

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет \_\_\_\_\_ комп'ютерних наук (або центр післядипломної освіти, або навчально-науковий центр заочної форми навчання)  
(повна назва)

Кафедра \_\_\_\_\_ програмної інженерії  
(повна назва)

## КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти \_\_\_\_\_ перший (бакалаврський)

Ігровий програмний застосунок у жанрі 2d Action-Adventure  
з елементами RPG  
(тема)

Виконав:  
студент 4 курсу, групи ПЗПІ-20-9

Кайданюк Г. С.  
(прізвище, ініціали)

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

Тип програми освітньо-професійна

Освітня програма Програмна інженерія  
(повна назва освітньої програми)

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

Допускається до захисту  
Зав. кафедри

\_\_\_\_\_  
(підпис)

З.В.Дудар  
(прізвище, ініціали)

2024 р.

## Харківський національний університет радіоелектроніки

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

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

« \_\_\_\_ » \_\_\_\_\_ 2024 р.

### ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові \_\_\_\_\_ Кайданюку Герману Сергійовичу \_\_\_\_\_  
 (прізвище, ім'я, по батькові)

1. Тема роботи Ігровий програмний застосунок у жанрі 2d Action-Adventure \_\_\_\_\_ з  
 елементами RPG \_\_\_\_\_

Затверджена наказом по університету від 20.05.2024р. № 471 Ст \_\_\_\_\_

2. Термін подання студентом роботи до екзаменаційної комісії 6.06.2024 \_\_\_\_\_

3. Вихідні дані до роботи Спроекувати та розробити ігровий програмний застосунок у жанрі 2d Action-Adventure RPG з усіма необхідними для повноцінного ігрового процесу механіками. \_\_\_\_\_

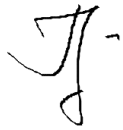
4. Перелік питань, що потрібно опрацювати в роботі

Вступ, аналіз предметної галузі, формування вимог до програмної системи, архітектура на проектування програмного забезпечення, опис прийнятих програмних рішень, тестування програмного забезпечення, впровадження програмного забезпечення. \_\_\_\_\_

## КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної галузі	10.03.2024	<i>виконано</i>
2	Створення специфікації ПЗ	17.03.2024	<i>виконано</i>
3	Проектування ПЗ	24.03.2024	<i>виконано</i>
4	Розробка ПЗ	10.05.2024	<i>виконано</i>
5	Тестування ПЗ	14.05.2024	<i>виконано</i>
6	Оформлення пояснювальної записки	20.05.2024	<i>виконано</i>
7	Підготовка презентації та доповіді	22.05.2024	<i>виконано</i>
8	Попередній захист	1.07.2024	<i>виконано</i>
9	Нормоконтроль, рецензування	1.07.2024	<i>виконано</i>
10	Здача роботи у електронний архів	5.07.2024	<i>виконано</i>
11	Допуск до захисту у зав. кафедри	6.07.2024	<i>виконано</i>

Дата видачі завдання 15 квітня 2024р.

Студент (ка)  Кайданюк Г. С.  
(підпис)

Керівник роботи \_\_\_\_\_ ст. викл. Кафедри ПІ Новіков Ю. С.  
(підпис) (посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи бакалавра: 63 стор., 38 рис., 1 табл., 14 джерел.

ДВОВИМІРНА ГРА, ІГРОВИЙ ПРОГРАМНИЙ ЗАСТОСУНОК, UNREAL ENGINE, ІГРОВА ІНДУСТРІЯ, PAPER ZD, PAPER 2D

Об'єкт розробки – ігровий програмний застосунок у жанрі 2d Action-Adventure з елементами RPG.

Мета розробки – ігровий дизайн та створення ігрового програмного застосунку.

Метод рішення – середовище розробки Unreal Engine 5, мови програмування C++ та технологія візуального програмування Blueprints.

У результаті розробки створено ігровий програмний застосунок у жанрі 2d Action-Adventure з елементами RPG.

2D GAME, GAME SOFTWARE, UNREAL ENGINE, GAME INDUSTRY, PAPER ZD, PAPER 2D

The object of development is a game software application in the genre of 2d Action-Adventure with RPG elements.

The solution method is the Unreal Engine 5 development environment, the C++ programming language, and visual scripting technology Blueprints.

As a result of the development, the game software application in the genre of 2d Action-Adventure with RPG elements was created.

Я, Кайданюк Герман Сергійович, студент гр. ПЗП-20-9, здобувач вищої освіти на першому (бакалаврському) рівні кафедри «Програмна інженерія»,

заявляю: моя кваліфікаційна робота на тему «Ігровий програмний застосунок у жанрі 2d Action-Adventure з елементами RPG», що буде представлена до екзаменаційної комісії для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу EIAr KhNURE. Усі запозичення з друкованих та електронних джерел мають відповідні посилання.

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

## ЗМІСТ

Вступ.....	8
1 Аналіз предметної галузі.....	9
1.1 Аналіз предметної галузі.....	9
1.2 Виявлення та вирішення проблеми.....	15
1.3 Постановка задачі.....	16
1.3.1 Постановка задачі.....	16
1.3.2 Цільова аудиторія.....	17
2 Формування вимог до програмної системи.....	18
2.1 Постановка мети.....	18
2.2 Загальний опис.....	18
2.3 Припущення та залежності.....	20
3 Архітектура та проектування програмного забезпечення.....	21
3.1 UML проектування.....	21
3.2 Проектування архітектурі ПЗ.....	24
3.3 Приклади цікавих алгоритмів та методів.....	26
3.4 Створення UI/UX.....	27
4 Опис прийнятих програмних рішень.....	33
4.1 Вибір інструментів програмної реалізації.....	33
4.2 Опис модуля головного персонажу.....	33
4.3 Опис модуля ворогів.....	36
4.4 Опис системи здібностей.....	38
5 Тестування програмного забезпечення.....	40
6 Впровадження програмного забезпечення.....	43
6.1 Наукове впровадження проекту.....	43
6.2 Практичне впровадження проекту.....	43
Висновки.....	44
Перелік джерел посилання.....	45
Додаток А. Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ.....	47

Додаток Б. Слайди презентації.....	48
Додаток В. Приклад програмного коду.....	54
Додаток Г. Тези доповіді для науково-практичної інтернет-конференції.....	57
Додаток І. Анотація для науково-практичної інтернет-виставки.....	62

## ВСТУП

Ігрова індустрія – це один з найбільших ринків програмного забезпечення у світі. Через пандемію тенденція зростання ігрового ринку дуже збільшилось, та все більше людей продовжують цікавитися цим видом розваг. Багато великих компаній витрачають мільйони на розробку та впровадження.

Серед великого набору різноманітних жанрів ігор жанр Action-Adventure з елементами RPG продовжує стабільно утримувати свою популярність. Серед представників жанру можна відзначити серію ігор “Dark Souls”, серію ігор “Monster Hunter” та багато інших.

Завдяки популярності жанру та ігрової індустрії в цілому кваліфікаційну роботу бакалавра, що виконується, можна вважати актуальною.

Ціль роботи з кваліфікаційної роботи бакалавра – розробка ігрового програмного застосунку у жанрі 2d Action-Adventure з елементами RPG. Можливою сферою використання даного застосунку може бути розміщення в онлайн-сервісі цифрового розповсюдження комп’ютерних ігор, наприклад, Steam.

Робота щільно пов’язана з навчальними дисциплінами з ігрового дизайну кафедри програмної інженерії.

## 1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

### 1.1 Аналіз предметної галузі

Предметна галузь ігрового програмного застосунку у жанрі 2d Action-Adventure з елементами RPG належить до ігрової промисловості. Хоча ігрова індустрія відносно нова, її швидкий розвиток кожного року збільшує потребу в нових ігрових програмних застосунках [1]. Перші комп'ютерні ігри почали з'являтися в середині 20 століття в якості університетського експерименту. Це була невелика гра, яка працювала на пристрої з ЕЛТ-трубкою.

Першими промисловими ігровими застосунками були невеликі аркадні ігри для автоматів. Їх відмінною рисою був нетривалий, але насичений ігровий процес. Згодом, разом з ігровою індустрією розвинулись і жанри комп'ютерних ігор. Так почали з'являтися такі жанри як Adventure, Action та RPG. Ігрова індустрія потерпіла великих змін за весь час свого існування та продовжує еволюціонувати. На рисунку 1.1 можна побачити частку номінального внутрішнього валового продукту ігрової індустрії в залежності від держави на 2022 рік.

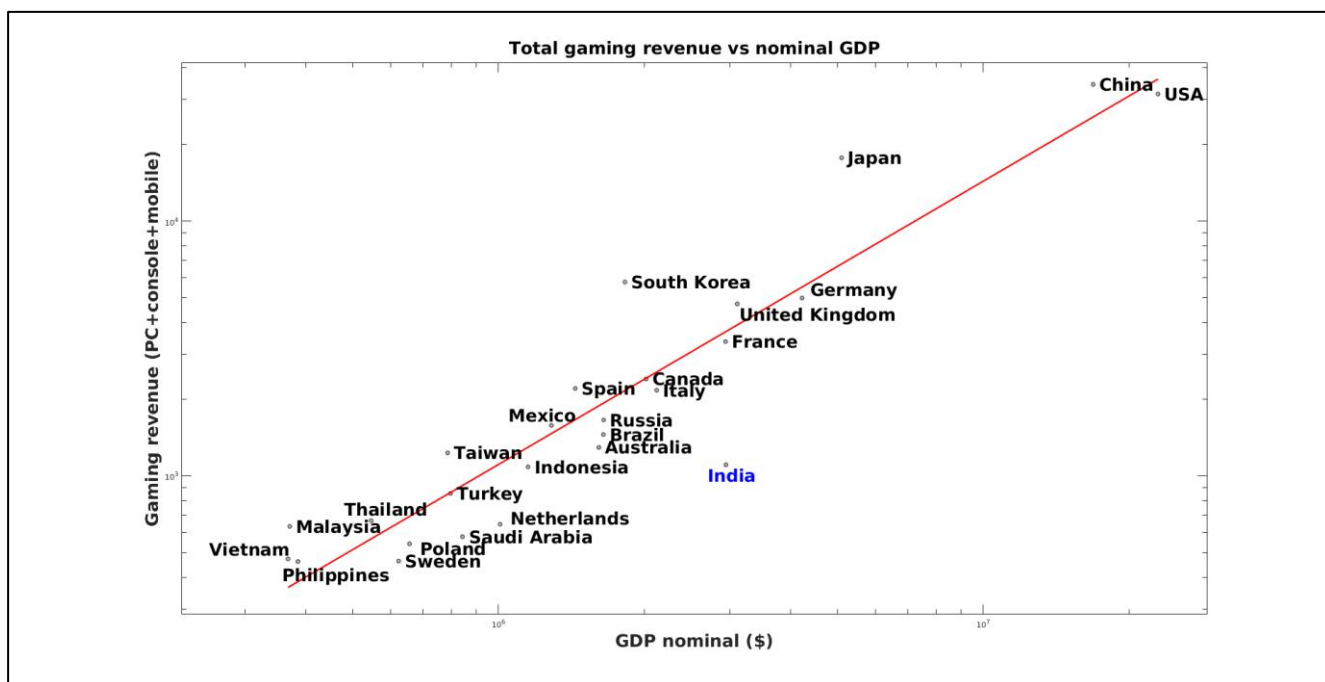


Рисунок 1.1 – частка номінального внутрішнього валового продукту ігрової індустрії в залежності від держави (за даними [2])

З кожним роком все більша кількість людей починають цікавитися іграми, тому збільшується потреба в розробниках та ігрових дизайнерів. За своєю суттю, ігри можна віднести до мистецтва, тому кожен новий розробник здатний додати щось нове у світ ігор [3].

Жанр 2d Action-Adventure з елементами RPG складається з декількох аспектів [4]. Під 2d мається на увазі вимірність елементів гри, будь то персонажі або об'єкти. Також у 2d іграх зазвичай гравець бачить ігровий світ від другого обличчя або збоку. Завдяки виду від другого обличчя, зазвичай нема потреби власноруч керувати камерою. Ігровий додаток власноруч рухає камеру разом з рухом головного персонажа, хоча, в деяких іграх гравець має можливість трохи "відсунути" камеру в певний бік. Також вид від другого обличчя дозволяє побачити більше інформації стосовно ігрового оточення, бо гравець водночас може побачити, що знаходиться з усіх боків від ігрового персонажа.

Взагалі, двовимірні ігри є гарною альтернативою тривимірним, бо дозволяють змінити ігровий досвід водночас зі зменшенням часу на розробку.

Під Action-Adventure мається на увазі суміш двох жанрів: Action та Adventure. Кожен з жанрів додає до гри свої механіки.

Жанр Action привносить в ігри фізичні випробовування. Тобто деякі перешкоди, рішення яких залежить від фізичних характеристик гравця. До фізичних характеристик гравця відносяться зір, швидкість реакції, координація руху. Комп'ютерні ігри з жанром Action відрізняються своєю динамікою та складністю, бо, зазвичай, у випадку коли гравець не справляється з випробовуванням його ігровий персонаж вмирає та втрачає деяку частину ігрового прогресу. В залежності від задуму розробників, перешкодами для гравця можуть бути вороги, яких контролює штучний інтелект, інші гравці або пастки, що розташовані в ігровому світі [5].

У випадку, коли ворогами є штучний інтелект, велику роль відіграє їх дизайн. Гравець завжди повинен мати шлях для того, щоб подолати ворогів. Зазвичай у кожного ворога є слабкість, яку гравець повинен знайти та застосувати на свою користь. Для того, щоб гравцю було легше визначити слабкі сторони ворога

існують архетипи супротивників. Архетип у випадку ворогів в іграх – це набір здібностей, які мають супротивники. Серед таких архетипів можуть бути вороги, які рухаються в бік гравця та завдають шкоди в ближній дистанції, або вороги, які тримаються на відстані від гравця та завдають шкоди поодаль. Архетипи можна доповнювати, змішувати або змінювати в залежності від необхідного ігрового досвіду.

Якщо ворогом є інший гравець, дуже важливий баланс, який забезпечить рівні шанси на перемогу обом сторонам.

Також в якості перешкод можуть бути пастки в ігровому світу, вони можуть мати один або декілька шляхів проходження в залежності від стилю гри користувача.

Іншим значним аспектом жанру Action є здібності ігрового персонажа. Набір здібностей дозволяє гравцю перемагати перешкоди різними шляхами. Серед здібностей можуть бути такі, які дозволять завдавати шкоди ворогам або пересуватися по ігровому світу. Також здібності можуть дозволяти гравцеві потрапити в ті частини ігрового світу, куди він раніше потрапити не міг.

Зазвичай кожна здібність збалансована таким чином, щоб не існувало той, що домінує, та гравець мусив використовувати усі інструменти, які йому надає гра. У кожній здібності є свої сильні та слабкі сторони, які зрозумілі для гравця, а результат їх використання передбачуваний для користувача [6].

Жанр Adventure привносить в ігри історію та інтелектуальні випробовування. Тобто деякі перешкоди, рішення яких залежить від інтелектуальних здібностей гравця. В якості перешкод зазвичай виступають головоломки або завдання. Ще одним з аспектів жанру Adventure є дослідження ігрового світу.

В залежності від бажаного ігрового досвіду акцент може бути зроблений або на складні логічні головоломки, або на великий та насичений ігровий світ, в будь-якому з перелічених випадків велику роль грає оповідання та атмосфера гри, яка занурює гравця у всесвіт гри.

Іншим важливим аспектом жанру Adventure є різноманітна кількість другорядних персонажів та діалогів, які дозволяють рухати сюжет гри. Існує декілька видів сюжету, серед яких лінійний та нелінійний [7].

У випадку лінійного сюжету гравець ніяк не впливає на події гри. Лінійний сюжет дозволяє дати гравцю той досвід, який був замислений ігровим дизайнером. Через відсутність альтернативних шляхів у розробників є більше часу та ресурсів на те, щоб більш ретельно продумати основний сюжет.

У випадку нелінійного сюжету гравець має змогу впливати на рух подій у грі в залежності від свого вибору. Розробка нелінійного сюжету коштує набагато більше ресурсів, так як необхідно не лише придумати альтернативні шляхи сюжету, але також підібрати альтернативи так, щоб більшість гравців змогла знайти підходящий для себе. Завдяки додатковим можливостям гравець може ще більше зануритися у світ гри та більш асоціювати себе з головним персонажем або відігравати якусь роль. Також нелінійний сюжет додає реграбельність гри, бо після одного проходження залишається можливість пройти гру ще раз, але виконуючи інші дії та отримуючи інший варіант сюжету.

Існує декілька підходів до розгалуження сюжету. Одним з підходів є створення непересічних сюжетних ліній. Розгалуження до конкретної лінії відбувається після вибору деякого варіанту діалогу або іншої дії в грі.

