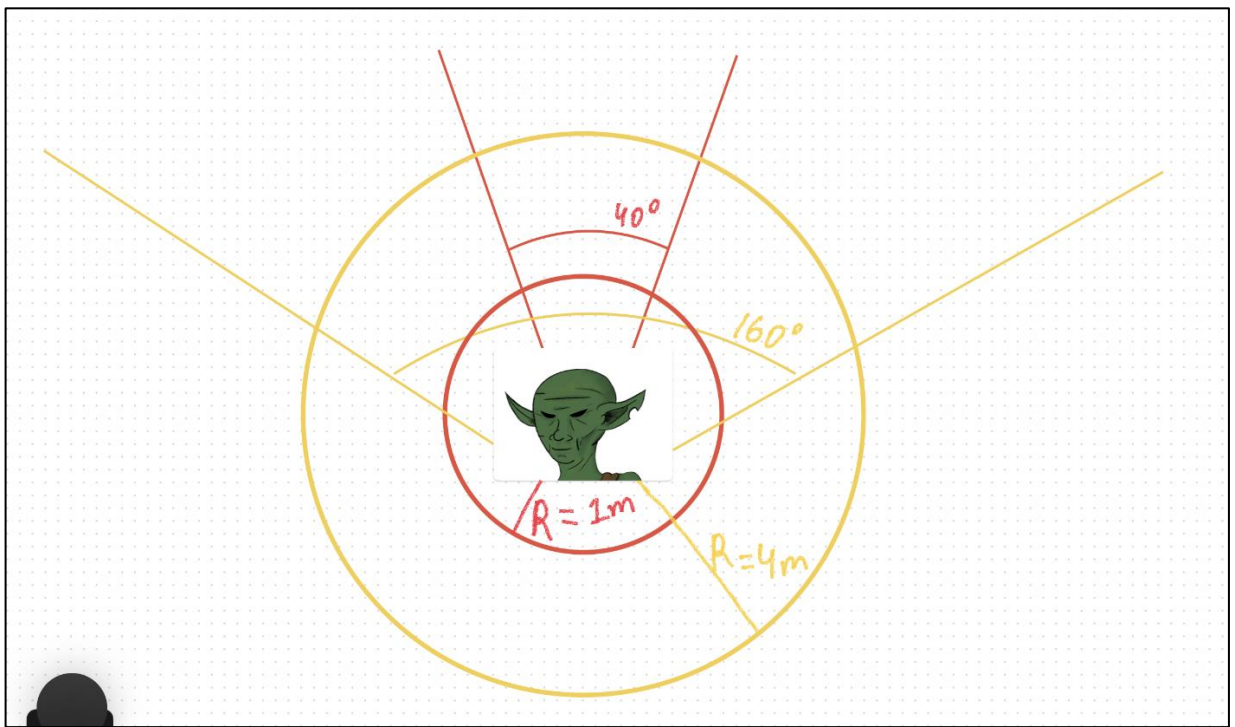


Рисунок 4.9 – Умови виявлення гравця

В обраній середі розробці ці параметри змінюються за допомогою налаштувань компоненту «AIPerception» (див. рис. 4.10). Мануально задаються параметри «Sight Radius» - максимальна дистанція для утримання гравця на виду, «Lose Sight Radius» - дистанція, на якій противник вже остаточно не бачить гравця та «PeripheralVisionHalfAngleDegrees» - кут огляду гобліна [7].

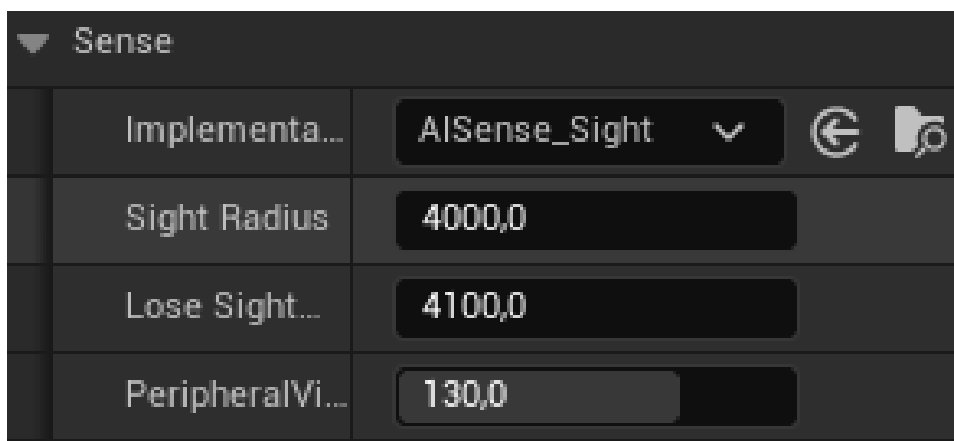


Рисунок 4.10 – Параметри налаштування сприйняття ворога

Сама поведінка штучного інтелекту ворогів розроблена за допомогою дерева поведінки (див. рис. 4.11) [8], що складається із селекторів, декораторів та тасків.

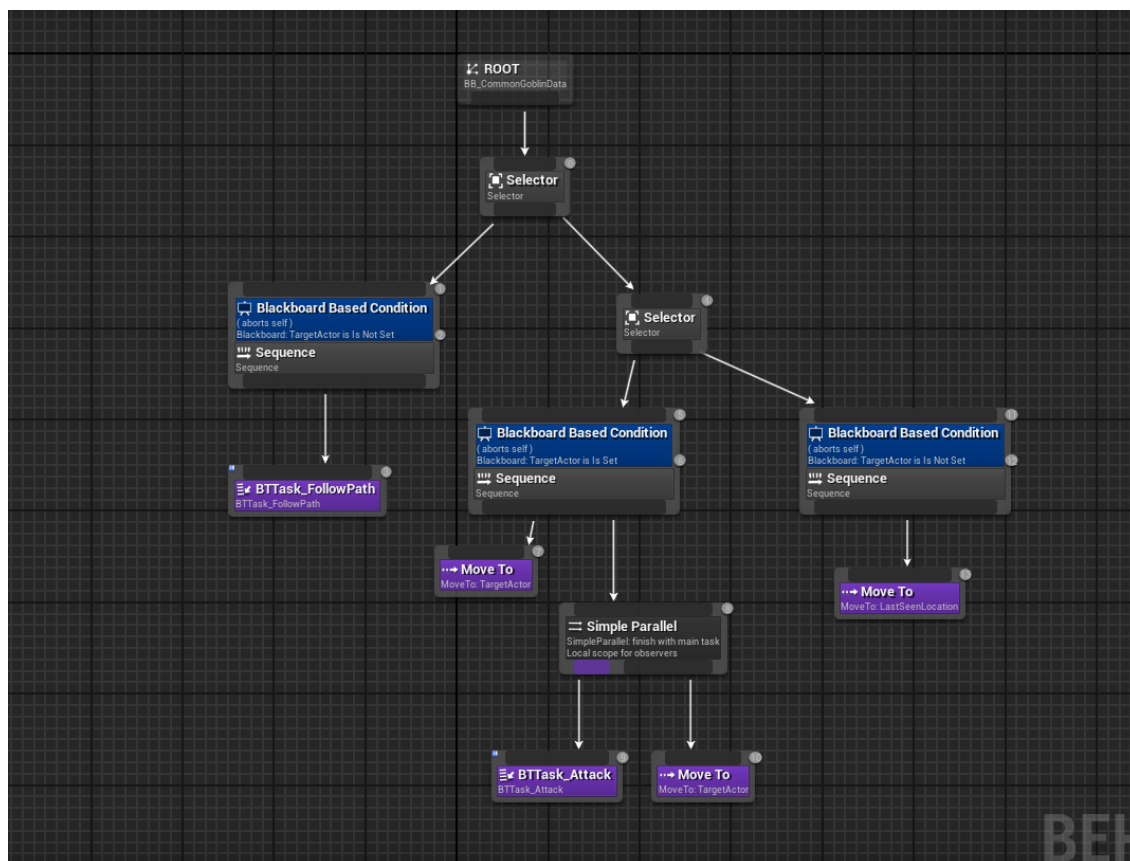


Рисунок 4.11 – Дерево поведінки ІІІ ворогів

Розглянемо задачу патрулювання «BTTask\_FollowPath» [9]. Маршрут для

патрулювання було зроблено за допомогою кривої (Spline), яку можна змінювати прямо у редакторі Unreal Engine 5. Зупинки були реалізовані через певний масив точок на кривій, які мають свою позицію (Time), та змінну, що відповідає за час, проведений ворогом на цій точці. Ще одним показником на зупинках є напрям, у якому буде стояти та дивитися ворог [10]. Він був реалізований за допомогою динамічного компоненту стрілки, яка вказує на напрямок, що редагується разом з іншими даними масиву (див. рис. 4.12) [11].



Рисунок 4.12 – Візуалізація маршруту патрулювання

Точки зупинок на маршруті коригуються з редактору Unreal Engine 5. Можна завдати час зупинки, локацію на кривій патрулювання та напрямок, в якому буде стояти та дивитися ворог на цій точці (див. рис. 4.13).

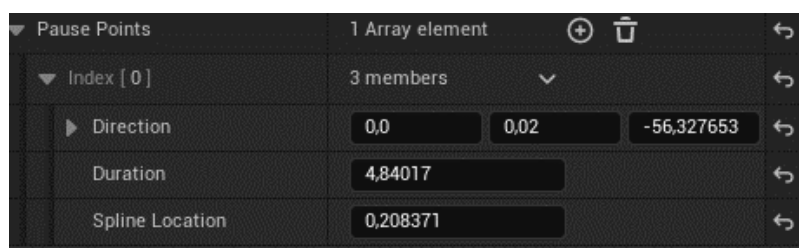


Рисунок 4.12 – Параметри налаштування зупинок на маршруті

Програмну складову цієї задачі можна побачити на малюнках 4.13 – 4.14. Так як малюнок дуже довгий, було прийнято рішення розрізати його навпіл.

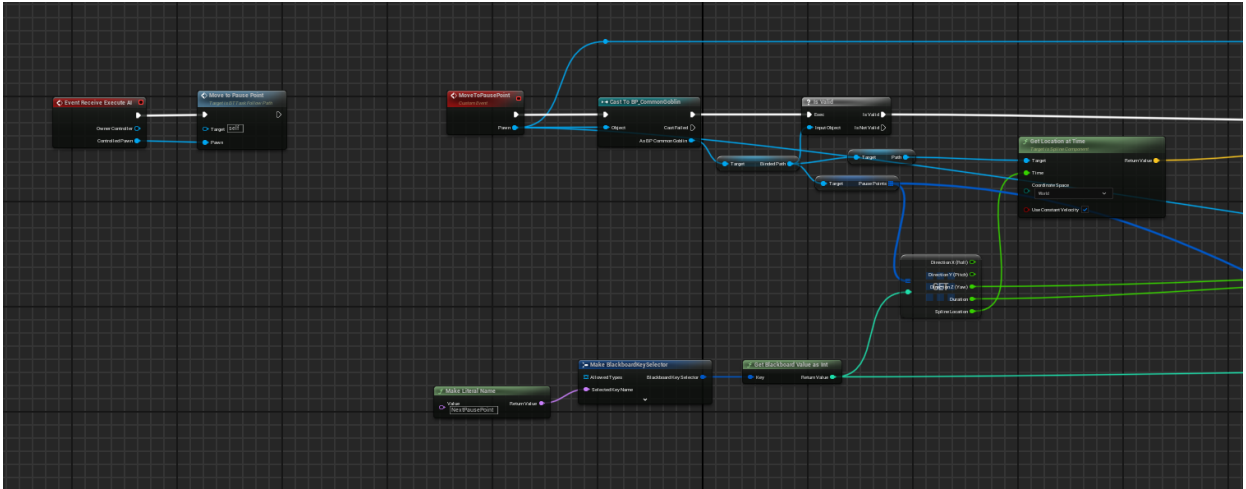


Рисунок 4.13 – Перша частина «BTTask\_FollowPath»

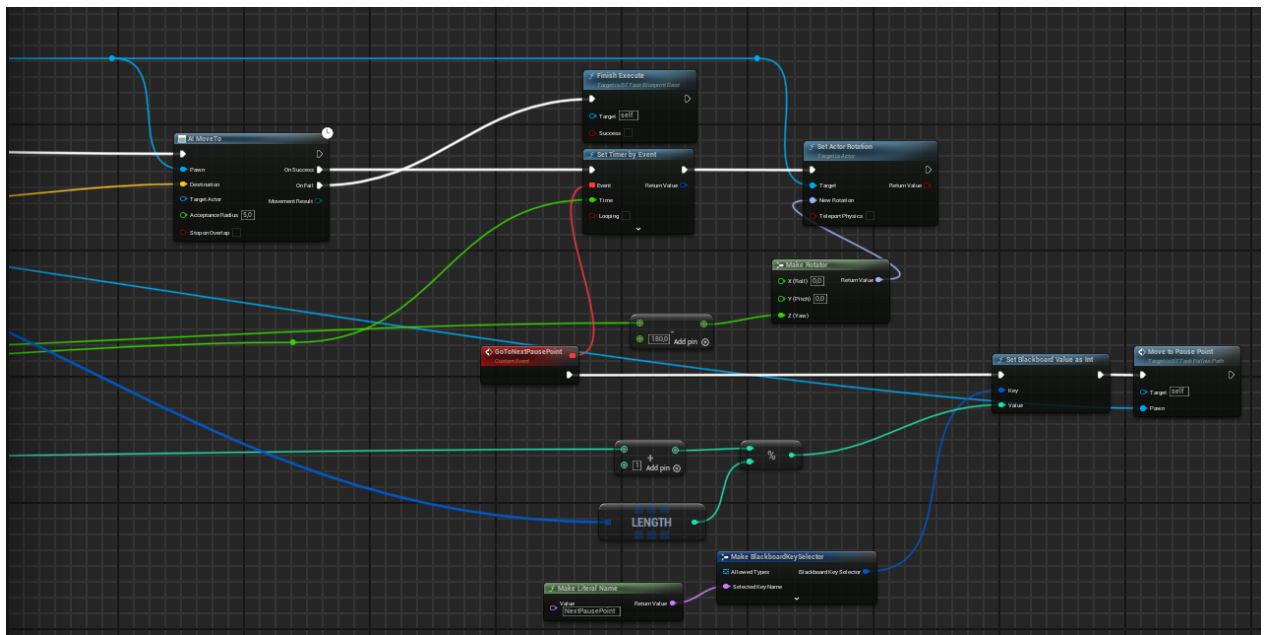


Рисунок 4.14 – Друга частина «BTTask\_FollowPath»

На малюнках вище було представлено «BTTask\_FollowPath», опис якого знаходиться у цьому розділі ще вище. Можна побачити, що код дуже довгий, тому для зручності були додані коментарі.



- переходи за допомогою клавіатури поміж елементами інтерфейсу;
- вікна з описом предметів в інвентарі;
- перевірка меню нотатків та записок;
- перевірка інтерфейсу налаштувань гри;
- перевірка інтерфейсу закінчення гри (сметрі).

## 2) Інтерфейс гри:

- перевірка відображення значень стану гравця, а саме: здоров'я, кількість мани.

## 3) Ігровий процес:

- перевірка механік стрілянини, а саме: завдання шкоди, фізичне відображення ефектів попадання стихійних снарядів;
- перевірка коректності завантаження текстур;
- перевірка поведінки мешів персонажів
- перевірка механіки поповнення здоров'я через їжу;
- проходження завдань;
- перевірка механіки керування предметами;
- перевірка коректності розподілення маси в предметах;
- перевірка нанесення урону гравцю самим гравцем (падіння з вікна);
- перевірка режиму стелсу;
- перевірка зони сприйняття ворогів (зір, слух);
- перевірка колізій на локації.

## 4) Штучний інтелект ворогів:

- перевірка патрулювання ворогів;
- перевірка системи пошуку гравця ворогом;
- перевірка коректності поведінки ворогів в різних випадках.

Враховуючи вищенаведені функції, що будуть тестуватися, розглянемо перелік

тестових випадків. Таблиці 5.1 – містять перелік тестових випадків.

Таблиця 5.1 – Тест у випадку з патрулем у гоблінів

Тест № 1	
Назва тесту:	Гобліни порушують шлях патруля
Опис тесту:	Гобліни неточно йдуть за шляхом патруля
Компонент системи:	Навігаційна система II/Поведінка ворогів.
Пріоритет:	P3 Середній
Критичність:	S1 Значний.
Кроки відтворення:	1. Привернути увагу гобліна. 2. Відбігти далеко. 3. Припинити переслідування. 4. Дочекатись поки гоблін повернеться на шлях патруля.
Очікуваний результат:	Гоблін точно йде за шляхом патруля.
Фактичний результат:	Гоблін йде за більшим колом, ніж повинен йти.

