

ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД  
«УНІВЕРСИТЕТ ЕКОНОМІКИ ТА ПРАВА «КРОК»

КВАЛІФІКАЦІЙНА РОБОТА

Тема: «Комп'ютерна гра "Demonic Hunter" з використанням алгоритмів інтелектуальної поведінки ботів»

Ступінь вищої освіти – бакалавр  
Спеціальність – 122 «Комп'ютерні науки»  
Освітня програма «Комп'ютерні науки»

ПОЯСНЮВАЛЬНА ЗАПИСКА

Виконав: здобувач 4 курсу  
групи КН-21  
Володимир ЛЯШИНСЬКИЙ

Керівник: зав. кафедри комп'ютерних наук,  
к.е.н., с.н.с., доцент,  
Сергій МІЧКІВСЬКИЙ

Засвідчую, що кваліфікаційна робота оформлена відповідно до ДСТУ 3008:2015 та не містить запозичень з праць інших авторів без відповідних посилань.

Здобувач: \_\_\_\_\_  
(підпис)

м. Київ – 2025 рік

ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД  
«УНІВЕРСИТЕТ ЕКОНОМІКИ ТА ПРАВА «КРОК»»

ЗАТВЕРДЖУЮ:  
завідувач кафедри  
комп'ютерних наук  
\_Сергій МІЧКІВСЬКИЙ  
« \_\_\_\_ » \_\_\_\_ 20 \_\_\_\_ р

ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ  
Ляшинський Володимир Олександрович

Тема роботи	Комп'ютерна гра "Demonic Hunter" з використанням алгоритмів інтелектуальної поведінки ботів (комплексна робота)
Номер та дата наказу про затвердження теми	№121-7 від 24 грудня 2024 року
Коротка постановка завдання	Розробити програмне забезпечення 2D-гри на Unity з елементами платформінгу, використанням інтелектуальних алгоритмів поведінки противників та динамічних візуальних ефектів.
Посилання на джерела інформації (не більше п'яти найменувань, які рекомендує науковий керівник)	1. Ваша перша 2D гра // Документація до Godot Engine 4.4 українською мовою – URL: <a href="https://docs.godotengine.org/uk/4.x/getting_started/first_2d_game/index.html">https://docs.godotengine.org/uk/4.x/getting_started/first_2d_game/index.html</a> , 2. Level-дизайн // Дизайн рівнів для 2D-ігор: основні патерни – URL: <a href="https://avada-media.ua/services/disain-urovney-dlya-2d-igr-osnovnye-patterny/">https://avada-media.ua/services/disain-urovney-dlya-2d-igr-osnovnye-patterny/</a> ,
Вимоги до кваліфікаційної роботи	Кваліфікаційна робота має передбачити теоретичне, системотехнічне або експериментальне дослідження складного спеціалізованого завдання або практичної проблеми в галузі комп'ютерних наук, яке характеризується комплексністю та невизначеністю умов і потребує застосування теорій і методів інформаційних технологій.

Дата видачі завдання 24 грудня 2024 р.

Керівник

Сергій МІЧКІВСЬКИЙ

Здобувач освітнього ступеня бакалавра

Володимир ЛЯШИНСЬКИЙ

## КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання	Примітка
<b>Підготовчий етап</b>			
1	Вибір напрямку дослідження	02.12.2024 р.	<i>виконано</i>
2	Формування теми та призначення керівника	16.12.2024 р.	<i>виконано</i>
3	Затвердження теми кваліфікаційної роботи	23.12.2024 р.	<i>виконано</i>
4	Затвердження завдання на кваліфікаційну роботу	27.12.2024 р.	<i>виконано</i>
<b>Основний етап</b>			
5	Розробка концепції кваліфікаційної роботи	13.01.2025 р.	<i>виконано</i>
6	Підбір та вивчення джерел інформації з напрямку дослідження. Огляд існуючих аналогів	20.01.2025 р.	<i>виконано</i>
7	Затвердження розширеної постановки завдання. Підготовка та подання керівникові розділу 1 кваліфікаційної роботи	10.03.2025 р.	<i>виконано</i>
8	Проектування. Підготовка та подання керівникові розділу 2 кваліфікаційної роботи	24.03.2025 р.	<i>виконано</i>
9	Підготовка доповіді для експертизи стану виконання кваліфікаційної роботи (проміжний контроль)	31.03-04.04.2025 р.	<i>виконано</i>
10	Реалізація. Підготовка та подання керівникові розділу 3 кваліфікаційної роботи	07.04.2025 р.	<i>виконано</i>
11	Підготовка та подання керівнику першого варіанту всієї кваліфікаційної роботи	14.04.2025 р.	<i>виконано</i>
12	Доопрацювання кваліфікаційної роботи з урахуванням зауважень керівника та представлення керівникові доопрацьованого варіанту кваліфікаційної роботи	21.04.2025 р.	<i>виконано</i>
<b>Завершальний етап</b>			
13	Представлення рукопису для перевірки на плагіат	28.04-04.05.2025 р.	<i>виконано</i>
14	Підготовка презентації та доповіді на передзахист	05.05-11.05.2025 р.	<i>виконано</i>
15	Передзахист кваліфікаційної роботи	12.05-16.05.2025 р.	<i>виконано</i>
16	Доопрацювання роботи за результатами передзахисту	19.05-06.06.2025 р.	<i>виконано</i>
17	Експертиза роботи керівником та зовнішнім експертом	09.06-15.06.2025 р.	<i>виконано</i>
18	Доопрацювання доповіді та презентації для захисту	09.06-15.06.2025 р.	<i>виконано</i>
19	Захист кваліфікаційної роботи	16.06-22.06.2025 р.	<i>виконано</i>

Керівник

Здобувач освітнього ступеня бакалавра

Сергій МІЧКІВСЬКИЙ

Володимир ЛЯШИНСЬКИЙ

*Ляшинський В.О. Комп'ютерна гра на Unity під назвою "Demonic Hunter" з використанням алгоритмів інтелектуальної поведінки ботів (комплексна робота)*

Пояснювальна записка кваліфікаційної роботи за спеціальністю 122 – Комп'ютерні науки (освітня програма – Комп'ютерні науки) СО Бакалавр. – ВНЗ «Університет економіки та права «КРОК», Навчально-науковий інститут інформаційних та комунікаційних технологій, кафедра комп'ютерних наук, Київ, 2025.

Розроблено гру в 2D форматі та піксель-арт, пост-апокаліптичному сетингу, з акцентом на ігрові механіки, атмосферу та унікальність стилю, з використанням алгоритмів інтелектуальної поведінки ботів та інструментів розробки Unity, Aseprite.

Ключові слова: платформер, Сюжет, 2D – платформер, апокаліпсис.

Рис. 47. Бібліограф.: 24 найм.

*Lyashinsky V.O. Computer game "Demonic Hunter" using algorithms of intelligent behavior of bots (complex work)*

Explanatory note of the qualification work in the specialty 122 – Computer Science (educational program – Computer Science) Bachelor's degree. – Higher Educational Institution “University of Economics and Law “KROK”, Educational and Scientific Institute of Information and Communication Technologies, Department of Computer Science, Kyiv, 2025.

The game was developed in 2D format and pixel art, in a post-apocalyptic setting, with an emphasis on game mechanics, atmosphere, and unique style, using algorithms for intelligent bot behavior and Unity and Aseprite development tools.

Keywords: platformer, Plot, 2D – platformer, apocalypse.

Fig. 47. Bibliography: 24 Items.

## ЗМІСТ

ВСТУП.....	6
РОЗДІЛ 1 ПОСТАНОВКА ЗАДАЧІ З РОЗРОБКИ 2D-ГРИ "DEMONIC HUNTER".....	8
1.1 Предметна область .....	8
1.2 Огляд аналогів .....	11
1.3 Постановка задачі.....	19
Висновок до розділу 1 .....	23
РОЗДІЛ 2 ПРОЄКТУВАННЯ .....	24
2.1 Діаграма варіантів використання Use Case Diagram.....	24
2.2 Діаграма діяльності Activiti Diagram.....	25
2.3 Діаграма класів Class Diagram .....	26
2.4 Опис архітектури програмного продукту.....	30
Висновок до розділу 2 .....	31
РОЗДІЛ 3 ПРОЄКТУВАННЯ .....	32
3.1 Дизайн Противників та Персонажів .....	32
3.2 Інтерфейс UI/UX.....	38
3.3 Дизайн наповнення рівнів та побічних предметів .....	43
3.4 Створення Анімацій до спрайтів .....	47
3.5 Дизайн предметів для взаємодії.....	49
Висновок до розділу 3 .....	51
ВИСНОВКИ .....	52
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ .....	53

## ВСТУП

*Актуальність теми.* Розробка відеоігор – одна з найцікавіших і багатообіцяючих областей сучасних інформаційних технологій. Вона об'єднує в собі багато сфер таких як: програмування, графічний дизайн, анімацію, створення ботів з штучним інтелектом. Це поєднання цих сфер і створює величезне різноманіття ігор, жанрів які є в наш час. Створення гри від задуму до повноцінного проекту, включаючи розробку механік, дизайну рівнів, персонажів та штучного інтелекту.

Актуальність теми пояснюється зростаючим попитом на незалежні інді-ігри та постійним удосконаленням технологій у сфері розробки відеоігор. Проєкт "Demonic Hunter" демонструє можливості невеликих команд створювати цікаві ігри, з оригінальними рішеннями які можуть зацікавити певну аудиторію гравців. Отже, у роботі буде розглянуто повний процес розробки гри: від концептуального задуму до його технічної реалізації з боку людини яка займалася графікою. Викладені результати можуть бути корисними для студентів, розробників-початківців та всіх, хто цікавиться створенням ігор на Unity.

*Мета дослідження.* Розробити 2D гру "Demonic Hunter" – платформер у пост-апокаліптичному стилі, який поєднує елементи виживання, бойових механік та інтерактивного середовища. Основний акцент зроблено на створенні унікальних алгоритмів поведінки ворогів за допомогою штучного інтелекту. Гра реалізована у стилі піксель-арт, що спрощує створення візуального наповнення у грі для не великих команд.

*Завдання на дослідження полягають в наступному;*

- проаналізувати існуючі аналоги гри;
- проаналізувати алгоритми інтелектуальної поведінки ботів;
- спроектувати елементи та архітектуру гри;
- створити оригінальний інтерфейс гри;
- розробити програмне забезпечення.

*Об'єктом дослідження* є ігровий процес платформера у пост-апокаліптичному сетингу.

*Предметом дослідження* є ігрова індустрія 2D піксель-арт платформерів з акцентом на ігрові механіки та алгоритми інтелектуальної поведінки ботів.

*Методи дослідження* це аналіз ринку ігор в такому жанрі, виокремлення декількох вже вдалих проектів, спроба відтворити свій візуальний стиль чи геймплейну частину. Аналіз алгоритмів інтелектуальної поведінки ботів.

*Практичне завдання.* Розроблена 2D гра на ігровому рушії Unity. В якій буде свій візуальний стиль також геймплей і впровадження інтелектуальної поведінки ботів.