Іншим підходом до розгалуження сюжету є відділення а потім поєднання сюжету в критичній точці, тобто в одному з найважливіших сюжетних моментів. У випадку з частково розгалуженим сюжетом у гравця менший вплив на рух сюжету, але для реалізації цього варіанту потребується менше ресурсів на відміну від непересічних сюжетних ліній.

Ще одним шляхом розгалуження сюжету є створення єдиної сюжетної лінії з розгалуженням кінцівок в залежності від дій гравця. Відділення сюжетних кінцівок потребує менше часу аніж частково або повністю розгалужений сюжет але надає та надає менше всього впливу діям гравця.

Поєднання відмінних рис жанрів Adventure та Action дозволяє значно розширити ігровий досвід гравця. Ігри жанру Action-Adventure мають елемент

дослідження ігрового світу, сутички з ворогами та головоломки, ці елементи додають різноманітності грі.

Елементи RPG можуть додавати до головного персонажа цифрові характеристики та розвиток, який зазвичай контролює гравець. Серед цифрових характеристик можуть бути показники як очки здоров'я, урону, сили, інтелекту. Елементи RPG дозволяють гравцю налаштувати персонажа під свій стиль гри.

Збільшення характеристик зазвичай відбувається після підвищення рівнів головного персонажу. Рівень може збільшуватися після подолання ворогів або виконання завдань. Також на характеристики може впливати екіпірування персонажу.

Ще одним елементом RPG є система завдань. Гравець може взяти завдання у другорядного персонажу, за виконання якого він зможе отримати різноманітні нагороди.

В ігровій промисловості популярні жанри часто змінюються, так з появою більш сучасних технологій жанр Action-Adventure замінив жанр Adventure, бо надавав більше можливостей для гравця. Також продовжують з'являтися нові жанри, наприклад, Battle Royale. Але навіть на фоні інших варіантів, жанр Action-Adventure залишається популярним.

Одним з найпопулярніших представників Action-adventure ігор з елементами RPG є "The Witcher 3: Wild Hunt", ця гра отримала значну популярність завдяки великому відкритому ігровому світу та нелінійному сюжету.

Action складова гри визначається наявністю бойової системи та великої різноманітності ворогів, багато з яких мають свої особливості та сильні сторони. Упродовж усієї гри гравцю доводиться перемагати складні випробовування, які потребують швидкої реакції.

Adventure складова гри визначається наявністю інтерактивного сюжету та елементу дослідження. Насичений ігровий світ має багато підземель та будівель, які гравець може відвідати впродовж свого проходження. Також гра має багато загадок та головоломок, які іноді можуть впливають на рух оповідання.

RPG складова гри визначається наявністю системи рівнів та здібностей, які гравець може відкривати та вдосконалювати в ході гри. Також у грі великий набір предметів екіпірування, які впливають на захист, атаку та здібності головного персонажу. Меню здібностей можна побачити на рисунку 1.2.



Рисунок 1.2 – меню здібностей гри “The Witcher 3: Wild Hunt” (за даними [8])

На зображенні зліва можна побачити варіанти посилень, які гравець може обирати за очки навичок, які здобуваються при отриманні нового рівня. Справа можна побачити комірки, в які можна помістити відкриті посилення задля отримання додаткових бонусів. Загалом, “The Witcher 3: Wild Hunt” надає гравцю можливість налаштування головного персонажу в залежності від його стилю гри, це не тільки поліпшує ігровий процес, але й надає можливість повторного проходження, використовуючи інші варіанти розвитку головного персонажу.

“The Witcher 3: Wild Hunt” була визнана грою року завдяки вдалому поєднанню різних ігрових механік та сюжетних рішень. Серед слабких сторін гри можна помітити погану та одноманітну механіку бою. На рисунку 1.3 можна побачити як виглядає гра.

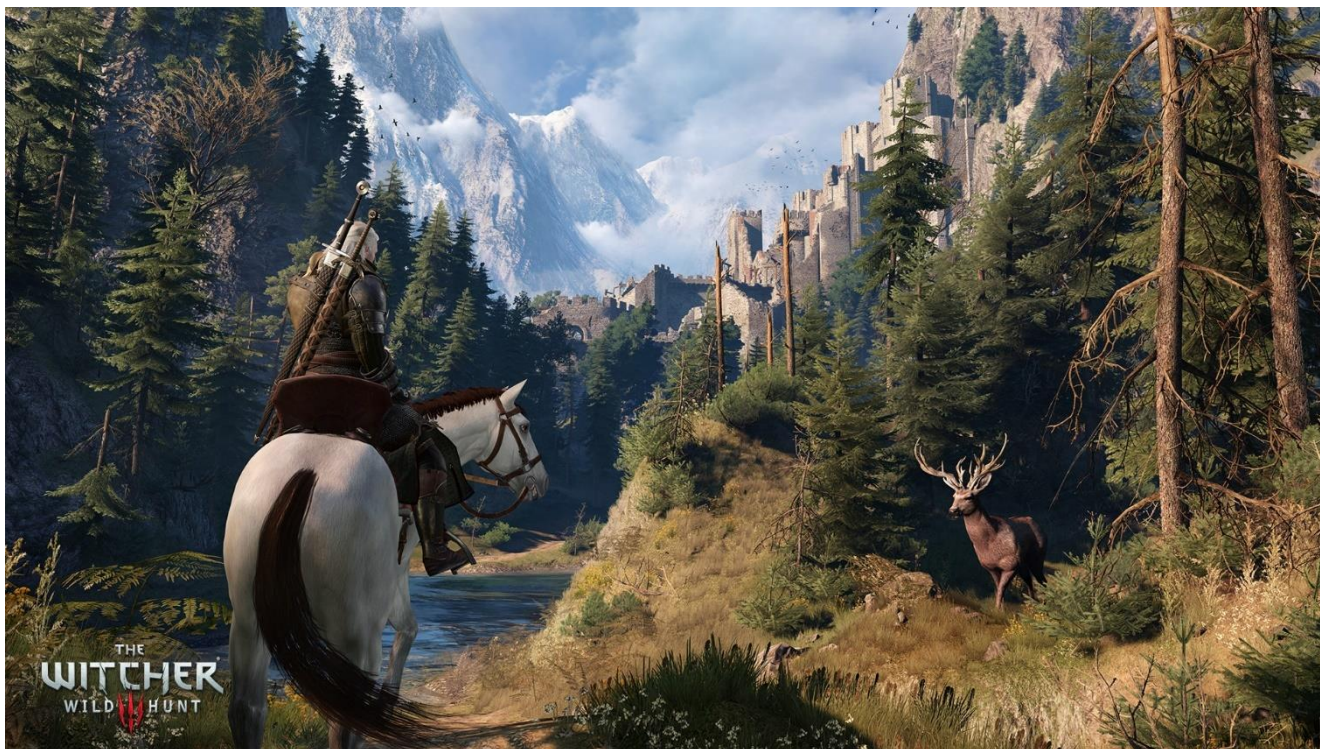


Рисунок 1.3 – Action-adventure гра з елементами RPG “The Witcher 3: Wild Hunt”  
(за даними [8])

Ігрова промисловість та жанр Action-adventure з елементом RPG продовжать розвиватися разом с технологіями та обладнанням. Більш потужні системи дозволять реалізовувати складні механіки та ворогів, саме тому в найближчому майбутньому цей жанр продовжить бути актуальним та популярним.

## 1.2 Виявлення та вирішення проблем

З часом все більша кількість людей цікавляться ігровими програмними застосунка, через це збільшується попит у нових іграх та ідеях. Ігровий програмний застосунок у жанрі 2d Action-Adventure з елементами RPG може задовільнити збільшеним потребам та надати альтернативу іншим сучасним іграм. В залежності від інтересів гравців можуть бути вирішені наступні проблеми:

- балансування ігрового процесу. Поширеною проблемою багатьох ігор є поганий баланс, через нього гра може бути або занадто складною, або занадто легкою. Необхідне ретельне балансування усіх ігрових механік задля досягнення цікавого

ігрового досвіду та складності. Для балансування ігрового процесу потрібне тестування різних аспектів ігрового застосунку та того, як ці частини працюють разом;

- оптимізація застосунку. Багато сучасних ігор потребують наявності потужних пристроїв. Завдяки рішення використувати двовимірну графіку зменшується потрібність в великих обсягах апаратних ресурсів;

- сюжетна складова гри. Однією з потенціальних проблем є створення та інтеграція сюжету в ігровий процес. Задля вирішення цієї проблеми необхідно ретельно сюжет та наративну складову ігрового застосунку.

Кожна з описаних проблем має великий вплив на загальний досвід користувача та повинна бути вирішена в ході проектування та розробки для забезпечення високої якості ігрового програмного застосунку.

### 1.3 Постановка задачі

#### 1.3.1 Постановка задачі

Задача даної роботи полягає в проектуванні та розробці ігрового програмного застосунку в жанрі 2d Action-Adventure з елементами RPG. Ігровий застосунок повинен відповідати усім зазначеним вимогам та надавати можливість гравцю отримати повноцінний ігровий досвід.

Основні задачі ігрового програмного застосунку включають:

- інтерфейс. Ігровий програмний застосунок повинен забезпечувати користувачів зручним інтерфейсом;

- коректна робота. Ігровий програмний застосунок повинен коректно та передбачувано працювати незалежно від дій користувача;

- ігровий процес. В ігровому програмному застосунку повинні бути реалізовані усі основні ігрові механіки;

- швидкодія. Ігровий програмний застосунок повинен оптимально використовувати системні ресурси та працювати з оптимальною частотою кадрів на більшості сучасного апаратного забезпечення.

Кожна з описаних задач має ітеративний підхід реалізації та потребує ретельного дослідження та проектування. Кожна ітерація повинна супроводжуватися відповідними тестами задля виявлення та усунення потенційних помилок. Також ігровий додаток повинен бути масштабованим та оптимізованим щодо використаних ресурсів.

### 1.3.2 Цільова аудиторія

Гра призначена для аудиторії віком від 17 років через наявність сцен насильства з перевагою чоловічої аудиторії. Верхньої межі віку немає. Гра розрахована на гравців, яким цікаві ігри з акцентом на динамічну бойову систему.

## 2 ФОРМУВАННЯ ВИМОГ ДО ПРОГРАМНОЇ СИСТЕМИ

### 2.1 Постановка мети

Метою роботи є розробка ігрового програмного застосунку у жанрі 2d Action-Adventure з елементами RPG. Застосунок спрямований на створення захоплюючого та інтерактивного ігрового досвіду, який має в собі аспекти відповідних жанрів.

Процес розробки включає в себе створення ідеї, проектування ігрового процесу, прототипування та реалізація інтерфейсу, кодування ігрового застосунку, тестування, балансування та впровадження гри.

Отриманий ігровий програмний застосунок повинен бути модульним та мати можливість подальшого розширення або змінення функціоналу або графічних елементів. Усі модулі гри повинні коректно працювати на операційних системах Windows. Усі графічні елементи повинні коректно відображатися на екранах різної розподільної здатності.

### 2.2 Загальний опис

У результаті кваліфікаційної роботи бакалавра необхідно створити програмний застосунок, що матиме в собі набір станів. До набору станів входить: головне меню, меню налаштувань, основний ігровий процес та меню паузи. Користувач повинен мати змогу вільно переходити від одного стану ігрового застосунку до іншого.

Головне меню повинно надавати доступ до основного ігрового процесу та меню налаштувань. Це перше, що бачить користувач при відчиненні гри. Також з головного меню можна зачинити додаток.

Меню налаштувань повинно давати функціонал для змінення та збереження основних параметрів ігрового застосунку.

Меню паузи необхідно для призупинення основного ігрового процесу та переходу до налаштувань або головного меню.

Основний ігровий процес повинен надавати гравцю основний функціонал та всі реалізовані ігрові механіки. Далі перераховані основні вимоги стосовно різних аспектів ігрового програмного застосування.

Технології та фреймворки:

- в якості основного ігрового двигуна обраний Unreal Engine 5;
- для роботи з двовимірною графікою використовується вбудована в двигун система Paper 2D;
- для роботи з анімаціями використовується плагін PaperZD;
- в якості мови програмування використовуються C++ та система візуального скрипту Blueprint.

Ігрові механіки:

- реалізація механіки віку персонажу, який збільшується після використання здібностей;
- реалізація системи здібностей, орієнтованої на комбінації;
- реалізація системи наборів заклинань, де в кожному наборі знаходиться декілька здібностей;
- реалізація бойової системи;
- реалізація переміщення по локаціям.

Інтерфейс:

- створення головного меню, меню налаштувань, екрану програшу та меню паузи;
- створення інтерфейсу основного ігрового процесу;

Масштабованість:

- забезпечення можливості зміни ігрового балансу та характеристик персонажів;
- забезпечення можливості зміни ігрових механік та додавання нових.

Документація та тестування:

- створення інструкції користувача із загальною інформацією щодо ігрового управління;

– проведення тестування ігрового програмного застосунку та опис баг-репорту.

Ігровий програмний застосунок у жанрі 2d Action-Adventure з елементами RPG складається із застосунку для персонального комп'ютера. Застосунок має працювати на операційній системі Windows з мінімальною версією Windows 7.

### 2.3 Припущення та залежності

Під час проектування та розробки ігрового програмного забезпечення розглядаються такі припущення та залежності:

– вихідний проект може бути відчинений з використанням рушію, версія якого менша за Unreal Engine 5.0. У даному випадку деякі модулі ігрового програмного застосунку та графічні елементи можуть поводити себе некоректно;

– ігровий програмний застосунок може бути відчинений на несумісній операційній системі. У даному випадку гра буде поводити себе непередбачувано;

– гра може бути запущена на малопотужному апаратному забезпеченні. У даному випадку може спостерігатися мала частота кадрів.

Означені припущення необхідно мати на увазі для коректної та зручної експлуатації ігрового програмного забезпечення.

## 3 АРХІТЕКТУРА ТА ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 3.1 UML проектування

Після аналізу та моделювання потреб користувача щодо функціоналу ігрового програмного застосунку у жанрі 2d Action-Adventure з елементами RPG була обрана діаграма використання (Use Case Diagram) для опису та визначення основної взаємодії користувача з системою. На рисунку 3.1 можна побачити розроблену діаграму.

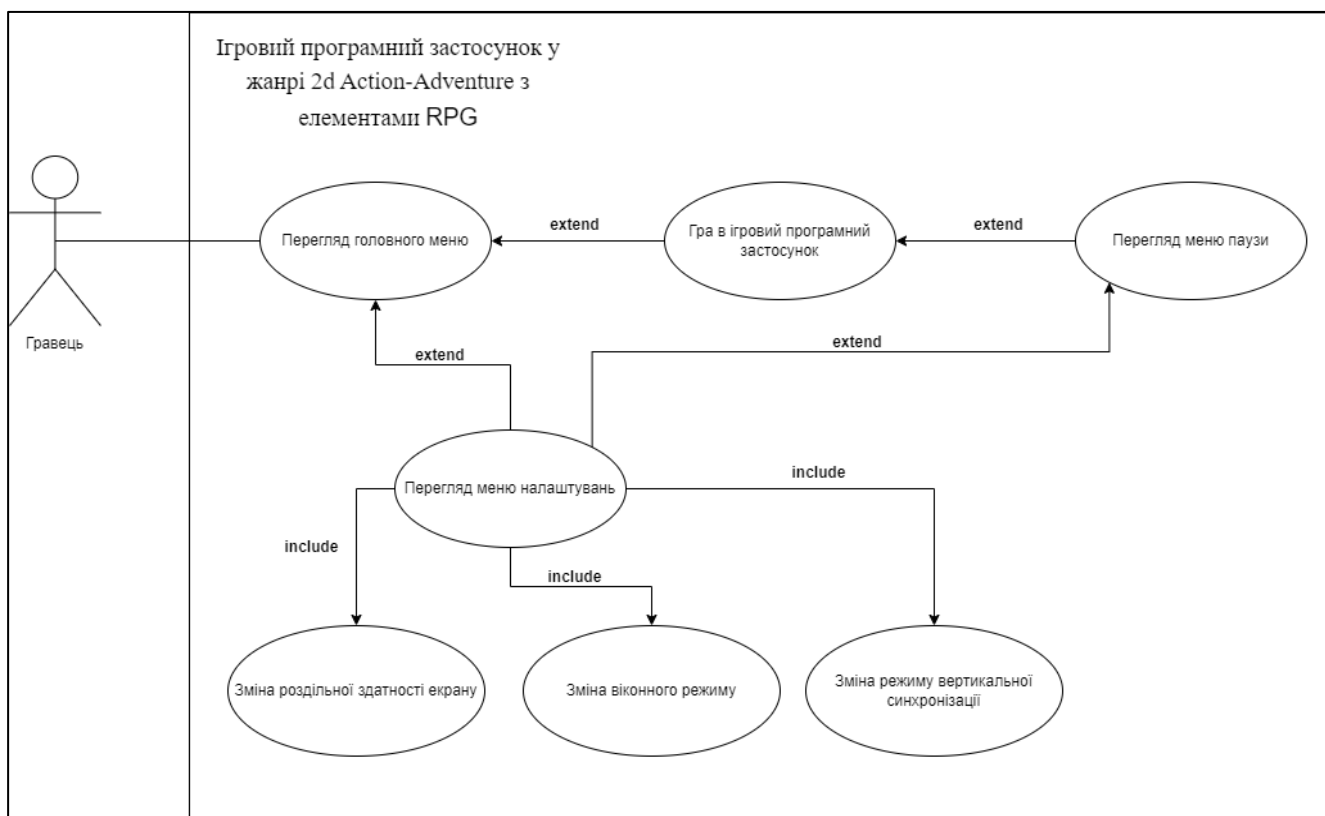


Рисунок 3.1 – Use-case діаграма ігрового застосунку (рисунок виконаний самостійно)

Гравець, зображений на діаграмі, – це основний користувач ігрового застосунку. Після запуску додатку користувач бачить головне меню гри.