Таблиця 5.2 – Тест у випадку зі здоров'ям ворога

Тест № 1	
Назва тесту:	Неочікуване збільшення здоров'я гобліна
Опис тесту:	Здоров'я гобліна іноді несподівано збільшується під час бою без використання додаткових методів.
Компонент системи:	II ворогів/Система здоров'я.
Пріоритет:	P3 Середній
Критичність:	S2 Значний.
Кроки відтворення:	1. Підпалити гобліна. 2. Дочекатись поки рівень його здоров'я буде майже на нулі. 3. Нанести одним ударом більше шкоди ніж максимальне здоров'я, що він має
Очікуваний результат:	Здоров'я гобліна зменшується адекватно нанесеним ударам.
Фактичний результат:	Гоблін продовжує бій з несподівано поновленим здоров'ям.

Інші тести, були впроваджені по ходу написання кодової частини ігрового застосунку. У таблицях наведені лише найбільш актуальні для частини дипломного проекту, що розробляється.

## ВИСНОВКИ

На основі проведеного аналізу та розробки ігрового застосунку в рамках дипломного проекту можна висловити декілька важливих пунктів, що відображають успіхи та виклики проекту. По-перше, імплементація штучного інтелекту ворогів була здійснена успішно, що сприяло створенню динамічного ігрового процесу. Системи патрулювання, пошуку гравця та адаптивної поведінки в різних ігрових ситуаціях були реалізовані з високою точністю, забезпечуючи гравцям складні та захоплюючі виклики, які підтримували їхній інтерес та залученість у процес гри.

Дизайн рівнів був виконаний з великою увагою до деталей, що дозволило створити глибоко інтерактивне та візуально привабливе ігрове середовище. Кожен рівень був ретельно спланований, з чітко визначеними цілями та ключовими зонами дії, що сприяло логічному і плавному ігровому потоку. Висока інтерактивність рівнів забезпечувала гравцям численні можливості для дослідження та взаємодії з ігровим світом.

По-третє, прототипування інтерфейсу користувача дозволило ефективно тестувати та вдосконалювати інтерактивні елементи управління, забезпечуючи, що інтерфейс є інтуїтивно зрозумілим та зручним у використанні. Ретельне тестування інтерфейсів на різних етапах розробки допомогло ідентифікувати та вирішити проблеми з навігацією та управлінням, забезпечуючи гравцям позитивний досвід взаємодії з програмним продуктом.

Для розробки технічної складової проекту була обрана система візуального програмування Blueprint, ігровий рушій – Unreal Engine 5. Система контролю версій Git.

На завершення, технічна реалізація проекту продемонструвала високу ефективність використання сучасних програмних технологій та методик розробки. Система була оптимізована для забезпечення стабільності та високої продуктивності, що є вирішальним для комерційного успіху ігрових застосунків. Використання

передових рішень у сфері програмування та графічного дизайну дозволило створити продукт, що не тільки відповідає сучасним вимогам до якості та функціональності, але й пропонує користувачам унікальні та захоплюючі ігрові можливості.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ


1. Use Case Diagrams | Unified Modeling Language (UML) URL: <https://www.geeksforgEEKS.org/use-case-diagram/> (дата звернення: 10.05.2024).
2. Anatomy of a Blueprint URL: <https://docs.unrealengine.com/4.26/en-US/ProgrammingAndScripting/Blueprints/Anatomy/> (дата звернення: 03.05.2024).
3. Blueprints Quick Start Guide URL: <https://docs.unrealengine.com/4.26/en-US/ProgrammingAndScripting/Blueprints/QuickStart/> (дата звернення: 11.05.2024).
4. Level Design Basics. Part 2: Level Creation. URL: <https://gamedev.dou.ua/blogs/level-design-basics-part-2/?hl=en> (дата звернення: 17.04.2024).
5. How to create, save, and open Level assets in UE4. URL: [https://dev.epicgames.com/documentation/en-us/unreal-engine/working-with-levels-in-unreal-engine?application\\_version=5.0](https://dev.epicgames.com/documentation/en-us/unreal-engine/working-with-levels-in-unreal-engine?application_version=5.0) (дата звернення: 25.03.2024).
6. A tutorial for adding Components to Actors. URL: [https://dev.epicgames.com/documentation/en-us/unreal-engine/adding-components-to-an-actor-in-unreal-engine?application\\_version=5.0](https://dev.epicgames.com/documentation/en-us/unreal-engine/adding-components-to-an-actor-in-unreal-engine?application_version=5.0) (дата звернення: 15.04.2024).
7. Blueprint Overview. URL: <https://docs.unrealengine.com/4.26/en-US/ProgrammingAndScripting/Blueprints/Overview/> (дата звернення: 13.05.2024).
8. Behavior Trees. URL: [https://dev.epicgames.com/documentation/en-us/unreal-engine/behavior-trees-in-unreal-engine?application\\_version=5.0](https://dev.epicgames.com/documentation/en-us/unreal-engine/behavior-trees-in-unreal-engine?application_version=5.0) (дата звернення: 25.03.2024).
9. Кісельгова М.Є., Методи та алгоритми балансування штучного інтелекту для ігор жанру FPS, 28 Міжнародний молодіжний форум «РАДІОЕЛЕКТРОНІКА ТА МОЛОДЬ В ХХІ ст.», м. Харків. 2024. С. 342-344.
10. Радіоелектроніка та молодь у ХХІ столітті. Т. 6 : Конференція "Інформаційні інтелектуальні системи" : матеріали 27-го Міжнар. молодіж. форуму, 10–11 трав. 2023 р. / М-во освіти і науки України, Харків. нац. унт радіоелектроніки.

Харків, 2023. 275 с.

11. O. Mazurova, O. Samantsov, O. Topchii and M. Shirokopetleva, A Study of Optimization Models for Creation of Artificial Intelligence for the Computer Game in the Tower Defense Genre, 2020 IEEE International Conference on Problems of Infocommunications. Science and Technology (PIC S&T), 2020, pp. 491-496, doi: 10.1109/PICST51311.2020.9468057.

## ДОДАТОК А

### Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ



Ім'я користувача:  
**Олійник Олена Володимирівна каф. ПІ**

Дата перевірки:  
25.05.2024 16:10:34 EEST

Дата звіту:  
25.05.2024 16:13:32 EEST

ID перевірки:  
1016282213

Тип перевірки:  
Doc vs Library

ID користувача:  
100012353

---

Назва документа: 2024\_Б\_ПІ\_ПЗПІ\_20\_5\_Кісельгова\_М\_Є\_скорочений

Кількість сторінок: 50 Кількість слів: 6759 Кількість символів: 53527 Розмір файлу: 3.82 MB ID файлу: 1016075191

---

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

**4.6%**  
**Схожість**

Найбільша схожість: 1.6% з джерелом з Бібліотеки (ID файлу: 1008270567)