*Структура роботи.* Кваліфікаційна робота складається зі вступу, трьох розділів, висновків та списку посилань (24 найменувань). Пояснювальна записка містить 47 рисунків. Загальний обсяг пояснювальної записки складає 55 сторінок, основний зміст викладено на 52 сторінках.

## РОЗДІЛ 1

### ПОСТАНОВКА ЗАДАЧІ З РОЗРОБКИ 2D-ГРИ "DEMONIC HUNTER"

#### 1.1 Предметна область

Платформер – це ігровий жанр в основі якого є проходження гри по так званих платформах, через перешкоди та супротивників. Гравець у платформері керує персонажем що може рухатися у чотирьох напрямках (вправо, вліво, вгору, вниз) рівні також можуть бути не тільки лінійно як у більш старих іграх. Основна ігрова зона прокладається платформами які можуть бути різних розмірів. Саме на такі платформи і виставляють дружніх ботів та супротивників. Протівники як правило володіють примітивним інтелектом, тому як тільки гравець потрапляє у поле зору (тригеру) моба той в свою чергу намагається зблизитися з гравцем та уразити його. Такі боти як правило не мають багато життів та швидко гинуть. Щоб ускладнити гру та створити більш цікаві елементи як правило використовують на рівні не тільки рядових ботів , але й босів які як правило знаходяться в кінці рівня щоб психологічно дати поняття головному герою що рівень добігає кінця. На протязі всього рівня потрібно не забувати збирати різні речі на рівні такі як : гроші , корисні модифікації які знадобляться в подальшому щоб покращувати свого героя.

Піксельна графіка – це стиль цифрового мистецтва, у якому зображення створюються та редагуються на рівні окремих пікселів. Вона використовує в іграх, анімації та цифровому дизайні, надаючи ретро-естетику, характерну для 8-бітних і 16-бітних консолей. Піксельна графіка проста у створенні, легко адаптується для різних роздільностей через що і користується попитом у інді розробників. Хоча здається що піксельна графіка не є популярною то це не так, хоча і великі компанії рідко використовують піксельну графіку понин на неї досі залишається великий, як від розробників так і для геймерів. Хоча здається що багато різного в пікселях не зробити , це здається тільки на перший погляд, є дуже багато віртуозів які створюють неймовірну графіку та атмосферу.

Головне не кількість пікселів, а поєднання та компоновання всіх елементів в одну композицію, що наче здається простою задачею проте це тільки здається.

Основні аспекти предметної області які розглядаються в цьому документі.

- план;
- жанр та стилістика;
- сюжетний контекст;
- ігровий процес та механіки;
- структура ігрового світу;
- візуальний стиль та атмосфера;
- технологічна реалізація;
- цільова аудиторія.

1. Жанр та стилістика- гра належить до жанру \*екшн-платформер , виживання та з елементами RPG\*. Стилiстично виконана в пiксель-арт форматi.

2. Сюжетний контекст - події розгортаються у світі, який постраждав від нашестя демонічних сил. Більшість цивілізації знищена або спотворена, міста перетворилися на руїни, а вцілілі люди змушені ховатися або боротися за своє існування. Головний герой – мисливець на демонів, який має власну мотивацію та ціль, пов'язану з винищенням нечисті та відновленням рівноваги.

3. Ігровий процес та механіки-гра поєднує декілька основних ігрових механік:

Платформер: стрибки, біг по платформах та подолання перешкод.

Бойова система: включає рукопашний бій, використання зброї та спеціальних навичок. Штучний інтелект ворогів демони можуть координувати атаки, ухилятися, змінювати тактику залежно від ситуації. Деякі вороги мають зграєву поведінку, уникають пасток або використовують оточення для переваги. Система розвитку персонажа ,покращення здібностей через досвід або спеціальні предмети, зброя та броня можуть бути модифіковані, спеціальні

навички, які дозволяють змінювати стиль гри (наприклад, приховані атаки, магичні здібності).

#### 4. Структура ігрового світу

Світ складається з різних регіонів: Міські руїни - залишки колишніх великих міст, захоплених демонами, підземелля та каналізація – небезпечні лабіринти, де ховаються сильніші вороги, лісиста місцевість - локація яка є переходом від штабу героя до міста, база - штаб в якому головний герой може відпочити щоб відновити сили, покращити зброю.

#### 5. Візуальний стиль та атмосфера

Гра має похмурий стиль із приглушеними кольорами, що підкреслює безнадійність світу. Оточення включає деталі розрухи – занедбані будівлі, покинуті машини, криваві сліди боїв.

#### 6. Технологічна реалізація – гра розробляється на базі рушія.

Алгоритми штучного інтелекту для створення складних та реалістичних ворогів. Фізичні механіки для руйнування оточення та бойових взаємодій. Оптимізація ресурсів для забезпечення плавного геймплею.

#### 7. Цільова аудиторія.

Гра орієнтована на любителів екшн-платформерів, RPG та виживання, а також фанатів пост-апокаліптичних світів та темного фентезі. Вона розрахована на середню складність та потребує тактичного підходу до боїв.

Розвиток дизайну відеоігор з 2000 по 2025 рік охоплює значні технологічні прориви, еволюцію візуального стилю, покращення ігрових механік та нові підходи до створення інтерактивного досвіду. Умовно цей розділ можна поділити на три часові проміжки.

#### 2000–2010: Перехід до тривимірної графіки та розширення жанрів

Початок 2000-х років ознаменувався масовим переходом від 2D-графіки до 3D, завдяки розвитку апаратного забезпечення та зростанню популярності рушіїв, таких як Unreal Engine та id Tech. Ключові особливості періоду:

Розвиток 3D-графіки – з'явилися детальніші текстури, реалістичні моделі персонажів та фізично коректне освітлення. Еволюція відкритого світу

– такі ігри, як Grand Theft Auto III (2001), започаткували тренд відкритих світів із свободою дій. Штучний інтелект у NPC – покращилися системи поведінки ворогів і союзників (Half-Life 2, 2004).

Масове впровадження онлайнного мультиплеєра – ігри, як World of Warcraft (2004), стали знаковими завдяки масштабним мережевим світам.

Стилізація та художній підхід – деякі проєкти, такі як Okami (2006) та Team Fortress 2 (2007), продовжували експериментувати з візуальним стилем.

Технології: DirectX 9, OpenGL 2.0, Havok Physics, Unreal Engine 3.

## 1.2 Огляд аналогів

Культові ігри цього часового відрізка гейм індустрії- Grand Theft Auto: San Andreas (2004), Half-Life 2 (2004), The Elder Scrolls IV: Oblivion (2006)

Grand Theft Auto: San Andreas (2004) – революційний відкритий світ із глибокою взаємодією, свободою дій і нелінійним геймплеєм. На рис 1.1 знаходиться скріншот гри на якому можливо побачити ботів з інтелектом, та можливість взаємодії з оточуючим світом.



*Рисунок 1.1 – Кадр з гри GTA S. A. 2004*

*Джерело [14]*

Half-Life 2 (2004) – просунутий штучний інтелект ворогів, фізично реалістичне оточення (завдяки рушійу Source) та кінематографічний сюжет.

На рис 1.2 зображено ботів які за допомогою AI можуть координувати атаки, та мають примітивну тактику. Рушій що використовується в грі дає змогу гравцю взаємодіяти з навколишнім середовищем, предметами, та додає функцію руйнування об'єктів.



*Рисунок 1.2 – Кадр з гри Half-Life 2 2004р.*

*Джерело [15]*

The Elder Scrolls IV: Oblivion (2006) – великий відкритий світ, процедурно згенеровані локації, нелінійність квестів та динамічна поведінка NPC. ШІ персонажі мають власні розклади, займаються справами, реагують на дії гравця.

Фотореалістична графіка (на той час) – деталізовані текстури, освітлення та фізичний рушій для взаємодії з предметами. рис. 1.3

2010–2020: Фотореалізм, процедурна генерація та інді-революція

У 2010-х роках розробники все більше фокусувалися на фотореалізмі, розширенні можливостей відкритих світів і більшій інтерактивності середовища. Ключові особливості періоду: фотореалізм, процедурна генерація світів, розвиток VR,

Процедурна генерація світів – Minecraft (2011) та No Man’s Sky (2016) продемонстрували потенціал алгоритмічного створення ігрових світів.



*Рисунок 1.3 – Скрін з гри The Elder Scrolls IV 2006р.*

*Джерело [16]*

Фотореалізм – завдяки новим рушіям, як CryEngine 3, Frostbite та Unreal Engine 4, ігри стали надзвичайно реалістичними (Red Dead Redemption 2, 2018). Процедурна генерація світів – Minecraft (2011) та No Man’s Sky (2016) продемонстрували потенціал алгоритмічного створення ігрових світів.

Розвиток VR-технологій – Half-Life: Alyx (2020) показала новий рівень взаємодії у віртуальній реальності. Інді-революція – завдяки Unity та Godot, невеликі студії почали створювати високоякісні проекти (Hollow Knight, Celeste, Undertale). Ігрові механіки та наратив – The Last of Us (2013) та The Witcher 3 (2015) підняли рівень сторітелінгу та кінематографічного підходу.

Технології: DirectX 11-12, Vulkan API, Unreal Engine 4, Unity, PhysX, AI-driven NPCs.

Культові ігри цього часового відрізка гейм індустрії-Red Dead Redemption 2 (2018),No Man's Sky (2016),The Last of Us Part II

Red Dead Redemption 2 (2018) рис 1.4. Революційний фотореалізм – детальні моделі персонажів, реалістична міміка, унікальна анімація для кожної дії. Динамічний світ – зміна погоди, поведінка NPC залежить від часу доби та вчинків гравця. Глибокий рівень деталізації – кожна дрібниця, від слідів у снігу до шерсті тварин, ретельно опрацьована.



*Рисунок 1.4 – Скрін з гри Red Dead Redemption*

*Джерело [17]*

No Man's Sky (2016, оновлення до 2020) рис 1.5. Процедурна генерація – мільярди унікальних планет, кожна з яких має свої біоми, флору, фауну та погодні умови. Унікальний художній стиль – яскраві кольори, інопланетні пейзажі, що натхненні науковою фантастикою 70-х. Живий всесвіт – розширені можливості дослідження, розбудови баз та інтерактивного світу.



*Рисунок 1.5 – Скрін з гри No Man's Sky*

*Джерело [18]*

The Last of Us Part II (2020) рис 1.6. Неймовірна деталізація персонажів – передова технологія сканування облич, складна анімація мимики та рухів. Реалістичне освітлення та фізика – технологія Global Illumination створює природні тіні та світлові ефекти.

Кінематографічний дизайн рівнів – ігрові локації поєднують інтерактивність та режисерський підхід до постановки сцен. Ці ігри суттєво вплинули на розвиток дизайну рівнів, графіки та інтерактивного оточення у відеоіграх.

2020–2025: AI, хмарний геймінг та нові стандарти якості

У 2020-х роках ігровий дизайн став ще більш реалістичним завдяки розвитку штучного інтелекту, машинного навчання та хмарних технологій.

