

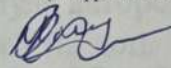
Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних наук

Бакалаврська кваліфікаційна робота на тему:

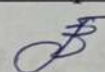
РОЗРОБКА ПРОГРАМНОГО МОДУЛЯ АВТОРСЬКОЇ 2D TOPDOWN ГРИ

Виконала: студентка 4 курсу, групи 4КН-216
спеціальності 122 – Комп'ютерні науки

(шифр і назва напрямку підготовки, спеціальності)

 Єрмолаєва М. А.
(прізвище та ініціали)

Керівник: к.т.н., доцент каф. КН

 Крилик Л. В.
(прізвище та ініціали)

«04» червня 2025 р.

Рецензент: к.т.н., доцент каф. АІТ

 Богач І. В.
(прізвище та ініціали)

«05» червня 2025 р.

Допущено до захисту

Зав. кафедри Яровий А. А. 

«06» червня 2025 р.

Вінниця ВНТУ - 2025 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних наук
Рівень вищої освіти перший (бакалаврський)
Галузь знань – 12 Інформаційні технології
Спеціальність – 122 Комп'ютерні науки
Освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ
Завідувач кафедри КН
проф., д.т.н. Яровий А. А.
«05» травня 2025 р.

ЗАВДАННЯ НА БАКАЛАВРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТЦІ

Єрмолаєвій Мілені Арсеніївні

1. Тема роботи: Розробка програмного модуля авторської 2D TopDown гри.
керівник роботи: Крилик Людмила Вікторівна, к.т.н., доц.

затверджені наказом вищого навчального закладу «20». 03.2025 року № 97

2. Термін подання студенткою роботи 30 травня 2025 р.

3. Вихідні дані до роботи: мова програмування – об'єктно-орієнтована; кількість можливих рівнів не менше 2; інтерфейс користувача має бути інтуїтивно зрозумілий; кількість прс не менше 2; кількість режимів гри не менше 2.

4. Зміст текстової частини (перелік питань, які потрібно розробити): вступ; обґрунтування доцільності розробки програмного модуля авторської 2D TopDown гри; проектування програмного модуля авторської 2D TopDown гри; програмна реалізація програмного модуля авторської 2D TopDown гри; висновки; список використаних джерел; додатки.

5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень): приклад ігрового процесу ігор; приклади графіки в різних іграх; roadmap авторської 2D TopDown гри з 2 ігровими режимами; UML-діаграма класів для персонажів Action gamemode; ієрархія класів персонажів для Action режиму; ієрархія класів персонажів та об'єктів для RPG режиму; схема алгоритму події взаємодії головного персонажа з іншими персонажами чи об'єктами; приклад роботи програми.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-3	Крилик Л. В., к.т.н., доц. каф. КН		

7. Дата видачі завдання 05 травня 2026р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1	Вступ. Обґрунтування доцільності розробки програмного модуля авторської 2D TopDown гри	05.05.25	07.05.25	
2	Проектування програмного модуля авторської 2D TopDown гри	08.05.25	11.05.25	
3	Програмна реалізація програмного модуля авторської 2D TopDown гри	12.05.25	15.05.25	
4	Висновки та аналіз отриманих результатів	16.05.25	26.05.25	
5	Оформлення додатків	27.05.25	29.05.25	
6	Розробка інструкції користувача	30.05.25	01.06.25	
7	Оформлення матеріалів до захисту БКР	02.06.25	04.06.25	

Студентка

(підпис)

Ермолаєва М. А.

(прізвище та ініціал)

Керівник роботи

(підпис)

Крилик Л. І.

(прізвище та ініціал)

АНОТАЦІЯ

Бакалаврська кваліфікаційна робота складається з 78 сторінок формату А4, які включають 48 рисунків і 1 таблицю. У списку використаних джерел зазначено 31 найменування.

Метою роботи є покращення ігрового процесу для авторської 2D TopDown гри з використанням ігрового рушія Unreal Engine 5.

У загальній частині роботи на основі аналізу існуючих відеоігор, методів та технологій доведена доцільність розробки програмного модуля авторської 2D TopDown гри. Розроблено алгоритм роботи та UML діаграми класів Action та RPG режимів авторської 2D TopDown гри, які спрощують розробку та розуміння структури програмного забезпечення. Програмний продукт розроблений з використанням сучасного ігрового рушія Unreal Engine 5 та Aseprite.

Розроблено програмний продукт та проведено його тестування. Результати тестування розробки вказують на те, що ігровий процес авторської 2D TopDown гри повноцінний та всі поставлені завдання було виконано.

Ключові слова: 2D TopDown гра, інді-гра, Unreal Engine 5, gamedev.

ABSTRACT

The bachelor's thesis consists of 78 A4 pages, including 48 figures and 1 table. The list of references includes 31 items.

The purpose of the work is to improve the gameplay for the author's 2D TopDown game using the Unreal Engine 5 game engine.

In the general part of the work, based on the analysis of existing video games, methods and technologies, the feasibility of developing a software module for the author's 2D TopDown game is proven. The algorithm of work and UML diagrams of the Action and RPG modes of the author's 2D TopDown game are developed, which simplify the development and understanding of the software structure. The software product is developed in the modern game engine Unreal Engine 5 and Aseprite.

The software product has been developed and tested. The results of the development testing indicate that the gameplay of the author's 2D TopDown game is complete and all the tasks have been completed.

Keywords: 2D TopDown game, indie game, Unreal Engine 5, gamedev.

ЗМІСТ

ВСТУП	4
1 ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ ПРОГРАМНОГО МОДУЛЯ АВТОРСЬКОЇ 2D TOPDOWN ГРИ.....	6
1.1 Огляд ігрової індустрії та жанрів відеоігор.....	6
1.2 Огляд графічної частини відеоігор.....	12
1.3 Середовища для реалізації програмного продукту	18
1.4 Формулювання вимог та постановка задачі	20
1.5 Висновок	21
2 ПРОЕКТУВАННЯ ПРОГРАМНОГО МОДУЛЯ АВТОРСЬКОЇ 2D TOPDOWN ГРИ.....	23
2.1 Розробка механік авторської 2D TopDown гри.....	23
2.2 Розробка UML-діаграми	25
2.3 Розробка схеми алгоритму роботи програмного модуля авторської 2D TopDown гри.....	30
2.4 Висновок	36
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ПРОГРАМНОГО МОДУЛЯ АВТОРСЬКОЇ 2D TOPDOWN ГРИ.....	38
3.1 Обґрунтування вибору ігрового рушія та мови програмування	38
3.2 Обґрунтування вибору програмного забезпечення для створення графічної частини	39
3.3 Розробка RPG gamemode.....	41
3.4 Розробка Action gamemode.....	45
3.5 Розробка компоненту системи здоров'я	50
3.6 Тестування роботи програми та аналіз результатів	52

3.7 Висновок	59
ВИСНОВКИ.....	61
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	62
ДОДАТКИ	65
ДОДАТОК А (обов’язковий) Протокол перевірки кваліфікаційної роботи	66
ДОДАТОК Б (обов’язковий) Графічна частина.....	67
ДОДАТОК В (довідниковий) Інструкція користувача.....	77

ВСТУП

Актуальність досліджень.