З головного меню користувач може перейти до меню налаштувань та змінити такі налаштування застосунку як роздільна здатність екрану, віконний режим та режим вертикальної синхронізації.

Також з головного меню можна перейти до основного ігрового процесу, в даному випадку гравець зможе керувати своїм ігровим персонажем.

У будь-який час ігрового процесу користувач може перейти до меню паузи, у даному випадку ігровий процес призупиняється та з'являється змога вийти до головного меню або меню налаштувань.

В ігровому програмному застосунку передбачена система здібностей, які користувач зможе використовувати.

Система здібностей повинна працювати наступним чином: під час ігрового процесу будуть відбуватися деякі події, в ході яких у головного персонажу будуть з'являтися набори зі здібностями.

Кожен набір зі здібностями має 3 заклинання, які головний персонаж може використати ціною свого віку. Кожне заклинання має свою анімацію, яку необхідно програвати під час його використання. Кожен набір здібностей має свою іконку та іконки усіх заклинань, які необхідно динамічно додавати до інтерфейсу гри.

Необхідно розробити набори зі здібностями, які можна інтегрувати до головного персонажу. Потрібно створити базові класи, які будуть надавати функції для використання здібностей, які будуть викликатися під час події натискання на відповідно клавішу в класі головного персонажа. Оскільки у наборі заклинань ми не знаємо, що саме буде відбуватися під час використання здібності, для здібностей повинен існувати базовий клас, інтерфейсом котрого буде користуватися набір здібностей. Подібна архітектура дозволяє досягти більшого рівня гнучкості та поліморфізму, бо усі реалізації методів, що викликаються, будуть визначатися саме у час їх використання в залежності від поточного набору здібностей. Задля полегшення розробки та розуміння була спроектована діаграма класів, що зображена на рисунку 3.2.

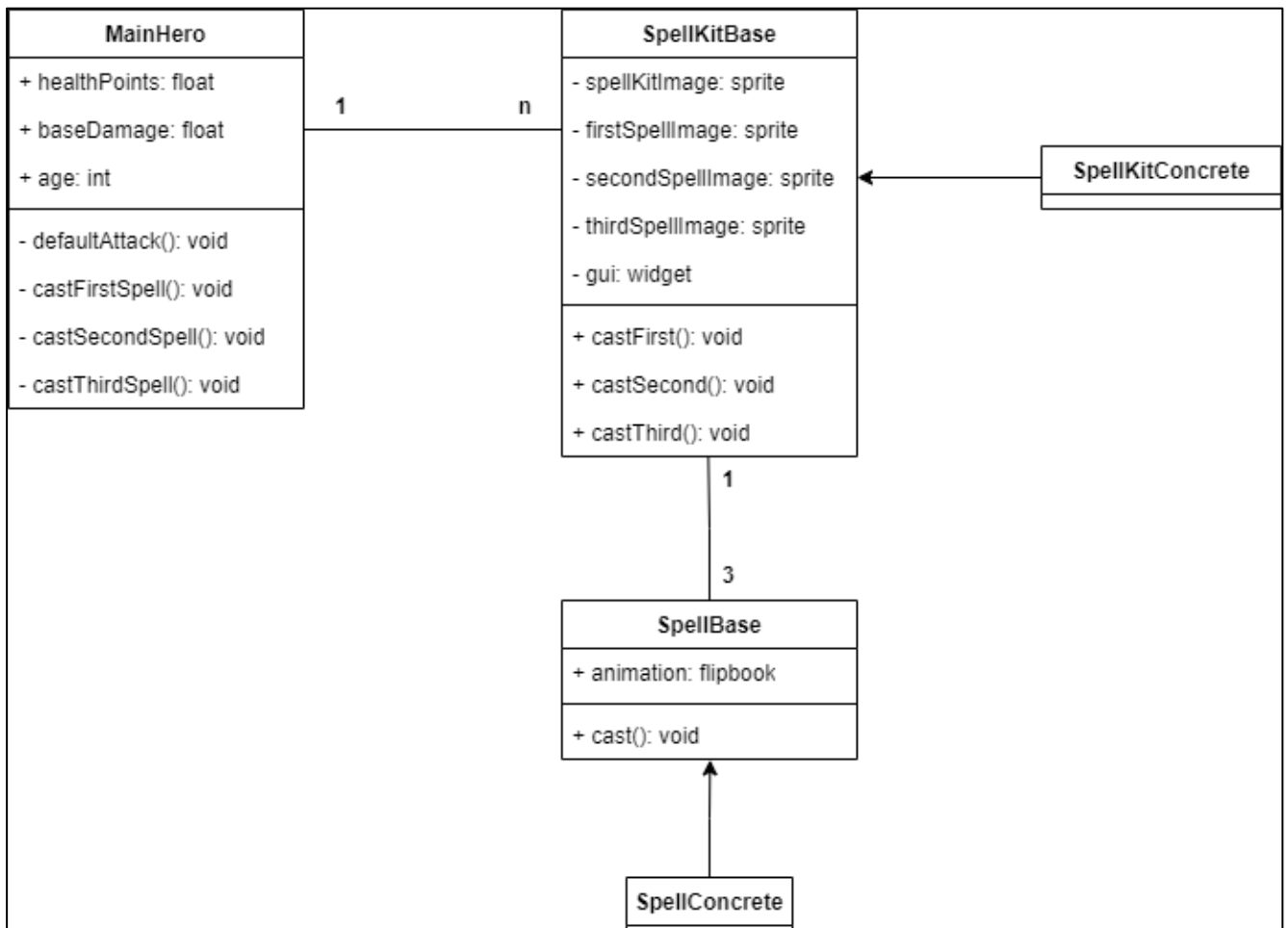


Рисунок 3.2 – діаграма класів для системи здібностей (рисунок виконаний самостійно)

Також необхідно спроектувати систему ворогів у грі. Система супротивників повинна бути гнучкою та з можливістю легкого додавання нових типів ворогів. Кожен ворог може відрізнятися за своїми здібностями, але в кожного будуть показники здоров'я та швидкість [9].

Також під час перевірки зіткнень система не повинна знати конкретний клас ворога, який зіштовхнувся з об'єктом зіткнення. Тому необхідно створити базовий клас супротивника, від якого будуть успадковуватися усі інші вороги. Усі основні властивості разом з реалізацією методів за замовчуванням необхідно виділити в базовий клас як показано на рисунку 3.3.

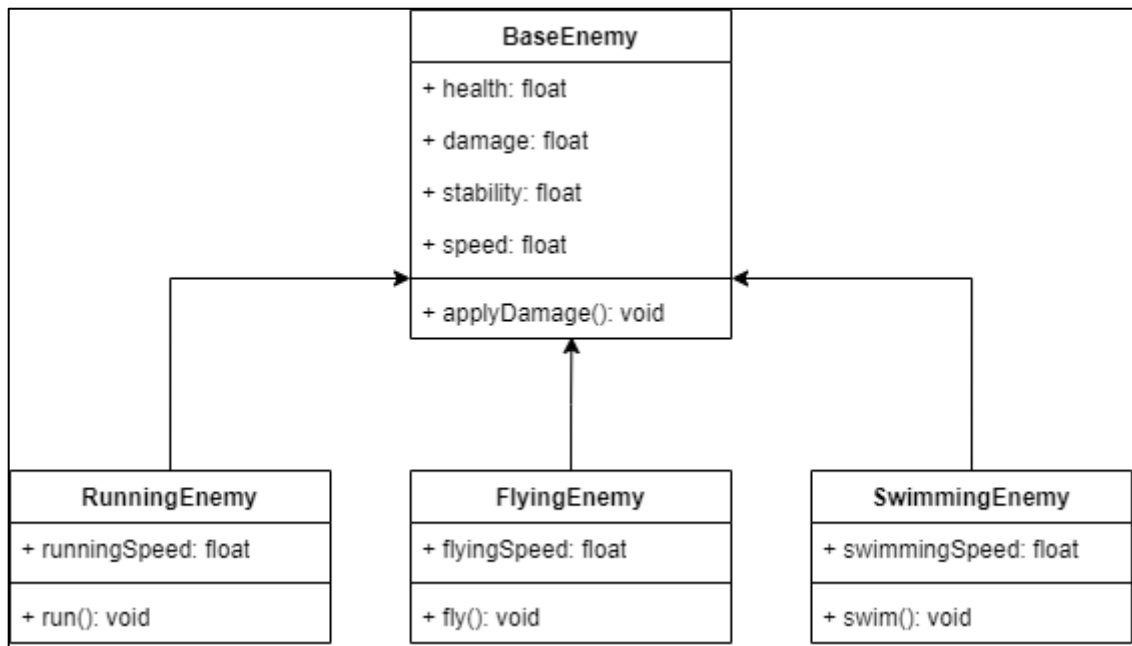


Рисунок 3.3 – діаграма класів для системи ворогів (рисунок виконаний самостійно)

Як можна побачити на діаграмі, різні типи ворогів зі своїми особливостями успадковуються від спільного класу, що дозволить узагальнити логіку взаємодії з супротивниками.

### 3.2 Проектування архітектури ПЗ

У розробці ігрового програмного застосунку у жанрі 2d Action-Adventure з елементами RPG використовується двигун Unreal Engine 5, який має свої архітектурні особливості [10].

Серед можливих варіантів була обрана архітектура заснована на компонентах. Ця архітектура є широко використовуваною в галузі ігрової розробки через свою гнучкість та ефективність. Заснована на компонентах архітектура дозволяє розбити гру на окремі функціональні блоки, що спрощує розробку та підтримку проекту. Кожен компонент може бути самодостатнім модулем, який має чітко визначені відповідальності та може бути легко модифікований або замінений без впливу на інші частини гри.

Через обрану архітектуру впливає велика кількість компонентів, які необхідно групувати в проекті. Задля зручної орієнтації в проекті, що розробляється, необхідно створити файлову структуру проекту.

Компоненти в проекті необхідно розділити на папки та підпапки в залежності від їх функціональності, щоб забезпечити належну організацію та зручний доступ до ресурсів та коду. Наприклад, необхідно створити каталоги для коду, графічних та звукових ресурсів. В каталозі для коду треба створити підкаталоги для головного персонажу, керування, ворогів, карт та здібностей. В каталозі для графічних ресурсів необхідно створити підкаталоги для кожної сутності, де будуть зберігатися відповідні текстури та анімації. В каталозі для звукових ресурсів треба створити підкаталоги для музики та ефектів.

У складі гри будуть включені різноманітні компоненти, які складаються з персонажів, ворогів, локацій, системи управління та інших. Кожен компонент буде мати свою відповідність та функціональність, що сприятиме гнучкості та розширюваності гри. Наприклад, персонажі можуть бути реалізовані як окремі компоненти з власними характеристиками, анімаціями та логікою поведінки, які можуть бути легко налаштовані та модифіковані [11].

Гра буде мати різні стани, які будуть керовані системою управління станами. Це дозволить гнучко керувати переходами між різними етапами гри, такими як головне меню, ігровий процес, екран паузи та інші. Управління станами буде здійснюватися через чітко визначені перехідні події та дії, що дозволить забезпечити плавні та безперебійні переходи між станами гри.

Для забезпечення високої продуктивності та швидкодії гри потрібно використовувати різні методи оптимізації та оптимізовані підходи до розробки. Це включає уникнення зайвого коду та операцій, використання швидкодіючих алгоритмів та структур даних, а також оптимізацію ресурсів та запитів до системи. Також треба звертати увагу на потенційні витрати даних та звільняти усі завантажені в пам'ять ресурси за непотреби. Такий підхід дозволить забезпечити високу продуктивність та плавну роботу гри навіть на великих обсягах даних та під час інтенсивного геймплею.

### 3.3 Приклади алгоритмів та методів

У ході проектування ігрового програмного застосунку необхідно звернути окрему увагу до алгоритмів, від яких залежать основні ігрові механіки. Одним з таких алгоритмів є алгоритм отримання шкоди ворогом. Кожен ворог має очки здоров'я та очки витривалості. Коли очки здоров'я ворога дорівнюють нулю – він видаляється з ігрового світу. Коли очки витривалості ворога дорівнюють нулю – він приголомшується після отримання шкоди. Під час оглушення ворог відштовхується в протилежний джерелу шкоди напрямок, та не може виконувати ніякі дії. Після отримання шкоди є деякий час Алгоритм завдання шкоди ворогу можна побачити на рисунку 3.4.

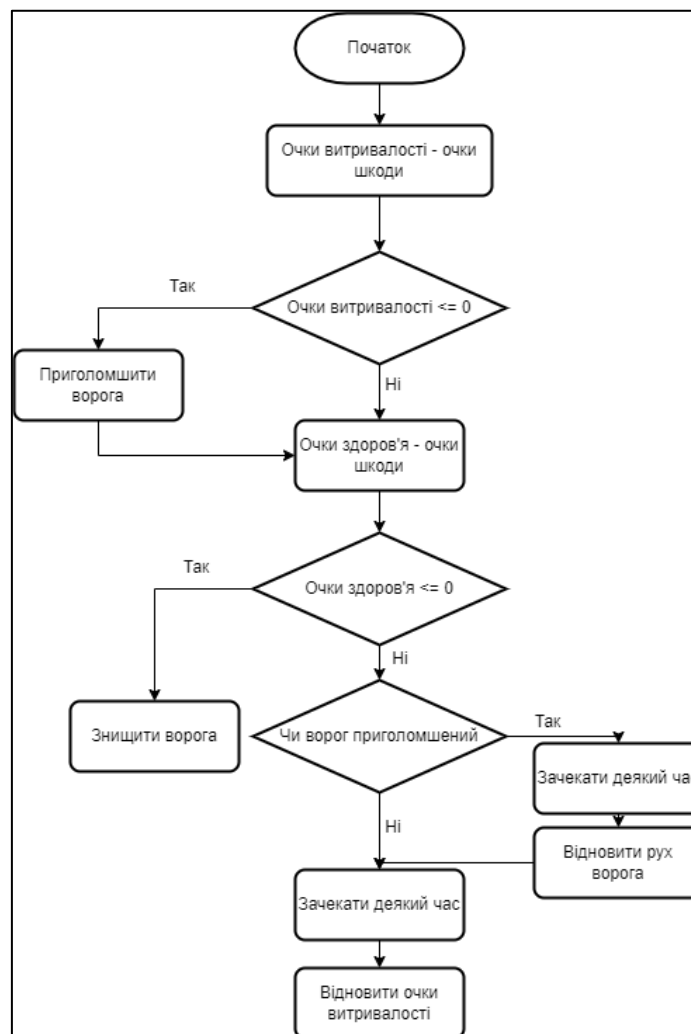


Рисунок 3.4 – блок-схема алгоритму отримання шкоди ворогом (рисунок виконано самостійно)

На зображенні можна побачити, що спочатку віднімаються очки витривалості та в залежності від результуючої кількості вирішується, чи може ворог бути оглушеним. Після цього віднімаються очки здоров'я, якщо їх кількість менша або дорівнює нулю – ворог видаляється з ігрового світу. Далі в залежності від того, чи був приголомшений ворог, через деякий час йому відновлюється можливість виконувати дії. У кінці супротивнику відновлюються очки витривалості.

### 3.4 Створення UI / UX

Проектування інтерфейсу користувача потребує ретельного аналізу потреб цільової аудиторії ігрових програмних застосунків з урахуванням особливостей жанру.

Для кращого розуміння задачі необхідно розглянути сильні сторони дизайну вдалих ігор жанру 2d Action-Adventure. На рисунку 3.5 можна побачити приклад інтерфейсу основного геймплею гри “ENDER LILIES: Quietus of the Knights”.