Культові ігри цього часового відрізка гейм індустрії- Cyberpunk 2077, Horizon Forbidden West (2022), The Matrix Awakens: An Unreal Engine 5 Experience



*Рисунок 1.6 – Скріншот з гри The Last of Us  
Джерело [19]*

Штучний інтелект та генеративні технології – NPC у сучасних іграх мають адаптивну поведінку, а AI допомагає у створенні рівнів, анімацій та навіть сценаріїв. Рей-трейсинг у реальному часі – нові покоління графічних карт (RTX) дозволяють створювати динамічне освітлення та реалістичні тіні. Хмарний геймінг – сервіси, як GeForce Now і Xbox Cloud Gaming, дозволяють грати у складні проєкти без потреби в потужному залізі. Перехід до процедурного дизайну AI-генерації контенту – ігрові світи стають все більш інтерактивними та унікальними завдяки алгоритмічним підходам. Гіперреалізм та стильові експерименти – поєднання високоякісного фотореалізму та художніх стилізацій для створення унікального візуального стилю.

Технології: DirectX 12 Ultimate, Unreal Engine 5, Machine Learning AI, Neural Networks in Games. Ігри які значно прогресували в графіці та дизайні завдяки новим технологіям.

Cyberpunk 2077 (2020) рис 1.7. Рей-трейсинг у реальному часі – використання технології RTX для динамічного освітлення, відображень та

тіней, що значно підвищує візуальний реалізм. Інтерактивні елементи світу – AI-система покращує поведінку NPC, створюючи адаптивні взаємодії з гравцем.



*Рисунок 1.7 – Скріншот з гри Cyberpunk 2077*

*Джерело [20]*

Horizon Forbidden West (2022) рис 1.8. Фотореалізм – значний прогрес у деталях персонажів, текстурах та анімаціях, особливо у відкритому світі з багатими пейзажами. Динамічне освітлення та тіні – завдяки Unreal Engine 5 та новим технологіям, освітлення природно змінюється в залежності від часу доби та погодних умов.

The Matrix Awakens: An Unreal Engine 5 Experience (2021) рис 1.9. Графіка нового покоління – тестування можливостей Unreal Engine 5, включаючи нові технології освітлення, текстур та геометрії, що дозволяють створювати неймовірно деталізовані міські пейзажі. Реалістичні анімації – поєднання сканування обличчя та рухів створює відчуття максимальної реалістичності персонажів.



*Рисунок 1.8 – Скріншот з гри Horizon Forbidden West  
Джерело [21]*



*Рисунок 1.9 – Скріншот з гри The Matrix Awakens: An Unreal Engine 5  
Experience  
Джерело [22]*

З 2000 до 2025 року ігровий дизайн пройшов шлях від примітивних 3D-ігрових світів до високо-якісних просторів з детально продуманими концептуально середовищами в грі. Графічні технології та дизайн відеоігор за останнє десятиліття зробили стрибок від реалізму до гіпер-реалізму, від інтерфейсів до повної інтерактивності з оточенням. Важливими факторами розвитку стали технологічний прорив у графіці, AI процедурна генерація та розвиток інструментів для інді-розробників, що полегшило їх роботу у сфері дизайну та і взагалом в розробці, проте ШІ ще досі інструмент людини що не може робити сам в процесі розробки проте допомагає розробникам через що і користується популярністю. Технології, такі як штучний інтелект, процедурна генерація контенту та новітні графічні рушії, змінили спосіб створення та сприйняття віртуальних світів, дозволяючи розробникам створювати більш живі, адаптивні і реалістичні ігрові досвіди. Це не лише збагачує ігровий процес, а й наближає ігри до кінематографічного рівня якості.

### **1.3 Постановка задачі**

Цілі проєкту "Demonic Hunter" з точки зору UI/UX розробника це розробка інтуїтивного та функціонального користувацького інтерфейсу. UI/UX у "Demonic Hunter" – створення зрозумілого, адаптивного та естетично привабливого інтерфейсу, що забезпечує комфортне занурення у гру. Гравець має легко орієнтуватися в меню, швидко отримувати необхідну інформацію та безперешкодно взаємодіяти з елементами інтерфейсу. Зокрема, необхідно продумати логіку розташування HUD-елементів: показники здоров'я, витривалості, боєприпасів, наявних ресурсів та поточних завдань. Важливо досягти балансу між інформативністю та ненав'язливістю, щоб UI не відволікав від ігрового процесу.

Адаптивність та зручність навігації. UI повинен бути адаптованим для різних роздільних здатностей екранів Меню та інші навігаційні елементи мають бути логічно структуровані: головне меню, екран інвентаря, та

налаштувань повинні бути доступними без зайвих складнощів. Особливу увагу слід приділити розробці швидких комбінацій клавіш.

Важливим аспектом UX є забезпечення чіткого зворотного зв'язку під час дій гравця. Кнопки та інші інтерактивні елементи повинні мати відповідні анімації натискання, звуковий супровід та візуальні ефекти, які дають зрозуміти, що команда виконана. Крім того, необхідно реалізувати системи підказок, які будуть допомагати гравцю адаптуватися до механік гри, але при цьому не перевантажувати екран зайвою інформацією.

Дизайн та естетика інтерфейсу. Враховуючи атмосферу постапокаліпсису, UI повинен мати відповідний стиль – темні тони, грубі текстури, елементи, що нагадують старі термінали або технології виживання. Важливо, щоб UI доповнював загальну стилістику гри, анімації були плавними, а ефекти підкреслювали напружену атмосферу. Використання мінімалістичних елементів із чітким контрастом допоможе створити виразний та ефективний дизайн.

Оптимізація та продуктивність. Інтерфейс повинен бути не лише красивим, а й оптимізованим для швидкої роботи навіть на слабких пристроях. Використання ефективних алгоритмів відображення графічних елементів, мінімізація зайвих анімацій та правильна організація ресурсів допоможуть забезпечити плавність гри без збоїв та затримок.

Тестування та покращення UI/UX. Перед фінальним випуском гри необхідно провести тестування користувацького досвіду. Гравці повинні випробувати інтерфейс у різних умовах, щоб виявити можливі незручності або проблеми з навігацією. Отриманий зворотний зв'язок дозволить внести корективи та покращити загальне сприйняття UI/UX у грі "Demonic Hunter".

Створення стилю піксельного малювання, що передаватиме похмурість, тяготу та безнадійність того світу, який поглинули декілька критичних світових подій. Гра буде зображуватися як максимально депресивний, тяжкий і похмурий світ у стилістиці піксель-арту, з акцентом на деталізацію занедбаних локацій, зруйнованих будівель, запустілих міст і темних лісів.

Колірна палітра складатиметься переважно з темних, холодних та затемнених відтінків, щоб передати всю важкість та напруженість, що витає в атмосфері планети, поглинутої хаосом.

Особливу увагу буде приділено дизайну персонажів і ворогів: головний герой виглядатиме як солдат в зброї який готовий до викликів світу який є виснаженим, зношеним після катастрофи, а мутанти, демони та бандити матимуть загрозливий вигляд, підкреслений агресивними рисами та викривленими формами. Візуальна стилістика та дрібні анімаційні деталі, такі як мерехтіння неонових вогнів у зруйнованих містах, вогонь у багаттях виживших або дрижання холодного повітря, допоможуть посилити ефект занурення гравця у цей світ.

Не лише створення красивої картинки, а й у формуванні емоційного впливу на гравця через атмосферу безвиході та боротьби за виживання. Завдяки цьому "Demonic Hunter" зможе донести свою історію та підкреслити напругу і небезпеку цього сповненого жахів світ

Гра "Demonic Hunter" належить до жанру 2D-платформерів та об'єднує елементи виживання, бойові механіки та інтерактивне середовище. Предметна сфера гри "Demonic Hunter" охоплює загальну концепцію постапокаліпсису, у якому мешкають демони, мутанти та інші ворожі створіння, яких необхідно ліквідувати для виживання.

Її події розгортаються у світі зануреному у хаос після ряду подій які відбулися протягом не великого проміжку часу що помістило життя на планеті в хаос, після чого почали процвітати та відчули розквіт волі різні мародери та вандали. Були як і поодинокі бандити так і цілі угруповання які керувалися більш розумнішими злочинцями які використовували свою структуру банди для наживи. Також і розгорілась вірусна небезпека, бушували віруси як і хвороб що можуть і вбивати дуже швидко так і віруси що вражали гени мутацією, через що живі створіння мутували та лишалися розуму. Гравцю доведеться досліджувати руїни цивілізації, битися з противниками та використовувати доступні ресурси для виживання.

У процесі виконання проекту вивчаються основи геймдизайну, середовище Unity, методи створення 2D-анімацій та інтеграція ігрових об'єктів. Проект виконується групою з двох розробників і спрямований на створення класичного 2D-екшн-платформера. Основний акцент зроблено на дизайні персонажів, супротивників і рівнів, без впровадження інноваційних рішень. Головною метою є забезпечення якісного геймплейного досвіду та збалансованої механіки гри. Відео ігри в наш час є важливим компонентом сучасної цифрової культури. Тому ми вирішили розробити власну гру, дивлячись на референси ігор які сподобалися нашій команді та додати своїх деталей.

Кваліфікаційна робота передбачає повний цикл розробки та впровадження основних UI/UX рішень для гри "Demonic Hunter". Головні завдання, які необхідно вирішити в рамках цієї роботи:

Проектування інтерфейсу гри, включаючи розташування основних HUD-елементів, меню, інвентаря та карт. Розробка адаптивного дизайну, що забезпечить зручність гри на різних пристроях та типах керування.

Оптимізація взаємодії гравця із системою, включаючи розробку підказок, анімацій зворотного зв'язку та навігаційних рішень. Візуальне оформлення користувацького інтерфейсу, що відповідатиме похмурій постапокаліптичній тематиці гри. Реалізація прототипів та їх тестування для покращення користувацького досвіду. Оптимізація продуктивності UI для мінімізації навантаження на систему.

Результатом роботи стане повноцінний інтерфейс гри, що буде не лише функціональним, але й гармонійно вписуватиметься в загальну атмосферу гри "Demonic Hunter". "Demonic Hunter" – це гра, яка поєднує динамічний бойовий процес, атмосферний пост-апокаліптичний світ та інтелектуальних ворогів. Вона пропонує унікальні виклики для гравців, які люблять глибокі бойові механіки та продуману систему розвитку персонажа.

## Висновок до розділу 1

Розвиток дизайну відеоігор з 2000 до 2025 року характеризується значними технологічними та концептуальними змінами, які вплинули на візуальний стиль, ігрові механіки та взаємодію з оточенням.

Перехід до 3D-графіки (2000–2010): Початок 2000-х ознаменувався переходом від 2D до 3D, покращенням текстур, фізичних рушіїв та інтерактивності. Значним проривом стали відкриті світи (GTA III), розширені можливості штучного інтелекту NPC (Half-Life 2) та розвиток онлайн-мультиплеєра (World of Warcraft).