Індустрія відеоігор зростає і розширюється з кожним роком. Також популярності набирають ігри розроблені невеликими компаніями, так звані інді-ігри, які є більш незалежними проєктами, ніж ігри від великих корпорацій. Тема «Розробка програмного модуля авторської 2D TopDown гри» є актуальною в сучасному світі. На актуальність теми вказують такі аспекти:

- Збільшення контенту і конкуренції. Популярних 2D TopDown ігор не так багато. Розробка такої гри може стати конкурентною на ринку відеоігор.

- Зручність для інді-розробки: інді-ігри, тобто ігри розроблені однією людиною або невеликою компанією не мають великих ресурсів і обмежені в бюджеті. Але багато 2D інді-ігор сподобались гравцям настільки, що стали відомі на весь світ.

- Ностальгія: 2D TopDown це старий жанр, наразі ринок відеоігор насичується більш 3D контентом, ніж 2D графікою. Жанр TopDown це класика, це відчуття комфорту, особливо в поєднанні з піксельною графікою. Це є гарною базою для поєднання з іншими жанрами, новими технологіями та можливостями сучасного ігрового рушія Unreal Engine 5.

- Ігровий рушій Unreal Engine 5 більш відомий для 3D проєктів, але існує плагін для роботи з 2D графікою. Поєднання відомих можливостей роботи зі світлом від Unreal Engine з 2D проєктом може дати дуже захоплюючий результат.

Отже, розробка програмного модуля авторської 2D TopDown гри на ігровому рушії Unreal Engine 5 є важливою та актуальною в контексті сучасних тенденцій розвитку ігрової індустрії і може бути цікавою для дослідження та провадження в практику.

Метою дослідження є розширення функціональних можливостей програмного модуля авторської 2D TopDown гри.

Об'єктом дослідження є ігровий процес та ігрова механіка.

Предметом дослідження є програмні засоби для реалізації програмного модуля авторської 2D TopDown гри на ігровому рушії Unreal Engine 5.

Задачі дослідження:

1. Провести аналіз існуючих ігор у жанрах 2D та TopDown, методів їх реалізації та обґрунтувати доцільність розробки;
2. Спроекувати програмний модуль авторської 2D TopDown гри;
3. Програмно реалізувати програмний модуль авторської 2D TopDown гри на ігровому рушії Unreal Engine 5;
4. Виконати тестування програми та провести аналіз отриманих результатів.
5. Розробити інструкцію користувача.

Апробація роботи.

Результати досліджень було апробовано на LIV Всеукраїнській науково-технічній конференції підрозділів Вінницького національного технічного університету (НТКП ВНТУ) м. Вінниці у 2025 р. [1].

Публікації: за основними результатами досліджень опубліковано тези доповіді на науково-технічній конференції [1] та статтю в журналі «Вісник Хмельницького національного університету» серія: «Технічні науки», який входить до переліку наукових фахових видань МОНУ (категорія Б) на тему: «Особливості розробки авторської 2D гри з використанням ігрового рушія UNREAL ENGINE 5» [2].

1 ОБГРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ ПРОГРАМНОГО МОДУЛЯ АВТОРСЬКОЇ 2D TOPDOWN ГРИ

1.1 Огляд ігрової індустрії та жанрів відеоігор

Протягом останніх 10 років індустрія відеоігор розвивається з шаленою швидкістю. Відеоігри є однією з найбільш популярних форм розваг у світі, пропонуючи гравцям інтерактивний досвід на різних платформах: персональних комп'ютерах (ПК) і консолях. Також існують різні пристрої для того, щоб урізноманітнити ігровий процес, наприклад джойстики, шоломи віртуальної реальності.

Відеоігри є одним з ключових рушіїв розвитку технологій, мистецтва та культури. Відеоігри є поєднанням графічної частини, аудіо мистецтва, письменства, сценічного мистецтва та програмування [1]. Розвиток цієї індустрії створив багато нових професій: гейм дизайнер, розробник ігор, художник персонажів, навколишнього середовища, пропсів, сценарист ігор та багато інших [2].

Розробка відеоігор починається з 1970-х років, тоді ж з'явилися аркадні ігри і консолі першого покоління. Пізніше стала популярна 3D-графіка і вимоги до якості зображення підвищилися. На розробку витрачалось все більше і більше грошей, почали з'являтися компанії з розробки відеоігор.

Однією з найкращих ігор всіх часів є японська гра в жанрі пригодницький бойовик The Legend of Zelda: Breath of the Wild, розроблена для консолей Nintendo. В цій грі можна досліджувати світ, їздити верхи, битись з монстрами, куховарити та створювати нові предмети, а також присутній основний сюжет гри.

Геймплей і механіка гри не такі як минулі ігри цієї серії. Особливістю цієї частини є відкритий світ та можливість створювати нові речі з ресурсів які добуваються. У грі присутні різноманітні квести та завдання за які можна отримати нагороду, а також можливість відкривати і досліджувати нові території. Також у цій частині є захоплюючі візуальні ефекти високої якості, озвучення

мови персонажів. Розробникам вдалось реалізувати відкритий світ дуже гарно, гравець може досліджувати те, що йому цікаво і немає потреби досліджувати абсолютно все в строгому порядку. Ця гра стала натхненням для багатьох інших ігор: Ghost of Tsushima, Elden Ring, The Witcher 3: Wild Hunt. На рисунку 1.1 наведено ігровий процес The Legend of Zelda: Breath of the Wild [3].



Рисунок 1.1 – Ігровий процес The Legend of Zelda: Breath of the Wild

Ігри Resident Evil популяризували такий жанр як survival horror. Survival horror – виживання серед жаху, обмежені ресурси, напружена атмосфера. Також, розповсюдженою рисою таких ігор є неможливість зберегти гру в будь-який момент. Особливістю цього жанру є поєднання двох жанрів survival (виживання) та horror. Основна механіка гри на виживання зосереджена на управлінні ресурсами, часто включає системи крафта або порятунку, які можна використовувати, щоб зберегти персонажа живим. Жахи – ще ширша категорія, яка, ймовірно, охоплює десятки ігор про виживання. Майже все, що містить зомбі, постапокаліптичну сюжетну лінію чи масу стрибків, вважається жахом. Ці назви також часто мають психологічний характер і використовують напругу, щоб

занурити гравця в гру. Ігри Survival horror в якомусь сенсі не справедливі до ворогів, тому що потребують грамотного розподілу ресурсів для виживання [4].

Наприклад, в ранніх іграх Resident Evil для зберігання потрібно було знаходити друкарська стрічка і зберегтись можна було тільки в місцях з друкарською машинкою. Також ця франшиза відома фіксованими камерами та танковим управлінням, яке в критичних ситуаціях додає напруги в грі. За всю свою історію ігри Resident Evil мали різноманітні схеми керування та ігрові механізми. З часом управління в іграх стало більш зручним і звичним. Розв'язування головоломок займає важливе місце в серії. На рисунку 1.2 наведено ігровий процес Resident Evil 1 [5].



Рисунок 1.2 – Ігровий процес Resident Evil 1

Пограти з друзями також можливо, існують як кооперативні так і мультиплеєрні ігри. Гарним прикладом кооперативної гри є It takes two (рис. 1.3).

Вона полюбилась гравцям через різноманітність цікавих механік, сюжет та графіку. Ця гра особлива тим, що вона поєднує багато цікавих ідей та механік, які гарно вписуються у сюжет та поєднані у кооперативній грі на двох. Основною

механікою є головоломки, які потрібно вирішувати за допомогою комунікації та роботи в команді. Інтерфейс поділений на два екрани для двох гравців. За сюжетом двоє друзів грають як сімейна пара, яка на грані розлучення, а їх дитина випадково перенесла батьків у магічний світ, де вирішуючи різні головоломки і працюючи разом можна проходити безліч рівнів. В грі присутній не тільки вид від третьої особи, а також рівні з видом збоку або зверху (TopDown) [6].