Пошук збігів з Інтернетом не проводився

4.6% Джерела з Бібліотеки 214 ..... Сторінка 52

**0% Цитат**

Вилучення цитат вимкнено

Вилучення списку бібліографічних посилань вимкнено

**0%**  
**Вилучень**

Немає вилучених джерел

**Модифікації**

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 2

Підозріле форматування 8 сторінок

## ДОДАТОК Б

### Слайди презентації

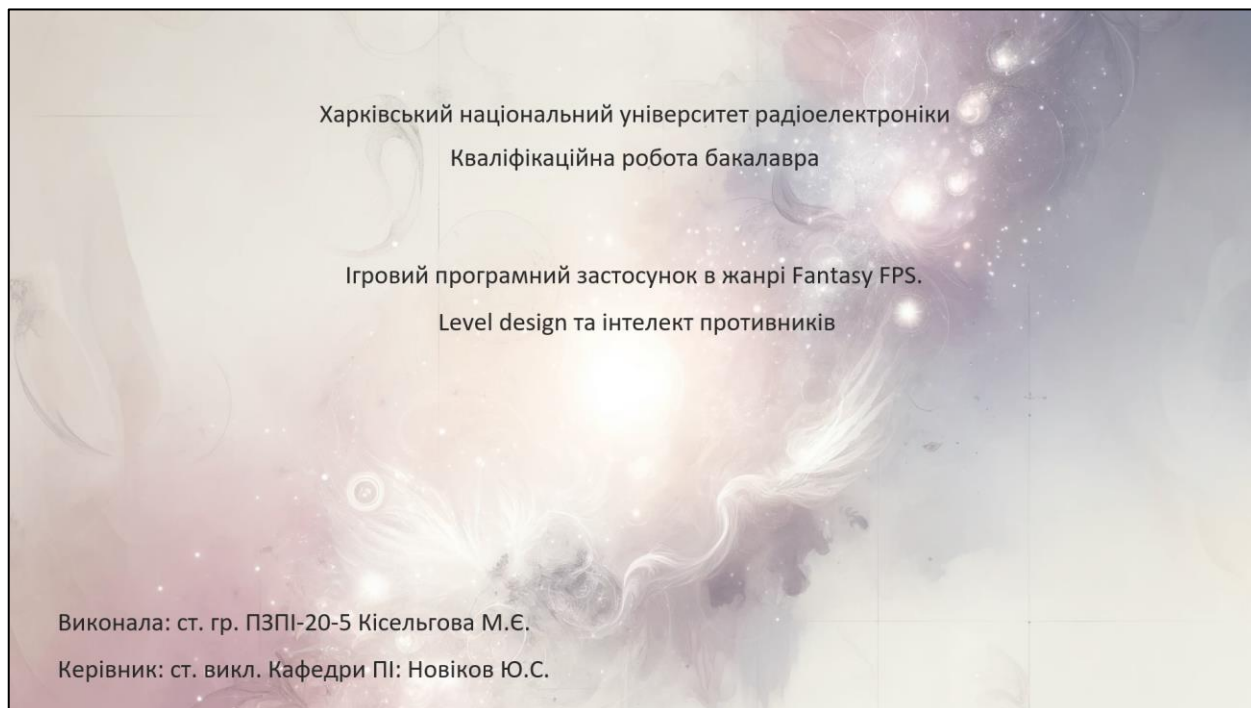


Рисунок Б.1 – Слайд 1

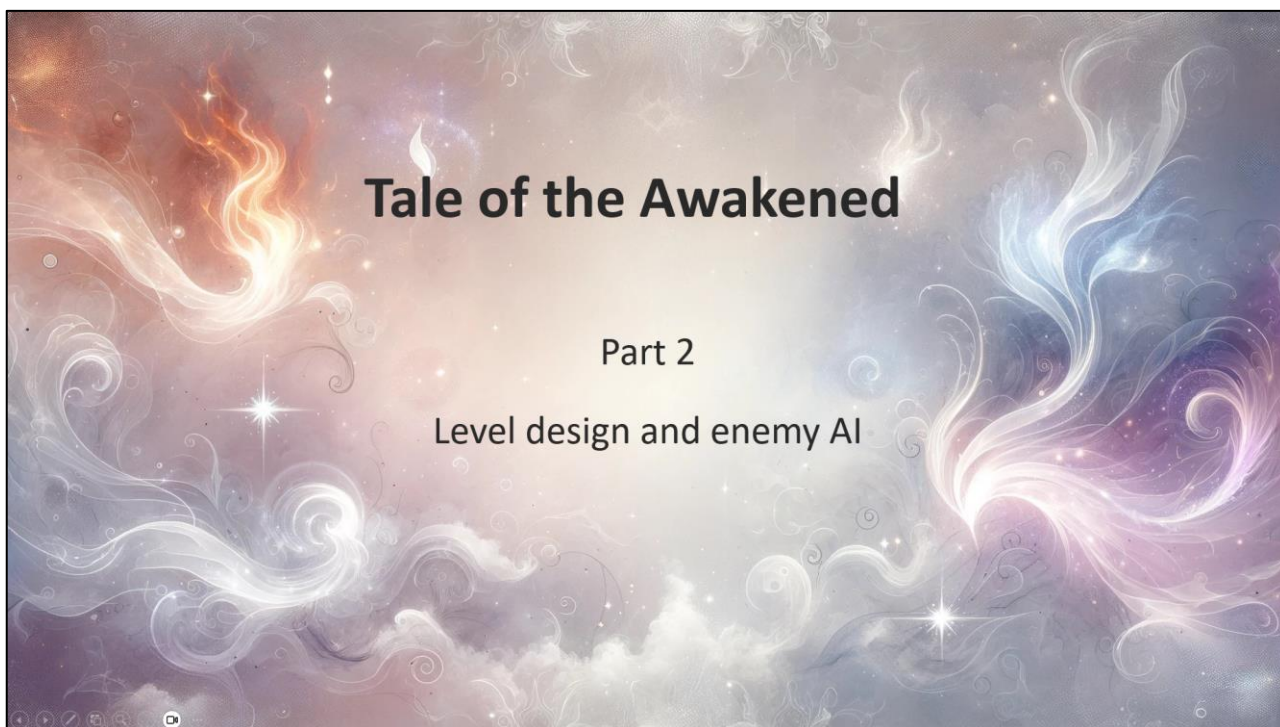


Рисунок Б.2 – Слайд 2

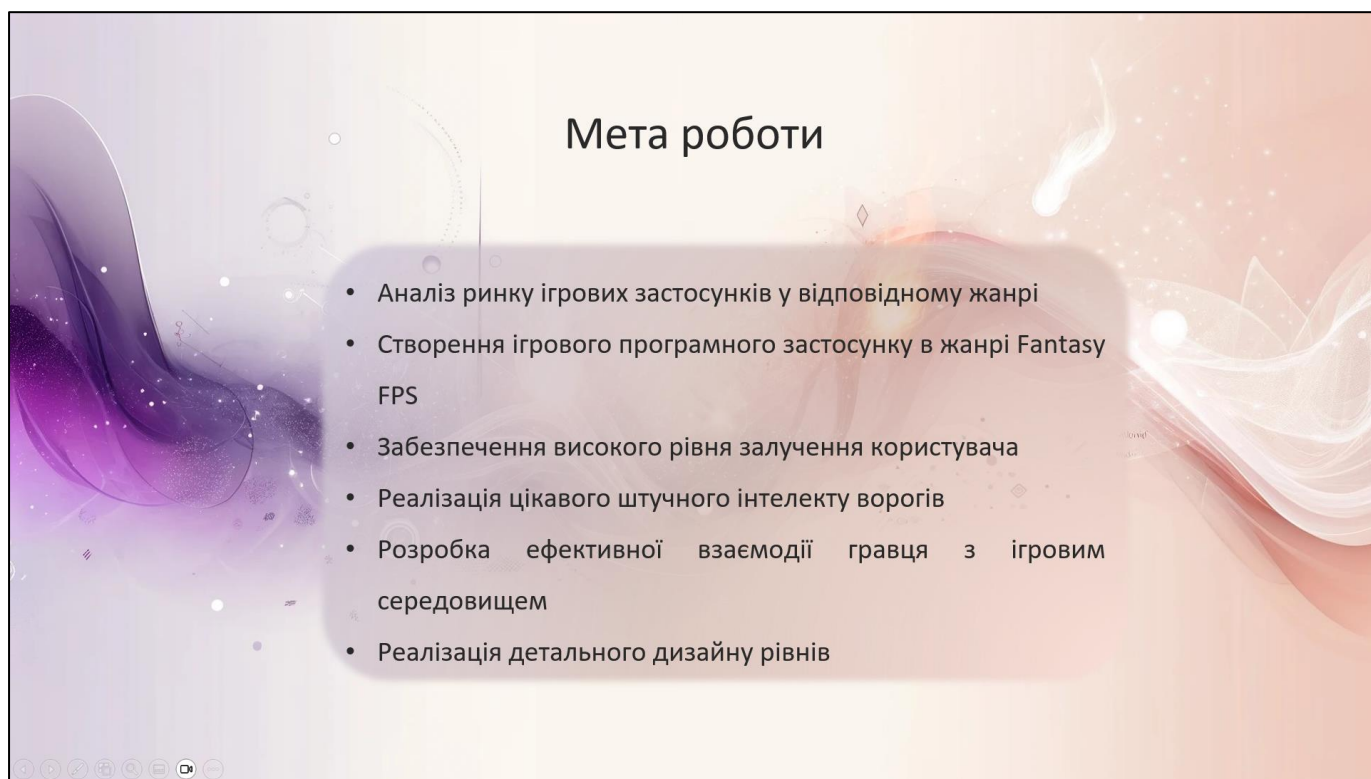


Рисунок Б.3 – Слайд 3

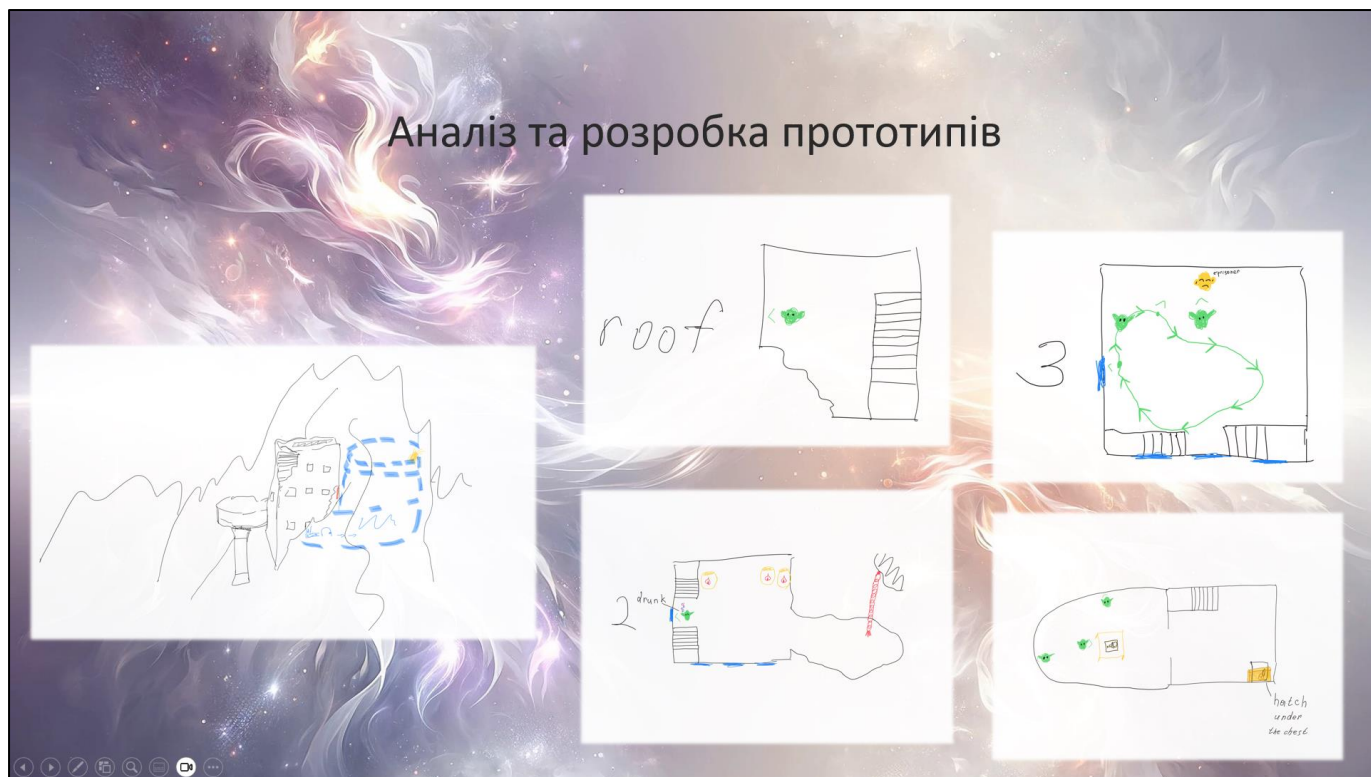


Рисунок Б.4 – Слайд 4

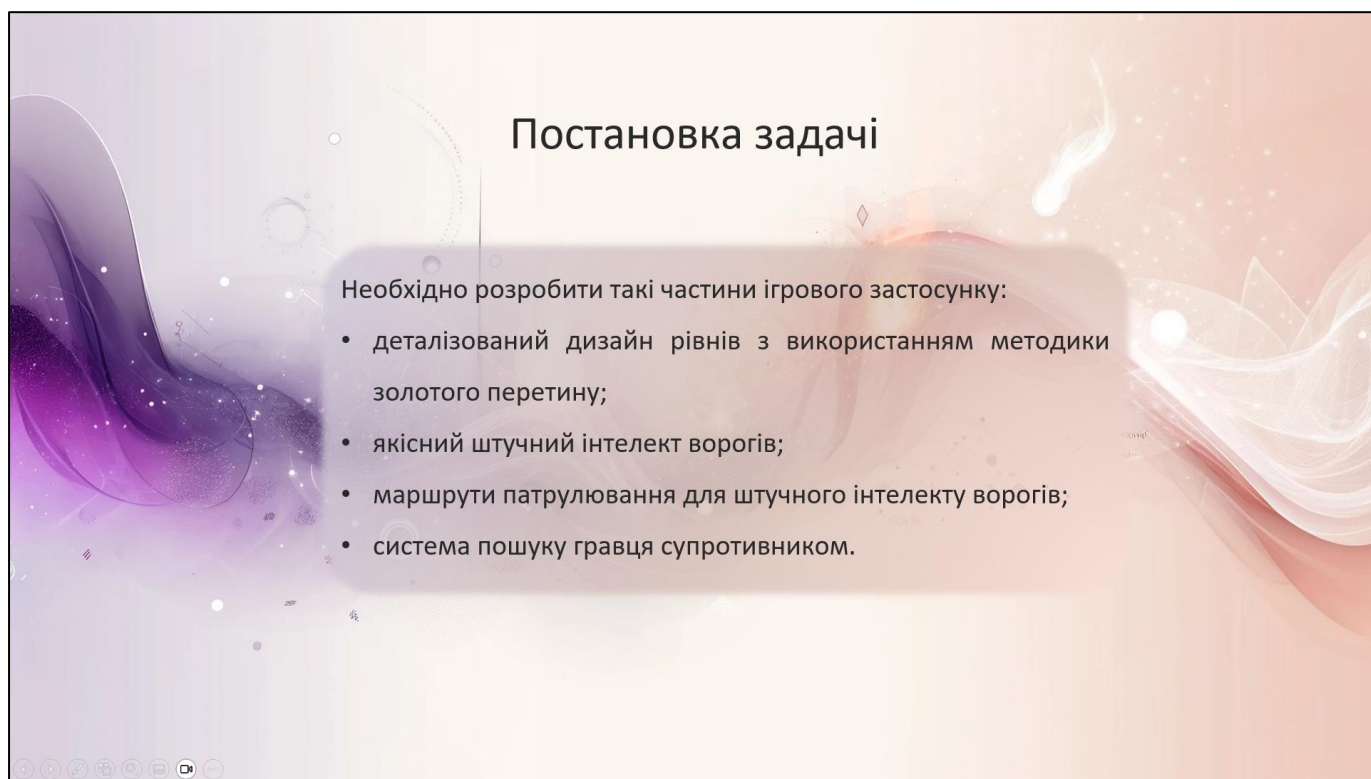


Рисунок Б.5 – Слайд 5



Рисунок Б.6 – Слайд 6



Рисунок Б.7 – Слайд 7

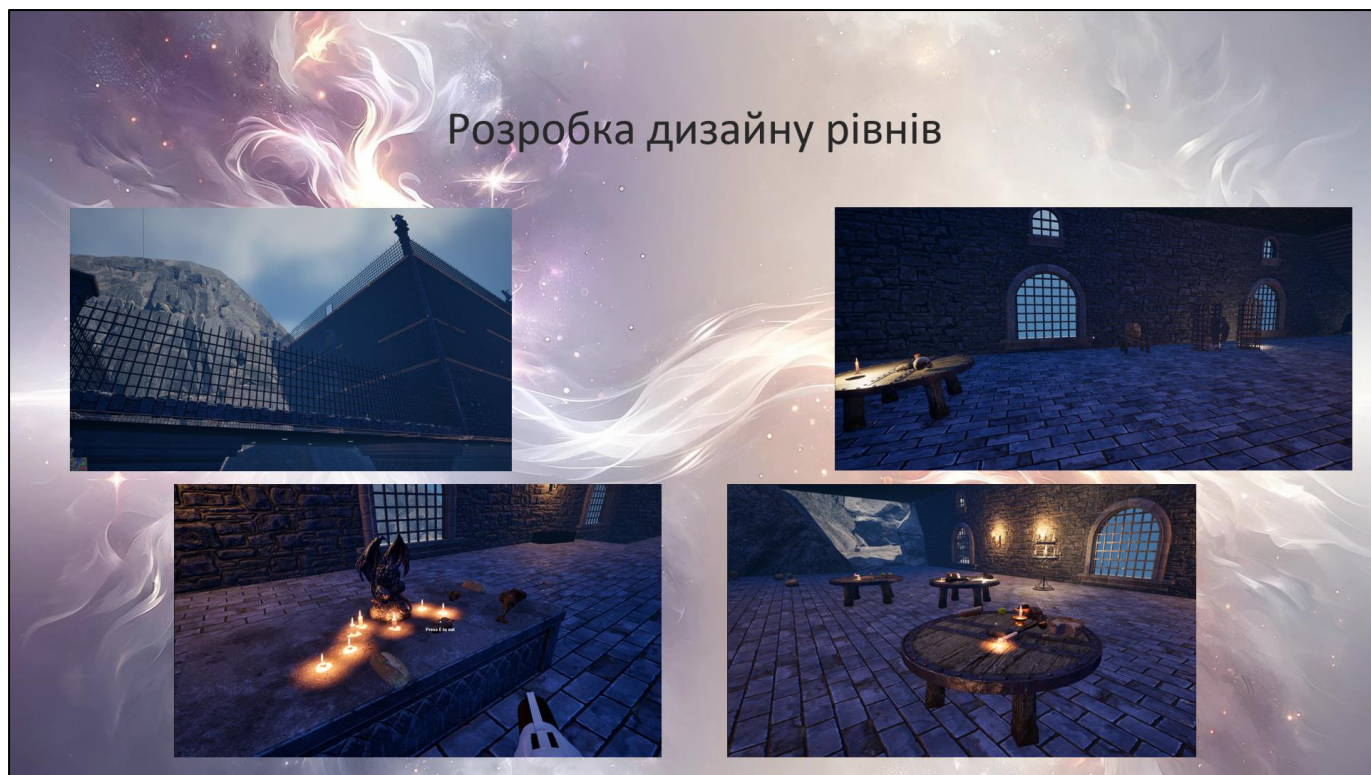


Рисунок Б.8 – Слайд 8

## Втілення принципу золотого перетину та використання технології Landscape

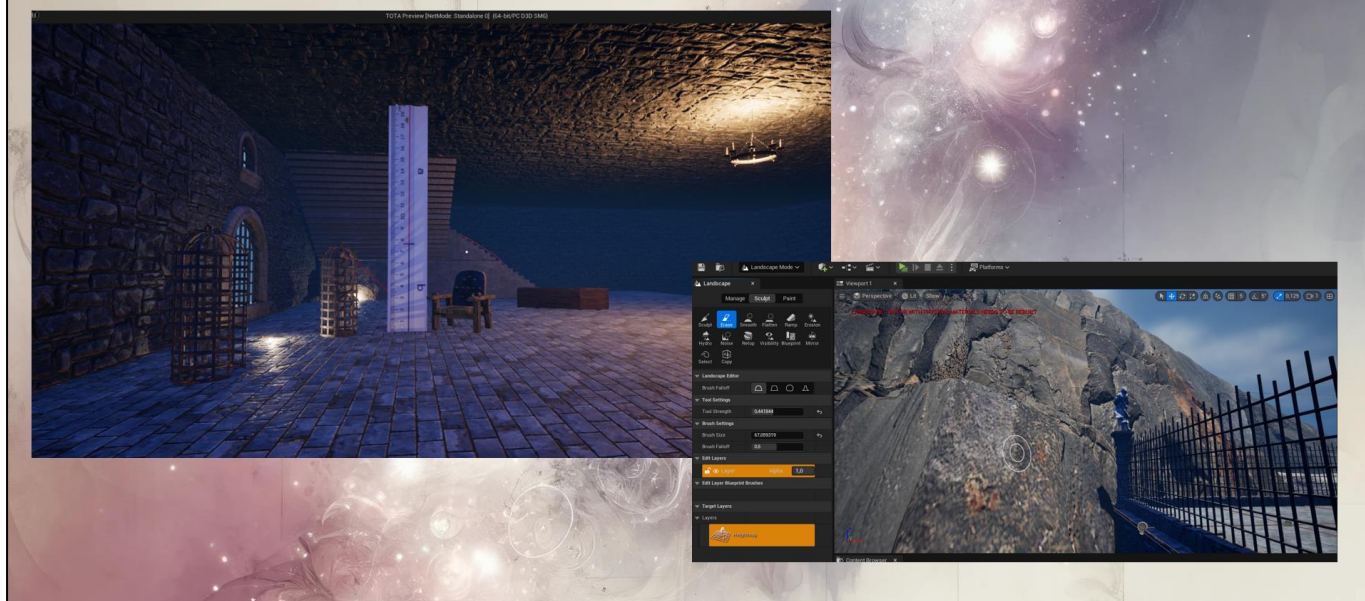


Рисунок Б.9 – Слайд 9

## Реалізація моделей поведінки ШІ ворогів

- 1** — Патрулювання

Маршрут патрулювання реалізований за допомогою кривої (Spline) з точками зупинки
- 2** — Пошук

Після втрати гравця, ворог шукає його в останньому відомому місці.
- 3** — Напрямок погляду

Напрямок погляду ворога реалізований за допомогою динамічного компонента стрілки.