Фотореалізм і процедурна генерація (2010–2020): У цей період індустрія відеоігор активно впроваджувала технології фотореалізму (Red Dead Redemption 2), процедурну генерацію ігрових світів (No Man's Sky, Minecraft) та розширені можливості штучного інтелекту NPC. Інді-ігри отримали потужний розвиток завдяки доступним рушіям (Unity, Unreal Engine 4), що сприяло появі унікальних проєктів (Hollow Knight, Celeste, Undertale).

Штучний інтелект та хмарний геймінг (2020–2025): Сучасні ігри використовують AI для покращення взаємодії з NPC, автоматичної генерації рівнів та динамічного освітлення (Cyberpunk 2077, Horizon Forbidden West). Рей-трейсинг та фотореалізм стали стандартом завдяки новим графічним рушіям (Unreal Engine 5). Хмарний геймінг розширив доступ до високобюджетних проєктів незалежно від потужності обладнання гравця.

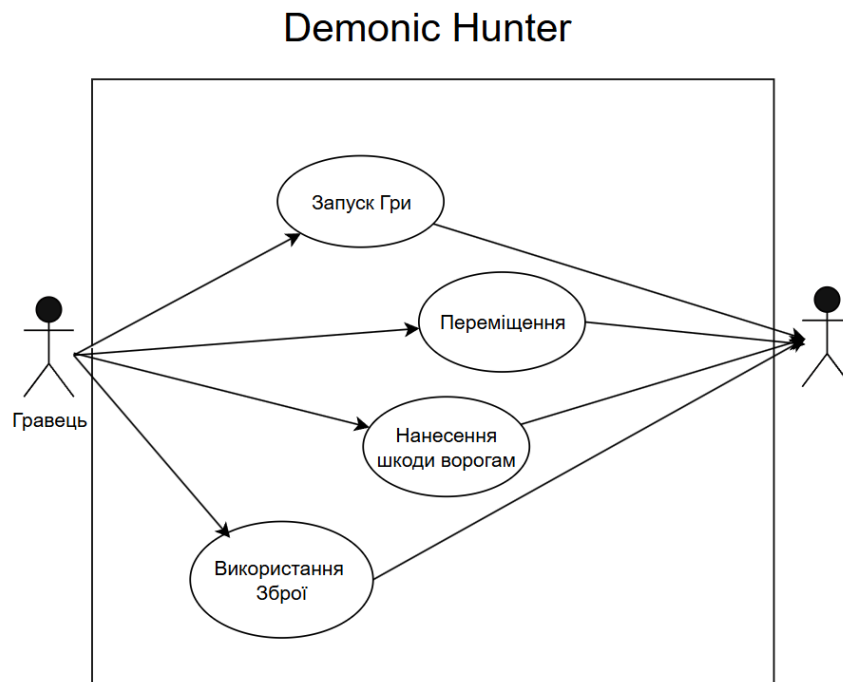
Значення технологічного прогресу: Відеоігри стали більш інтерактивними, деталізованими та кінематографічними. Впровадження AI, процедурної генерації та високоякісних текстур дозволило створювати глибші ігрові світи. Хоча штучний інтелект значно покращив розробку, він залишається інструментом, що допомагає розробникам, а не повністю автоматизує процес створення ігор.

Загалом, еволюція ігрового дизайну демонструє перехід від традиційних підходів до комплексних технологічних рішень, що робить відеоігри більш захопливими та реалістичними.

## РОЗДІЛ 2 ПРОЄКТУВАННЯ

### 2.1 Діаграма варіантів використання Use Case Diagram

На рис. 2.1 діаграма прецедентів (Use Case Diagram).



*Рисунок 2.1 – Діаграма Use Case  
Джерело: розроблене автором*

Вказівки до рис. 2.1:

- гравець: відзначений як основний об'єкт;
- коло: символізує функцію, яку може виконати користувач;
- стрілка, спрямована до функції: вказує на запит, зроблений користувачем;
- стрілка від функції: ілюструє результат для гравця в рамках проєкту.

Як користувач, під час запуску гри ми користуємося певними алгоритмами. Починаючи з головного меню, користувач може налаштувати параметри та обрати доступні інтерфейси. Після натискання кнопки "Грати"

розпочинається завантаження рівня. Коли користувач переміщується, подаючи команди гри, вона відображає результати. Аналогічним чином можна пояснити дії зі зброєю: при натисканні відповідної клавіші запит обробляється, після чого на екрані з'являється постріл.

## 2.2 Діаграма діяльності Activity Diagram

На рис. 2.2 представлено діаграму активності (Activity Diagram).

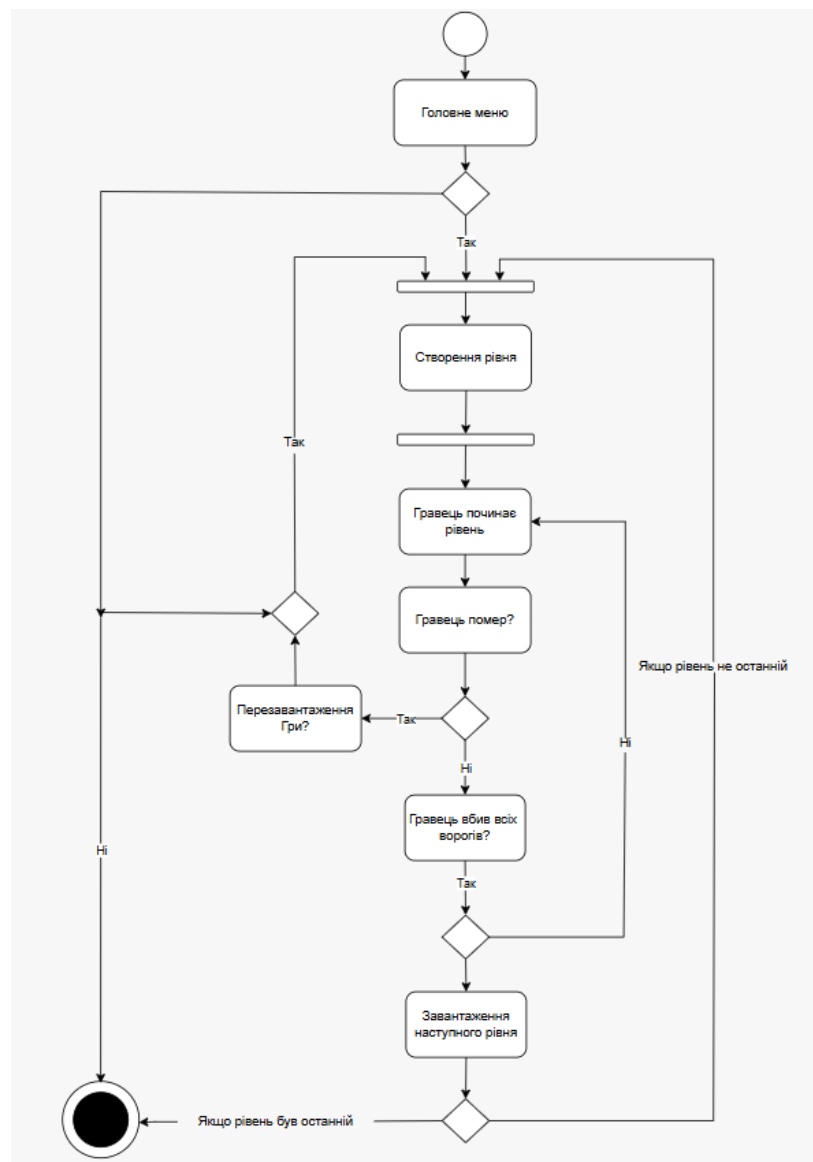


Рисунок 2.2 – Activity Diagram

Джерело: розроблене автором

Позначення в рис. 2.2:

- чорний круг представляє гравця на початку його дій;
- ромб у діаграмі вказує на перевірку умов для двох станів;
- стрілочки демонструють послідовність операцій відповідно до дій гравця;
- прямокутник показує вже визначені дії, які гравець може активувати;
- білий круг у рамці означає завершення всіх можливих дій та алгоритмів.

Ця діаграма рис. 2.2 пояснює, що користувач бачить стартове меню, де може обрати "Почати гру" або "Вийти на робочий стіл" через відповідні кнопки. Наступний етап передбачає натискання кнопки "Play". Потім гравець потрапляє на попередньо підготовлений рівень з ворогами та іншими викликами. Після початку гри для гравця активуються два можливі алгоритми:

- Якщо гравець помирає (не справляється з викликами), відображається меню, де він може натиснути кнопку, щоб розпочати рівень спочатку.

Якщо гравцеві вдалося перевершити ворогів, на локації з'являється вихід, при дотику до якого персонаж переходить на наступний рівень. Далі є два варіанти розвитку подій: - Якщо рівень не останній, запускається алгоритм створення нового рівня. - Якщо рівень був останнім, гра завершується.

### 2.3 Діаграма класів Class Diagram

На рис. 2.3 представлено діаграму класів.

Коротке пояснення цих змінних у кодї гри. У кодї Hero, до якого підв'язані два дочірніх класи, реалізовано такий принцип роботи: Anim відповідає за анімації персонажа в залежності від ситуації. Змінна JumpForce визначає силу стрибка, Rigidbody 2D дозволяє гравцеві рухатися. Методи, які відповідають за анімації сердець і завантаження відповідних меню, коли серця закінчуються, прописані в PlayerHealth. Gun створює префаб – заготовлений об'єкт на сцені – і точку, через яку відбувається стрілянина.