Рисунок 1.3 – Ігровий процес It takes two

Іншим прикладом гарної кооперативної гри є франшиза We were here, де можна грати за двох друзів експедиторів, які вирішуючи головоломки досліджують історії місць, де вони знаходяться. Гра насичена унікальними рівнями та гарною графікою. Особливістю цієї гри є те, що для комунікації гравці використовують рацію. Також обидва гравці майже ніколи не знаходяться в одній і тій самій кімнаті та повинні пояснювати своє оточення, щоб розв'язати чергову головоломку. Гейм дизайн цієї гри виконаний на високому рівні. Ця гра справді вчить гравців працювати в команді та вміти слухати один одного та навчитись

давати можливість вимовити свої думки один одному, не перебиваючи. На рисунку 1.4 наведено ігровий процес We were here forever [7].



Рисунок 1.4 – Ігровий процес We were here forever

Однією з найбільш відомих інді ігор є Minecraft. Який був розроблений однією людиною. Проєкт став настільки успішний, що зараз ним володіє компанія Microsoft. Ця гра жанру виживання, пісочниця з відкритим світом, де можна робити що завгодно. Minecraft не має встановленої мети, і в нього можна грати як завгодно! Ось чому її іноді називають «грою з пісочницею». Гра без мети може здатись нецікавою та нудною, але суть цієї гри полягає в тому, що кожен гравець самостійно ставить собі мету та досягає її. Світ Minecraft дозволяє епічні пригоди, спокійні медитації та все, що між ними [8]. Minecraft також є прикладом затишної гри. Не дивлячись на можливість досліджувати світ та боротись з монстрами та драконом, у грі також можна спокійно будувати, вирощувати їжу та доглядати за тваринами під затишну ембієнт музику. Також існує багато контенту розробленого фанатами, так звані моди, які допомагають урізноманітнити гру або змінити її до невпізнанності [9].

Приклад ігрового процесу Minecraft наведено на рисунку 1.5.



Рисунок 1.5 – Ігровий процес Minecraft

Прикладом 2D Topdown гри є Stardew Valley. Комфортна піксельна графіка, фермерське життя, можливість завести сім'ю та прийняти участь в міському житті та різних фестивалях.

Ця гра була розроблена однією людиною протягом 4 років. Головним натхненням для проєкту стала гра Harvest Moon. Згодом автор мріяв створити найкращу версію Harvest Moon – гри, яка була б однаково заспокійливою та ідилічною, але більш масштабною та амбітнішою. Stardew Valley є однією з найпопулярніших затишних ігор, не зважаючи на факт, що у грі присутні окремі локації з монстрами з якими потрібно битись, це протиставляється всьому концепту жанрів затишних ігор [10].

На рисунку 1.6 наведено ігровий процес Stardew Valley



Рисунок 1.6 – Ігровий процес Stardew Valley

1.2 Огляд графічної частини відеоігор

Графіка у відеоіграх є важливим аспектом, хоч і не найголовнішим. Вона не обов'язково має бути ідеальною з найпродвинутішими останніми технологіями, щоб гравці її любили. За допомогою різних стилів можна підкреслювати атмосферу гри і навіть сюжет.

Піксельне мистецтво – це мистецтво та естетичний стиль, який ми все ще можемо знайти у візуальному дизайні гри. До середини 90-х цей вид мистецтва був єдиним, який можна було застосувати до цифрових ігрових консолей і ПК. Його форму можна порівняти з вишивкою хрестиком або цифровою мозаїкою. Це обумовлено з проблемами, пов'язаними з обмеженнями обладнання в епоху 8- і 16-бітних обчислень, коли роздільна здатність мала і кількість кольорів, які могли відображатися була дуже обмеженою.

Піксельна графіка це відчуття ностальгії та затишку. Як унікальний стиль мистецтва, в піксельному мистецтві деталізація часто обмежується за допомогою роздільної здатності. Зокрема, розмитий візерунок додає образу сміливий та

виразний стиль, і надає можливість дизайнерам персонажів і медіа-дизайнерам вільно експериментувати.

Завдяки фіксованій палітрі та обмеженню доступних кольорів, піксель-арт дизайнери персонажів розробили кілька технік для роботи з піксель-артом.

Хоча беззаперечно, що 3D-зображення стали популярним мейнстримним дизайном, піксель-арт стиль все ще має сильний характерний вплив. Піксельний дизайн, як дослідження між ретро і трендом також приваблює серця багатьох нових споживачів своєю унікальністю. У дизайні піксельний стиль незалежний і пронизує всі сторони життя людей. Одиниці блоків об'єднані, щоб абстрактно показати різні форми, але вони також мають неповторну красу [11].

На рисунку 1.7 наведено приклад використання простої піксельної графіки у грі Undertale.



Рисунок 1.7 – Приклад використання простої піксельної графіки у грі Undertale

Не дивлячись на просту графіку гри Undertale гра стала легендарною через гарний музичний супровід та сюжетну історію.

На рисунку 1.8 наведено приклад використання більш деталізованої піксельної графіки у грі The Siege and the Sandfox.



Рисунок 1.8 – Приклад використання більш деталізованої піксельної графіки у грі The Siege and the Sandfox

Також існують різні види перспектив в іграх. Для 2D графіки характерно вид збоку або зверху, так звані TopDown ігри. На рисунку 1.9 наведено приклад ігри з TopDown графікою [12].

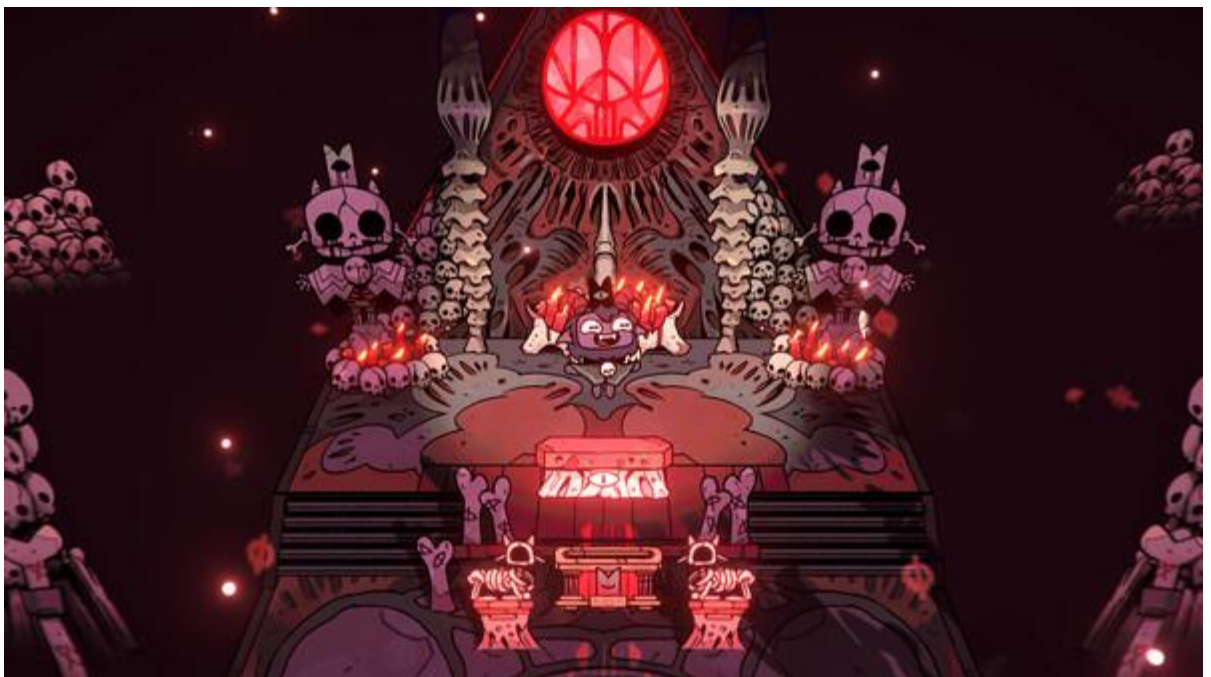


Рисунок 1.9 – Приклад ігри з TopDown графікою

На рисунку 1.10 наведено приклад гри з боковим видом і піксельною графікою.