Рисунок Б.10 – Слайд 10



Рисунок Б.11 – Слайд 11

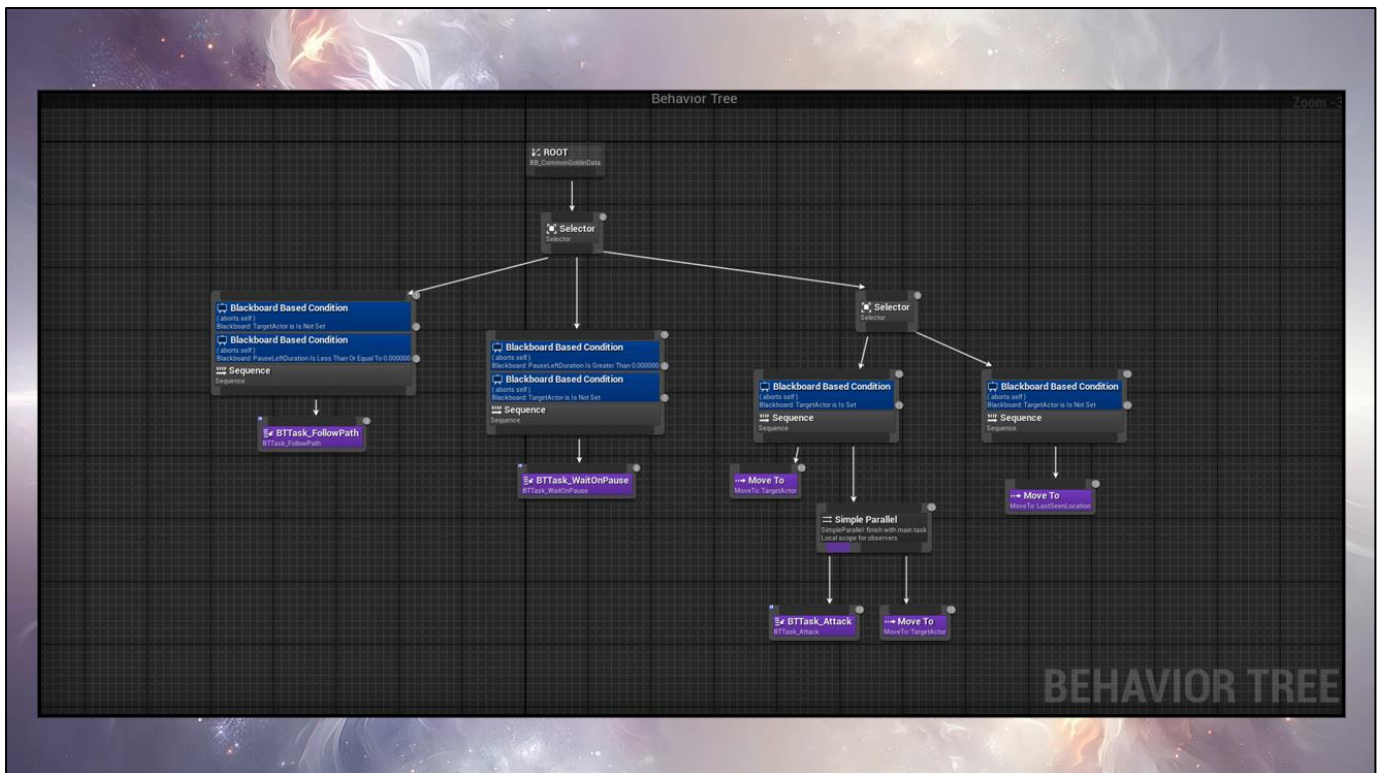


Рисунок Б.12 – Слайд 12

## Програмна реалізація «BTTask\_FollowPath»

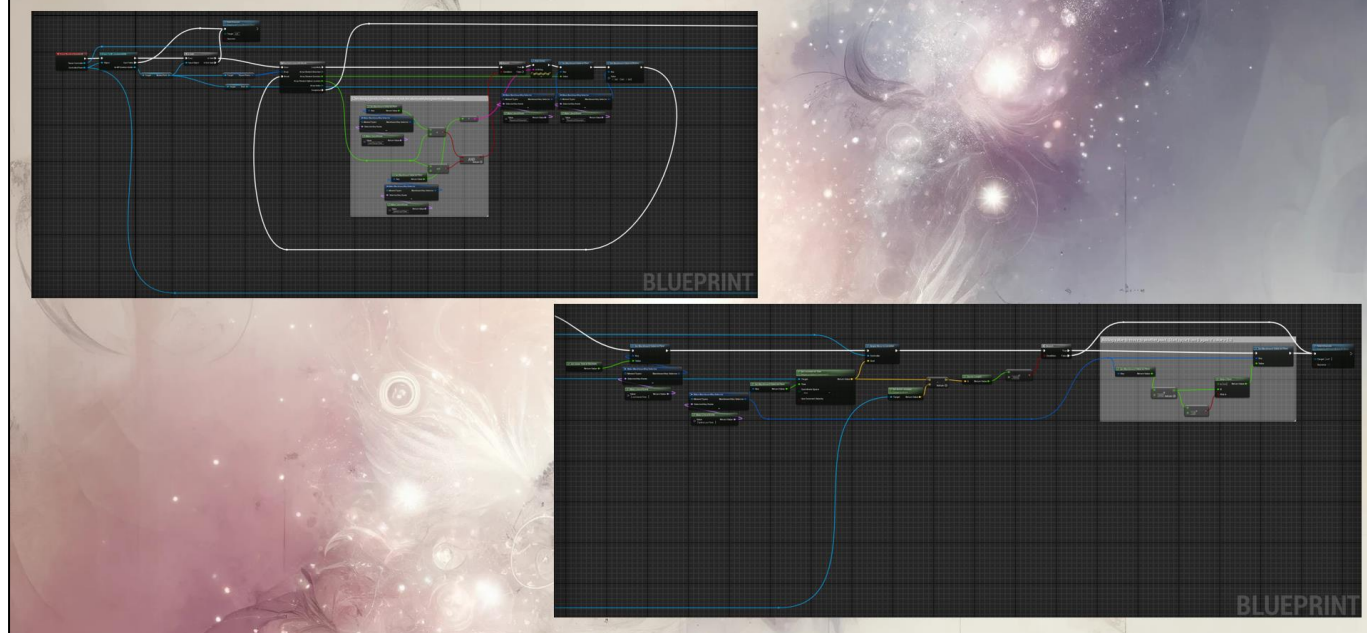


Рисунок Б.13 – Слайд 13

## Тестування

Тест № 1	
<b>Назва тесту:</b>	Гобліни порушують шлях патруля
<b>Опис тесту:</b>	Гобліни неточно йдуть за шляхом патруля
<b>Компонент системи:</b>	Навігаційна система II/Поведінка ворогів.
<b>Пріоритет:</b>	P3 Середній
<b>Критичність:</b>	S2 Значний.
<b>Кроки відтворення:</b>	1. Привернути увагу гобліна. 2. Відбігти далеко. 3. Припинити переслідування. 4. Дочекайтесь поки гоблін повернется на шлях патруля.
<b>Очікуваний результат:</b>	Гоблін точно йде за шляхом патруля.
<b>Фактичний результат:</b>	Гоблін йде за більшим колом, ніж повинен йти.

Тест № 2	
<b>Назва тесту:</b>	Гобліни блокуються об'єктами-привідами
<b>Опис тесту:</b>	Гобліни блокуються великими об'єктами (не мають змоги дістатись гравця)
<b>Компонент системи:</b>	Навігаційна система II/Поведінка ворогів.
<b>Пріоритет:</b>	P3 Середній
<b>Критичність:</b>	S2 Значний.
<b>Кроки відтворення:</b>	1. Прибрати об'єкт з дороги гобліна. 2. Привернути увагу гобліна.
<b>Очікуваний результат:</b>	Гобліни не бачать перешкод там де їх фактично немає.
<b>Фактичний результат:</b>	Деякі гобліни застряють за об'єктами-привідами, втрачаючи можливість переслідування

Рисунок Б.14 – Слайд 14

## Тези XXVIII міжнародного молодіжного форуму «РАДІОЕЛЕКТРОНІКА ТА МОЛОДЬ У XXI СТОЛІТТІ»

УДК 004.89.004.946

### МЕТОДИ ТА АЛГОРИТМИ БАЛАНСУВАННЯ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ІГОР ЖАНРУ FPS

Кісільова М. С.  
Науковий керівник – ст. викл. Новіков Ю. С.  
Харківський національний університет радіоелектроніки, каф. ПІ  
м. Харків, Україна  
e-mail: mshk@ict.kharkivuniv.edu.ua

This research focuses on optimizing artificial intelligence (AI) for enemy behavior in FPS (First-Person Shooter) games to enhance the gaming experience. AI plays a pivotal role in creating realistic and engaging enemy characters, directly influencing game difficulty and player immersion. This study conducts a comparative analysis of existing AI balancing algorithms and methods used for enemy behavior, proposing an optimized approach based on the analysis of player preferences. By selecting the most relevant balancing parameters, this research aims to develop a more adaptive and challenging AI, contributing to a deeper and more captivating gameplay.

Штучний інтелект (ШІ) відіграє ключову роль у створенні реалістичності і цікавих ворожих персонажів в іграх жанру FPS (First-Person Shooter). Ефективність ШІ безпосередньо впливає на складність гри, рівень заперення і загальне враження гравця [1].

В якості вступу до порівняння різних характеристик методів балансування та алгоритмів штучного інтелекту використовується власна розроблена гра у жанрі FPS «Tale of the Awakened».

Балансування ШІ в іграх жанру FPS можна здійснювати за різними параметрами, включно з агресивністю, швидкістю реакції, точністю стрільби та здатністю до навчання [2]. ґрунтуючись на аналізі вподобань цільової аудиторії та сплайнів ігрового процесу, було проведено порівняння характеристик основних методів балансування.

Таблиця 1 – Порівняльна характеристика методів балансування

Метод	Параметри для балансування	Переваги	Недоліки	Актуальність для жанру FPS
Динамічна складність	Рівень майстерності гравця, швидкість проходження гри, частота смерті	Адаптивність ШІ до рівня гравця, корисність ігрового досвіду	Може виконувати відсутня дослідження у складних іграх, висока мовна в складності не простор	Дуже висока

342

Продовження таблиці 1 – Порівняльна характеристика методів балансування

Метод	Параметри для балансування	Переваги	Недоліки	Актуальність для жанру FPS
Обмежені стратегії	Дальність пострілу ворогів (ШІ), кут огляду, здатність чухи шуму	Збільшує реалізм, підвищує логічність обмеження стратегії	Може привести до використання тільки ворогів ШІ	Висока
Симонавання ШІ	Здатність до адаптивної стратегії, зміна поведінки ворогів	ШІ може адаптуватися до ігрових стратегій гравця	Реалізм ШІ, що самонавчається, може бути надібно складно	Середня

На основі цього аналізу можна зробити висновок, що динамічна складність являє собою найбільш універсальний і ефективний метод балансування ШІ в іграх FPS для розробки одиного гравця.

Штучному інтелекту ворогів в іграх жанру FPS, з можливостю відомого перебігання гри на карті, протязомі так модаль поведінки [2].

1. Патрулювання: ШІ здійснює маршрут до виявленої границі.

2. Пошуку: Після втрати гри з поля зору, ШІ шукає його в останньому відомому місці розташування.

3. Використання зброї: ШІ може утримати від вогню гравця в атаку в безпечному місці.

4. Командні дії: ШІ координує дії з іншими ШІ для оточення або захисту на границі.

У власній розробленій грі «Tale of the Awakened» були реалізовані дві моделі поведінки ШІ: патрулювання, пошуку. Маршрути для патрулювання було зроблено за допомогою крайової (Spine), яку можна завантажити прямо у редакторі Unreal Engine 5. Зупинивши її реалізовані через певний час ігрового часу на карті, які мають свою логіку (Time), та швидку, що відповідає за час, проведений ворогом на цій точці. Ще одним покращенням на зупинках є напрям, у якому буде стояти та дивитися ворог. Він був реалізований за допомогою динамічного контролю стрільби, на випадок наварення, що реагується разом з іншими даними масиву.

Для реалізації штучного інтелекту ворогів у грі «Tale of the Awakened» було проведено порівняльну характеристику алгоритмів та обрано оптимальний.

343

Таблиця 2 – Порівняльна характеристика алгоритмів ШІ

Алгоритм	Критерії порівняння	Обмеження	Придатність для жанру FPS
Дерева поведінки	Висока гнучкість і детальність поведінки	Складність реалізації та відсутність обмеження гнучкості у складних ситуаціях	Висока
Машинні станні	Простота реалізації та розробки	Обмеження гнучкості у складних ситуаціях	Середня
Генетичні алгоритми	Автоматизація оптимізації та адаптації	Висока вартість часу і ресурсів на «навчання»	Середня
Нейронні мережі	Здатність до навчання та адаптації	Складність встановлення і висока вартість часу обробки даних для навчання	Середня

На основі цієї порівняльної характеристики та з огляду на високі вимоги жанру FPS, де важливими критеріями є швидка адаптація до дій гравця і деталізована поведінка ворогів, найкращим вибором у розробці стали дерева поведінки. Цей алгоритм забезпечує високу гнучкість у моделюванні різноманітної поведінки ШІ, від простого патрулювання до складних тактичних дій у бою.

Розглянувши порівняльну таблицю, можна дійти висновку, що найкращим методом балансування ШІ для розробки ігрових ворогів у жанрі FPS є динамічна складність, а найкращим алгоритмом виявилася логіка поведінки. Ці підходи забезпечують збалансовану реалізму та інтерактивності ігрового процесу.

Список використаних джерел:  
1. O. Malinova, O. Samoilov, O. Terchii and M. Shirokopteva, A Study of Optimization Models for Creation of Artificial Intelligence for the Computer Game in the Tower Defense Genre, 2020 IEEE International Conference on Problems of Informatics, Science and Technology (PIC S&T), 2020, pp. 491-496, doi: 10.1109/PICST51311.2020.9468057.

2. Радіоелектроніка та молодь у XXI столітті, Т. 6 : Конференція «Інформаційні технології та системи»: матеріали 27-го Міжнародного молодіжного форуму, 10-11 трав. 2023 р. / М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. Харків, 2023. 275 с.

344

Рисунок Б.15 – Слайд 15

## Висновки

- У ході роботи було створено ігровий додаток у жанрі Fantasy FPS, який інтегрує сучасні ігрові механіки та забезпечує високий рівень залучення користувача.
- Розроблено систему штучного інтелекту для ворогів, що використовує дерева поведінки, що підвищує реалістичність та динамічність ігрового процесу.
- Створено продуманий дизайн рівнів гри, які відповідають сучасним вимогам до геймдизайну, забезпечуючи баланс між естетикою та функціональністю.
- Проведено тестування ігрового застосунку.
- Результати роботи створюють базу для подальшого розширення ігрового світу, введення нових механік та можливостей для гравців, що сприяє розвитку проекту у майбутньому.

Рисунок Б.16 – Слайд 16

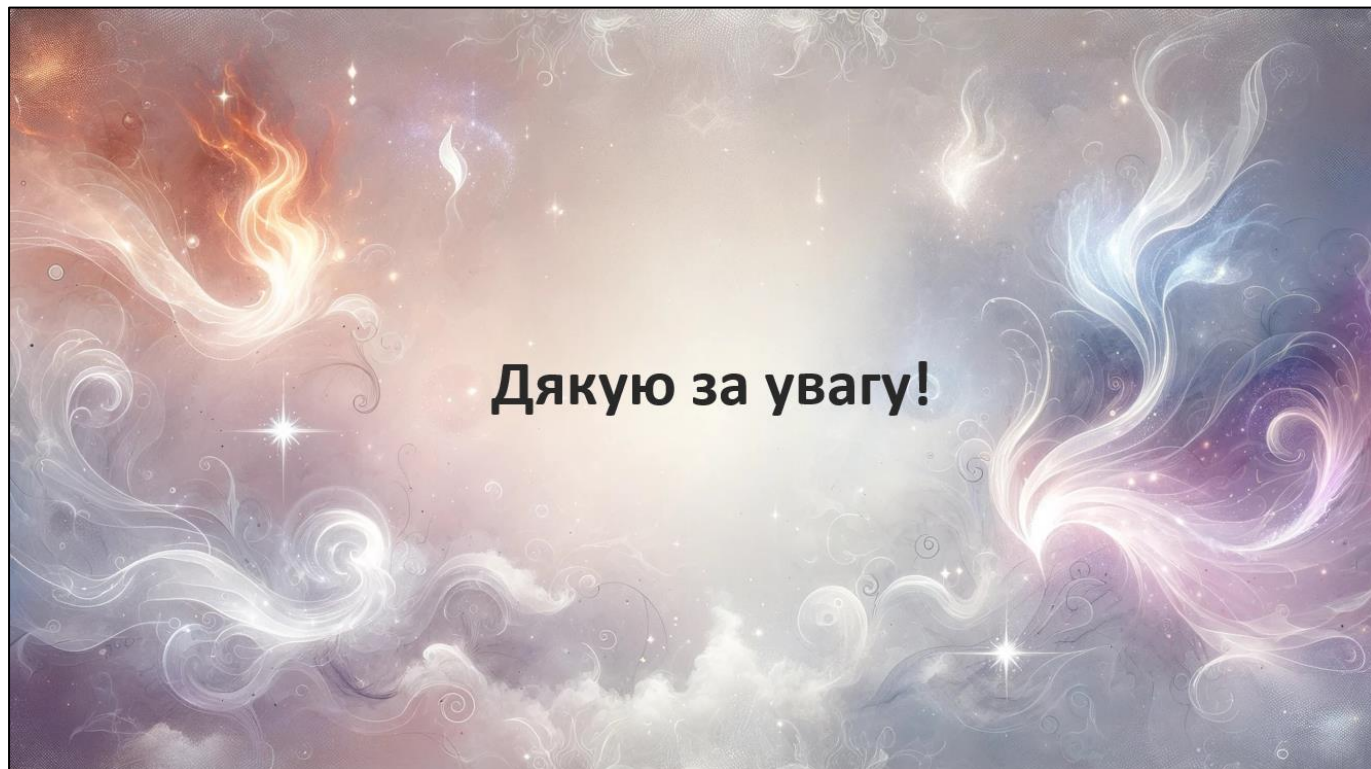


Рисунок Б.17 – Слайд 17

## ДОДАТОК В

### Геймдизайн-документ

#### **Тетра**

Технологія: ПК, консолі

Механіка: шутер від першої особи

Історія: ГГ – звичайна людина у світі магії

Естетика: зовсім трішки magicpunk та середньовічне фентезі

Рисунок В.1 – Геймдизайн-документ сторінка 1

## Лор

Даний світ дуже нагадує звичні нам середньовічні фентезійні світи за виключенням магії. Зазвичай, магу для використання своїх магічних здібностей необхідно використовувати стихійник - спеціальний мінерал, що перетворює чисту ману на стихію: вогонь, лід, вода, блискавка та похідні від них. Завдяки цій магії, працює багато виробництв, а військові в свою чергу мають спеціальну "вогнепальну" зброю, що працює на основі стихійників. Попри все, магією володіє далеко не кожен (приблизно 0,2% населення). Маги є дуже поважними особами у суспільстві, адже як раз завдяки ним, їх світ не переживає постійні глобальні війни, епідемії, голод та інші біди, що спіткали наш. Якщо казати коротко, то немає критичної недостачі життєво необхідних ресурсів.