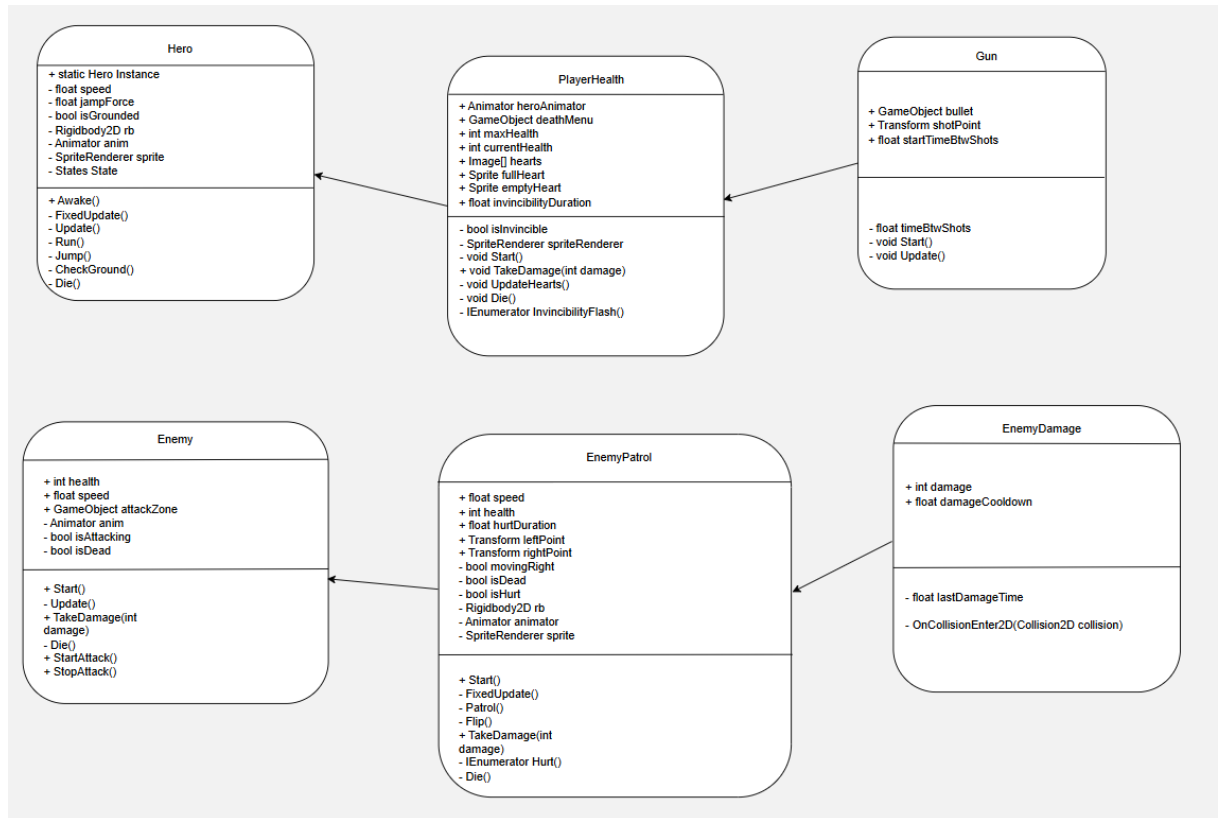


Рисунок 2.3 – Діаграма Класів

Джерело: розроблене автором

Позначки до рис. 2.3:

- заокруглений квадрат символізує набір перемінних та функцій;
- плюс вказує, що змінна є публічною;
- мінус позначає, що клас є приватним;
- стрілочка означає прив'язку до батьківського об'єкта.

У класі Enemy можна побачити дві змінні для атаки: одна активується, коли ворог бачить гравця, інша – коли гравця не видно і ворог патрулює. Також у ворога вказана кількість здоров'я та завдану шкоду. Патрулювання здійснюється між двома невидимими для гравця точками.

Детальніше розглянемо класи Hero, PlayerHealth та Gun: Параметри головного героя Hero:

- static Hero Instance об'єкт героя на сцені, щоб скрипти могли звертатися до нього.

- float speed задає швидкість героя.
- float jumpForce регулює силу стрибка героя.
- bool isGrounded перевіряє, чи герой стоїть на землі.
- Rigidbody2D rd додає фізику переміщення по сцені.
- Animator anim відповідає за анімації в залежності від дій гравця.
- SpriteRenderer sprite використовує спрайт для відображення на сцені.
- States State перераховує стани героя.
- Awake() перевіряє появу героя на сцені при натисканні кнопки Play.
- FixedUpdate() перевірка бігу персонажа при натисканні клавіш.
- Update() перевірка натискання клавіш.
- Run() функція для бігу гравця.
- Jump() функція для стрибка.
- Die() виконується у разі смерті героя відсутності здоров'я. Пункти про

здоров'я PlayerHealth:

- Animator heroAnimator анімації, пов'язані з отриманням шкоди. -  
GameObject deathMenu меню при смерті.

- int maxHealth максимальне здоров'я персонажа.
- int currentHealth поточне здоров'я персонажа.
- Image[] hearts масив сердечок у інспекторі.
- Sprite fullHeart, emptyHeart спрайти повного та порожнього сердечка,

налаштовуються через інспектор.

- float invincibilityDuration затримка після шкоди, щоб гравець міг відійти від противника.

- Start() встановлює максимальне здоров'я у графічному інтерфейсі.
- TakeDamage(int damage) перевіряє кількість отриманої шкоди.
- UpdateHearts() оновлює серця в інтерфейсі згідно зі шкодою.
- Die() перевірка смерті при відсутності здоров'я.

- `InvincibilityFlash()` забезпечує спалах героя при отриманні шкоди.

#### Зброя Gun:

- `GameObject bullet` прив'язаний об'єкт кулі через інспектор.
  - `Transform shotPoint` точка пострілу, прив'язана до зброї.
  - `float startTimeBtwShots` затримка між пострілами.
  - `Start()` – ініціалізація таймера для пострілу.
  - `int health` кількість здоров'я ворога.
  - `GameObject attackZone` зона атаки противника.
  - `Animator anim` – підключення анімацій.
  - `bool isAttacking` перевірка стану атаки ворога. - `bool isDead` перевірка смерті ворога.
  - `Start ()` налаштування противника для програвання анімації коли починається сцена;
  - `Update ()` тут йде перевірка на ситуацію в сцені та зміна анімацій від події;
  - `TakeDamage(int damage)` перевірка на отримання шкоди від гравця;
  - `StartAttack ()` тут позначається атака ворога коли нас бачить;
  - `StopAttack ()` зупиняємо атаку для патрулювання коли не бачим героя;
- Поясним як працює патрулювання `EnemyPatrol`:
- `float speed` зазначимо швидкість для противника;
  - `int health` скільки в противника буде здоров'я;
  - `float hurtDuration` тут йде невеличка затримка в анімація, щоб гравець побачив зміну анімацій в противника при отриманні шкоди;
  - `Transform leftPoint, rightPoint` тут вказані 2 невидиміх для гравця точки по яким пересувається противник;
  - `bool movingRight` тут задаємо параметр для свапу спрайта по Y в залежності в яку сторону йде противник;
  - `bool isDead, bool isHurt` позначимо в цих двох пунктах статуси ворога при якому він отримає шкоду і помре;
  - `Rigidbody2D rb` додаємо противнику фізику для переміщення;

- Animator animator, SpriteRenderer sprite в першому позначаємо анімації які перед цим настроїли в аніматорі інший буде Картинка яка позначатиметься на сцені;

- Start () налаштування для патруля;
- FixedUpdate () тут йдуть дві точки для переміщення противника по ним;
- Patrol () коли противник бачить героя він зупиняється і атакує, а коли втрачає його с поля зору повертається до патрулювання;

- Flip () відповідає за зміну напрямку ворога від сторони в яку він прямує;
- TakeDamage(int damage) перевірка на отримання шкоди;
- Hurt () оброблення шкоди від гравця;
- Die () ворог помирає коли закінчується життя;

Лишилось пояснити, як працює шкода від ворога EnemyDamage:

- int damage позначаємо кількість шкоди ворогу;
- float damageCooldown робим затримку між атаки що противник не атакував постійно

- OnCollisionEnter2D(Collision2D collision) тут позначений метод при якому ворог наносить шкоду персонажу, якщо він зітнеться з персонажем;

## 2.4 Опис архітектури програмного продукту

Все починається з рушія Unity, де зосереджені всі скрипти, ассети та звуки. Налаштування сцени в проєкті стартує з наявності чотирьох локацій: три з них бойові, а одна призначена для відпочинку. Конфігурація сцени здійснюється через скрипт Parallax, що відповідає за налаштування кожного шару, щоб при русі героя задній фон рухався з різною швидкістю, створюючи ефект глибини.

Після запуску гри гравець бачить фон із кнопками та логотипом гри. У меню є п'ять кнопок: play для початку гри; options для налаштування гучності звуків і музики; save відкриває меню для збереження гри або його закриття; vaccine демонструє мету гри – знайти кейс із антивірусом і зібрати склянки; quit для виходу з гри. Головний персонаж має один батьківський об'єкт, до

якого приєднано п'ять дочірніх об'єктів, кожен зі своєю роллю: основний об'єкт Hero, на якому прикріплені важливі скрипти - Hero.cs, PlayerInventory, PlayerHealth.cs, Inventory.cs; дочірній об'єкт з усіма анімаціями героя; точка плеча Shoulder, до якої прикріплено руки. Спрайт рук обертається за напрямком миші. Зброя, до якої прив'язаний скрипт Gun.cs; там налаштовані об'єкт кулі та точка вильоту кулі. Таким чином працює базовий алгоритм руху та стрільби персонажа. Персонаж також має інтерфейс з кнопкою для призупинення гри або переходу в меню. На екрані гравця видно рюкзак з інвентарем: коли натискаєш на іконку рюкзака, відкривається меню з п'ятьма слотами, також видно здоров'я гравця та кількість зібраних монет.

Кожен ворог складається з трьох об'єктів: основний об'єкт із прикріпленими скриптами - Enemy.cs, EnemyPatrol.cs, EnemyDamage.cs; об'єкт спрайта з анімаціями; зона атаки – це невелика площа, при вході в яку ворог починає атакувати гравця, а при виході гравця з цієї зони продовжує патрулювати. Перехід між локаціями відбувається через спеціальні зони на сцені, візуально позначені інтерактивними елементами та предметами. У кожній локації будуть противники й монети для збору. Гравець рухатиметься вперед, доки не знайде кейс із вакциною.

## **Висновок до розділу 2**

У цьому розділі було розглянуто важливі діаграми, що демонструють архітектуру гри, діаграму Use Case, діаграму діяльності Activiti, діаграма Class. Діаграма варіантів використання відображає взаємодію гравця з основними функціями гри, наприклад, початок гри, управління інвентарем та збереження прогресу. Діаграма активності описує послідовність дій гравця, зокрема переміщення між локаціями або взаємодію з ворогами.

Діаграма класів представляє структуру базових класів проекту та їхні зв'язки. Усі ці діаграми сприяють кращому розумінню логіки роботи гри та взаємодії її ключових елементів.

## РОЗДІЛ 3

### ПРОЄКТУВАННЯ

#### 3.1 Дизайн Противників та Персонажів

У процесі роботи над проектом моєю основною задачею було створення візуального стилю гри, зокрема дизайну і оформлення ігрових персонажів та противників. Я прагнув досягти гармонії між ігровою механікою та візуальним сприйняттям, щоб герой і антагоніст виглядав яскраво, впізнавано та відповідав загальній атмосфері гри. У цьому розділі представлено головних персонажів гри, а також різноманітних противників, з якими зустрінеться гравець. Кожен герой має свій унікальний зовнішній вигляд, характерні риси та особливості стилю. Противники ж оформлені з урахуванням того, щоб їх легко можна було ідентифікувати у геймплеї та водночас зберігалася естетична єдність. Під час розробки я звертав увагу на колористику, деталі та анімаційну адаптацію. Було створено кілька варіантів образів, після чого обрано найкращі відповідно до тестування та відгуків.

Герої та допоміжні фігури . Ми наведемо готові модельки персонажів та прототипи які ми також розглядали на роль текстур для тих чи інших юнітів. Приклади спочатку ми наведемо приклади головного героя та його першого чергового прототипу, потім побачим дружнього персонажа , нижче буде наведено приклади ворогів які ми можемо зустріти у грі та головного антагоніста гри на якому зав'язується сюжет. В цьому пункті ми розберемо дружніх персонажів та головного.