Рисунок 1.10 – Приклад гри з боковим видом і піксельною графікою

На рисунку 1.11 наведено приклад гри з ізометричною піксельною графікою.



Рисунок 1.11 – Приклад гри з ізометричною піксельною графікою

Наразі великі компанії прагнуть зробити ігри все більш схожими на реальний світ чи кіно. За останні 10 років графіка розвилась до неймовірно реалістичного рівня.

Unreal Engine 5 відомий своєю роботою з освітленням, що наближає життя до локації у іграх [13]. Також для деяких ігор використовують поєднання роботи акторів та комп'ютерної графіки. Наприклад, для гри Detroit: Become Human було відзнято обличчя та емоції акторів, які використовувались для створення персонажів. Це було дуже гарним рішенням, враховуючи сюжет гри, де роботи починають відчувати людські емоції, а не просто є сталевими бездушними прислугами для людей.

На рисунку 1.12 наведено 3D графіку з гри Detroit: Become Human.



Рисунок 1.12 – 3D графіка з гри Detroit: Become Human

Гра важлива не тільки графікою, існують й інші важливі аспекти, які впливають на успішність гри. Як можна помітити з прикладів вище, графіка не обов'язково має бути ідеальною, щоб гра подобалась гравцям. Є такі аспекти як цікавий сюжет, механіка гри, естетика.

Механіка: це процедури та правила відеогри. Механіка описує мету гри, як гравці можуть і не можуть намагатися її досягти, і що відбувається, коли вони намагаються. Якщо порівняти ігри з більш лінійними розважальними досвідами (книги, фільми тощо), можна помітити, що хоча лінійні досвіди включають технології, історію та естетику, вони не включають механіку, оскільки саме механіка робить гру грою. При виборі набору механізмів як вирішальні для ігрового процесу, потрібно вибрати технологію, яка їх підтримуватиме, естетику, яка чітко підкреслить їх для гравців та історію, яка дозволить ігровій механіці мати сенс для гравців.

Сюжет: це послідовність подій, які розгортаються у грі. Він може бути лінійним і заздалегідь визначеним, або він може бути розгалуженим та емерджентним. Коли є історія, яку потрібно розповісти у грі, то потрібно вибрати механіку, яка одночасно посилить цю історію та дозволить їй з'явитися. Як будь-який оповідач, краще обрати естетику, яка допоможе посилити ідеї історії і технологію, яка найкраще відповідає конкретній історії, яка вийде з гри.

Естетика: це те, як гра виглядає, звучить, пахне, смакує та відчувається. Естетика є надзвичайно важливим аспектом ігрового дизайну, оскільки вона безпосередньо пов'язана з досвідом гравця. Якщо є певний вигляд або тон, який хочеться, щоб гравці відчули та занурилися в нього, потрібно буде вибрати технологію, яка не лише дозволить реалізувати естетику, але посилить і підсилить її. Тоді краще обрати механіку, яка змушує гравців відчувати себе у світі, визначеному естетикою та історію з набором подій, які дозволять обраній естетиці розвиватися в правильному темпі та справляти найбільший вплив.

Технологія: мається на увазі не лише «високі технології», але й будь-які матеріали та взаємодії, які роблять гру можливою, як-от папір та олівець, пластикові фішки чи потужні лазери. Технологія, яку можна обрати для гри, дозволяє їй виконувати певні дії та забороняє інші дії. Технологія, по суті, є середовищем, у якому відбувається естетика, у якому відбуватиметься механіка та за допомогою якого розповідатиметься історія [14].

1.3 Середовища для реалізації програмного продукту

Ігрові рушії – це інструменти, які прискорюють складний процес розробки ігри. Наразі існує багато різних рушіїв для різних цілей і типів ігор. Деякі з них дають можливість створювати ігри навіть без навичок програмування. Також є більш складні для роботи з якими потрібно витратити багато часу на навчання та опановування. Такі ігрові рушії як Unreal Engine та Unity розроблені на базі C++ та C# відповідно і потребують деяких знань з сфери комп'ютерних наук для роботи з ними.

Unity це ігровий рушій для створення крос-платформних ігор, розроблений з підтримкою об'єктно-орієнтованої мови програмування C#. Цей рушій підтримує створення 2D та 3D ігор. Існує багато додаткових плагінів, що розширюють можливості Unity. Також є велика спільнота розробників та багато платних і безкоштовних уроків.

Unreal Engine 5 – це один з найбільш популярних ігрових рушіїв, написаний мовою C++ . Система візуальних сценаріїв Blueprint у Unreal Engine – це повна система сценаріїв ігрового процесу, заснована на концепції використання інтерфейсу на основі вузлів для створення елементів ігрового процесу в Unreal Editor. Як і в багатьох поширених мовах сценаріїв, він використовується для визначення об'єктно-орієнтованих (ОО) класів або об'єктів у механізмі.

Для реалізації 2D ігор існує безкоштовний плагін PaperZD, який значно полегшує роботу з двовимірною графікою.

Ця система є надзвичайно гнучкою та потужною, оскільки надає дизайнерам можливість використовувати практично повний спектр концепцій та інструментів, які зазвичай доступні лише програмістам. Крім того, спеціальна розмітка Blueprint, доступна в C++ реалізації Unreal Engine, дозволяє програмістам створювати базові системи, які можуть бути розширені дизайнерами [15].

Залежно від стилю для візуальної частини використовують різні програми. Для створення 3D графіки для ігор часто використовуються програмні забезпечення як Autodesk Maya, ZBrush, 3D Max. В залежності від мети, іноді використовують поєднання декількох програм для реалізації гарного продукту. Всі ці програми є платними, але існує безкоштовна програма Blender за допомогою якої можна створити не менш якісний контент.

Для 2D графіки використовують Adobe Photoshop, Adobe Illustrator, Clip Studio, Krita.

Для піксельної графіки найбільш популярною програмою є Aseprite. Це програмне забезпечення поєднує в собі всі необхідні функції для створення піксель-арту та анімації. Також є можливість встановити додаткові плагіни, які називаються скрипти, що можуть значно полегшити роботу. Aseprite легко опанувати новачку, але також є багато безкоштовних уроків на Youtube. Ця програма гарно підходить для тих, хто вперше вчиться анімації персонажів та піксель-арту [16].

На рисунку 1.13 наведено інтерфейс програми Aseprite.