Рисунок 3.5 – інтерфейс основного геймплею гри “ENDER LILIES: Quietus of the Knights” (за даними [12])

На зображенні можна побачити, що головний персонаж, найбільш важливий елемент, знаходиться близько до центру екрана. Далі в лівому верхньому куту знаходяться показники очок здоров'я, рівень персонажу та кількість аптечок, друга найбільш важлива інформація. У лівому нижньому куту знаходяться навички, які може використовувати гравець, менш важлива інформація. У правому верхньому куту знаходиться карта. Решту екрана займає ігрове оточення.

Комп'ютерна гра "ENDER LILIES: Quietus of the Knights" має зручний та зрозумілий інтерфейс користувача бо більш важлива інформація помітніша за усю іншу. Даний варіант розташування елементів дуже поширений серед двовимірних ігор, саме тому гравці, які вже мали досвід гри в 2d ігрові програмні застосунки, зможуть легше та швидше освоїтися.

Для ігрового програмного застосунка в жанрі 2d Action-Adventure з елементами RPG були визначені основні ресурси, які повинні бути доступні гравцю. Ресурси в порядку важливості:

- головний персонаж;
- ігрове оточення;
- кількість очок здоров'я персонажу;
- вік персонажу;
- кількість очок здоров'я супротивника;
- завдання шкоди;
- отримання шкоди;
- поточний набір здібностей;
- здібності з набору.

Також були сформовані канали інформації, завдяки яким користувач зможе отримувати поточний стан ресурсів. Канали інформації разом з ресурсами, що розподілені по ним:

- основна площа екрану – ігрове оточення;
- біля центру екрану – головний персонаж;
- лівий верхній кут екрану: кількість очок здоров'я персонажу, вік персонажу;

- лівий нижній кут екрану: поточний набір здібностей, здібності з набору;
- звукові ефекти: завдання шкоди, отримання шкоди;
- над зображенням супротивника – кількість очок здоров'я ворога.

Для деяких з ресурсів необхідно сформувати систему виміру інформації для більш представлення. Серед таких ресурсів є:

- кількість очок здоров'я персонажу вимірюється смугою, що залита червоним на відсоток від максимальної кількості очок здоров'я
- кількість очок здоров'я ворога вимірюється смугою, що залита червоним на відсоток від максимальної кількості очок здоров'я

У результаті отриманого аналізу та моделювання були отримані макети інтерфейсу основних станів, наявних у грі. Макет інтерфейсу основного геймплею можна побачити на рисунку 3.6.

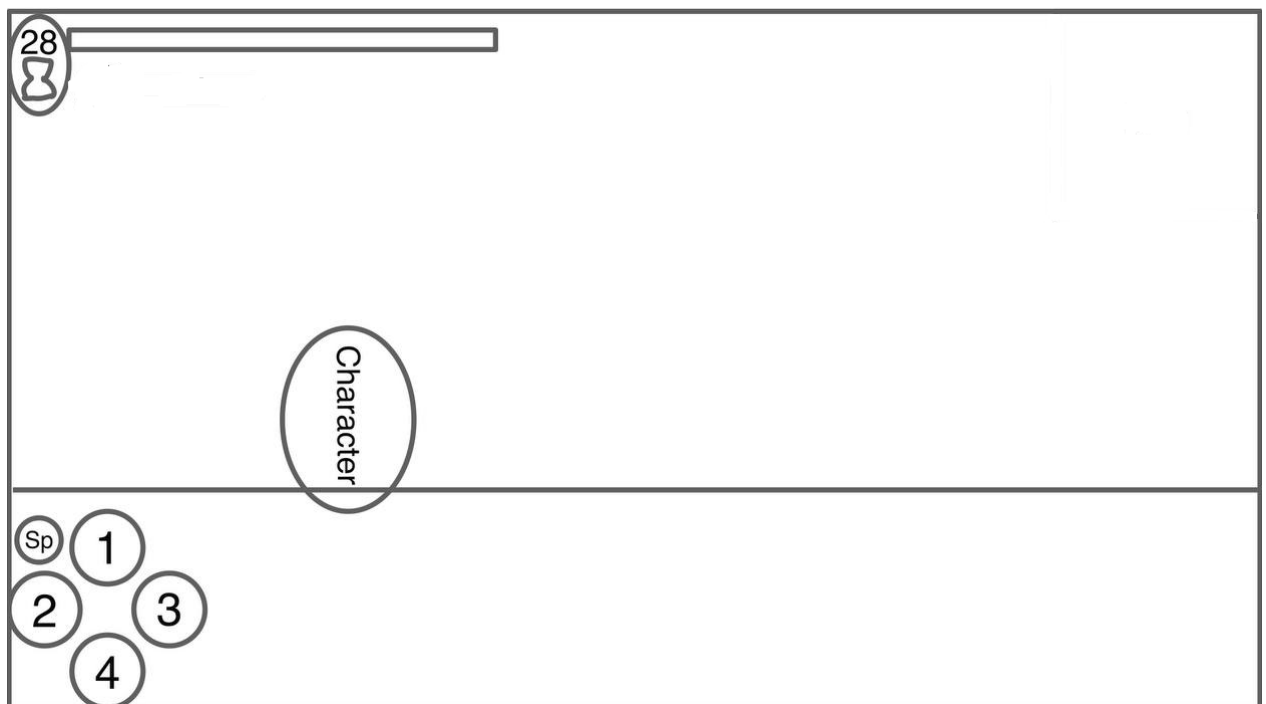


Рисунок 3.6 – макет інтерфейсу основного ігрового процесу (рисунок виконаний самостійно)

На даному рисунку можна побачити як розташовані усі необхідні гравцю ресурси у відповідності до каналів інформації. На рисунку 3.7 можна побачити макет головного меню.

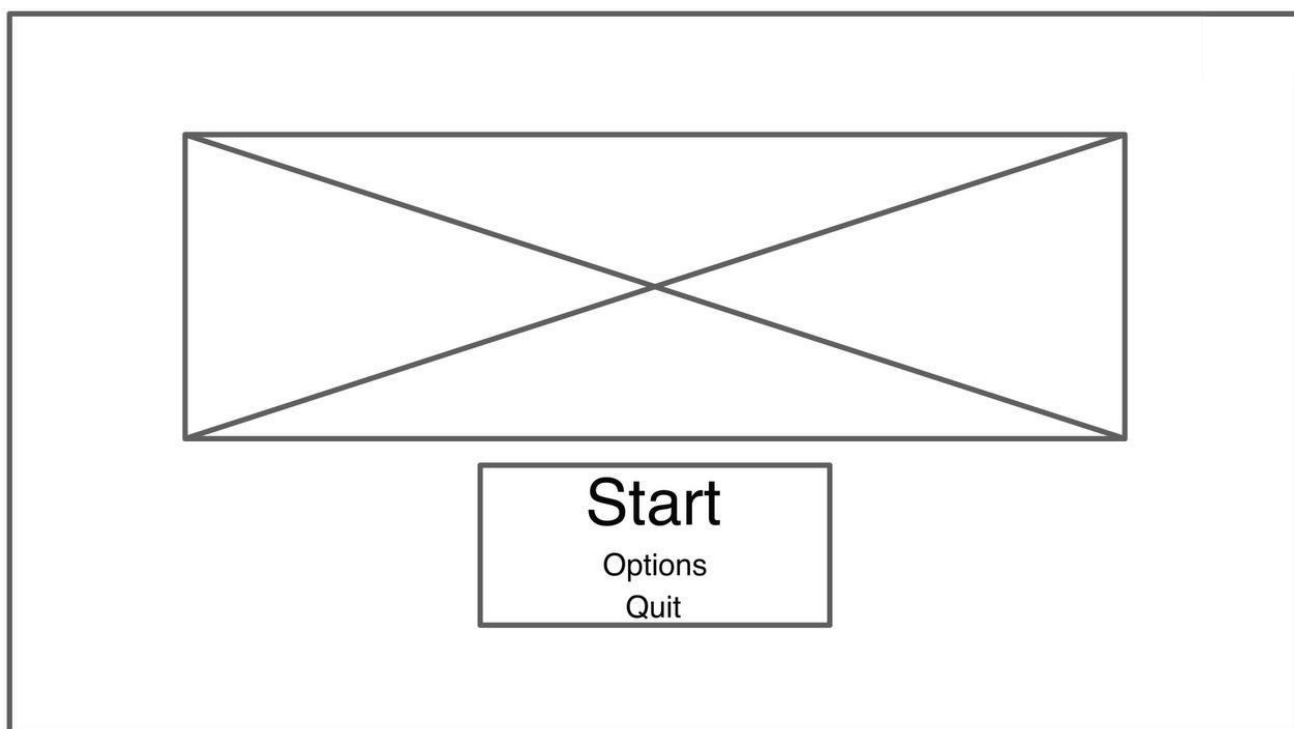


Рисунок 3.7 – макет інтерфейсу головного меню (рисунок виконаний самостійно)

На зображенні можна побачити, що при вході в ігровий програмний застосунок гравець може перейти до основного геймплею, перейти до меню налаштувань або закрити гру.

Ще одним важливим елементом інтерфейсу є меню налаштувань, досить часто це є другим після головного меню, до якого звертається гравець на початку ігрового процесу. У цьому меню користувач може змінити роздільну здатність екрану, віконний режим, режим вертикальної синхронізації, гучність звуку та інші опції.

Доступ до меню налаштувань повинен бути в будь-який момент ігрового процесу. Також усі категорії параметрів повинні бути відсортовані по відповідним підменю, щоб максимально зменшити час, який необхідний користувачу на пошук налаштування, що він бажає змінити. На рисунку 3.8 можна побачити макет меню налаштувань.

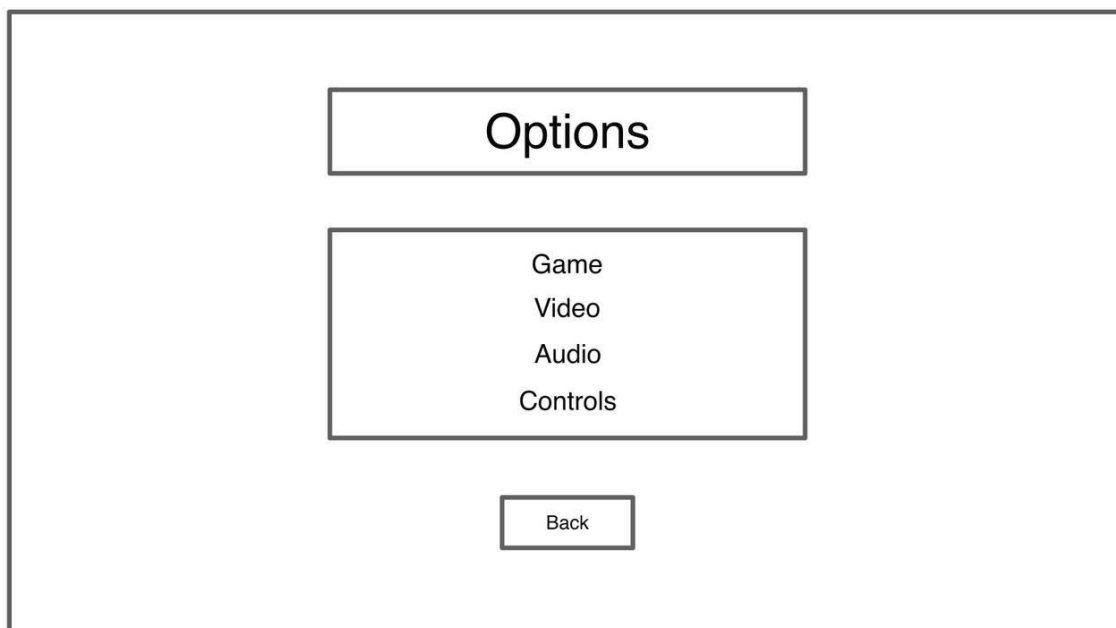


Рисунок 3.8 – макет інтерфейсу меню налаштувань (рисунок виконаний самостійно)

Також важливим станом ігрового застосунку є меню паузи. Його макет можна побачити на рисунку 3.9.

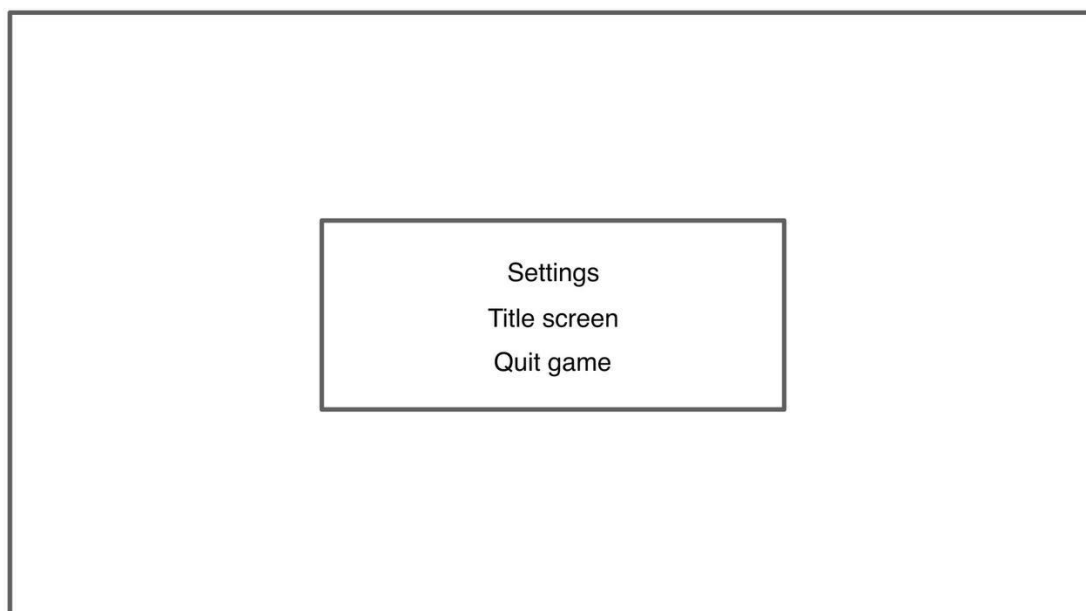


Рисунок 3.9 – макет інтерфейсу меню паузи (рисунок виконаний самостійно)

У меню паузи можна потрапити в будь-який час основного ігрового процесу. Через меню паузи можна потрапити до меню налаштувань або головного меню,

також є можливість вийти з гри. Коли користувач відчиняє меню паузи, ігровий процес повинен призупинитися до моменту, коли гра не буде відновлена.

Створені макети допоможуть у розробці зручного та зрозумілого для користувача UI / UX дизайну. Розробка інтерфейсу – ітеративний процес, бо після реалізації кожної частини функціоналу необхідно протестувати його на коректність роботи та відображення [13].

## 4 ОПИС ПРИЙНЯТИХ ПРОГРАМНИХ РІШЕНЬ

### 4.1 Вибір інструментів програмної реалізації

В якості основної технології для розробки ігрового програмного застосунку був обраний рушій Unreal Engine 5.1. Unreal Engine є одним з найпотужніших та найсучасніших інструментів для розробки ігор різноманітних жанрів, що забезпечує високий рівень швидкодії та надає широкий спектр можливостей для реалізації ігрових програмних застосунків різної складності.

Для більш легкої роботи з двовимірною графікою використовуються плагіни Paper 2D та PaperZD. Paper 2D – це розроблений Epic Games набір функціоналу, який дозволяє створювати з двовимірних зображень анімації та елементи ігрового всесвіту. PaperZD був створений задля доповнення функціоналу рушія, він дозволяє створювати класи станів ігрових сутностей та більш зручно керувати зміною анімацій елементів ігрового світу.

В якості інтегрованої системи розробки було використано Visual Studio 2022 від Microsoft.

### 4.2 Опис модуля головного персонажу

Головний персонаж – це одна з найважливіших сутностей ігрового програмного застосунку. До цієї сутності входять клас актора, класи анімацій, текстури та налаштування керування.

Більшу частину ігрового процесу користувач буде керувати головним персонажем. Модуль головного персонажу реалізує функціонал з переміщення, нанесення шкоди, використання здібностей та керування станом гри, наприклад станом паузи.

Текстуру головного персонажу та хітбоксі його моделі разом з хітбоксом базової атаки можна побачити на рисунку 4.1.