На рис. 3.1 зображено прототип героя який ми хотіли використати у грі але він на не підійшов ні візуально, ні концептуально. Цей персонаж не підходив під історію яку задали для головного героя, він не схожий на колишнього професійного військового який готовий приймати виклики у вимірі який поглинув хаос.

На рис 3.2 вже затверджений головний герой якого ми використовуємо в грі він більше схожий на професійного військового ніж прототип цей головний

герой справляє враження сильного та підготовленого військовослужбовця який готовий прийняти виклики які йому кине навколишнє середовище. Колишній солдат, який служив в армії, а згодом, після багатьох років служби, отримав високе звання. На момент подій у грі був звільнений і відгукнувся на пропозицію яку пропонувала адміністрація та армія разом з лікарями щодо допомоги в поверненні важливого вантажа назад за достойне винагородження.



*Рисунок 3.1 – Прототип головного героя*

*Джерело: розроблене автором*



*Рисунок 3.2 – Головний герой*

*Джерело: розроблене автором*

Спочатку персонаж (рис. 3.3) розроблявся як якийсь найманець який нам трапляється по ходу гри але ми вирішили що така роль за-мала для нього і він став продавцем в магазині. Він так і залишився як і планувалось досвідченим мандрівником, просто він після травми яка заважала йому займатись колишніми справами вирішив почати торгувати різними речами які можуть знадобитися у екстримальних умовах бо він прекрасно знав як колишній найманець що завжди необхідно у тих умовах і що важко отримати, тож розпочав торгувати.



*Рисунок 3.3 – Продавець на базі відпочинку*

Джерело: розроблене автором

На рис. 3.4 зображено прототип головного антагоніста який мав бути у гри проте він також не підійшов нашому ігровому всесвіту по історії. Він не виглядає як головний суперник, а якщо і виглядає то як чаклун проте в нашому вимірі немає магії.

На рис 3.5 зображено затвердженого суперника який є нашим фінальним суперником який виглядає загрозово. Цей герой має дуже сильні фізичні показники це мутована в колишньому людина з генами летючої миші. Головному герою доведеться подолати його бо він буде знаходитися біля

вантажу який втратила армія, і за яким полює наш герой, саме цей мутант в збив груз з цінним вантажем.



*Рисунок 3.4 – Прототип головного анти-героя*

*Джерело: розроблене автором*



*Рисунок 3.5 – Головний анти-герой*

*Джерело: розроблене автором*

На рис, 3.6 наведено прототип який ми розглядали на роль головного суперника, але він нам не так сильно сподобався як той якого ми затвердили.

Суперники без історії (рис. 3.7-3.9), які просто стануть на заваді нашому герою.



*Рисунок 3.6 – Вовк мутант  
Джерело: розроблене автором*



*Рисунок 3.7 –Бандит  
Джерело: розроблене автором*



*Рисунок 3.8 –Гоблін*

*Джерело: розроблене автором*



*Рисунок 3.9 –Зомбі-бомбун*

*Джерело: розроблене автором*

## 3.2 Інтерфейс UI/UX

В цьому розділі можна буде ознайомитись з елементами ігрового UI - головним меню, кнопками, панеллю здоров'я, а також іншими інформаційними вікнами та елементами управління на малюнках –



*Рисунок 3.10 – Фрагмент головного меню*

*Джерело: розроблене автором*

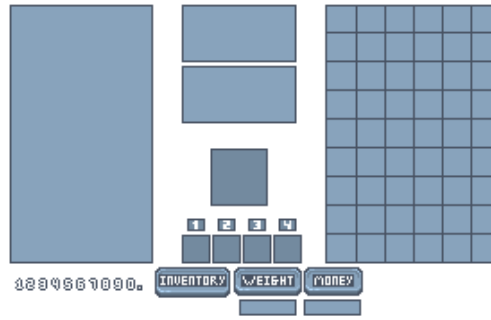
На рис. 3.2 показано лого проекту та кнопки інтерфейсу їх приблизне розташування та зовнішній вид. У фінальній версії ці кнопки матимуть анімацію натискання, а за кожною з них буде закріплений певний функціонал.

На рис. 3.12 ми бачимо інвентар в якому будуть розміщатися предмети та будуть писатися кількість грошей, та рахуватиметься вага.

На рис. 3.13 інвентар в якому будуть розміщатися предмети з об'єктів з якими можна взаємодіяти (бочки кейси, сундуки)

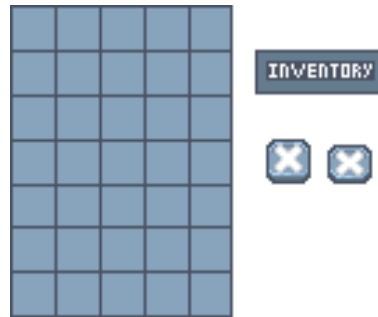
На рис. 3.14 меню з торговцем в якому будуть розміщатися як і героя так і продавця.

На рис 3.15 – 3.17 наведено зразки UI/UX з головного меню.



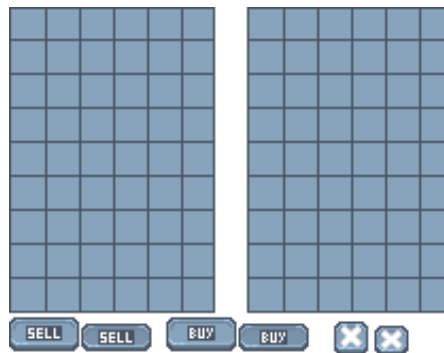
*Рисунок 3.12 – Інвентар*

*Джерело: розроблене автором*



*Рисунок 3.13 – Інвентар в сундуках, кейсах*

*Джерело: розроблене автором*



*Рисунок 3.14 – Інвентар в магазині*

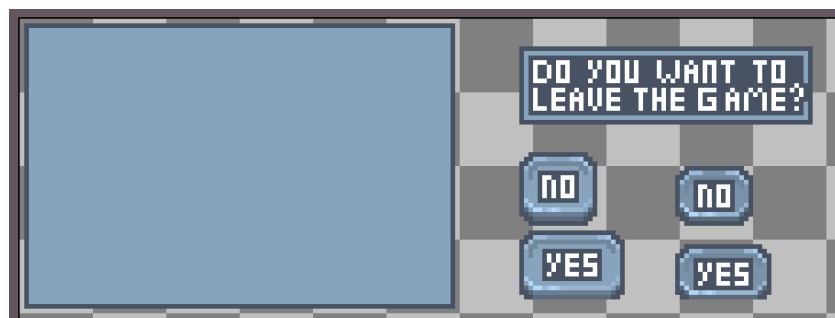
*Джерело: розроблене автором*



*Рисунок 3.15 – Кнопка Options  
Джерело: розроблене автором*



*Рисунок 3.16 – Кнопка Save  
Джерело: розроблене автором*



*Рисунок 3.17 – Кнопка Quit  
Джерело: розроблене автором*

На рис. 3.18 представлено функціонал кнопки Vaccine, яка є ключовим елементом гри. Основна мета гравця – зібрати всі частинки вакцини, щоб у підсумку створити антивірус. Після досягнення цієї цілі на екрані з'явиться повідомлення: "Ви впорались!"

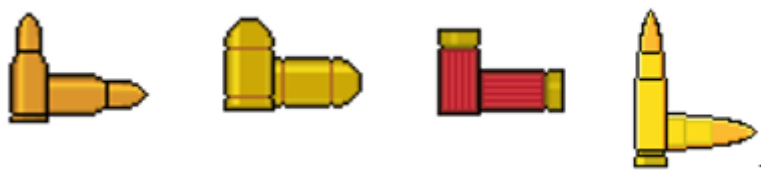


*Рисунок 3.18 – Кнопка Vaccine*

*Джерело: розроблене автором*

Інтерфейс оновлюється автоматично після кожного успішного проходження рівня, коли гравцеві вдається зібрати всі необхідні елементи.

На рис 3.19 наведено приклади набоїв які будуть у вас в інвентарі. Вони діляться на типи для: Пістолета, гвинтівки, автоматів ,рушниць.



*Рисунок 3.19 – Іконки набоїв*

*Джерело: розроблене автором*



Рисунок 3.20 – Кнопка Vaccine  
Джерело: розроблене автором



Рисунок 3.21 – Кнопка Vaccine  
Джерело: розроблене автором

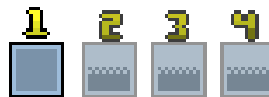
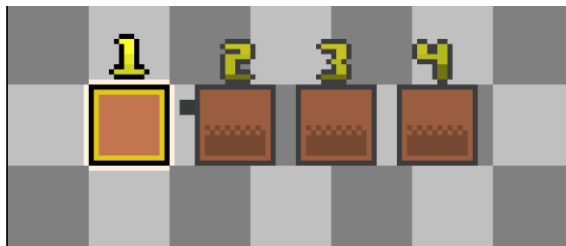


Рисунок 3.22 – Комірки для речей  
Джерело: розроблене автором



*Рисунок 3.23 – Номер комірки*

*Джерело: розроблене автором*



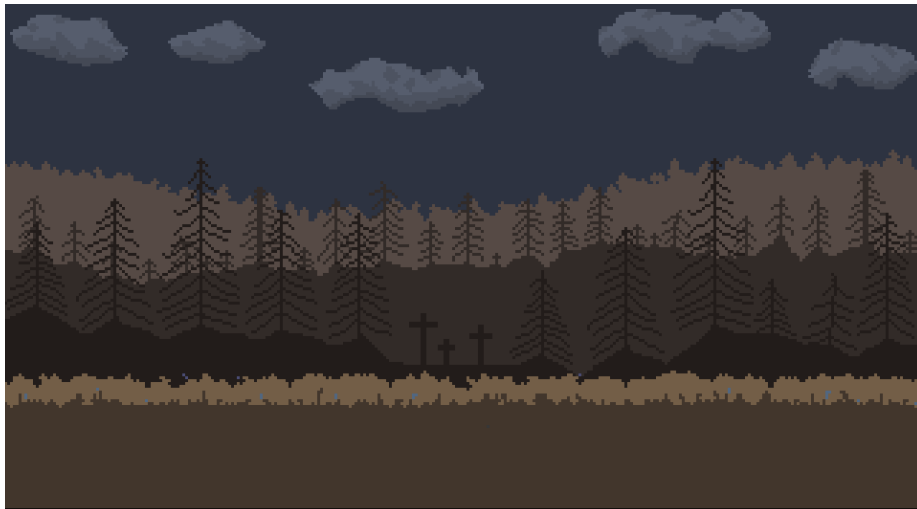
*Рисунок 3.22 – Приціл*

*Джерело: розроблене автором*

### **3.3 Дизайн наповнення рівнів та побічних предметів**

Створення ігрових рівнів вимагало ретельного опрацювання як основних, так і другорядних елементів, що формують атмосферу гри. У цьому розділі представлено наповнення рівнів – декоративні та функціональні об'єкти, які не завжди є обов'язковими для проходження, але суттєво впливають на сприйняття простору, занурення у світ гри та загальну динаміку. Побічні предмети, що з'являються на рівнях, мають різне призначення: від посилення ігрового досвіду до прихованих підказок і бонусів. Під час їхнього дизайну я звертав увагу на візуальну різноманітність, стильову відповідність та логіку розташування в межах кожного рівня.