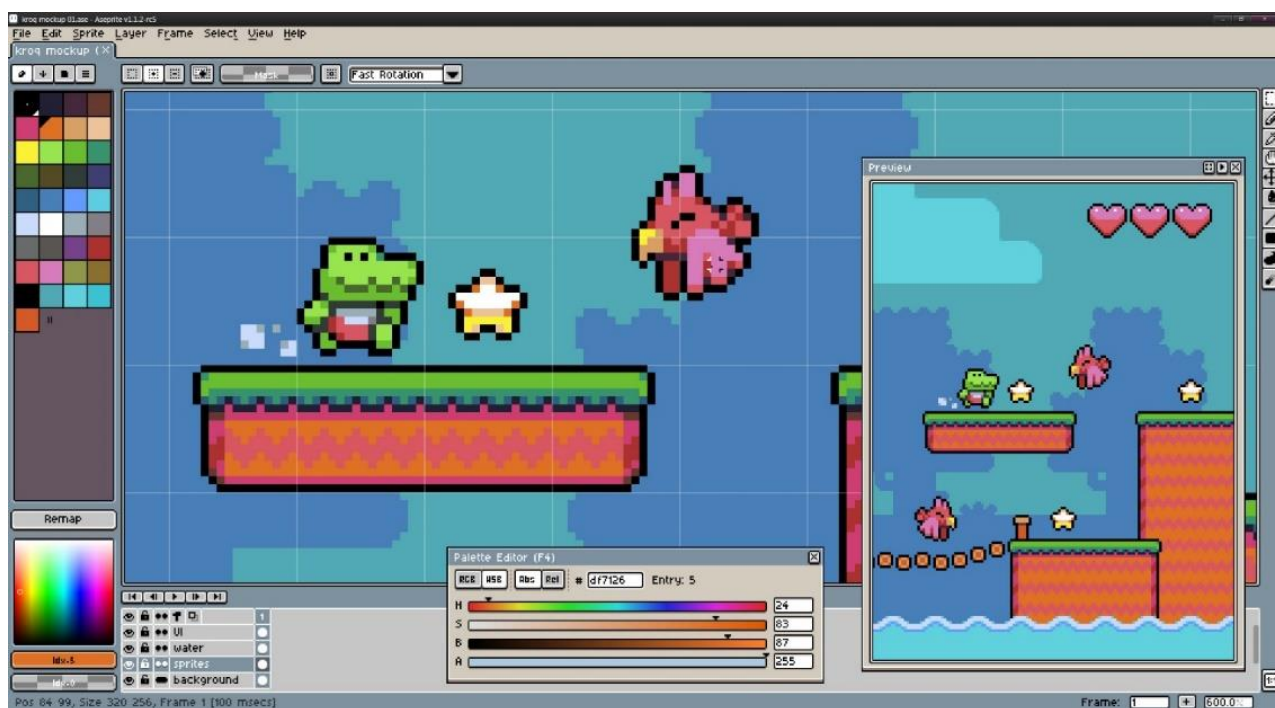


Рисунок 1.13 – Інтерфейс програми Aseprite

Але для інді-розробки створення графічної складової може бути важкою, особливо, якщо немає навичок малювання. Для таких випадків існує платформа Itch.io. Це велика спільнота інді-розробників зі всього світу. Також багато художників діляться там своїми артами і дають дозвіл на використання їх творів у своїх проєктах безкоштовно або за невелику плату [17].

На цій платформі можна знайти не тільки готові асети для гри, але й звуковий супровід, додаткові інструменти, уроки. Також можна викласти свою гру та знайти свою аудиторію. На рисунку 1.14 наведено інтерфейс Itch.io.

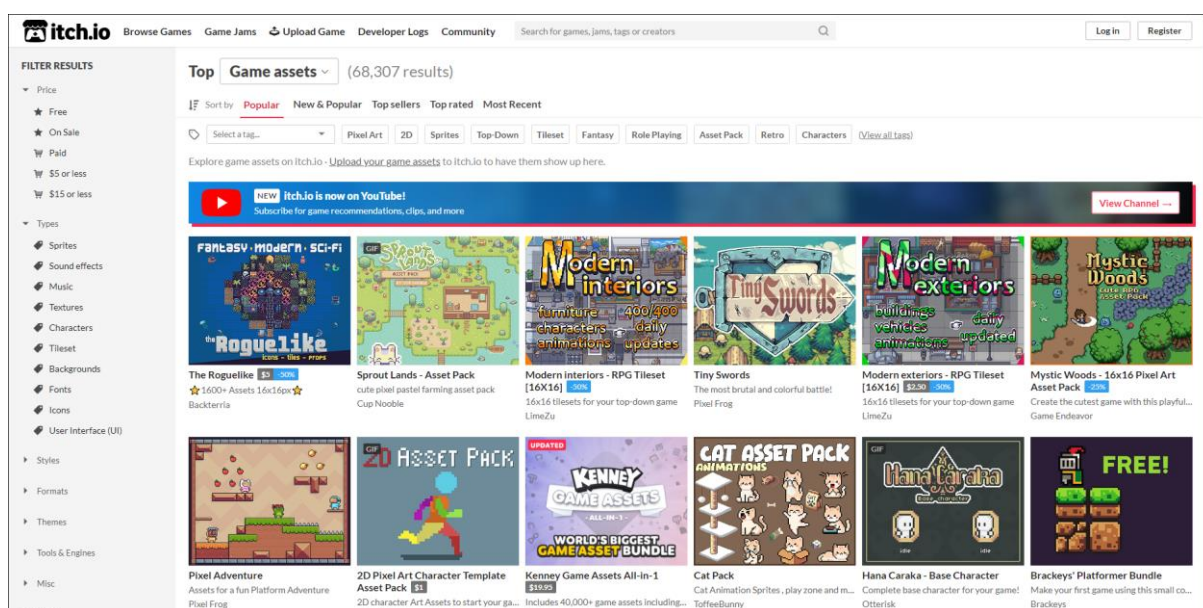


Рисунок 1.14 – Інтерфейс програми Itch.io

1.4 Формулювання вимог та постановка задачі

Наразі існує вже багато різноманітних ігор на будь-який смак, але все ж таки важко знайти щось, щоб зачепило з перших секунд і було б цікаво грати.

Основною ідеєю є створення гри, атмосфера якої допоможе розслабитись і не буде напружувати. Для цього ігровий процес має бути спокійним, а графічна частина бути привабливою.

Також має бути можливість розмовляти з неігровими персонажами (NPC), що допоможе дізнаватись сюжет гри.

Багато схожих ігор було зроблено, але в основному для реалізації схожих ігор було використано ігровий рушій Unity. Крім того, найпопулярніша гра у жанрі 2D TopDown з піксельною графікою була реалізована на мові програмування Python.

Основною задачею є реалізація 2D TopDown гри на ігровому рушії Unreal Engine 5. Цей ігровий рушій має свої переваги та недоліки, але для полегшення роботи з 2D іграми буде використано внутрішні функції Paper2D в поєднанні з плагіном PaperZD.

Для реалізації програмного модуля найкраще застосувати 2D графіку в поєднанні з піксель-артом та видом зверху (TopDown). Це ідеальний варіант, тому що для його реалізації не потрібно мати багато ресурсів та часу на обробку 3D зображень. Для реалізації буде використано і створено графічні елементи у програмі Aseprite з поєднанням асетів з безкоштовною ліцензією з сайту Itch.io. Обрано для розробки ігровий рушій Unreal Engine 5, через його систему візуальних сценаріїв Blueprint, яка прийнятна для новачків.