Усюди панує раса людей, але при цьому разом з ними співіснують й інші, такі як ельфи та метри (назва пов'язана з їх зростом, що зазвичай дорівнює близько одному метру). На момент гри, людству відомі два материки Фівенія (в народі Земля Творця) та Вардус (в народі Темні Землі). Фівенію населяють усі миролюбні раси (зазначені раніше люди, ельфи та метри), а Вардус - гобліни, тролі, дракони та інші потвори. Дуже давно союз людей, ельфів та метрів прогнали гоблінів з Фівенії до Вардусу та возвели незламну стіну (див. мапу), що забезпечило мир для їх нащадків, але, як виявилось, не назавжди. Багато років потому у королівській сім'ї з'явився хлопчик, що володів надпотужною магією. Завдяки ній, він міг підкорити собі майже будь-кого, від кішки до людини (чим розумніша істота - тим важче її підкорити). Ця сила хлопчика, котрого назвали Вальцем, не дала йому стати порядною людиною, він виріс в найжорстокішого принца, що знала історія. Через це, помираючи, король надав перевагу у виборі спадкоємця своєму молодшому сину. Ця ситуація розлютила Вальца, через що він вирішив відправитись у Темні Землі, де зміг підкорити собі народ гоблінів. Згодом він почав розвивати їх армію, підкоряти тролів у допомогу гоблінам, могутніх елементалей та навіть одного дракона. Згодом, через надвелику кількість водних елементалей (вони є одним цілим із усією водою у світі, вони підтримують баланс. Чим більше існує водних елементалей - тим менше води та навпаки), котрі будучі завербованими злим принцом, перестали щезати, рівень

океану опустився, що дало гоблінам, котрі не ладнають з наукою і будівництвом кораблів, змогу обійти незламну стіну по новоствореним берегам. Після цього гобліни на чолі із Вальцем напали на Королівство кинджалів (батьківщина Вальца) із наміром захопити світ.

Рисунок В.3 – Геймдизайн-документ сторінка 3

## Сюжет гри

Звуть нашого героя Кіт (в українській локалізації через це його всі і називають котом, а в англійській перекичують на kid (дитина)). Ще в дитинстві він помітив, що удача завжди на його стороні, тому обрав найлегший для себе метод заробітку - крадіжка та обман. Виконуючи підозріле замовлення, що само по собі не є новинкою для героя, він вперше став жертвою невдачі і розбив артефакт, котрий мав вкрасти, який містив у собі запечатану силу, що здатна наділити будь-яку істоту здібностями до магії, після чого вона перейшла до найближчої живої істоти - Кіта. Артефакт було зроблено в давнину як експеримент, котрий, як виявилось, потребує забагато жертв для створення. На створення одного подібного артефакту було витрачено більше десятка життів немагів. Злочин головного героя не залишився без покарання. Охорона конвою, яка повинна була доставити даний артефакт до придворного мага на ім'я Зерхен, тепер доставить новоствореного мага-крадія.

Після пробудження від довгого сну, Кіт одразу опиняється у камері для підсудних. Згодом, його визивають до суду перед самим королем (Леопольдом, молодшим сином старого короля). Беручи до уваги ситуацію в країні (війна, нестача магів), король прийняв єдине вірне рішення - зберегти Кіту життя, але і відпустити він його не міг. Було вирішено навчити Кіта володіти своєю новою силою і відправити його на фронт - на допомогу в один із прифронтових гарнізонів. Спокутою для Кіта повинно було стати служіння в армії до перемоги, або смерті.

Незадоволений своїм вироком, головний герой вирішує, що дезертує у будь-який зручний момент. Тут удача відвертається від нього вразі - в першу ж ніч його перебування там, армія супротивника робить ривок та за лічені години дістається до укріплень, в одному з яких знаходиться Кіт. Побачивши смерть невинних людей, він міняє свій світогляд і вирішує боротись за тих, кого обкрадав усе своє життя до цього.

Як єдиний бойовий маг, що знаходився в тому гарнізоні, Кіт мав звітувати придворному магу (за сумісництвом, генералу армії) про все, що той бачив.

Нічого з того, що він йому розказав, не здивувало Зерхена. Зваживши ситуацію, генерал вирішив відправити головного героя до Вардусу - дослідити підозрілу активність гоблінів біля одного з древніх храмів. Обрано було саме Кіта через його минуле, адже хто, як не крадій, зможе прошмигнути до бази супротивника і повернутись звідти живим?

Обстеживши руїни храму, де знаходились гобліни, Кіт знайшов документ - наказ від Вальца, в якому було сказано шукати "корону стійкого розуму". Після чергового звіту, головний герой разом із Зерхеном розуміють, що мова йде про артефакт, що згадується в одній із казок. Якщо вірити тому, що в ній написано - цей артефакт дозволяв королям у давнину не ставати жертвами нечесних радників, що могли намагатись маніпулювати ними. Стає ясно, що Вальц боїться, що саме даний артефакт може його зупинити. Чого Вальц не знав - так це того, що даний артефакт вважається навіки загубленим у пустелі елементалей - безлюдній місцевості, де вже понад тисячу років, без перерви тривають бої елементалей різних стихій. Коли один елементаль помирає - світовий баланс стихій згодом повертає його назад.

Для того щоб все ж знайти потрібний артефакт посеред пустелі, було створено спеціальний інструмент, що завдяки стихійнику, генерує воду у напрямку найближчої мани. Для того щоб знайти артефакт, спершу Кіту та команді, що була створена спеціально для даної місії, потрібно вбити всіх елементалей на території і прислухатись до найменших коливань води у приборі. Даний процес доводиться повторювати кілька разів, адже кожного разу елементалі повертаються. В результаті, команда знаходить корону, але вона виявляється розбитою.

Після даної операції було прийнято рішення - провести ще одну, в ході якої Кіт з напарниками мають визволити з полону єдиного у світі коваля-мага, що, потенційно, міг би перекувати корону. Дана місія проходить успішно, але навіть після визволення від Вальца, коваль ні на що не реагує - він просто втратив глузд після тривалого впливу магії підкорення. Єдиним шансом на лікування для нього є - зілля повного зцілення, що може надати лише могутня та нарцистична відьма

- Марла. Вона є найкращим зіллеваром серед усіх королівств Фівенії, але її послуги коштують дорого, дуже дорого.

Платою за зілля повного зцілення, Марла просить дістати їй інгредієнти для зілля, що поверне їй красу та молодість. Знайти їх можна лише в Темних Землях. Робити тут нічого - Кіт вкотре відправляється до Вардусу - шукати необхідні трави та чудовиськ, рештки яких теж потрібні для зілля. Хоч і не без проблем, але головний герой знаходить усі інгредієнти та отримує натомість зілля повного зцілення.

Зілля лікує коваля. При цьому загоюються навіть старі шрами та опіки, що він постійно отримує. Придворний маг повідомляє йому про його місію - відремонтувати корону. Без зайвих слів, коваль відправляється виконувати свою роботу. Зерхен повідомляє Кіту, що саме йому доведеться боротися із Вальцем, адже головний герой, не дивлячись на недосвідченість, зарекомендував себе як одного з найсильніших бійців, що здатний вижити майже в будь-якій ситуації.

Перед початком свого, вірогідно, останнього завдання, Кіт вирішує помститися генералу, що вів армію на гарнізон котрий був знищений в самому початку історії. Потай від усіх, він споряджається на Вардус - на базу, де, за останніми даними, знаходиться цей генерал. Після успішної ліквідації, він з боєм виривається із лігва та повертається до свого королівства, де коваль як раз до того часу встиг перекувати корону. Його майстерності вистачило щоб зробити з неї браслет - більш зручний артефакт для носіння на полі бою.

Головного героя споряджають на останню битву - віддають браслет, найкращу зброю, що тільки є у королівстві, зілля мани та інше. Після - відправляють до підводного храму (котрий вже не є підводним через зниження рівня моря), лігва злодія. Кіт успішно ліквідує дракона - головного охоронця Вальца, після чого починає бій із самим ватажком гоблінів. Після запеклого бою, в ході якого Кіт міг загинути безліч разів, Вальц помирає від його руки. Після цього чари маніпуляції розумом зникають і елементалі одразу щезають, тим самим нормалізуючи рівень моря, в ході чого головний герой ледве не тоне, але

в останню мить на кораблі йому на допомогу приходять друзі, яких він знайшов під час своїх небезпечних пригод.

Рисунок В.7 – Геймдизайн-документ сторінка 7

## **Цільова аудиторія**

Основна цільова аудиторія гри – молоді люди 16-30 років, оскільки ігрові аспекти, що розглядаються в нашій грі, достатньо гарно задовольняють і відповідають їх бажанням і потребам для розваг. Гра має вікове обмеження 16+, через середній ступінь жорстокості вбивств.

Для залучення аудиторії в цій грі розраховані такі аспекти: отримання емоцій, достатній перелік різної зброї, різнобарвність бійок, іммерсивність, жорстокість.

**USP**

- Здолайте могутнього злодія.
- Знайдіть неочікуваний шлях вирішення проблеми.
- Експериментуйте зі стихіями.
- Здолайте ворога у відкритій динамічній битві.
- Пройдіть повз ворога будучи непоміченим.
- Виконуйте додаткові завдання, що допоможуть з основними.

Рисунок В.9 – Геймдизайн-документ сторінка 9

## **Час ігрової сесії**

Основний сюжет гри буде складатися з 7 повноцінних місій та 1 вступного завдання, під час якого буде проводитися навчання.

Сайдквести – в розробці.

Приблизне проходження однієї місії буде займати 2-3 години. Отже загальна тривалість гри складе десь 14-20 годин

## Технологічні характеристики

Для Windows

<b>Мінімальні</b>	<b>Рекомендовані</b>
ОС: 64-bit Windows 10	ОС: 64-bit Windows 10
Процесор: Intel Core i5-6600 (3.3Ghz), AMD Ryzen 5 1400 (3.2Ghz)	Процесор: Intel Core i7-8700 (3.2Ghz) or AMD Ryzen 5 3600 (3.6 Ghz)
Оперативна пам'ять: 8 GB	Оперативна пам'ять: 8 GB
Відеокарта: NVIDIA GeForce GTX 960 4GB, AMD Radeon RX 470 4GB	Відеокарта: NVIDIA GeForce 1080 Ti or AMD Radeon RX 5700 XT or INTEL Arc A770
Місце на диску: ? GB	Місце на диску: ? GB

Рисунок В.11 – Геймдизайн-документ сторінка 11

## Опис гри

### Опис ігрового процесу

Гра має лінійний сюжет, але також в ній є розгалуження у вигляді додаткових квестів. Основний сюжет складається з 7 місій, для кожної з яких розроблена окрема локація. Додаткові квести інколи мають окремі локації, але найчастіше базуються на локаціях з основного сюжету. Гра є шутером від першої особи, де гравець грає за мага, що перетворює свою магічну енергію на стихійну завдяки особливому виду зброї. Вибір спорядження/місій та сюжетні діалоги проходить у хабі – кабінеті придворного мага. Завдяки іммерсивності гри, гравець може обирати нові, несподівані, винахідливі шляхи для виконання свого завдання: від лінійного проходження до створення імпровізованих сходів із льоду, або вбивства супротивника шляхом скидання із прірви завдяки повітряній пушці.

### Механіки

Бій:

- Найчастіше в битві використовується магічна «вогнепальна» зброя, що дозволяє використовувати певні стихії.
- Холодна зброя, що дозволяє виконати моментальне вбивство із стелсу, або наносити удари у відкритій битві.
- Шкала здоров'я – після отримання поранень, кількість очків здоров'я знижується і це відображається на шкалі. При досягненні 0 – настає моментальна смерть. Поступово, лише поза битвою, очки здоров'я відновлюються.
- Шкала мани – потрібна для стрільби із магічної зброї. Різна зброя використовує різну кількість очків мани. Поступово відновлюється під час битви, але поза неї – набагато швидше.

Вплив стихій на навколишнє середовище:

- Лід: застигає, створюючи колізію, попадаючи на статичні об'єкти. Завдяки цьому гравець може створити імпровізовану барикаду, сходи і т.д. При попаданні на живу істоту – підморожує її, тим самим уповільнюючи її рухи. Час дії – 12 секунд, після чого додаткова колізія зникає, а істоти перестають отримувати штраф до рухливості. Наносить середню кількість урону.
- Вогонь: підпалює легкозаймисті предмети (спеціальні бочки з горючим, факели, свічки, тощо). Підпалює живих істот, тим самим,

наносючи ефект горіння, що триває 3 секунди (під час дії цього ефекту, вони отримують урон). Наносить велику кількість урону.

- Вода: Створює водні поверхні, які можна наелектролізувати, тим самим створивши пастку. Можна використати проти живої істоти, тим самим зробивши її більш вразливою до електрики. Наносить низьку кількість урону.
- Електрика: немає впливу на статичні об'єкти окрім водяних поверхонь, які він наелектролізує. Жива істота застигає від цієї стихії на 1 с. Наносить середню кількість урону.
- Вітер: відштовхує фізичні об'єкти та істоти, дозволяючи скинути їх з обриву, або швидко перемістити. Наносить низьку кількість урону.
- Кислота: Не має впливу на статичні об'єкти. При потраплянні на живу істоту, накладає на неї ефект роз'їдання, що триває 1 хв. Наносить низьку кількість моментального урону, але за весь період дії ефекту накопичується велика кількість урону.
- Плазма: Не має впливу на статичні об'єкти. При потраплянні на живу не створює додаткових ефектів. Наносить найбільшу кількість урону.

#### Стелс:

- Умова виявлення гравця: У противника є спеціальні колізії: сферична (відповідає слуху противника) та конічна (відповідає зору). Якщо гравець заходить в них, або робить шумні/підозрілі дії – у противника починає заповнюватись шкала тривоги. Якщо джерело тривоги не щезає – через 3 секунди гравця буде помічено. Також є додаткові колізії – з меншим радіусом для сфери та кутом для конуса. В цих зонах гравця помічають одразу. У режимі стелсу, колізії тривоги (жовті на малюнку) не працюють, якщо не шуміти. Колізії моментального помічання (червоні на малюнку) працюють без змін.

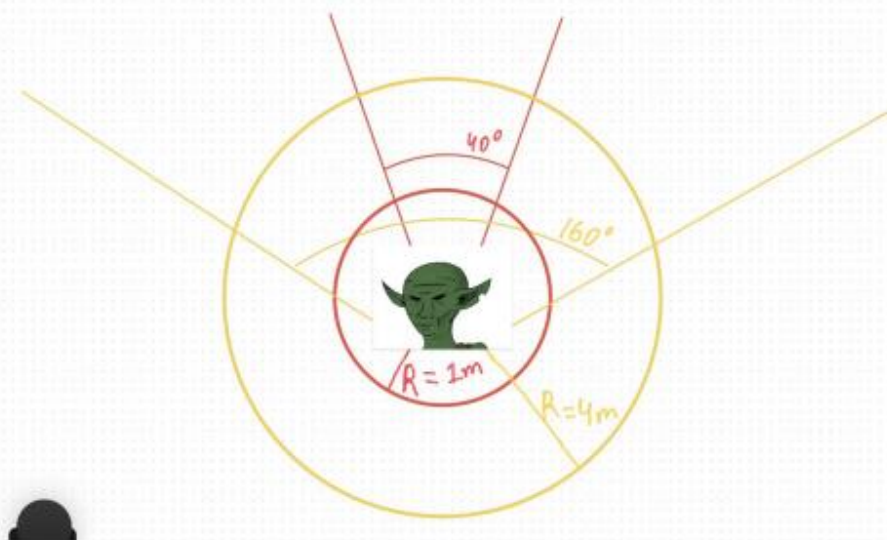


Рисунок В.13 – Геймдизайн-документ сторінка 13

- Моментальне вбивство: Якщо гравця не було помічено і противник знаходиться в полі досягнення – його можна моментально вбити холодною зброєю не здіймаючи галасу.

#### Іммерсивність:

- З деякими ігровими об'єктами (такими як важіль, двері та інше) можна взаємодіяти.
- Часто зустрічаються місця до яких можна потрапити лише якщо провзаємодіяти з іншими об'єктами, або створити собі шлях туди, або прибрати перешкоди.
- Гравець може підіймати неважкі об'єкти та переміщувати їх (як у Скайрім).
- Деякі об'єкти можна підібрати (ніяк не відображається у hud), що дозволяє взаємодіяти з новими об'єктами. Наприклад, таким чином можна підібрати недостатню деталь для механізму, або ключ від закритої двері.
- Ігрові персонажі можуть отримувати урон від прискорення (гальмування). Тобто, якщо їх моментальне прискорення завелике, що в рамках гри може бути досягнуто лише за рахунок падіння, удару об стіну.