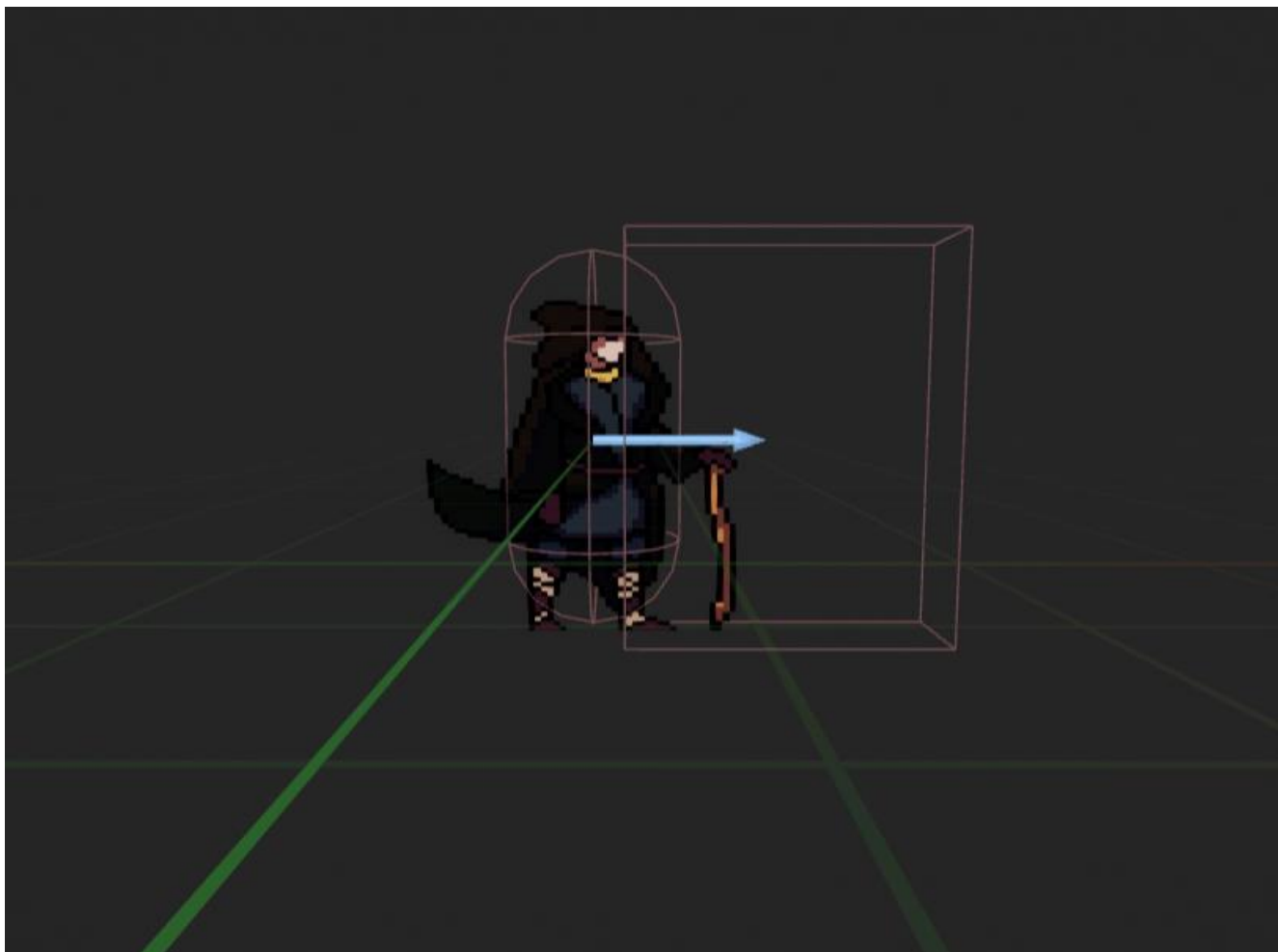


Рисунок 4.1 – модель головного персонажу з хітбоксами (рисунок виконаний самостійно)

На зображенні можна побачити текстуру персонажу в стані спокою, вектор напрямлення та два хітбокси. Капсула, в якій розташований головний персонаж – це основний хітбокс, який вирішує усі зіткнення з елементами ландшафту та іншими сутностями. Паралелепіпед поруч з персонажем – це хітбокс його атаки, під час натискання на відповідну клавішу відбувається перевірка зіткнень між цим хітбоксом та усіма ворожими сутностями.

Клас головного персонажу представлений у виді візуального скрипту Blueprint. Скрипт складається з ініціалізації графічного інтерфейсу та базових властивостей, налаштування дій на кожну з клавіш керування, логіки отримання шкоди, перехід до стану паузи та допоміжних функцій. Частина скрипту, що відповідає за базову атаку можна побачити на рисунку 4.2.

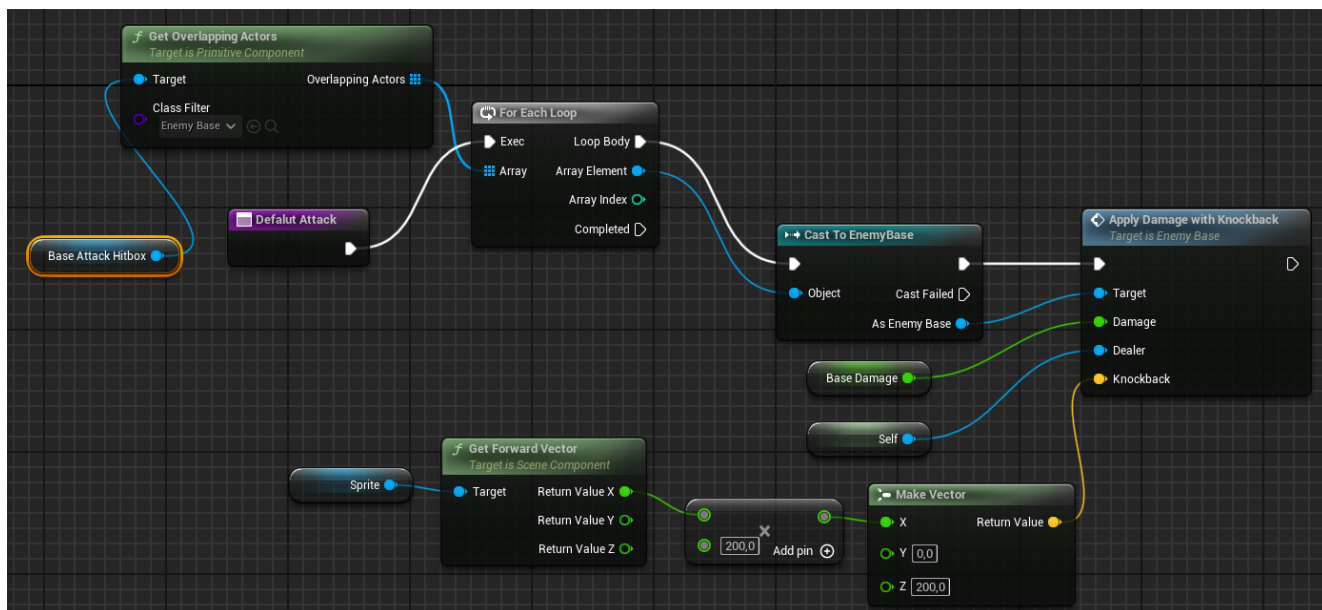


Рисунок 4.2 – скрипт базової атаки (рисунок виконано самостійно)

На рисунку можна побачити, що для того, щоб коректно здійснити атаку, необхідно спочатку перевірити, чи існує зіткнення між хітбоксом атаки та базовим класом ворогів – EnemyBase, після цього кожній сутності ворога завдається шкода разом з вектором відштовхування.

Для керування анімаціями головного персонажу був використаний функціонал плагіну PaperZD, завдяки якому усі варіанти анімацій були представлені в якості машини станів. У машині станів існує два типи вузлів – вузол переходу та вузол стану. Коли сутність головного персонажу з'являється в ігровому світі він автоматично переходить до вузла переходу Out, до якого під'єднаний стан Idle. Завдяки цьому, перш за все відображається анімація стану спокою.

В залежності від натиснутих клавіш, стан головного персонажу буде постійно змінюватись, але майже всі можливі анімації переходять з часом назад до анімації спокою.

Завдяки використанню машини станів удалось більш зручно та інтуїтивно побудувати логіку візуальної складової головного персонажу. Усі можливі стани головного персонажу можна побачити на рисунку 4.3.

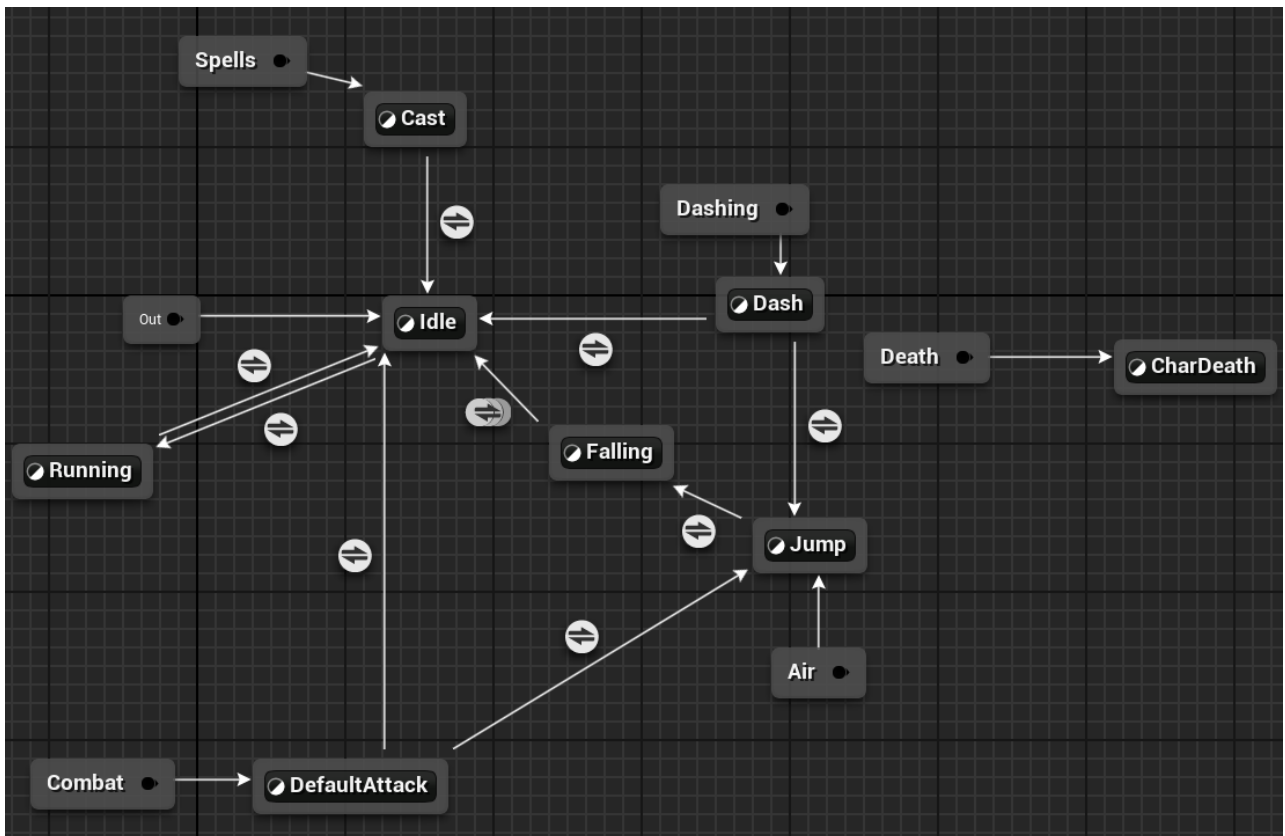


Рисунок 4.3 – машина станів головного персонажу (рисунок виконано самостійно)

Окрему увагу слід звернути на вузол переходу Death – перехід до цього вузлу відбувається у випадку, коли очки здоров'я головного персонажу сягають нулю. У даному випадку відображається меню програшу.

### 4.3 Опис модуля ворогів

При розробці ворогів окрема увага була звернена на принципах ООП, в особливості на поліморфізму та успадкуванні. Усі вороги є спадкоємцями базового класу – EnemyBase. Такий підхід дозволяє не тільки виділити весь базовий функціонал до батьківського класу, але й побудувати спільну логіку взаємодії ігрового світу з ворожими сутностями.

Кожен ворог повинен мати свій хітбокс, очки здоров'я, очки витривалості, функціонал з отримання шкоди та допоміжні функції. Частина скрипту, яка відповідає за виявлення головного персонажу можна побачити на рисунку 5.4.

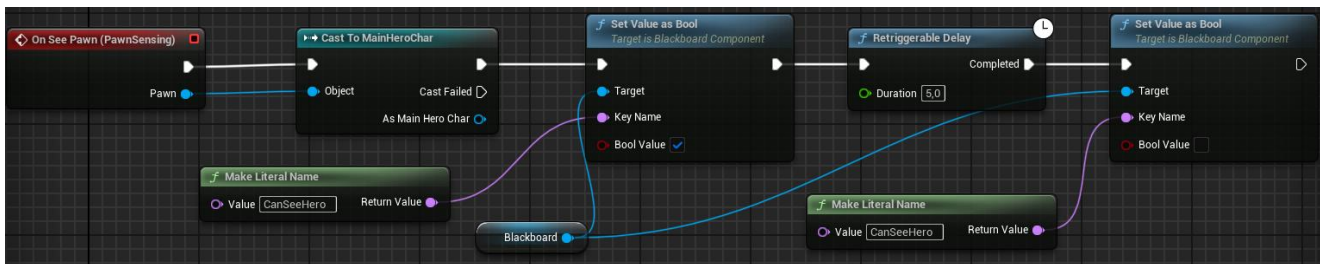


Рисунок 4.4 – скрипт виявлення головного персонажу (рисунок виконаний самостійно)

З зображення можна побачити, що клас супротивника очікує виклик події свого компонента, який визначає характеристики зору. Коли деяка сутність потрапляє в радіус зору ворога, викликається подія On See Pawn. Далі необхідно перевірити, чи потрапивший у поле зору об'єкт відповідає класу головного персонажу, якщо це так, то відповідні логічні поля помічаються для того, щоб в подальшому використати їх в прийнятті рішення штучним інтелектом.

Штучний інтелект ворожих сутностей реалізован у вигляді дерева поведінки, яке вирішує, котрі дії буде виконувати керований об'єкт. Дерево поведінки для ворога слизу можна побачити на рисунку 4.5

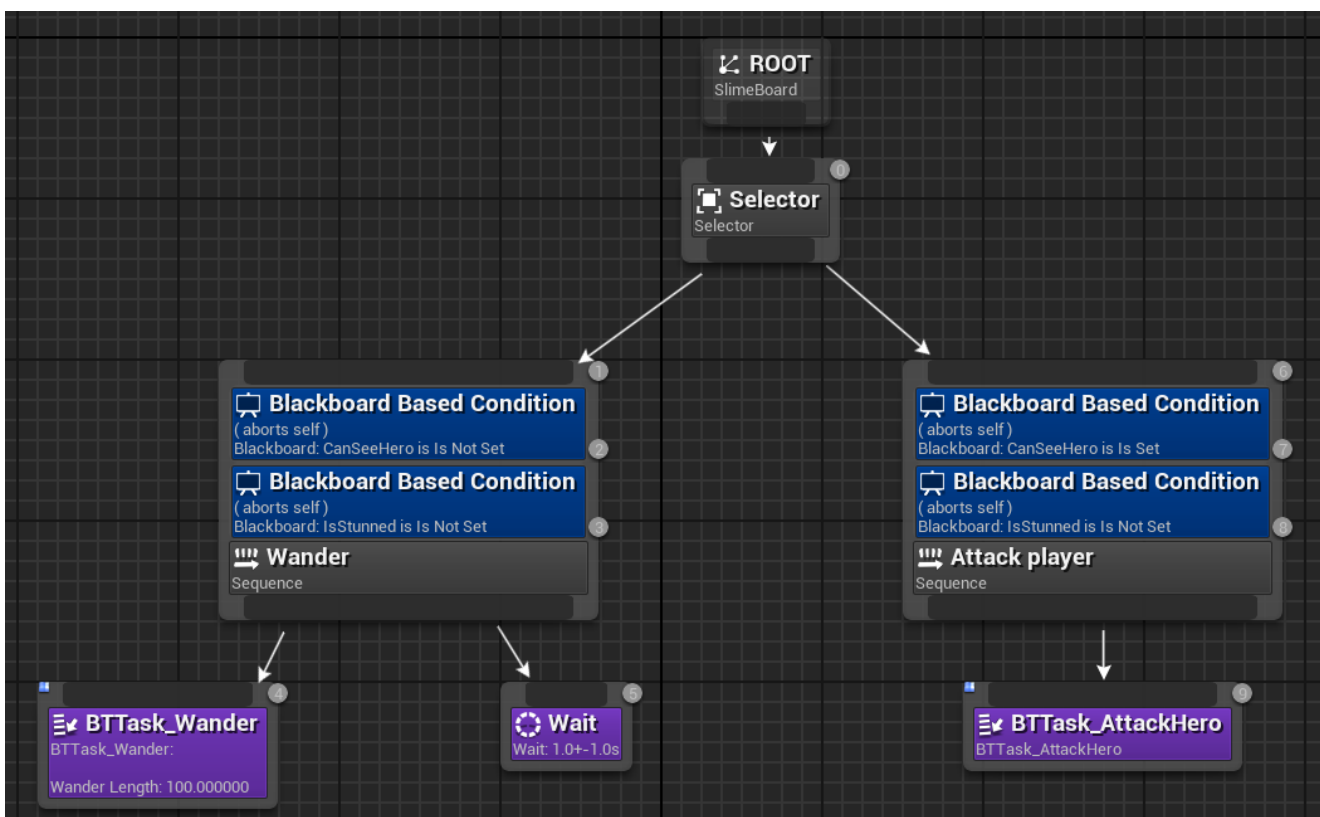


Рисунок 4.5 – дерево поведінки для ворога слизу (рисунок виконано самостійно)

На зображенні можна побачити, що дерево починається з кореня, далі розташовується селектор, який розгалужує дві можливих послідовності дій, в залежності від наступних умов:

- ворог на бачить головного персонажа, у даному випадку він буде пересуватися з однієї точки в іншу з періодичністю в секунду;
- ворог бачить головного персонажа, тоді він почне рухатися в бік гравця та завдавати йому шкоди при контакті.