Для розробки коректного програмного модуля потрібно вирішити такі задачі:

- Налаштувати середовище для роботи з 2D графікою;
- Розробити та підготувати графічну частину гри. Графіка має бути комфортна і релаксна;
- Розробити рівень гри;
- Розробити ігрову механіку для гри;
- Розробити інтерфейс для гри;
- Здійснити тестування програмного модуля.

1.5 Висновок

У цьому розділі було проаналізовано існуючі жанри відеоігор та ігрову індустрію. Також було проаналізовано графіку, яку використовують в іграх та її

вплив на загальну атмосферу в іграх. Проведено аналіз популярних програм та сервісів для розробки відеоігор, які прийнятні для інді-розробки.

Для розробки програмного модуля гри обрано стиль графіки 2D піксель-арт з ігровою перспективою TopDown та ігровий рушій Unreal Engine 5.

Розробка програмного модуля 2D гри з піксель-арт графікою на ігровому рушії Unreal Engine 5 є актуальною, тому що це розробка класичного поєднання жанрів за допомогою нових інструментів. Такі ігри рідко створюються за допомогою Unreal Engine 5, тому поєднання його функції для розробки 2D гри є основною задачею для розробки.

Розробка є актуальною, оскільки багатьом гравцям подобається піксельна графіка та прості ігри з цікавою подачею. Цей програмний продукт допоможе людям гарно провести вільний час та зняти стрес після важкого дня.

В цьому розділі також було поставлено задачі для розробки програмного модуля авторської 2D TopDown гри.

2 ПРОЕКТУВАННЯ ПРОГРАМНОГО МОДУЛЯ АВТОРСЬКОЇ 2D TOPDOWN ГРИ

2.1 Розробка механік авторської 2D TopDown гри

Ігрова механіка це одна з складових розробки ігор, яка додає особливості та впливає на жанр гри. Іноді в іграх застосовують поєднання простих ігрових механік в поєднанні з основною механікою гри для урізноманітнення ігрового процесу.

Вторинна механіка додає глибини, смаку та різноманітності. Гра все одно працювала б чудово без них, але їх застосування додає нових вражень і відкриває нові виклики.

До ігрової механіки входять правила та системи, які регулюють ігровий процес і визначають досвід гравця. Від руху персонажа до бойової механіки, механіка ігрового дизайну є основою, на якій будується гра. Гра все ще може бути успішною без приголомшливої графіки та унікального художнього стилю (це все ж таки велика перевага), але не без привабливої ігрової механіки.

Основна механіка є основою гри, але часто вона виражається кількома способами. У Portal ви починаєте із взаємодії з існуючими порталами, потім отримуєте порталну гармату, яка посилається на існуючий портал, а потім оновлюється до створення всіх порталів самостійно. Це три способи гри з основною механікою порталу. Крім того, використання вашого порталу підтримується або заперечується багатьма іншими механізмами, додаючи різноманітності.

Вторинна механіка, з іншого боку, зазвичай обмежена простою формою і не має підтримки з боку інших механік [18].

При розробці авторської 2D TopDown гри головним натхненням є гра Stardew Valley. Ця гра є гарним прикладом жанру затишних ігор, але також ця гра має локації з бойовою частиною. Для покращення ігрового досвіду гравця таке

поєднання двох різних ігрових механік буде гарним рішенням. Для цього гру поділено на два gamemode, які переключаються за командою гравця. Один з них це RPG gamemode, інший Action gamemode.

Основна особливість RPG ігор це те, що основний фокус наведений на сюжет та історію. Гравець досліджує світ, виконує завдання в яких дізнається більше про сюжет та події у грі, часто навіть з можливістю вплинути і змінити хід подій.

Натомість Action ігри мають особливості як бойова механіка з різними типами зброї, система здоров'я, вороги, можливість їх вбивати та отримувати шкоду від них, спецефекти та більш напружена атмосфера.

На рисунку 2.1 представлено roadmap авторської 2D TopDown гри з 2 ігровими режимами.

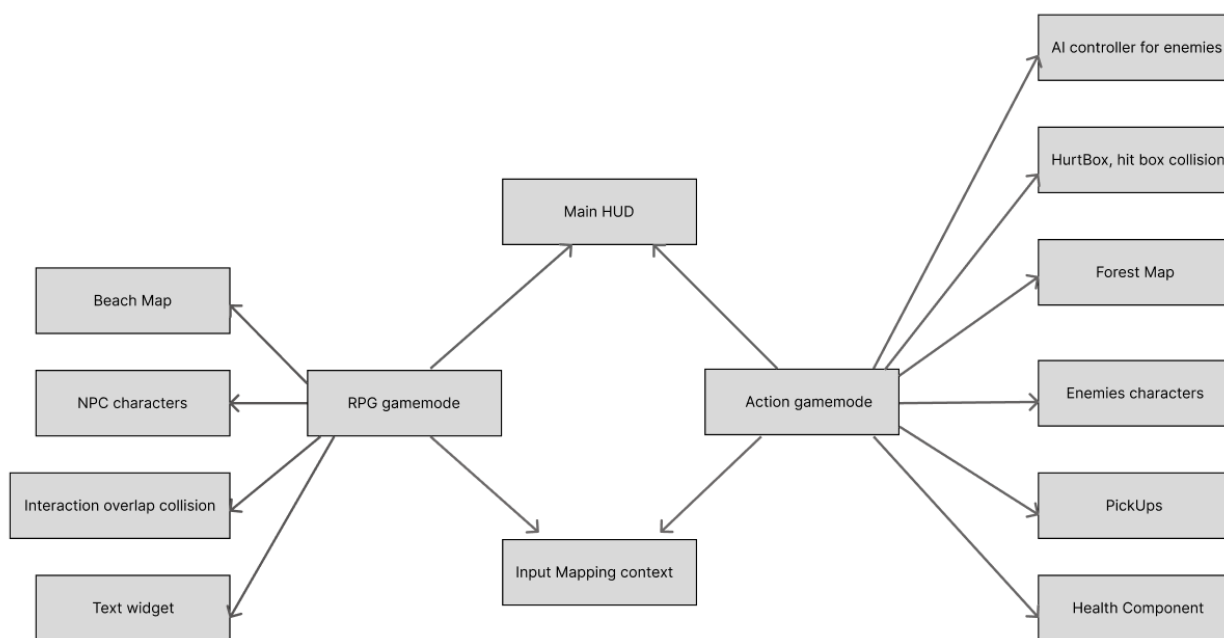


Рисунок 2.1 – Roadmap авторської 2D TopDown гри з 2 ігровими режимами

Ці ігрові режими мають різні ігрові механіки. До RPG режиму входить карта з пляжем, неігрові персонажі з якими можна тільки розмовляти, але немає можливості завдати їм шкоди чи отримати її, також інші налаштування колізій для можливості взаємодії з неігровими персонажами.

До Action режиму входить бойова система з двома видами зброї, наявність колізій для отримання шкоди, колізія меча та стріл для завдання шкоди іншим ворогам, карта лісу, додавання простого штучного інтелекту для контролю ворогами, щоб вони постійно переслідували гравця. Крім того, має бути система здоров'я та запровадження системи отримання бонусів (додаткове здоров'я чи зброя) при піднятті певних предметів.