На рисунку 3.23 представлено фон локації під назвою Мертвий ліс. Одразу впадають в око приглушені сірі тони та могили, що чітко виділяються на тлі похмурого лісового пейзажу. Пожовкла трава, що втратила свою природну насиченість, підкреслює атмосферу занепаду.

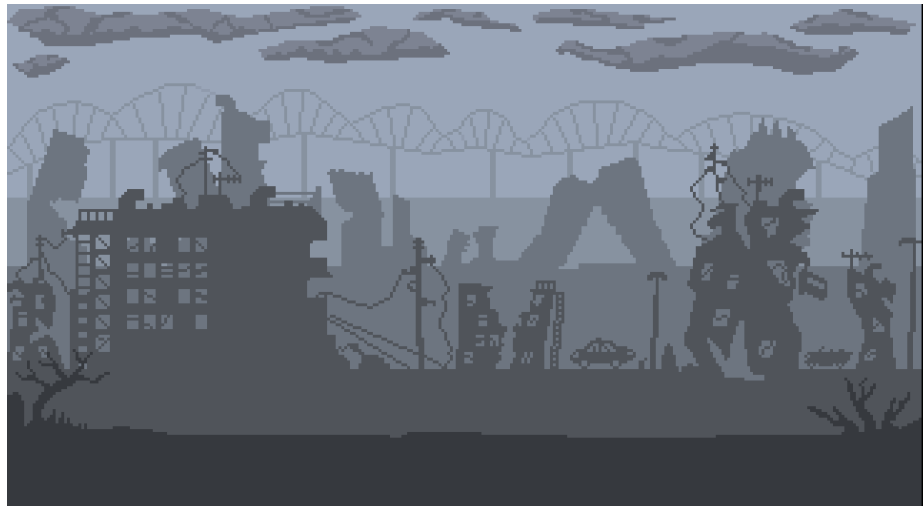


*Рисунок 3.23 – Фон для локації ліс*

*Джерело: розроблене автором*

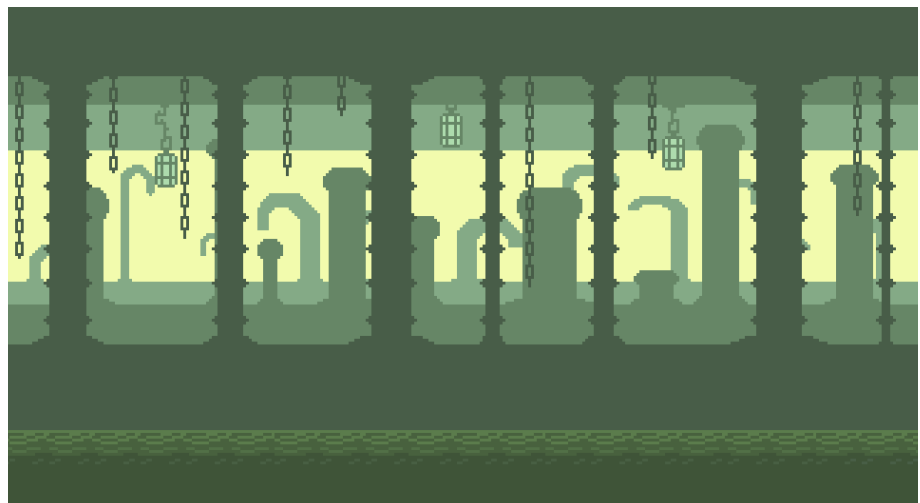
На рисунку 3.24 представлено фон локації Втрачене місто. Першим, що привертає увагу, є залишки зруйнованих будівель, які поступово занепали під дією стихій та часу. Колись живе місто нині повністю зупинило своє існування – через вірус усі споруди були покинуті, втратили догляд і перетворилися на мовчазні свідчення трагедії. Похмурий настрій і почуття тривоги супроводжують кожен крок у цьому занедбаному світі, що став наслідком фатального наукового експерименту.

Зображення на рисунку 3.25 ілюструє задній фон локації Каналізація. Це середовище одразу викликає відчуття небезпеки – темна, болотиста палітра кольорів створює атмосферу отруєного, ворожого простору, де, ймовірно, мешкають монстри. Окремо варто звернути увагу на потік токсичної рідини, що протікає в кадрі. Його вигляд чітко вказує на небезпеку – торкатися цієї речовини без потреби не варто, адже вона може завдати шкоди гравцеві. Усе це підсилює напругу та додає локації гнітючої, загрозливої атмосфери.



*Рисунок 3.24 – Фон для локації місто*

*Джерело: розроблене автором*



*Рисунок 3.25 – Фон для локації Каналізація*

*Джерело: розроблене автором*

На рис.3.26-3.28 зображені побічні предмети для наповнення рівнів. На перших слайдах цього розділу зображувалися локації, а в рисунках нижче зображено наповнення для рівнів щоб збільшити різноманітність локацій та для кожної додати своєї індивідуальності та шарму.



На рис. 3.28 зображено будинок через який здійснюватиметься перехід з локації лісу до загубленого міста.



*Рисунок 3.28 – Дім перехід*

*Джерело: розроблене автором*

### **3.4 Створення Анімацій до спрайтів**

Анімація спрайтів є невіддільною частиною будь-якої динамічної гри, адже саме завдяки їй персонажі та об'єкти виглядають живими, а ігровий процес – насиченим і плавним. У рамках цього проєкту мною було розроблено кілька видів анімацій для основних героїв, противників і предметів. Анімаційні переходи охоплюють різні стани: рух, атаки, взаємодію з об'єктами, а також реакцію на події (наприклад, отримання шкоди або перемогу). Для досягнення бажаного ефекту я використав покадрову анімацію, при цьому особливу увагу приділяв плавності рухів, логіці змін і відповідності стилю гри. Для живих істот було розроблено по декілька анімацій такі як

(idle,attack,death). На рис (6.1,6.2,6.3) наведено приклад як виглядає анімація та їх вигляд у папці.

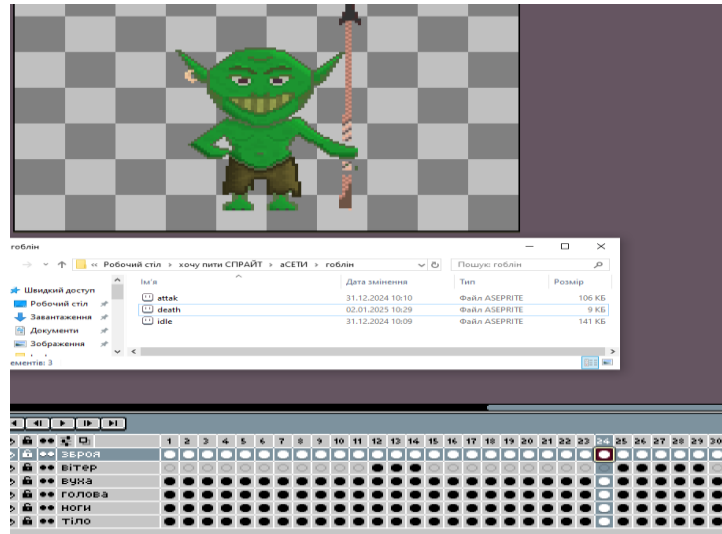


Рисунок 3.29 – Анімація гобліна

Джерело: розроблене автором

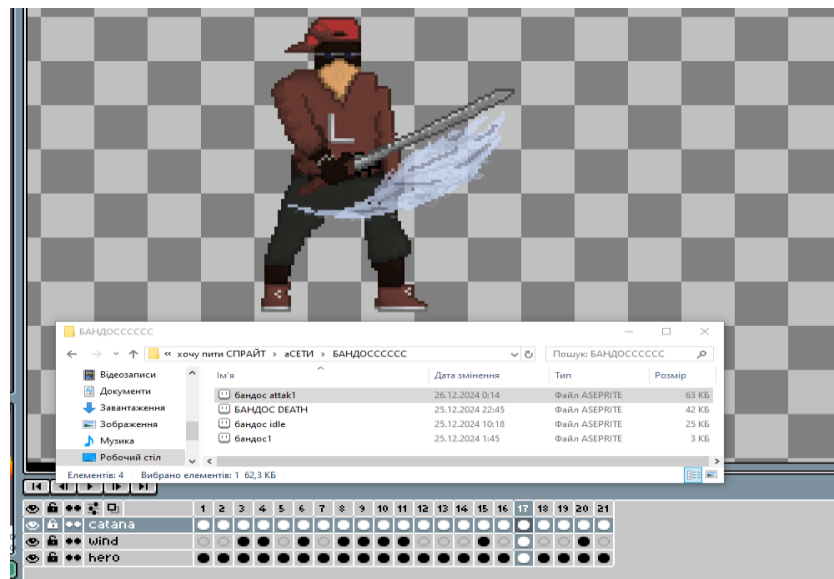
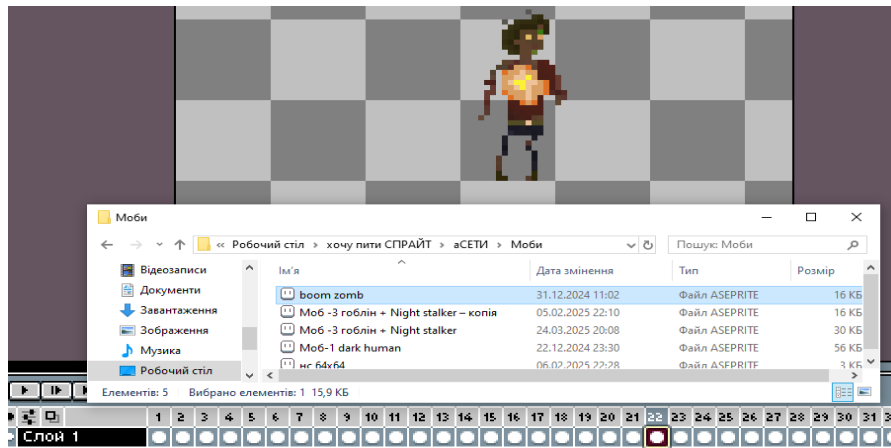


Рисунок 3.30 – Анімація бандита

Джерело: розроблене автором



*Рисунок 3.31 – Анімація зомбі-бомбера*

*Джерело: розроблене автором*

### 3.5 Дизайн предметів для взаємодії

У цьому розділі розглядаються об'єкти, з якими гравець може взаємодіяти під час проходження гри. Вони відіграють важливу роль у геймплеї, адже не лише урізноманітнюють ігровий процес, а й сприяють досягненню основної мети. Під час розробки таких елементів я прагнув зробити їх візуально привабливими, інтуїтивно зрозумілими та логічно вписаними в загальну стилістику гри.

На рис 3.32 ми бачимо зброю з якою головний герой зможе взаємодіяти та брати їх на завдання.

На рис 3.33 зображено як виглядатимуть трофеї з вбитих мобів та лікувальні препарати що допомагатимуть нам вижити.