Змінні з умовними значеннями зв'язані з кожним об'єктом класа супротивника та можуть бути змінені в будь-який момент ігрового процесу, що дозволяє керувати поведінкою ворога в залежності від дій гравця та оточення.

### 4.3 Опис системи здібностей

Здібності в ігровому застосунку являють собою класи, об'єкти яких створюються, коли гравець натискає відповідну клавішу. Здібності розташовані в набори по три екземпляра. Усі набори здібностей успадковуються від спільного класу `SpellKit`. У ігровому програмному застосунку класи наборів здібностей відіграють роль компонентів, які здатні існувати лише в межах класу ігрової сутності.

У базовому класі реалізований такий спільний функціонал, як ініціалізація важливих полів та функції для початку та кінця використання здібностей. Інтерфейс, який надає базовий клас дозволяє класу головного персонажу використовувати необхідні функції не зважаючи уваги на те, який саме набір здібностей буде реалізовувати інтерфейс. Це дозволяє досягти більшого поліморфізму.

Від базового класу був створений клас-спадкоємець, який реалізовує необхідний інтерфейс. Також набори здібностей додають графічні елементи в інтерфейс користувача, щоб гравцю було легше зрозуміти, які здібності він може використати в поточний час. На рисунку 4.6 можна побачити частину скрипту, що відповідає за використання здібності.

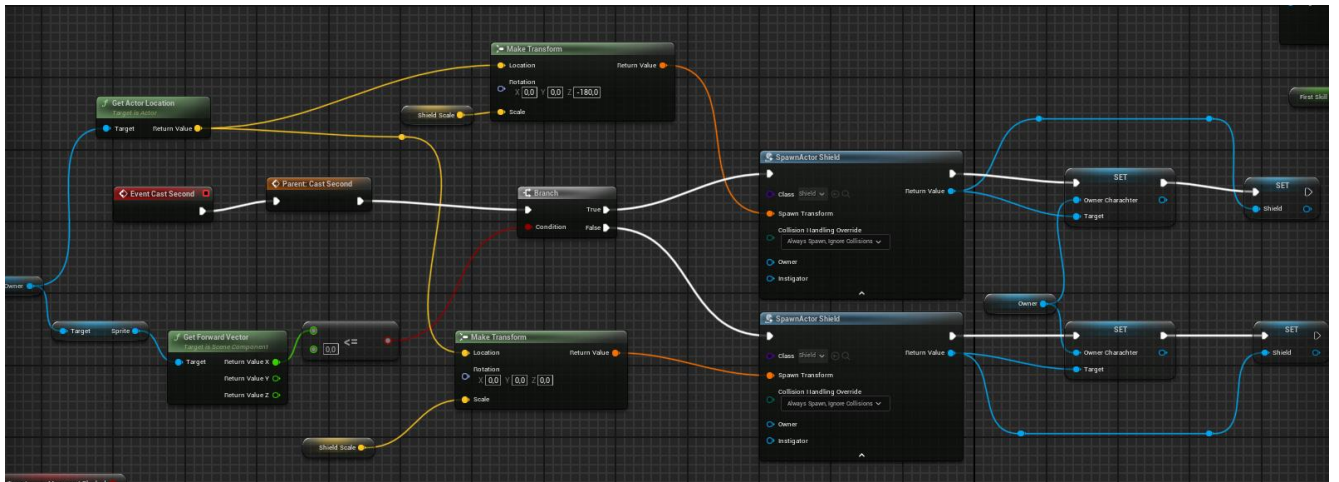


Рисунок 4.6 – скрипт використання здібності

Клас головного персонажу зберігає існуючі набори здібностей в контейнері разом з поточно обраним набором. При натисканні відповідної клавіши викликається подія обраного набору, на який далі покладається відповідальність щодо створення сутності заклинання.

## 5 ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

У ході розробки, відповідно до створеного тест-плану, що знаходиться в додатках, були створені тестові випадки. Тестування відбувалося за методологією BlackBox.

Кожен тест-кейс, або тестовий випадок, містить назву, опис тесту, компонент, тестування якого було проведено, пріоритет тестового випадку, критичність, кроки, що необхідні для відтворення тесту, очікуваний та фактичний результати. Пріоритет тест-кейсу вважається найвищим при значенні P1, найменшим – при P4. Критичність потенційного багу оцінюється від S1 (некритичний) до S4 (руйнує геймплейний процес). У таблиці 5.1 наведено список створений тест-кейсів.

Таблиця 5.1 – Список тестових випадків ігрового програмного забезпечення

Тест № 1	
Назва тесту:	Варіанти розширення екрану
Опис тесту:	Відображаються варіанти розширення екрану, які не підходять до екрану користувача.
Компонент системи:	Меню налаштувань
Пріоритет:	P3
Критичність:	S5
Кроки відтворення:	Запустити гру. Відкрити меню налаштувань. Обрати розширення екрану
Очікуваний результат:	Будуть лише сумісні з екраном розширення
Фактичний результат:	Розширення не завжди сумісні.
Результат:	Знайдено баг, виправлено 4.05.2024

## Продовження таблиці 5.1

Тест № 2	
Назва тесту:	Неправильний напрямок відштовхування
Опис тесту:	Персонажа відштовхує у напрямку руху, а не проти нього, коли він наступає на шипи.
Компонент системи:	Демо рівень.
Пріоритет:	P2
Критичність:	S3
Кроки відтворення:	Запустити демо рівень, стрибнути на один з кінців шипів рухаючись у той же напрямок
Очікуваний результат:	Персонажа відштовхне у протилежний руху напрямок
Фактичний результат:	Персонажа відштовхує у напрямок руху
Висновок:	Знайдено баг, виправлено 05.05.2024
Тест № 3	
Назва тесту:	Некоректні стрибки
Опис тесту:	Іноді, при стрибках в упор до стін, на які не можна застрибнути, персонаж «прослизає» на них.
Компонент системи:	Демо рівень.
Пріоритет:	P2
Критичність:	S3
Кроки відтворення:	Запустити демо рівень, затиснути кнопку стрибка біля високої стіни.
Очікуваний результат:	Персонаж не зможе застрибнути на стіну
Фактичний результат:	Персонаж «прослизає» на стіну
Висновок:	Знайдено баг

## Продовження таблиці 5.1

Тест № 4	
Назва тесту:	Некоректна фізика щита
Опис тесту:	Коли ворог рухається на щит з певної сторони, він «перестрибує» його, а не відштовхується.
Компонент системи:	Демо рівень.
Пріоритет:	P1
Критичність:	S2
Кроки відтворення:	Поставити щит спиною до ворога.
Очікуваний результат:	Ворог відштовхнеться.
Фактичний результат:	Ворог «перестрибує» щит.
Висновок:	Знайдено баг, виправлено 07.05.2024
Тест № 5	
Назва тесту:	Некоректна поведінка камери
Опис тесту:	При зістрибуванні з поверхонь камера може поводити себе некоректно.
Компонент системи:	Головний персонаж
Пріоритет:	P2
Критичність:	S4
Кроки відтворення:	Зістрибнути с поверхні та почати рухатися в протилежному напрямку
Очікуваний результат:	Камера ніяк не буде змінювати своє положення відносно головного персонажу.
Фактичний результат:	Камера наближається до персонажу
Висновок:	Знайдено баг, виправлено 09.05.2024

Окрім обзначених тестів були проведені інші. У ході розробки після кожного реалізованого модуля ігрового програмного застосунку було проведене ручне тестування задля знаходження помилок в роботі гри.

## **6 ВПРОВАДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

### **6.1 Наукове впровадження проекту**

За тематикою кваліфікаційної роботи було написано наукову роботу у вигляді тез для XXVIII Міжнародного молодіжного форуму «Радіоелектроніка та молодь у XXI столітті», що проходив з 16 по 18 травня 2024 року в онлайн режимі на базі ХНУРЕ.

Представлену роботу було опубліковано у відповідному збірнику конференції №6 під заголовком «Проектування бойової системи для ігор жанром 2d Action-Adventure» [14]. У роботі було обговорено основні аспекти проектування бойових систем у двовимірних пригодницьких відеоіграх з наголосом на їх вплив на залучення гравців. Робота розглядала розробки бойових систем, включаючи такі фактори, як жанр гри, розмірність гри, досвід гравця та бажану складність. Робота підкреслювала важливість здібностей персонажа, архетипів ворога балансу геймплею гравець проти гравця, механіку руху та внутрішньоігрової економіки в бойовій системі.

### **6.2 Практичне впровадження проекту**

За матеріалами кваліфікаційної роботи було прийнято участь у виставці технічної творчості молоді 2024 року на базі ХНУРЕ. Виставку було проведено онлайн, презентовано та захищено роботу за темою «Ігровий програмний застосунок у жанрі 2d Action-Adventure з елементами RPG» у секції «Ігрові технології». Виставка проходила з 16 по 18 травня 2024 року.

## ВИСНОВКИ

У процесі виконання кваліфікаційної роботи бакалавра було проведено глибокий аналіз предметної області, що включав у себе ретельне дослідження існуючих програмних рішень. Цей аналіз був спрямований на виявлення як позитивних, так і негативних аспектів, щоб зрозуміти поточні тенденції в ігровій індустрії та визначити основні вимоги і завдання для успішної розробки ігрового програмного застосунку.

Під час вибору технологій для реалізації проекту, було прийнято рішення скористатися Unreal Engine 5 - потужним інструментом для розробки сучасних ігор. Основні переваги цього двигуна полягають у великій кількості функціональних можливостей та високій підтримці з боку Epic Games. Для забезпечення більшого рівня функціональності були використані плагіни Paper 2D та PaperZD, спрямовані на роботу з двовимірною графікою.

Отже, ігровий програмний застосунок, що розробляється, не лише відповідає поточним стандартам і вимогам галузі, але й використовує передові технології для забезпечення найкращого ігрового досвіду. Він може займати значне місце в розважальній сфері, виступаючи частиною ігрової індустрії, та пропонувати гравцям нові можливості для насолоди захоплюючим ігровим випробуванням. Такий проект відображає прогресивний підхід у використанні передових технологій та демонструє бажання створювати інноваційні рішення в сфері інтерактивних розваг.

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. Аналіз ігрового ринку [Електронний ресурс] – URL: <https://www.mordorintelligence.com/ru/industry-reports/global-gaming-market> (дата звернення: 18.05.2024)
2. Video game industry – Wikipedia [Електронний ресурс] – URL: [https://en.wikipedia.org/wiki/Video\\_game\\_industry](https://en.wikipedia.org/wiki/Video_game_industry) (дата звернення: 18.05.2024)
3. Video games as an art form – Wikipedia [Електронний ресурс] – URL: [https://en.wikipedia.org/wiki/Video\\_games\\_as\\_an\\_art\\_form](https://en.wikipedia.org/wiki/Video_games_as_an_art_form) (дата звернення: 18.05.2024)
4. Video game genre – Wikipedia [Електронний ресурс] – URL: [https://en.wikipedia.org/wiki/Video\\_game\\_genre](https://en.wikipedia.org/wiki/Video_game_genre) (дата звернення: 18.05.2024)
5. Action game – Wikipedia [Електронний ресурс] – URL: [https://en.wikipedia.org/wiki/Action\\_game](https://en.wikipedia.org/wiki/Action_game) (дата звернення: 18.05.2024)
6. The fundamental pillars of a combat system [Електронний ресурс] – URL: <https://www.gamedeveloper.com/design/the-fundamental-pillars-of-a-combat-system> (дата звернення: 18.05.2024)
7. Adventure game – Wikipedia [Електронний ресурс] – URL: [https://en.wikipedia.org/wiki/Adventure\\_game](https://en.wikipedia.org/wiki/Adventure_game) (дата звернення: 18.05.2024)
8. The Witcher 3: Wild Hunt [Електронний ресурс] – Official Website – URL: <https://www.thewitcher.com/us/en/witcher3#home> (дата звернення: 18.05.2024)
9. 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.
10. Unreal Architecture [Електронний ресурс] – URL: <https://docs.unrealengine.com/4.26/en-US/ProgrammingAndScripting/ProgrammingWithCPP/UnrealArchitecture/> (дата звернення: 18.05.2024)

11. Gameplay Architecture [Електронний ресурс] – URL: <https://docs.unrealengine.com/4.27/en-US/ProgrammingAndScripting/GameplayArchitecture/> (дата звернення: 18.05.2024)

12. ENDER LILIES: Quietus of the Knights on Steam [Електронний ресурс] – URL: [https://store.steampowered.com/app/1369630/ENDER\\_LILIES\\_Quietus\\_of\\_the\\_Knights/](https://store.steampowered.com/app/1369630/ENDER_LILIES_Quietus_of_the_Knights/) (дата звернення: 18.05.2024)

13. Джессі Шелл. Геймдизайн: як створити гру, в котру будуть грати всі: книга. Альпіна Паблішер, 2021. – 640 с.

14. Т. 5. Каталог виставки технічної творчості молоді [Електронний ресурс] – URL: <https://openarchive.nure.ua/handle/document/26351> (дата звернення: 20.05.2024).