Ці два ігрові режими поділяють спільний модуль інтерфейсу, який динамічно оновлюється в залежності від того в якому режимі знаходиться гравець. А також спільний Input Mapping Context, який відповідає за призначення клавіш, керування персонажем та налаштуваннями для контролера.

2.2 Розробка UML-діаграми

Об'єктно-орієнтоване програмування (ООП) у розробці відеоігор дозволяє структурувати дизайн за допомогою класів та об'єктів, віддзеркалюючи сутності реального світу. В іграх такі об'єкти, як гравці, вороги та предмети, можна моделювати як класи, інкапсулюючи властивості (наприклад, здоров'я, положення) і поведінку (наприклад, рух, атака). Успадкування дозволяє мати спільні риси між порогами та дружніми персонажами, оскільки вони успадковують базовий клас «Персонаж». ООП допомагає розробляти ігри модульно та більш гнучко. Це допомагає уникнути повторення коду, повторно використовуючи функції [19].

UML-діаграми класів допомагають презентувати та показати структуру модулів у грі. Діаграми класів є одними з найкорисніших типів діаграм в UML, оскільки вони чітко відображають структуру конкретної системи шляхом моделювання її класів, атрибутів, операцій і зв'язків між об'єктами.

На рисунку 2.2 представлено UML-діаграма класів для персонажів Action gamemode.

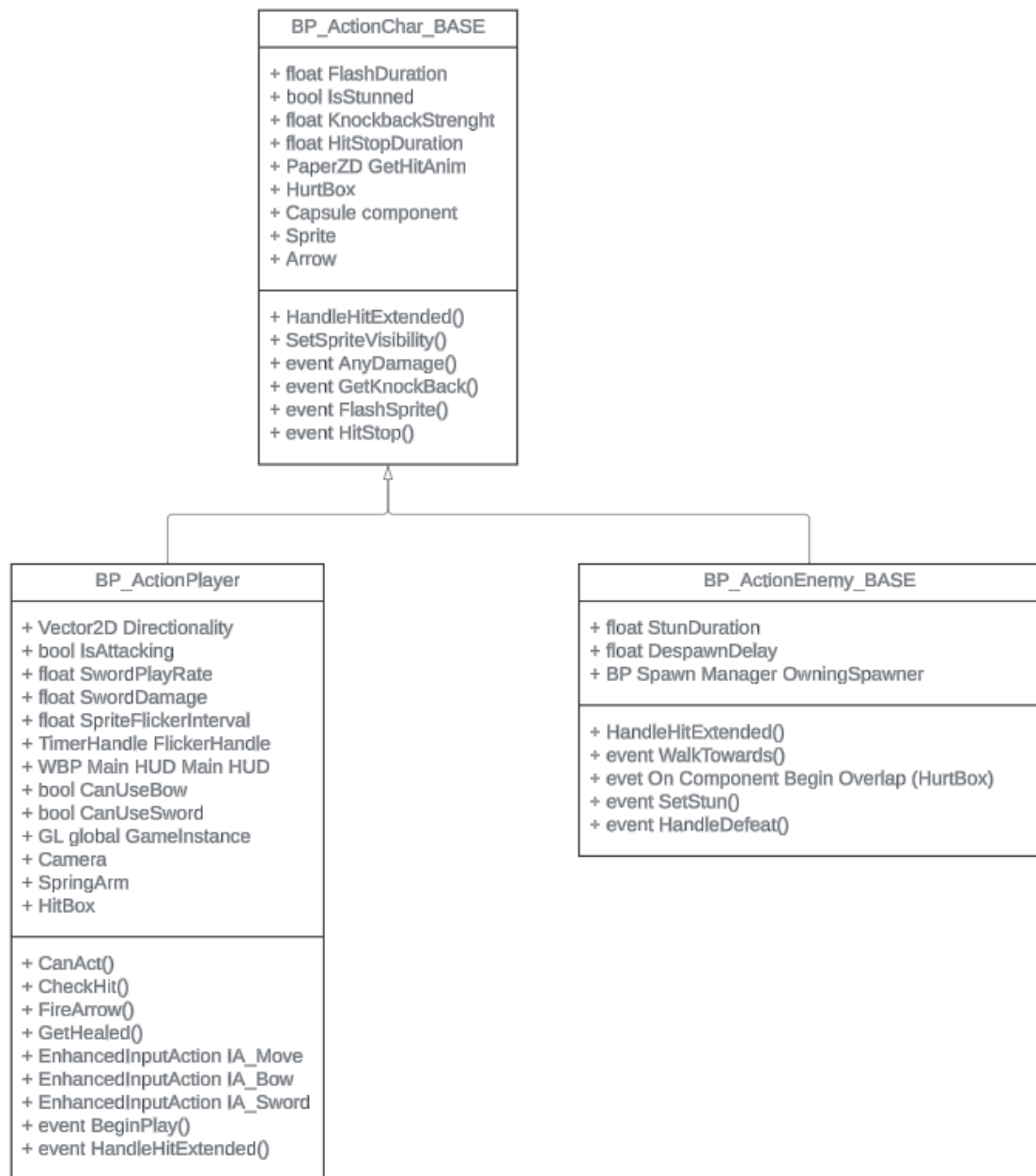


Рисунок 2.2 – UML-діаграма класів для персонажів Action gamemode

BP_ActionCharBase це батьківський клас, в якому буде функція, яка відповідає за отримання шкоди, функція відкидування після отримання удару від іншого персонажа, функція HitStop, яка буде зупиняти персонажа на деякий час після отримання шкоди. Також у цьому класі представлена функція, яка змінює видимість персонажа, цю функцію потім буде використано для встановлення мерехтіння після отримання шкоди. Такий візуальний ефект в поєднанні з

функцією, яка змінює кольоровий тон персонажа дає гравцеві відчуття, що шкода нанесена.

BP_ActionPlayer це клас ігрового персонажа в Action режимі. В цьому класі налаштовано отримання контролю клавіатурою або геймпадом, що відповідає за можливість управління персонажем. Також тут налаштовано управління мечом, луком та функція, що обробляє отриманий удар від ворогів.

BP_ActionEnemy_BASE це загальний клас ворогів. У цьому класі налаштовано функцію переслідування головного героя та його ураження у разі наближення. Також налаштовано функції оглушення ворога після удару та поява VFX ефекту у разі смерті ворога.

На рисунку 2.3 представлено ієрархію класів персонажів для Action режиму.