## Інтерфейси

Екран запуску гри



Екран загрузки рівня

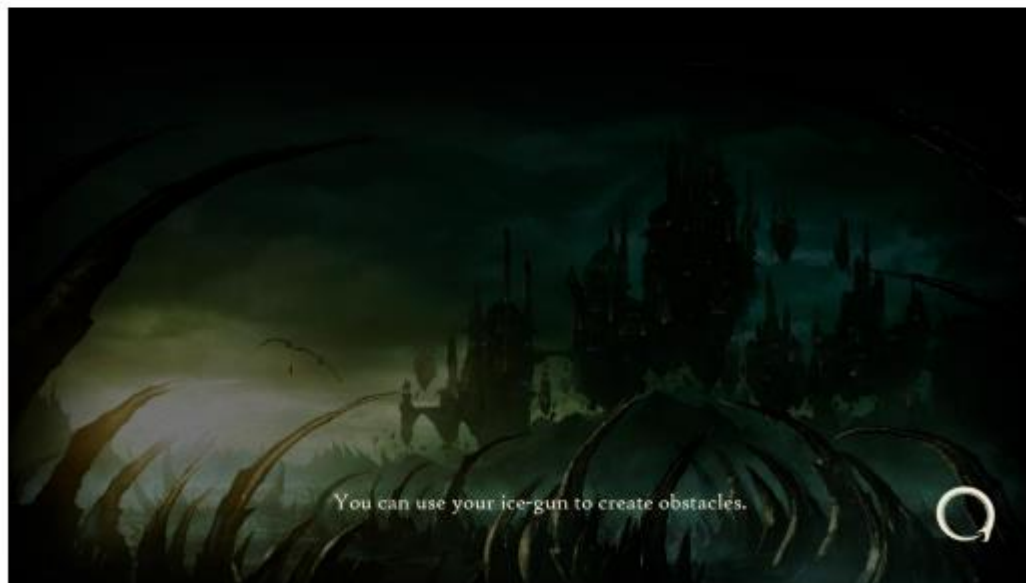


Рисунок В.15 – Геймдизайн-документ сторінка 15

Геймплей під час битви



Геймплей поза битвою

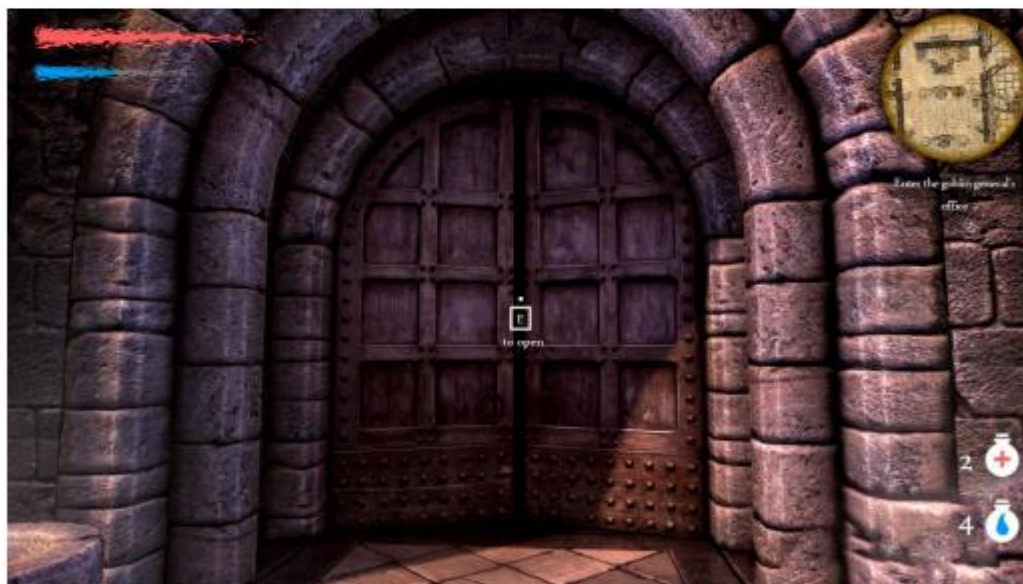


Рисунок В.16 – Геймдизайн-документ сторінка 16

Хаб (місце для діалогів з персонажами, вибору рівня, екіпіровки, тощо)



Інтерфейс інвентаря

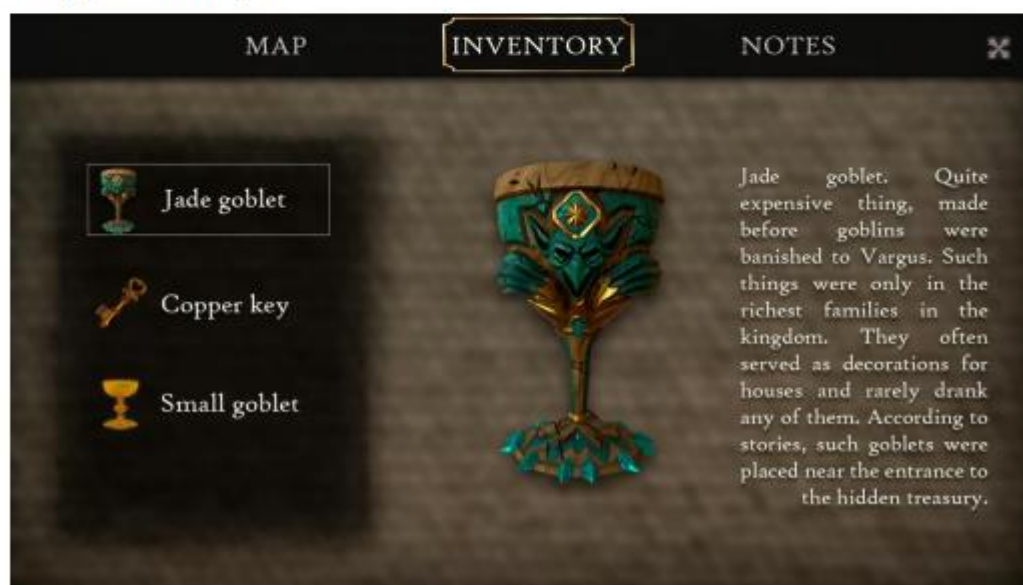


Рисунок В.17 – Геймдизайн-документ сторінка 17

## Інтерфейс з мапою



## Інтерфейс із записками

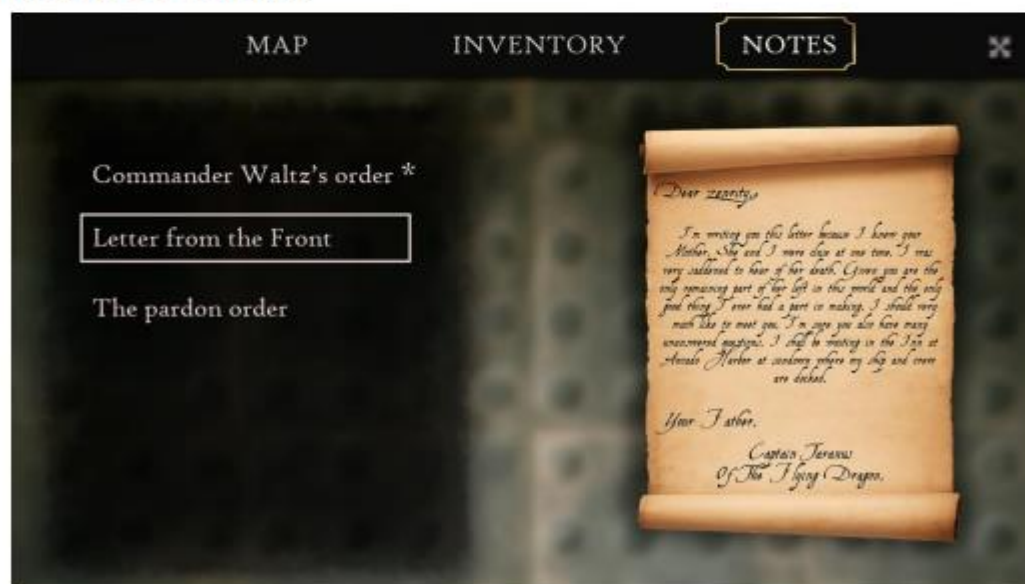


Рисунок В.18 – Геймдизайн-документ сторінка 18

**Візуал****ГГ**

<https://docs.google.com/presentation/d/1p0ElywFLtYaHxAAQaiij2w4imadesnKYySpUzfF2rIU/edit?usp=sharing>

**Локації**

<https://docs.google.com/presentation/d/1zzaVCnDaXkSvSHeo7MkjOxBQWm6YxWE7OahR9zywB6Y/edit?usp=sharing>

**Другорядні персонажі**

<https://docs.google.com/presentation/d/1AwujuGxxOABpTjDZgNPuL0yl4EWIFYFMmDVcQIyuKNo/edit?usp=sharing>

**ДОДАТОК Г**

Тест-план

**Tale of the Awakened****План тестування  
(Test Plan)****Версія 1.0  
(Version 1.0)**

Рисунок Г.1 – Тест-план сторінка 1

**Зміст**

<b>1. Вступ (Introduction)</b> .....	4
1.1. Мета (Purpose).....	4
1.2. Довідкова інформація (Background).....	4
1.3. Галузь застосування (Scope).....	5
1.4. Визначення проекту (Project Identification).....	5
<b>2. Вимоги до тестування (Requirements for Test)</b> .....	7
2.1. Функціональні вимоги .....	7
<b>3. Типи тестування. Стратегія тестування (Test Strategy)</b> .....	9
3.1. Типи тестування (Testing Types).....	9
<b>4. Ресурси (Resources)</b> .....	14
4.1. Ролі (Roles) .....	14
4.2. Система (System) .....	14
<b>5. Етапи проекту (Project Milestones)</b> .....	16
<b>6. Кінцевий продукт (Deliverables)</b> .....	17
6.1. Тестова модель .....	17
6.2. Звіти з дефектів (Defect Reports) .....	17

Рисунок Г.2 – Тест-план сторінка 3

## 1. Вступ (Introduction)

### 1.1. Мета (Purpose)

Метою даного плану тестування є детальний опис процедур і підходів до тестування гри "Tale of the Awakened". Документ має на меті забезпечити високу якість продукту через систематичне виявлення та виправлення помилок, покращення геймплею та інтерфейсу. Цей документ призначений для координації діяльності команди тестувальників, а також для інформування розробників про методи та результати тестування.

### 1.2. Довідкова інформація (Background)

Головний персонаж гри - Кіт, звичайна людина, що несподівано для себе відкриває здатність до магії після розбиття давнього артефакту. Незважаючи на своє походження, він знаходить у собі силу стати могутнім магом, що володіє різноманітними стихіями. Його минуле - це історія про звичайне життя у світі, де магія є привілеєм небагатьох, але одна подія радикально змінює його долю.

Гра має лінійний сюжет, але також в ній є розгалуження у вигляді додаткових квестів. Основний сюжет складається з 7 місій, для кожної з яких розроблена окрема локація. Додаткові квести інколи мають окремі локації, але найчастіше базуються на локаціях з основного сюжету.

- Гра пропонує широкий арсенал зброї. Найчастіше в битві використовується магична «вогнепальна» зброя, що дозволяє використовувати певні стихії.
- Холодна зброя, що дозволяє виконати моментальне вбивство із стелсу, або наносити удари у відкритій битві.

Шкала здоров'я – після отримання поранень, кількість очків здоров'я знижується і це відображається на шкалі. При досягненні 0 – настає моментальна смерть. Поступово, лише поза битвою, очки здоров'я відновлюються.

Шкала мани – потрібна для стрільби із магичної зброї. Різна зброя використовує різну кількість очків мани. Поступово відновлюється під час битви, але поза неї – набагато швидше.

Система урону в "Tale of the Awakened" базується на стихійному впливі та фізичних атаках. Вороги та гравець можуть отримувати урон від різних стихій, що впливають на них по-різному: вогонь може підпалити та завдати тривалого урону; лід - заморозити та сповільнити; блискавка - паралізувати ціль. Ефективність урону залежить від вразливостей ворогів до певних стихій, що додає стратегічний елемент у виборі зброї та магії для бою.

### 1.3. Галузь застосування (Scope)

Метою тестування гри "Tale of the Awakened" є перевірка коректної роботи її функціоналу та зручності для користувача. Результатом процесу тестування повинен стати розгорнутий огляд, що надасть розробникам, менеджерам і користувачам даного продукту картину якості ігрового процесу та юзабіліті. Для аналізу системи використовується методика "чорної скриньки", (мануальне, функціональне та нефункціональне тестування). Гра перевіряється на платформі Windows 11.

Ризики і непередбачувані обставини, що можуть зустрітися у проекті: недостатня база знань для розробки проекту.

### 1.4. Визначення проекту (Project Identification)

У таблиці 1.1 наведено документацію та її готовність, для розробки плану тестування.

Таблиця 1.1 – Документація

Документ і версія / дата	Створено або доступно	Отримано або перевірено	Автор або ресурс	Примітка
Специфікація вимог	Так	Так	Бізнес-аналітик	
Функціональна специфікація	Так	Так	Бізнес-аналітик	
План проекту	Так	Так	Project manager	
Специфікація дизайну	Ні	Ні	UX дизайнер	
Прототип	Так	Так	Команда розробників	
Керівництво користувача	Так	Так	Усі розробники та бізнес-аналітик, тестер	
Бізнес модель	Так	Так	Бізнес-аналітик	
Модель даних	Так	Так	Команда розробників	
Бізнес-функції	Так	Ні	Команда розробників	
Оцінка ризиків	Так	Так	QA	

Рисунок Г.5 – Тест-план сторінка 6

## 2. Вимоги до тестування (Requirements for Test)

### 2.1. Функціональні вимоги

Перелік функцій системи, які будуть тестуватися:

- Інтерфейс меню:

- а) анімації переходів між елементами інтерфейсу;
- б) переходи за допомогою клавіатури поміж елементами інтерфейсу;
- в) вікна з описом предметів в інвентарі;
- г) перевірка меню нотатків та записок;
- д) перевірка інтерфейсу налаштувань гри;
- е) перевірка інтерфейсу закінчення гри (сметрі).

- Інтерфейс гри:

- а) перевірка відображення значень стану гравця, а саме: здоров'я, кількість мани.

- Ігровий процес:

- а) перевірка механік стрілянини, а саме: завдання шкоди, фізичне відображення ефектів попадання стихійних снарядів;
- б) перевірка коректності завантаження текстур;
- в) перевірка поведінки мешів персонажів
- г) перевірка механіки поповнення здоров'я через їжу;
- г) проходження завдань;
- д) перевірка механіки керування предметами;
- е) перевірка коректності розподілення маси в предметах;
- е) перевірка нанесення урону гравцю самим гравцем (падіння з вікна);
- і) перевірка режиму стелсу;
- ї) перевірка зони сприйняття ворогів (зір, слух);
- й) перевірка іммерсивності рівня;

Рисунок Г.6 – Тест-план сторінка 7

- к) перевірка взаємодії з оточуючим середовищем;
- л) перевірка колізій на локації;
- м) перевірка відображення попадання стихійних снарядів в певні предмети на локації.

- Штучний інтелект ворогів:

- а) перевірка патрулювання ворогів;
- б) перевірка системи пошуку гравця ворогом;
- в) перевірка коректності поведінки ворогів в різних випадках.

Рисунок Г.7 – Тест-план сторінка 8

### 3. Типи тестування. Стратегія тестування (Test Strategy)

У процесі оцінки системи будуть застосовані наступні методики тестування:

- Функціональне Тестування: Зосереджене на перевірці функцій гри, щоб забезпечити їх правильну роботу згідно з вимогами.
- Тестування Інтерфейсу Користувача: Спрямоване на оцінку інтуїтивності, зручності та візуальної привабливості інтерфейсу гри.
- Тестування Безпеки та Контролю Доступу: Має на меті виявити потенційні вразливості у захисті даних гри та систем контролю доступу.
- Тестування Відмовостійкості та Відновлення: Перевіряє здатність гри протистояти помилкам та відновлювати роботу після збоїв.
- Тестування Конфігурації: Оцінює сумісність гри з різними системними налаштуваннями та обладнанням.
- Тестування Інсталяції: Переконується, що процес встановлення гри проходить безперешкодно на підтримуваних платформах.

Стратегія Тестування:

На першому етапі проводиться комплексне системне тестування, яке базується на аналізі взаємодії користувача з інтерфейсом гри від моменту запуску. Метою цього етапу є ідентифікація потенційних недоліків у дизайні інтерфейсу та внесення пропозицій щодо його оптимізації для покращення користувацького досвіду.

На другому етапі, в процесі функціонального тестування, планується виявлення помилок, які впливають на ігровий процес, шляхом тестування нетипових ситуацій та перевірки основних ігрових сценаріїв.

#### 3.1. Типи тестування (Testing Types)

##### Функціональне тестування (Function Testing)

Функціональне тестування перевіряє, наскільки гра відповідає заданим вимогам. Цей метод зосереджений на перевірці результатів обробки даних, імітуючи реальне користування системою без врахування її внутрішньої структури. Він орієнтований на очікування та потреби користувачів. Детальний план тестування представлено у таблиці 3.1.