## ДОДАТОК А

## Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ

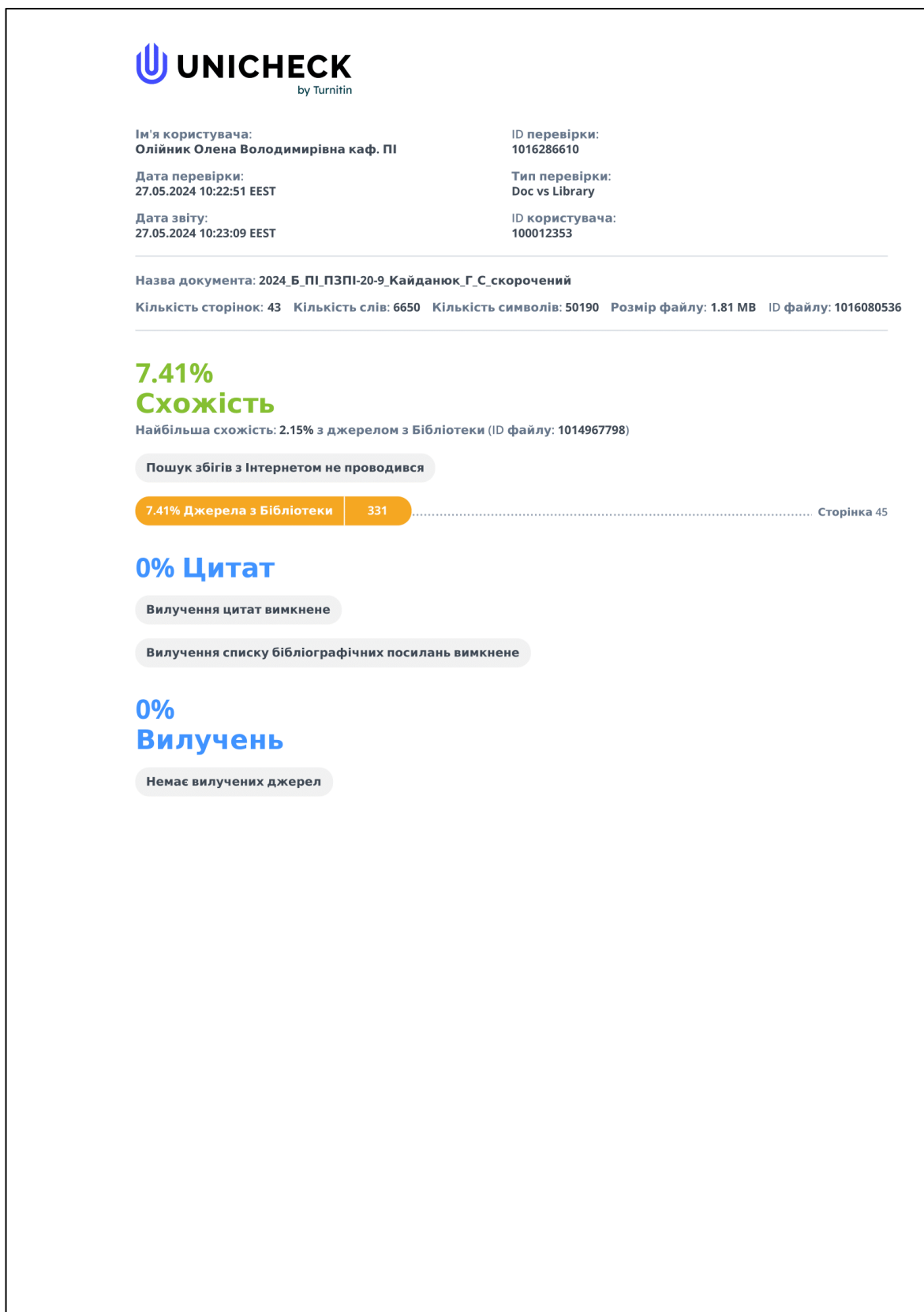


Рисунок А.1 – звіт з результату перевірки на унікальність тексту

ДОДАТОК Б  
Слайди презентації

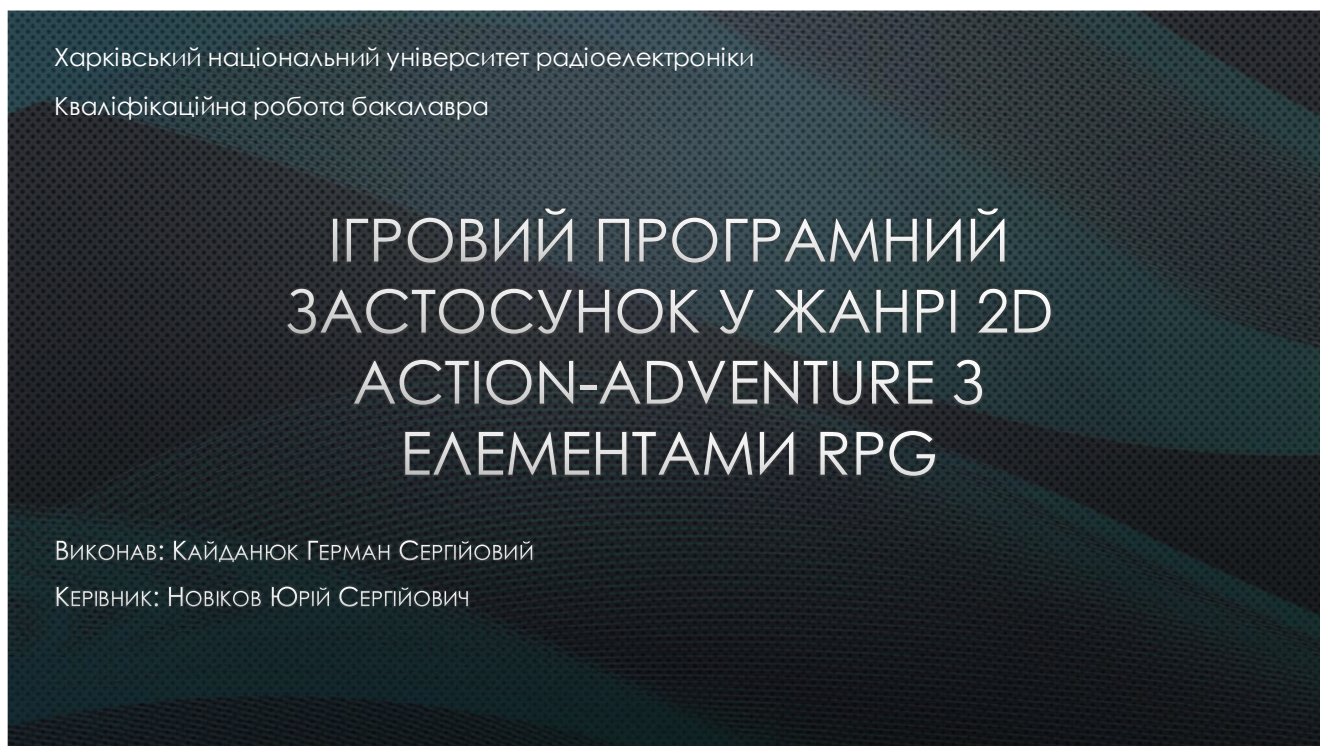


Рисунок Б.1 – Слайд 1

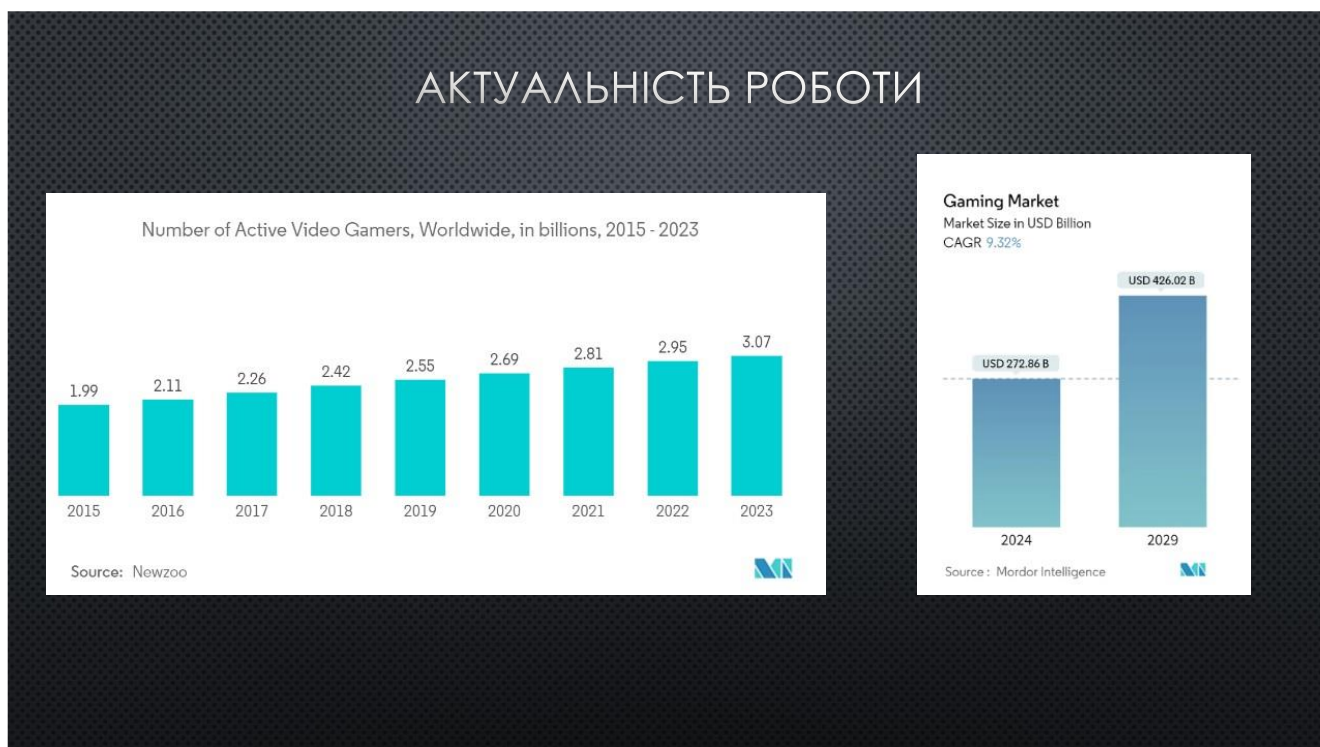


Рисунок Б.2 – Слайд 2

## МЕТА РОБОТИ

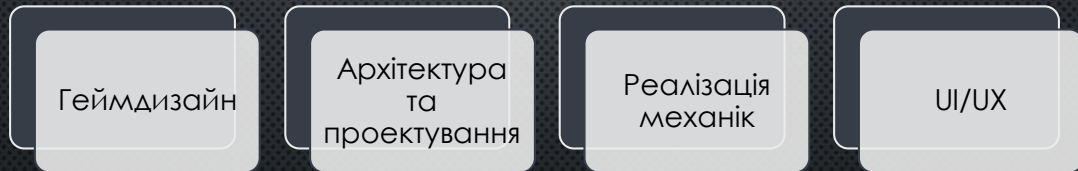


Рисунок Б.3 – Слайд 3

## ПРОЕКТУВАННЯ

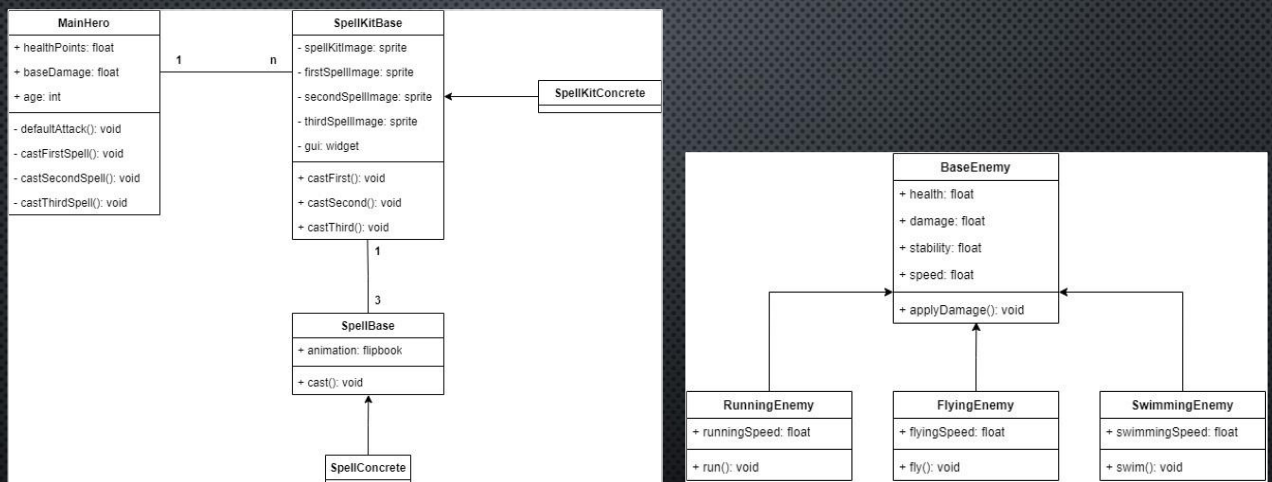


Рисунок Б.4 – Слайд 4

## СИСТЕМА МАГІЇ

- НАБОРИ ПО 3 ЗАКЛИНАННЯ
- МОЖНА ПЕРЕМΙΚАТИСЯ ДИНАМІЧНО ПОСЕРЕД ІГРОВОГО ПРОЦЕСУ
- НОВІ НАБОРИ ВІДКРИВАЮТЬСЯ ПО ХОДУ ПРОХОДЖЕННЯ
- НА ДАНИЙ ЧАС РЕАЛІЗОВАНО БАЗОВИЙ НАБІР

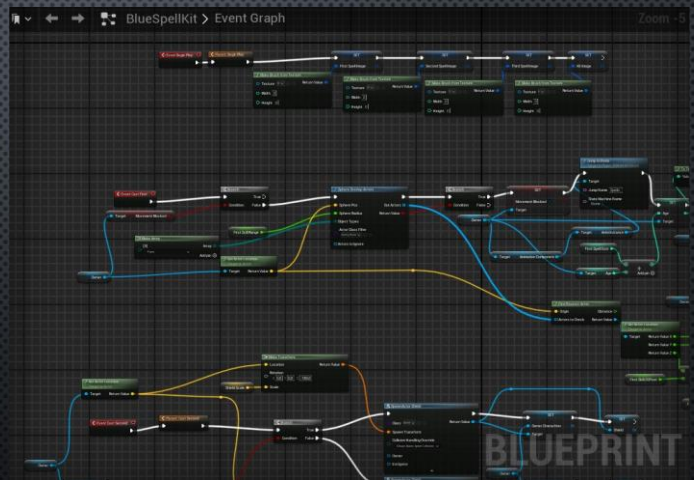


Рисунок Б.5 – Слайд 5

## БОЙОВА СИСТЕМА

- СКЛАДАЄТЬСЯ З ВИКОРИСТАННЯ БАЗОВОЇ АТАКИ ТА ЗАКЛИНАНЬ
- НЕОБХІДНО БАЛАНСУВАТИ ЗАТРАТИ ВІКУ, ЩОБ УНИКНУТИ СМЕРТІ



Рисунок Б.6 – Слайд 6



Рисунок Б.7 – Слайд 7



Рисунок Б.8 – Слайд 8

## НАУКОВЕ ВПРОВАДЖЕННЯ

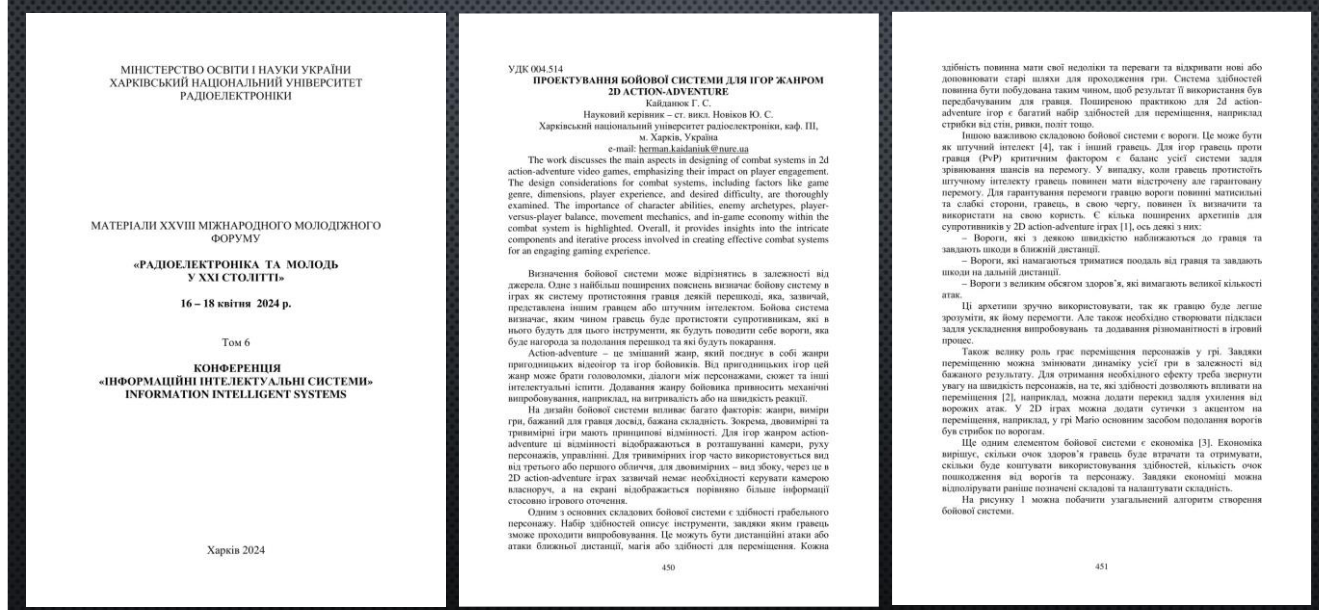


Рисунок Б.9 – Слайд 9

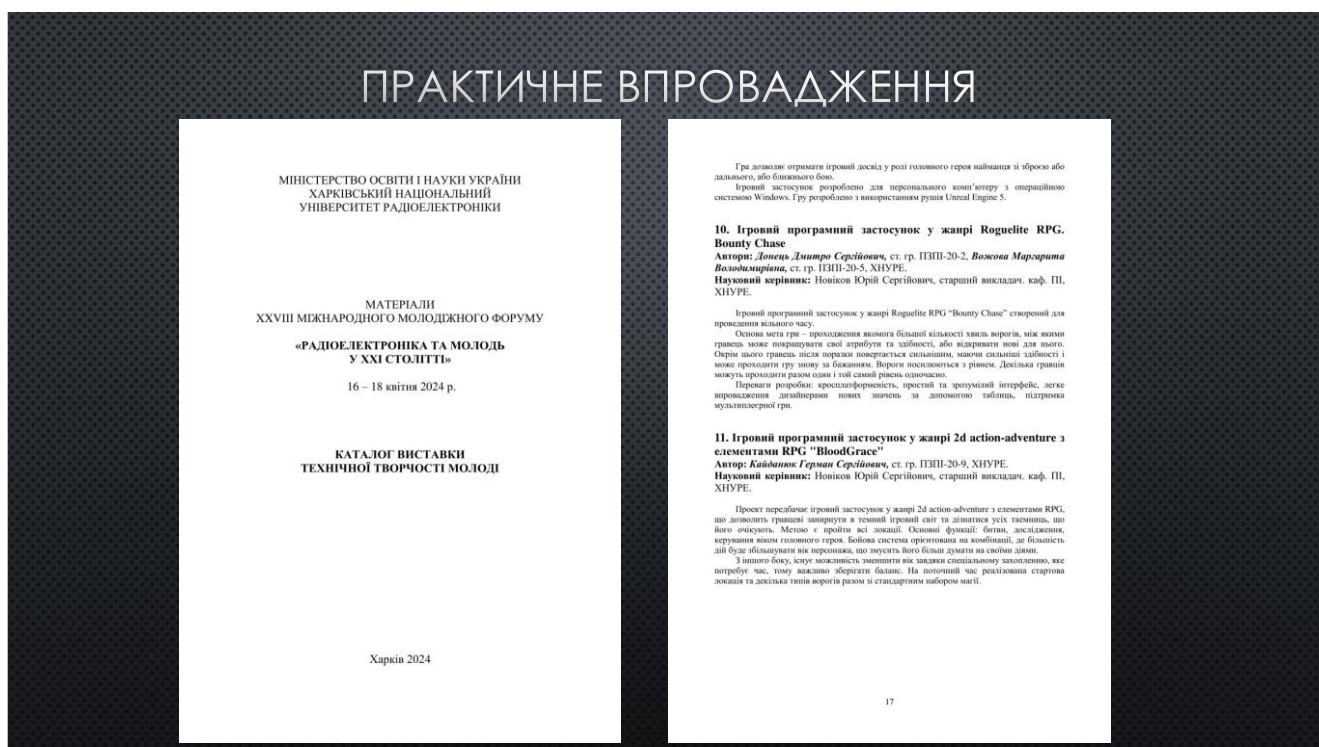


Рисунок Б.10 – Слайд 10

## ВИСНОВКИ

- Розроблено ігровий програмний застосунок
- Спроектовано модульну систему з можливістю подальших змін та розширення
- Реалізовано механіки бою, систему ворогів та навичок
- Розроблено графічний інтерфейс до гри