На рис 3.34 ми бачимо дві текстури для грошей, це одна буде ледани на карті у грі, а інша це вигляд у нашому інвентарі.

На рис 3.35 зображено дві фази сундука: закрыта та відкрита.



*Рисунок 3.32 – Знарядження для персонажа*

*Джерело: розроблене автором*



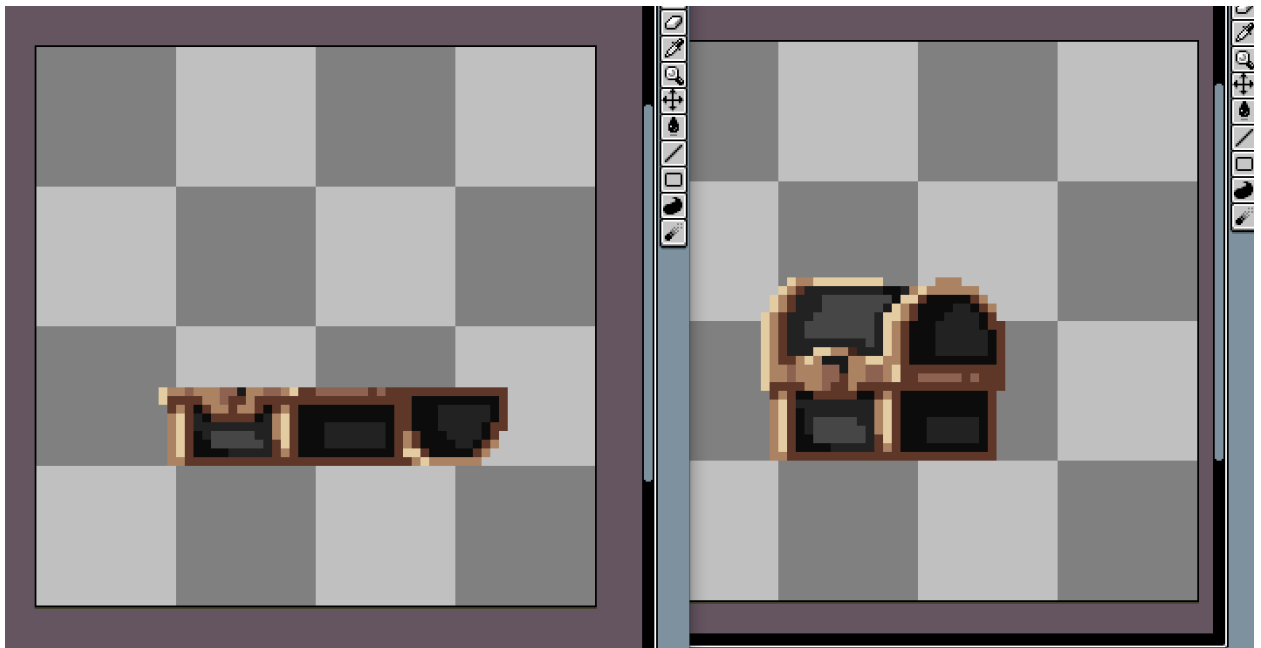
*Рисунок 3.33 – Ліки та трофеї з противників*

*Джерело: розроблене автором*



*Рисунок 3.34 – Дизайн грошей*

*Джерело: розроблене автором*



*Рисунок 3.35 – Дизайн сундука*

*Джерело: розроблене автором*

### **Висновок до розділу 3**

У процесі розробки дизайну гри було створено повноцінну візуальну систему, що охоплює головних персонажів, противників, інтерфейсні елементи та предмети для взаємодії. Особливу увагу приділено стилістичній цілісності, читабельності об'єктів і зручності для користувача. Дизайн предметів і наповнення рівнів сприяє глибшому зануренню у світ гри, додає гравцеві мотивації досліджувати простір і досягати поставлених цілей.

Результатом стало формування цілісного ігрового середовища, де кожен елемент виконує свою роль, як функціонально, так і естетично. Цей підхід дозволив забезпечити баланс між ігровою механікою та візуальним сприйняттям, що є ключовим фактором у створенні привабливого ігрового досвіду.

## ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було розроблено 2D-гра "Demonic Hunter" .

У першому розділі було проаналізовано предметну область, еволюцію геймдизайну з початку 2000-х до 2025 року, сформульовано мету, цілі та завдання проєкту, що дозволило визначити основні вимоги до гри.

Другий розділ був присвячений етапу проєктування, де за допомогою UML-діаграм описано структуру, логіку та архітектуру гри, що дало змогу чітко уявити організацію роботи системи.

У третьому розділі зосереджено увагу на візуальній частині проєкту: розроблено персонажів, противників, інтерфейс, ігрові рівні, анімації та інтерактивні об'єкти. Це забезпечило створення ігрового середовища, яке відповідає сучасним вимогам до 2D-ігор.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Створення візуальних ефектів у Unity: посібник для розробників  
URL: <https://foxminded.ua/ru/vizualnie-effekti-v-unity/>
2. ОСНОВИ ІГРОВОГО ДИЗАЙНУ : ПРАКТИЧНИЙ ПОСІБНИК  
URL: <https://www.based.ua/uk/osnovy-igrovogo-dyzajnu-praktychnyj-posibnyk/>
3. РОЗРОБКА 2D - ГРИ НА ПЛАТФОРМІ UNITY URL:  
<https://krs.chmnu.edu.ua/jspui/bitstream/123456789/2394/1/402%20%D0%A7%D0%B5%D1%80%D0%BD%D0%BE%D0%B2%20%D0%86%D0%BB%D0%BB%D1%8F%20%D0%86%D0%B3%D0%BE%D1%80%D0%BE%D0%B2%D0%B8%D1%87.pdf>
4. ДИЗАЙН ПЕРСОНАЖІВ 2D URL: <https://avada-media.ua/services/dyzaun-personazhey-2d/>
5. Як створити цікавого ворога в грі. Розповідають геймдизайнери  
URL: <https://gamedev.dou.ua/articles/creating-enemies-in-the-game/>
6. Інтеграція ШІ для динамічного рівневого дизайну в 2D-іграх URL:  
<https://peerdh.com/uk/blogs/programming-insights/integrating-ai-for-dynamic-level-design-in-2d-games>
7. Мічківський С. Microsoft Office (Word, Excel, Outlook ...) : навч. посіб. / С. Мічківський, Д. Балдик, В. Головань; Східноукр. нац. ун-т ім. В. Даля, Аграр. ф-т. – Київ : [Вид-во Східноукр. нац. ун-т ім. В. Даля], 2023. – 128 с. – URL: <https://dspace.snu.edu.ua/handle/123456789/1723>
8. ІСТОРІЯ ГЕЙМДЕВУ ХХ СТОЛІТТЯ: ВІД SPACEWAR! ДО SONIC THE HEDGEHOG URL: [https://skvot.io/uk/blog/istoriya-geymdevu-hh-stolittya-vid-spacewar-do-sonic-the-hedgehog?utm\\_source](https://skvot.io/uk/blog/istoriya-geymdevu-hh-stolittya-vid-spacewar-do-sonic-the-hedgehog?utm_source)
9. Історія розвитку ігор: від расман до wow URL:  
[https://training.qatestlab.com/blog/technical-articles/game-history-from-rasman-to-wow/?utm\\_source](https://training.qatestlab.com/blog/technical-articles/game-history-from-rasman-to-wow/?utm_source)
10. Ігри що змінили індустрію: топ-5 проєктів, які перевернули світ геймінгу URL: <https://gameverse.com.ua/blogs/artovmi/ihry-shcho-zminyly-industriiu-top-5-proiektiv-yaki-perevernuly-svit-heiminhu-artovmi/>

11. Хронологія еволюції графіки ігор з важливими етапами апаратного забезпечення URL: <https://peerdh.com/uk/blogs/programming-insights/timeline-of-game-graphics-evolution-with-hardware-milestones-1>
12. Еволюція відеоігор: огляд найпопулярніших ігор усіх часів URL: <https://mxm.com.ua/interesting/evolyutsiya-videoigor-oglyad-najpopulyarnishyh-igor-usih-chasiv/>
13. Grand Theft Auto: San Andreas (2004) URL: [https://uk.wikipedia.org/wiki/Grand\\_Theft\\_Auto:\\_San\\_Andreas](https://uk.wikipedia.org/wiki/Grand_Theft_Auto:_San_Andreas)
14. Half-Life 2 (2004) URL: <https://www.half-life.com/uk/home/>
15. The Elder Scrolls IV: Oblivion (2006) URL: [https://elderscrolls.fandom.com/uk/wiki/The\\_Elder\\_Scrolls\\_IV:\\_Oblivion](https://elderscrolls.fandom.com/uk/wiki/The_Elder_Scrolls_IV:_Oblivion)
16. Redemption 2 (2018) URL: [https://reddead.fandom.com/wiki/Red\\_Dead\\_Redemption\\_2](https://reddead.fandom.com/wiki/Red_Dead_Redemption_2)
17. No Man's Sky (2016) URL: [https://nomanssky-archive.fandom.com/wiki/No\\_Man%27s\\_Sky](https://nomanssky-archive.fandom.com/wiki/No_Man%27s_Sky)
18. The Last of Us Part II URL: [https://thelastofus.fandom.com/wiki/The\\_Last\\_of\\_Us\\_Part\\_II](https://thelastofus.fandom.com/wiki/The_Last_of_Us_Part_II)
19. Cyberpunk 2077 (2020) URL: [https://store.steampowered.com/app/1091500/Cyberpunk\\_2077/](https://store.steampowered.com/app/1091500/Cyberpunk_2077/)
20. Horizon Forbidden West (2022) URL: [https://horizon.fandom.com/uk/wiki/Horizon\\_Forbidden\\_West](https://horizon.fandom.com/uk/wiki/Horizon_Forbidden_West)
21. The Matrix Awakens: An Unreal Engine 5 Experience URL: <https://www.playstation.com/ru-ua/games/the-matrix-awakens--an-unreal-engine-5-experience/>
22. Технології, які зробили революцію в іграх + 5 тенденцій майбутнього URL: <https://www.showmetech.com.br/uk/%D1%82%D0%B5%D1%85%D0%BD%D0%BE%D0%BB%D0%BE%D0%B3%D1%96%D1%97%2C-%D1%8F%D0%BA%D1%96-%D0%B7%D1%80%D0%BE%D0%B1%D0%B8%D0%BB%D0%B8->

<https://dspace.krok.edu.ua/items/6e1488e1-9e21-4282-af10-2e3bca48d2bc>

23. Vysochyn I., Michkivskyu S. Using machine learning for translation and speech generation in e-book reading applications. Держава, регіони, підприємництво: інформаційні, суспільно-правові, соціально-економічні аспекти розвитку: матеріали VI Міжнародної наукової конференції (5-6 грудня 2024 р., м. Київ). Київ: Університет "КРОК", 2024. С.62-64 – URL: <https://dspace.krok.edu.ua/items/6e1488e1-9e21-4282-af10-2e3bca48d2bc>