Таблиця 3.1 – Функціональне тестування

Мета випробування	Забезпечення відповідності функціональності гри заданим вимогам. Метою є переконатися, що всі функції працюють коректно, відповідають визначеним специфікаціям та задовольняють потреби користувачів. (використовується підхід «чорний ящик»).
Технічний прийом	Використання заздалегідь визначених сценаріїв та тестових випадків, що відтворюють різноманітні умови використання програми користувачами, без залучення інформації про внутрішню структуру програми. Основна увага приділяється тестуванню кожної функції гри на відповідність очікуваному результату.
Критерії завершення	Всі функції програми протестовано і підтверджено їхню правильну роботу в усіх тестових сценаріях. Відсутність помилок, що впливають на основні операції користувачів, і забезпечення, що всі виявлені дефекти були виправлені та перевірені.
Спеціальні рекомендації	Головний функціонал системи потрібно тестувати на різних ОС, різних екранах, на різному «залізі».

### Тестування інтерфейсу користувача (User Interface Testing)

Тестування інтерфейсу користувача — це процес тестування продукту інтерфейсу користувача для забезпечення його відповідності до специфікації (див. табл. 3.2).

Таблиця 3.2 – Тестування інтерфейсу користувача

Мета випробування	Перевірка на інтуїтивність, зручність використання і відповідність інтерфейсу користувача до встановлених вимог, а також ідентифікація можливих незручностей у взаємодії користувача з програмою.
Технічний прийом	Використовувати тестові сценарії, що наслідують реальну поведінку користувачів при роботі з інтерфейсом, та застосування евристичного аналізу для оцінювання зручності користування інтерфейсом..
Критерії завершення	Усі плановані тестові сценарії виконані без виявлення критичних помилок, забезпечена відповідність очікувань користувачів до фактичної роботи інтерфейсу, отримано позитивні відгуки від тестувальників щодо інтуїтивності та зручності використання..
Спеціальні рекомендації	Запрошувати до тестування різноманітні групи користувачів для охоплення широкого спектру можливих взаємодій з інтерфейсом. Використовувати інструменти для відстеження руху очей, щоб аналізувати, наскільки ефективно розміщені елементи інтерфейсу з точки зору користувача.

### Тестування Відмовостійкості та відновлення (Failover and Recovery Testing)

Цей вид тестування гарантує працездатність гри або збереження даних.

Відмово стійка система має:

- точки відновлення;
- локалізацію пошкоджень у пошкодженому компоненті;
- обмеження розповсюдження несправності;

- доступність режимів реверсії.

Таблиця 3.3 - Тестування відмовостійкості та відновлення

Мета випробування	Забезпечення стабільності та надійності гри, здатності системи ефективно реагувати на помилки та відновлювати свою роботу після збоїв або втручань ззовні.
Технічний прийом	Симуляція різноманітних збоїв та відмов, включаючи раптове вимкнення живлення, втрату мережевого з'єднання або збої програмного забезпечення. Моніторинг системи на предмет її здатності автоматично відновлювати функціональність без втрати даних користувача.
Критерії завершення	Усі симульовані сценарії збоїв та відмов пройдені, при цьому система продемонструвала здатність до швидкого відновлення роботи з мінімальними втратами даних. Записано час відновлення для різних типів збоїв.
Спеціальні рекомендації	Розробити комплексний план відновлення після збоїв, що включає автоматичне створення резервних копій даних гри та їх швидке відновлення. Проводити регулярні тренування команди технічної підтримки для ефективного реагування на критичні збої.

### Тестування інсталяції (Installation Testing)

Інсталяційне тестування зосереджено на перевірці процесів успішного встановлення, налаштування, а також оновлення або деінсталяції гри. У сучасному ігровому середовищі загальноприйнятою практикою є розгортання ігор через інсталяційні програми на ігрових платформах, які також вимагають ретельного тестування.

Таблиця 3.4 - Тестування інсталяції

Мета випробування	Забезпечити, що гра може бути успішно встановлена, налаштована, оновлена або видалена з системи користувача без помилок та перешкод.
Технічний прийом	Виконання серії тестів інсталяції через стандартні інсталяційні процедури, які включають використання інсталяторів на різних ігрових платформах, перевірка налаштувань за замовчуванням та користувацьких налаштувань, а також тестування процесів оновлення та видалення програми.
Критерії завершення	Усі процеси інсталяції, налаштування, оновлення та видалення пройдені без виявлення критичних помилок або збоїв. Програма успішно інсталюється та функціонує на підтримуваних операційних системах згідно з документацією.
Спеціальні рекомендації	Проведення тестів на різноманітних конфігураціях апаратного забезпечення та версіях операційних систем, щоб охопити можливі варіанти використання кінцевими користувачами. Врахування особливостей інсталяційних процедур на різних ігрових платформах для виявлення специфічних для платформи помилок.

## Інструменти (Tools)

Таблиця 3.5 – Інструменти

Процес	Інструмент
Створення тест кейсів	Гугл форма
Трекінг багів	Гугл таблиця
Виконання тест кейсів	Мануально
Структура проекту	Mind Map

## 4. Ресурси (Resources)

### 4.1. Ролі (Roles)

Таблиця 4.1 показує припущення щодо кадрового забезпечення проекту.

Таблиця 4.1 – Припущення кадрового забезпечення проекту

Працівник	Рекомендований мінімальний обсяг осіб	Конкретні обов'язки або коментарі
Тест-менеджер, менеджер з тестування	1	Забезпечує управління наглядом. Обов'язки: - технічна підтримка; - придбання відповідних ресурсів; - забезпечення управлінської звітності.
Проектувальник тестів	1	Визначення, пріоритетів, і реалізація тестів. Обов'язки: - створення плану тестування; - генерація тестових моделей; - оцінка ефективності тестових зусиль.
Тестувальник	1	Виконання тестів. Обов'язки: - виконання тестів; - журнал результатів; - відновлення в журналі реєстрації після помилок.
Тестовий системний адміністратор	1	Забезпечує тестове середовище і управління активами. Обов'язки: - адміністрування тестової системи управління; - встановлення і управління доступом до тест-системи.

### 4.2. Система (System)

Рисунок Г.13 – Тест-план сторінка 14

Таблиця 4.2 – Система

Мінімальні	Рекомендовані
ОС: 64-bit Windows 10	ОС: 64-bit Windows 10
Процесор: Intel Core i5-6600 (3.3Ghz), AMD Ryzen 5 1400 (3.2Ghz)	Процесор: Intel Core i7-8700 (3.2Ghz) or AMD Ryzen 5 3600 (3.6 Ghz)
Оперативна пам'ять: 8 GB	Оперативна пам'ять: 8 GB
Відеокарта: NVIDIA GeForce GTX 960 4GB, AMD Radeon RX 470 4GB	Відеокарта: NVIDIA GeForce 1080 Ti or AMD Radeon RX 5700 XT or INTEL Arc A770
Місце на диску: 30 GB	Місце на диску: 30 GB

## 5. Етапи проекту (Project Milestones)

Тестування має охоплювати тестові процедури для кожного виду випробувань, описаних у попередніх секціях. Важливо встановити конкретні фази проекту для забезпечення звітності про прогрес стану проекту.

Таблиця 5.1 – Етапи проекту

Цільове завдання	Обсяг робіт	Дата початку	Дата закінчення
План випробувань	5 год	25.02.2024	25.02.2024
Тест – дизайн	10 год	26.02.2024	01.02.2024
Реалізація випробувань	50 год	01.03.2024	21.03.2024

## 6. Кінцевий продукт (Deliverables)

### 6.1. Тестова модель

Таблиця 6.1 – Етапи проекту

Test Plan	Повний документ планування, який містить підхід, ресурси, графік.
Test Cases	Група передумов виконання, очікуваних виконання умов та результатів. Розроблюється для тестового сценарію.
Requirements Traceability Matrix	Матриця відповідності вимог використовується QA-інженерами для валідації покриття вимог щодо продукту тестами. Мета документу полягає в тому, щоб з'ясувати: – які вимоги покриті тестами, а які ні; – надлишковість тестів. □
Test Data	Дані, які існують до виконання тесту. Використовуються для виконання Test Cases.
Defect Report	Технічний документ, який містить в собі повний опис бага, що включає інформацію, як про сам баг, так і про умовах виникнення даного бага.
Test Summary Report	Документ високого рівня, який підсумовує проведені тестові дії.

### 6.2. Звіти з дефектів (Defect Reports)

#### Bug-report

## ДОДАТОК Г

Тези доповіді для науково-практичної інтернет-конференції

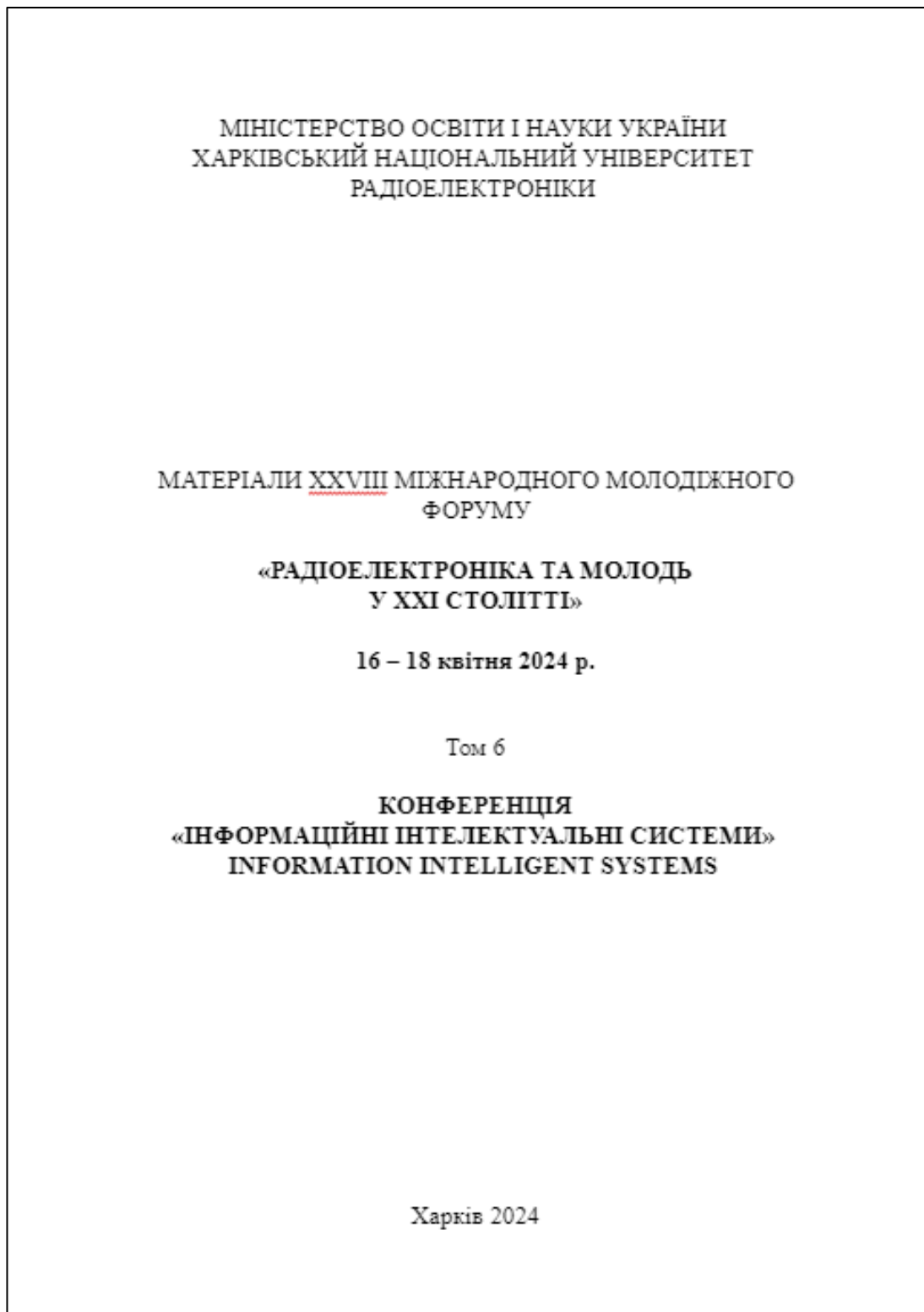


Рисунок Г.1 – Обкладинка збірника

- Гриб А. С., 714  
 Григор'єв О. В., 945  
 Гринишина С. О., 923  
 Гриньов С. А., 121  
 Гриньова О. С., 13, 16, 24  
 Гриньова О.С., 5, 7  
 Грішаєва А. М., 64  
 Громенко А. І., 158, 160  
 Груздо І. В., 116, 502, 846  
 Гулієв Н. Б., 508  
 Гуркін В. С., 894
- Д**
- Давиденко А. Л., 160  
 Данилов А. Д., 456  
 Данилейко С. І., 730  
 Двугрошев А. О., 369  
 Дегтяр В. Е., 544  
 Дейнеко А. О., 35  
 Дейнеко Ж. В., 948, 950  
 Дем'яненко М. С., 487  
 Демиденко С. О., 246  
 Демченко М. О., 493  
 Денисюк В. М., 336  
 Дергачова Д. К., 140  
 Деркач К. Ю., 481  
 Дехадрай Д. Р., 796  
 Дідуць О. П., 850  
 Добудько А. М., 902  
 Домніч Д. В., 840  
 Донець Д. С., 338  
 Драконова О. О., 348  
 Дробицький Д. С., 124  
 Дубок В. Ю., 425  
 Дуванов А. К., 630  
 Дудар З. В., 399  
 Дудка М. В., 794  
 Дукельська К.Б., 24  
 Дюжев М. Л., 162
- Є**
- Євланов М. В., 168, 275, 284  
 Євменкін Д. К., 164  
 Єгорова І. М., 925  
 Єлтишев П. І., 861  
 Єльчанінов Д. Б., 74, 80  
 Ємельянов А. В., 695  
 Єрохін А. Л., 326, 329  
 Єрохін М. А., 166  
 Єрошенко С. О., 317, 605
- Ж**
- Жаркіх С. С., 16  
 Жемчужний Р. І., 869  
 Женило К.О., 5  
 Жирко К. В., 158, 160  
 Жмур Д. М., 112
- З**
- Забийворота М. А., 892  
 Заворіна М.А., 30  
 Загнойко І. Ю., 421  
 Задніпровський Д. Б., 168  
 Задорожний А. Ю., 568  
 Запара О. С., 300  
 Заполочний А. Д., 170  
 Звєгінцев А. В., 532  
 Златкін С. С., 173
- І**
- Іванов В. Г., 581, 583, 618,  
 650, 667, 673, 704, 800  
 Іванов С. О., 130  
 Іванова А. І., 175  
 Іванова О. С., 44  
 Ігнатюк С. О., 160  
 Ільїн І. О., 810  
 Імангулова З. А., 133, 644,  
 663, 755, 796, 806, 812,  
 848, 855  
 Іпполітова В. Є., 920  
 Ісаєнко С. С., 177  
 Іткін Д. О., 180
- К**
- Казимов Л. Б., 183  
 Кайданюк Г. С., 450  
 Калайда Н. С., 697, 734, 746,  
 766, 784, 810, 892, 900  
 Калита Н. І., 738, 836  
 Калінін Д. В., 757  
 Калінін Д.В., 10  
 Калініченко О. Ю., 363  
 Кальний С. А., 865  
 Калюжний О. Д., 702  
 Камсюк Д. О., 830  
 Канінець А. А., 945  
 Кардаш Д. М., 52  
 Каряка В. В., 185  
 Кастиркін Д. Р., 880  
 Каук В. І., 353, 366, 428, 472,  
 493  
 Кащенко Ю. С., 656  
 Кириченко І. В., 112  
 Кирсанов О. О., 59  
 Кієнко Д. В., 187  
 Кієу Куанг Хієп, 753  
 Кікоть М. С., 189  
 Кісельгова М. С., 342  
 Кітов А. В., 54  
 Кіценко Ю. О., 435  
 Климова І. М., 205, 726, 882  
 Клішов М. Р., 886  
 Клочко Є. С., 654  
 Ключанський Є. Г., 194
- Коваленко А. І., 587, 589,  
 591, 595, 599, 601, 608,  
 620, 675, 871  
 Коваленко О. А., 635  
 Коваленко О. О., 933  
 Коваль О. О., 677  
 Ковальов І. М., 768  
 Ковальов М. М., 667  
 Козирев А. Д., 437  
 Козорог І. Г., 818  
 Колєндовська М. М., 933,  
 938  
 Колесник Л. В., 699  
 Коломоєць К. В., 710  
 Коломойцев П. А., 326  
 Комзолєв М. О., 786  
 Комін А. С., 102  
 Кондратьєв О. В., 914  
 Коновалова М. Д., 898  
 Константинов Б. С., 855  
 Копейчиков І. Ю., 197  
 Коптілов Н. С., 632  
 Корзун В. Р., 741  
 Коріненко В. Д., 842  
 Косенко Б. А., 413  
 Котелевєць К. А., 107  
 Котенко І. І., 571  
 Кошарний Є. Ю., 612  
 Кошель В.О., 42  
 Кравець Н. С., 345, 387, 405,  
 425, 440  
 Кравцов Д. О., 526  
 Кравченко В. Д., 744  
 Кравченко Є. О., 396  
 Кравченков Т. П., 583  
 Кривенко С. А., 59  
 Круц О. О., 900  
 Крюкова М. М., 925  
 Кубай Р. В., 475  
 Кудрявський Д. А., 681  
 Кудрявцева М. С., 10, 140,  
 183, 248, 757  
 Кузнецов Р. О., 340  
 Кузьміна П. О., 896  
 Куліш Є. І., 217  
 Кулішова Н. Є., 905, 907, 920  
 Кулішова Н.Є., 40  
 Кульмінський Я. К., 479  
 Купенко М. І., 200  
 Кучеренко Д., 931
- Л**
- Лавриненко Р. М., 91  
 Лавриненко С. Р., 86, 91  
 Лаврінєнко В. В., 712  
 Лановий О. Ф., 303, 541, 565  
 Ларченко Л. В., 290  
 Ларченко С. О., 565  
 Латинцев О. О., 581

Рисунок Г.2 –Алфавітний покажчик збірника

УДК 004.89:004.946

## МЕТОДИ ТА АЛГОРИТМИ БАЛАНСУВАННЯ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ІГОР ЖАНРУ FPS

Кісельгова М. Є.