Рисунок Б.11 – Слайд 11

ДЯКУЮ ЗА УВАГУ

Рисунок Б.12 – Слайд 12

## ДОДАТОК В

## Приклад програмного коду (інтерфейс класу головного персонажа)

```

// MainHeroChar.h: //

#pragma once

#include "Misc/FileHelper.h"
#include "CoreMinimal.h"

UCLASS(Blueprintable, BlueprintType)
class AMainHeroChar : public APaperZDCharacter
{
    GENERATED_BODY()

public:
    UFUNCTION(BlueprintCallable)
    void DamageTaken(double Damage);

    UFUNCTION(BlueprintCallable)
    void DefalutAttack();

    UFUNCTION(BlueprintCallable)
    void FlickSprite();

    UFUNCTION(BlueprintCallable)
    void UpdateAgeGUI();

public:
    UPROPERTY(BlueprintReadOnly, VisibleAnywhere, Category="Default")
    TObjectPtr<UBlueSpellKit_C> BlueKit;

    UPROPERTY(BlueprintReadOnly, VisibleAnywhere, Category="Default")
    TObjectPtr<UBoxComponent> BaseAttackHitbox;

    UPROPERTY(BlueprintReadOnly, VisibleAnywhere, Category="Actor")
    TObjectPtr<UCameraComponent> Camera;

    UPROPERTY(BlueprintReadOnly, VisibleAnywhere, Category="Default")
    TObjectPtr<USpringArmComponent> SpringArm;

    UPROPERTY(BlueprintReadOnly, VisibleAnywhere, Category="MainHeroChar")
    TObjectPtr<UTimelineComponent> Timeline;

    UPROPERTY(BlueprintReadWrite, EditDefaultsOnly, Category="Default")
    bool CanDash;

```

```
UPROPERTY(BlueprintReadWrite, EditDefaultsOnly, Category="Default")  
bool IsDashing;
```

```
UPROPERTY(BlueprintReadWrite, EditAnywhere, Category="Default")  
double HealthPoints;
```

```
UPROPERTY(BlueprintReadWrite, EditDefaultsOnly, Category="Default")  
TObjectPtr<UMainCharacterGUI_C> GUI;
```

```
UPROPERTY(BlueprintReadWrite, EditDefaultsOnly, Category="Default")  
bool MovementBlocked;
```

```
UPROPERTY(BlueprintReadWrite, EditAnywhere, Category="Default")  
double Attack cooldown;
```

```
UPROPERTY(BlueprintReadWrite, EditAnywhere, Category="Default")  
float Base Damage;
```

```
UPROPERTY(BlueprintReadWrite, EditDefaultsOnly, Category="Default")  
bool CanBeDamaged;
```

```
UPROPERTY(BlueprintReadWrite, EditAnywhere, Category="Default")  
double MaxHealth;
```

```
UPROPERTY(BlueprintReadWrite, EditAnywhere, Category="Default")  
double ImmortalityTimer;
```

```
UPROPERTY(BlueprintReadWrite, EditAnywhere, Category="Default")  
double FlickingPeriod;
```

```
UPROPERTY(BlueprintReadWrite, EditAnywhere, Category="Default")  
int32 Age;
```

```
UPROPERTY(BlueprintReadWrite, EditAnywhere, Category="Default")  
double Max Health;
```

```
UPROPERTY(BlueprintReadWrite, EditAnywhere, Category="Default")  
double GrabDuration;
```

```
UPROPERTY(BlueprintReadWrite, EditAnywhere, Category="Default")  
TArray<USpellKit_C*> Kits;
```

```
UPROPERTY(BlueprintReadWrite, EditAnywhere, Category="Default")  
int32 KitNum;
```

```
UPROPERTY(BlueprintReadWrite, EditDefaultsOnly, Category="Default")
bool IsPauseMenuOpen;

UPROPERTY(BlueprintReadWrite, EditDefaultsOnly, Category="Default")
TObjectPtr<UPauseMenu_C> PauseMenuWG;
};
```

## ДОДАТОК Г

Тези доповіді для науково-практичної інтернет-конференції

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
РАДІОЕЛЕКТРОНІКИ

МАТЕРІАЛИ ХХVІІІ МІЖНАРОДНОГО МОЛОДІЖНОГО  
ФОРУМУ

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

**16 – 18 квітня 2024 р.**

Том 6

**КОНФЕРЕНЦІЯ  
«ІНФОРМАЦІЙНІ ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ»  
INFORMATION INTELLIGENT SYSTEMS**

Харків 2024

Рисунок Г.1 – обкладинка збірника

Гриб А. С., 714  
 Григор'єв О. В., 945  
 Гринишина С. О., 923  
 Гриньов С. А., 121  
 Гриньова О. Є., 13, 16, 24  
 Гриньова О.Є., 5, 7  
 Грishaєва 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  
 Данілейко 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  
 Дідусь 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. Є., 16  
 Жемчужний Р. І., 869  
 Женило К.О., 5  
 Жирко К. В., 158, 160  
 Жмур Д. М., 112

### З

Забийворога 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 Є. О., 130  
 Іванова А. І., 175  
 Іванова О. С., 44  
 Ігнатюк Є. 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  
 Кайданок 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. Є., 656  
 Кириченко І. В., 112  
 Кирсанов О. О., 59  
 Кієнко Д. В., 187  
 Кієу Куанг Хієп, 753  
 Кікоть М. С., 189  
 Кісельгова М. Є., 342  
 Кітов А. В., 54  
 Кіщенко Ю. О., 435  
 Климова І. М., 205, 726, 882  
 Клішов М. Р., 886  
 Клочко Є. С., 654  
 Клованський 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  
 Колендовська М. М., 933,  
 938  
 Колесник Л. В., 699  
 Коломоєць К. В., 710  
 Коломойцев П. А., 326  
 Комзолов 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  
 Котелевець 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  
 Лавріненко В. В., 712  
 Лановий О. Ф., 303, 541, 565  
 Ларченко Л. В., 290  
 Ларченко С. О., 565  
 Латишев О. О., 581  
 Лахтін В. В., 105

УДК 004.514

**ПРОЕКТУВАННЯ БОЙОВОЇ СИСТЕМИ ДЛЯ ІГОР ЖАНРОМ  
2D ACTION-ADVENTURE**

Кайданюк Г. С.

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

Харківський національний університет радіоелектроніки, каф. ПІ,  
м. Харків, Україна

e-mail: [herman.kaidaniuk@nure.ua](mailto:herman.kaidaniuk@nure.ua)

The work discusses the main aspects in designing of combat systems in 2d action-adventure video games, emphasizing their impact on player engagement. The design considerations for combat systems, including factors like game genre, dimensions, player experience, and desired difficulty, are thoroughly examined. The importance of character abilities, enemy archetypes, player-versus-player balance, movement mechanics, and in-game economy within the combat system is highlighted. Overall, it provides insights into the intricate components and iterative process involved in creating effective combat systems for an engaging gaming experience.

Визначення бойової системи може відрізнитись в залежності від джерела. Одне з найбільш поширених пояснень визначає бойову систему в іграх як систему протистояння гравця деякій перешкоді, яка, зазвичай, представлена іншим гравцем або штучним інтелектом. Бойова система визначає, яким чином гравець буде протистояти супротивникам, які в нього будуть для цього інструменти, як будуть поводити себе вороги, яка буде нагорода за подолання перешкод та які будуть покарання.

Action-adventure – це змішаний жанр, який поєднує в собі жанри пригодницьких відеоігор та ігор бойовиків. Від пригодницьких ігор цей жанр може брати головоломки, діалоги між персонажами, сюжет та інші інтелектуальні іспити. Додавання жанру бойовика привносить механічні випробовування, наприклад, на витривалість або на швидкість реакції.

На дизайн бойової системи впливає багато факторів: жанри, виміри гри, бажаний для гравця досвід, бажана складність. Зокрема, двовимірні та тривимірні ігри мають принципові відмінності. Для ігор жанром action-adventure ці відмінності відображаються в розташуванні камери, руху персонажів, управлінні. Для тривимірних ігор часто використовується вид від третього або першого обличчя, для двовимірних – вид збоку, через це в 2D action-adventure іграх зазвичай немає необхідності керувати камерою власноруч, а на екрані відображається порівняно більше інформації стосовно ігрового оточення.

Одним з основних складових бойової системи є здібності грабельного персонажу. Набір здібностей описує інструменти, завдяки яким гравець зможе проходити випробовування. Це можуть бути дистанційні атаки або атаки ближньої дистанції, магія або здібності для переміщення. Кожна

здібність повинна мати свої недоліки та переваги та відкривати нові або доповнювати старі шляхи для проходження гри. Система здібностей повинна бути побудована таким чином, щоб результат її використання був передбачуваним для гравця. Поширеною практикою для 2d action-adventure ігор є багатий набір здібностей для переміщення, наприклад стрибки від стін, ривки, політ тощо.

Іншою важливою складовою бойової системи є вороги. Це може бути як штучний інтелект [4], так і інший гравець. Для ігор гравець проти гравця (PvP) критичним фактором є баланс усієї системи задля зрівнювання шансів на перемогу. У випадку, коли гравець протистоїть штучному інтелекту гравець повинен мати відстрочену але гарантовану перемогу. Для гарантування перемоги гравцю вороги повинні матисильні та слабкі сторони, гравець, в свою чергу, повинен їх визначити та використати на свою користь. Є кілька поширених архетипів для супротивників у 2D action-adventure іграх [1], ось деякі з них:

- Вороги, які з деякою швидкістю наближаються до гравця та завдають шкоди в ближній дистанції.
- Вороги, які намагаються триматися поодаль від гравця та завдають шкоди на дальній дистанції.
- Вороги з великим обсягом здоров'я, які вимагають великої кількості атак.

Ці архетипи зручно використовувати, так як гравцю буде легше зрозуміти, як йому перемогти. Але також необхідно створювати підкласи задля ускладнення випробовувань та додавання різноманітності в ігровий процес.

Також велику роль грає переміщення персонажів у грі. Завдяки переміщенню можна змінювати динаміку усієї гри в залежності від бажаного результату. Для отримання необхідного ефекту треба звернути увагу на швидкість персонажів, на те, які здібності дозволяють впливати на переміщення [2], наприклад, можна додати перекид задля ухилення від ворожих атак. У 2D іграх можна додати сутички з акцентом на переміщення, наприклад, у грі Magic основним засобом подолання ворогів був стрибок по ворогам.

Ще одним елементом бойової системи є економіка [3]. Економіка вирішує, скільки очок здоров'я гравець буде втрачати та отримувати, скільки буде коштувати використання здібностей, кількість очок пошкодження від ворогів та персонажу. Завдяки економіці можна відполірувати раніше позначені складові та налаштувати складність.

На рисунку 1 можна побачити узагальнений алгоритм створення бойової системи.

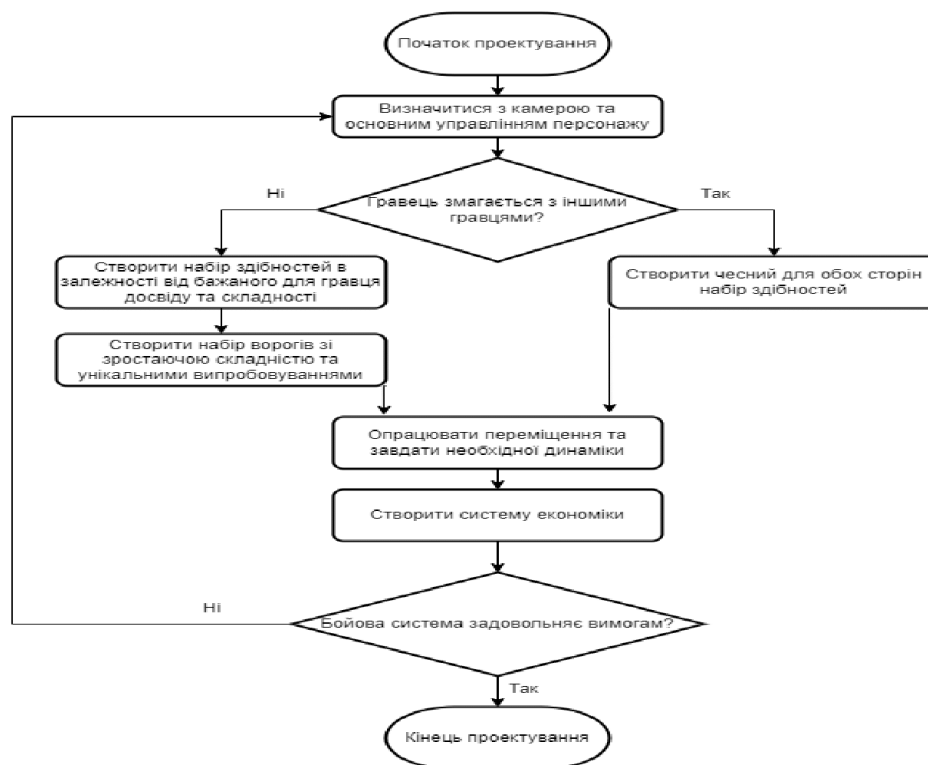


Рисунок 1 – Схема алгоритму проектування бойової системи

Процес проектування бойових систем досить складно описати єдиною формулою, бо кожна гра та кожний дизайнер може мати свій підхід. Слід пам'ятати, що це досить складний та ітеративний процес, основні деталі котрого були проаналізовані в даних тезах.

Список використаних джерел:

1. Sébastien Lambottin. (2012, 15 серпня). The fundamental pillars of a combat system. <https://www.gamedeveloper.com/design/the-fundamental-pillars-of-a-combat-system>
2. amalia.pomian. (2016, 22 липня). About combat system design. <https://www.evozon.com/combat-system-design/>
3. book.leveldesignbook.com. (2023, 7 липня). Combat – the level design book. <https://book.leveldesignbook.com/process/combat>
4. Mazurova O., Samantsov O., Topchii O., Shirokopetleva M., 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 Г

Анотація для науково-практичної інтернет-виставки

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ  
УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

МАТЕРІАЛИ  
XXVIII МІЖНАРОДНОГО МОЛОДІЖНОГО ФОРУМУ

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

16 – 18 квітня 2024 р.

**КАТАЛОГ ВИСТАВКИ  
ТЕХНІЧНОЇ ТВОРЧОСТІ МОЛОДІ**

Харків 2024

Рисунок Г.1 – обкладинка каталогу виставки

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

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

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

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

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

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

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

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

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

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

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

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

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