Науковий керівник – ст. викл. Новіков Ю. С.

Харківський національний університет радіоелектроніки, каф. ПІ

м. Харків, Україна

e-mail: [marharyta.kiselhova@nure.ua](mailto:marharyta.kiselhova@nure.ua)

This research focuses on optimizing artificial intelligence (AI) for enemy behavior in FPS (First-Person Shooter) games to enhance the gaming experience. AI plays a pivotal role in creating realistic and engaging enemy characters, directly influencing game difficulty and player immersion. This study conducts a comparative analysis of existing AI balancing algorithms and methods used for enemy behavior, proposing an optimized approach based on the analysis of player preferences. By selecting the most relevant balancing parameters, this research aims to develop a more adaptive and challenging AI, contributing to a deeper and more captivating gameplay.

Штучний інтелект (ШІ) відіграє ключову роль у створенні реалістичних і цікавих ворожих персонажів в іграх жанру FPS (First-Person Shooter). Ефективність ШІ безпосередньо впливає на складність гри, рівень занурення і загальне враження гравця [1].

В якості прикладу для порівняння різних характеристик методів балансування та алгоритмів штучного інтелекту використовується власна розроблена гра у жанрі FPS «Tale of the Awakened».

Балансування ШІ в іграх жанру FPS можна здійснювати за різними параметрами, включно з агресивністю, швидкістю реакції, точністю стрільби та здатністю до навчання [2]. Грунтуючись на аналізі вподобань цільової аудиторії та специфіки ігрового процесу, було проведено порівняння характеристик основних методів балансування.

Таблиця 1 – Порівняльна характеристика методів балансування

Метод	Параметри для балансування	Переваги	Недоліки	Актуальність для жанру FPS
Динамічна складність	Рівень майстерності гравця, швидкість проходження рівнів, частота смертей	Автоматичне адаптування ШІ до рівня навичок гравця, покращення ігрового досвіду	Може зменшити відчуття досягнення у недосвідчених гравців, якщо зміни в складності не прозорі	Дуже висока

Продовження таблиці 1 – Порівняльна характеристика методів балансування

Метод	Параметри для балансування	Переваги	Недоліки	Актуальність для жанру FPS
Обмежене сприйняття	Дальність зору ворогів (ШІ), кут огляду, здатність чути шуми	Збільшує реалізм, імітуючи людські обмеження сприйняття	Може призвести до непередбачуваної поведінки ШІ	Висока
Самонавчання ШІ	Здатність до адаптації стратегій, зміна поведінкових патернів	ШІ може адаптуватися до ігрових стратегій користувача	Реалізація ШІ, що самонавчається, може бути надмірно складною	Середня

На основі цього аналізу можна зробити висновок, що динамічна складність являє собою найбільш універсальний і ефективний метод балансування ШІ в іграх FPS для розробки однією людиною.

Штучному інтелекту ворогів в іграх жанру FPS, з можливістю вільного переміщення гравця по карті, притаманні такі моделі поведінки [2]:

1. Патрулювання: ШІ слідує заданим маршрутом до виявлення гравця.
2. Пошук: Після втрати гравця з поля зору, ШІ шукає його в останньому відомому місці розташування.
3. Використання укриттів: ШІ шукає укриття від вогню гравця й атакує з безпечного місця.
4. Командні дії: ШІ координує дії з іншими ШІ для оточення або засідки на гравця.

У власній розробленій грі «Tale of the Awakened» були реалізовані дві моделі поведінки ШІ: патрулювання, пошук. Маршрут для патрулювання було зроблено за допомогою кривої (Spline), яку можна змінювати прямо у редакторі Unreal Engine 5. Зупинки були реалізовані через певний масив точок на кривій, які мають свою позицію (Time), та змінну, що відповідає за час, проведений ворогом на цій точці. Ще одним показником на зупинках є напрям, у якому буде стояти та дивитися ворог. Він був реалізований за допомогою динамічного компоненту стрілки, яка вказує на напрямок, що редагується разом з іншими даними масиву.

Для реалізації штучного інтелекту ворогів у грі «Tale of the Awakened» було проведено порівняльну характеристику алгоритмів та обрано оптимальний.

Таблиця 2 – Порівняльна характеристика алгоритмів ШІ

Алгоритм	Критерії порівняння	Обмеження	Придатність для жанру FPS
Дерева поведінки	Висока гнучкість і деталізація поведінки	Складність реалізації та підтримки	Дуже висока
Машини станів	Простота реалізації та розуміння	Обмежена гнучкість у складних ситуаціях	Висока
Генетичні алгоритми	Автоматична оптимізація та адаптація	Потребують багато часу і ресурсів на "навчання"	Середня
Нейронні мережі	Здатність до навчання та адаптації	Складність налаштування і вимога великого обсягу даних для навчання	Середня

На основі цієї порівняльної характеристики та з огляду на вимоги жанру FPS, де важливими критеріями є швидка адаптація до дій гравця і деталізована поведінка ворогів, найкращим вибором у розробці стали дерева поведінки. Цей алгоритм забезпечує високу гнучкість у моделюванні різноманітної поведінки ШІ, від простого патрулювання до складних тактичних дій у бою.

Розглянувши порівняльні таблиці, можна дійти висновку, що найкращим методом балансування ШІ для розробки інді гри у жанрі FPS є динамічна складність, а найкращим алгоритмом виявилися дерева поведінки. Ці підходи забезпечують збільшення реалізму та інтерактивності ігрового процесу.

Список використаних джерел:

1. O. Mazurova, O. Samantsov, O. Topchii and M. Shirokopetleva, A Study of Optimization Models for Creation of Artificial Intelligence for the Computer Game in the Tower Defense Genre, 2020 IEEE International Conference on Problems of Infocommunications. Science and Technology (PIC S&T), 2020, pp. 491-496, doi: 10.1109/PICST51311.2020.9468057.

2. Радіоелектроніка та молодь у ХХІ столітті. Т. 6 : Конференція "Інформаційні інтелектуальні системи" : матеріали 27-го Міжнар. молодіж. форуму, 10–11 трав. 2023 р. / М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. Харків, 2023. 275 с.

## ДОДАТОК Д

Тези доповіді для науково-практичної інтернет-виставки

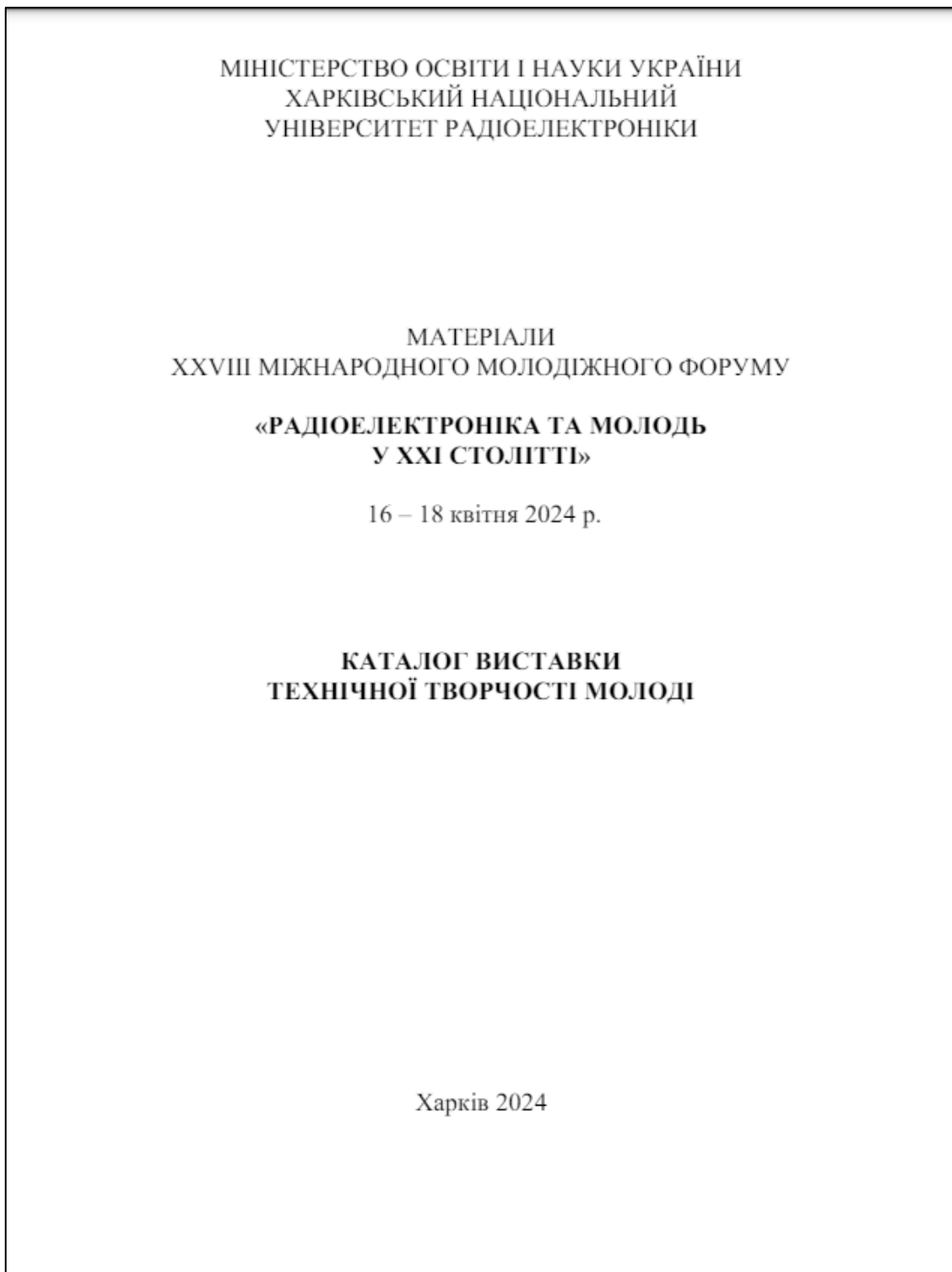


Рисунок Д.1 – Обкладинка збірника

## **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 проекті 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.

Рисунок Д.2 – Опис роботи для презентації

## ДОДАТОК Е

### Приклад програмного коду Блупринт AI\_CommonGoblin

```

/** Please add a class description */
UCLASS(Blueprintable, BlueprintType)
class AAI_CommonGoblin : public AAIController
{
    GENERATED_BODY()
public:
    /** Please add a function description */
    UFUNCTION(BlueprintCallable)
    void HandleSightSense(AActor* Actor, FAIStimulus Stimulus);
public:
    /** Please add a variable description */
    UPROPERTY(BlueprintReadOnly, VisibleAnywhere, Category="Default")
    TObjectPtr<UAIPerceptionComponent> AIPerception;
};

```

### Блупринт BP\_CommonGoblin

```

/** Please add a class description */
UCLASS(Blueprintable, BlueprintType)
class ABP_CommonGoblin : public ACharacter
{
    GENERATED_BODY()
protected:
    /** Please add a function description */
    UFUNCTION(BlueprintCallable)
    void WeldSkeletalComponent(USkeletalMeshComponent* Skeletal Mesh Component);

    /** Please add a function description */
    UFUNCTION(BlueprintCallable)
    void AddRandomTopClothes();

    /** Please add a function description */
    UFUNCTION(BlueprintCallable)
    void AddRandomBottomClothes();

    /** Please add a function description */
    UFUNCTION(BlueprintCallable)

```

```

void AddRandomWeapon(UChildActorComponent* WeaponComponent);

/** Please add a function description */
UFUNCTION(BlueprintCallable)
void AddRandomHelmet();

/** Please add a function description */
UFUNCTION(BlueprintCallable)
void AddRandomBracers();

/** Please add a function description */
UFUNCTION(BlueprintCallable)
void AddRandomLegArmor();
public:
    /** Please add a function description */
    UFUNCTION(BlueprintCallable)
    void Debug_Run(double Speed);

    /** Please add a function description */
    UFUNCTION(BlueprintCallable)
    void StartJump();

    /** Please add a function description */
    UFUNCTION(BlueprintCallable)
    void Attack();

    /** Please add a function description */
    UFUNCTION(BlueprintCallable)
    void AttackSlash();

    /** Please add a function description */
    UFUNCTION(BlueprintCallable)
    void Rage();
public:
    /** Please add a variable description */
    UPROPERTY(BlueprintReadOnly, VisibleAnywhere, Category="Default")
    TObjectPtr<UWidgetComponent> Widget;

    /** Please add a variable description */
    UPROPERTY(BlueprintReadOnly, VisibleAnywhere, Category="Default")
    TObjectPtr<UBP_DeathComponent_C> DeathComponent;

    /** Please add a variable description */
    UPROPERTY(BlueprintReadOnly, VisibleAnywhere, Category="Default")

```

```

TObjectPtr<UBP_FlammableComponent_C> FlammableComponent;

/** Please add a variable description */
UPROPERTY(BlueprintReadOnly, VisibleAnywhere, Category="Default")
TObjectPtr<UBP_HealthComponent_C> HealthComponent;

/** Please add a variable description */
UPROPERTY(BlueprintReadWrite, EditDefaultsOnly, Category="Default")
bool JumpStarted;

/** Please add a variable description */
UPROPERTY(BlueprintReadWrite, EditDefaultsOnly, Category="Default")
TObjectPtr<UABP_CommonGoblin_C> AnimInstance;

/** Please add a variable description */
UPROPERTY(BlueprintReadWrite, EditDefaultsOnly, Category="Default")
bool IsAttacking;

/** Please add a variable description */
UPROPERTY(BlueprintReadWrite, EditAnywhere, Category="Default")
TArray<FPause> PausePoints;

/** Please add a variable description */
UPROPERTY(BlueprintReadWrite, EditInstanceOnly, Category="Default")
TObjectPtr<ABP_AI_PatrolPath_C> BindedPath;
};

```

## БЛУПРИНТ UBTTask\_MoveRandomLocation

```

/** Please add a class description */
UCLASS(Blueprintable, BlueprintType)
class UBTTask_MoveRandomLocation : public UBTTask_BlueprintBase
{
    GENERATED_BODY()

public:
    /** Please add a variable description */
    UPROPERTY(BlueprintReadWrite, EditAnywhere, Category="Default")
    float Radius;
};

```

## Блупринт BP\_AI\_PatrolPath

```
/** Please add a class description */
UCLASS(Blueprintable, BlueprintType)
class ABP_AI_PatrolPath : public AActor
{
    GENERATED_BODY()
public:
    /** Please add a variable description */
    UPROPERTY(BlueprintReadOnly, VisibleAnywhere, Category="Default")
    TObjectPtr<USplineComponent> Path;

    /** Please add a variable description */
    UPROPERTY(BlueprintReadOnly, VisibleAnywhere, Category="Default")
    TObjectPtr<USceneComponent> DefaultSceneRoot;

    /** Please add a variable description */
    UPROPERTY(BlueprintReadWrite, EditDefaultsOnly, Category="Default")
    TArray<FPause> PausePoints;
};
```