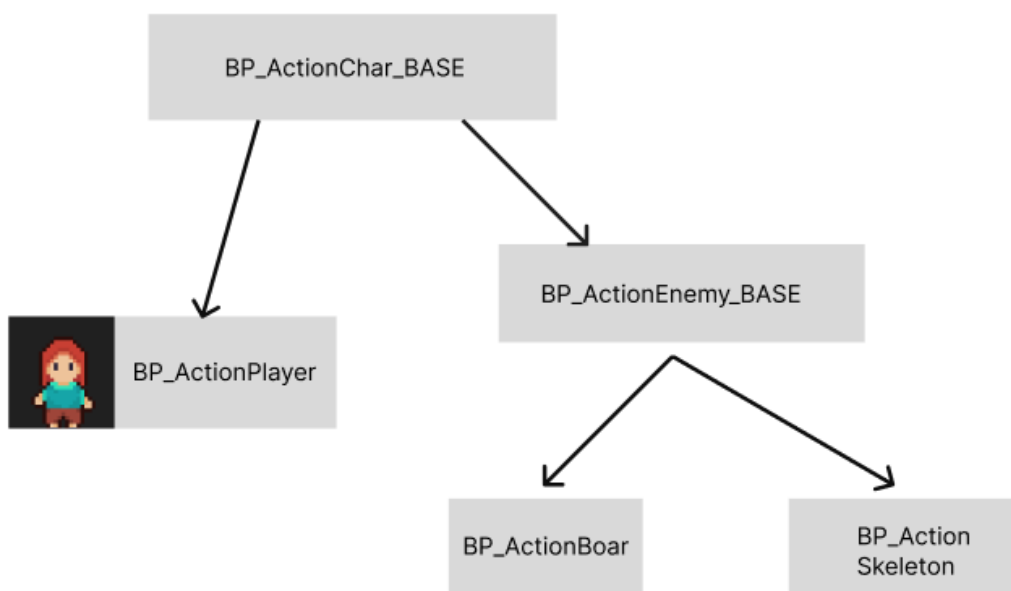


Рисунок 2.3 – Ієрархія класів персонажів для Action режиму

З рисунку 2.3 видно, що клас BP_ActionEnemy_BASE має два дочірні класи. Ці два класи наслідують функції від BP_ActionEnemy_BASE та відрізняються тільки спрайтами, тобто виглядом персонажів. Тобто є два види ворогів з однаковою механікою.

Також для Action режиму налаштована система PickUp. Є батьківський клас BP_PickUp_BASE, в якому прописана функція якщо колізія головного персонажу накладається на колізію предмета, то предмет зникає та накладається функція з дочірнього класу, яка переписується для кожного з дочірнього класу.

На рисунку 2.4 наведено UML-діаграму класів PickUp системи для Action режиму.

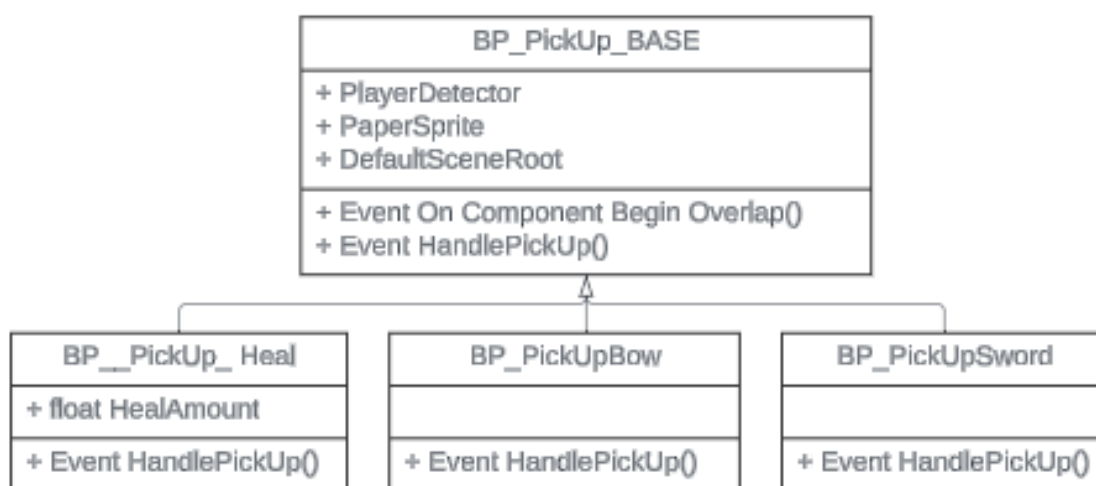


Рисунок 2.4 – UML-діаграма класів PickUp системи для Action режиму

На рисунку 2.5 представлено UML-діаграму класів для персонажів RPG режимі.

BP_RPGChar_BASE клас це батьківський клас, який має всі основні функції для майбутньої взаємодії персонажів з іншими персонажами або об'єктами. В цьому класі налаштована механіка для показу тексту в вікні діалогу.

BP_RPGPlayer це клас ігрового персонажу в RPG режимі. В цьому класі так само як в BP_ActionPlayer налаштовано отримання контролю клавіатурою або геймпадом, що відповідає за можливість управління персонажем. Також налаштовано доступ до глобальної інстанції для зберігання значень деяких змінних при переході з одного ігрового режиму в інший. Водночас налаштовано функції руху та інтеракцій, показу діалогу та закінчення діалогу.

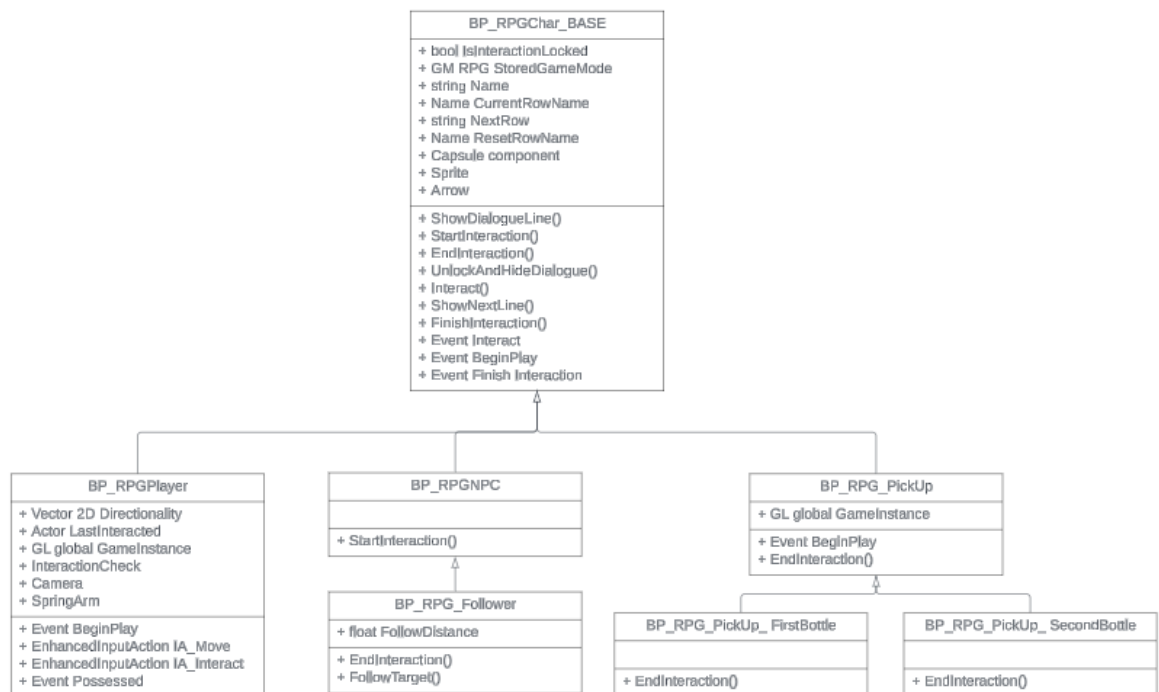


Рисунок 2.5 – UML-діаграма класів для персонажів RPG режимі

У класі BP_RPGNPC переписано функцію початку інтеракції з батьківського класу та враховано, що при діалозі з неігровим персонажем він має бути повернутий до того, з ким він говорить.

BP_NPCFollower налаштовано функцію переслідування персонажа, це буде майбутній друг головного героя.

На відмінну від PickUp системи для Action режиму, для RPG вона налаштована по-іншому. Клас BP_RPGPickUp_BASE наслідується від батьківського BP_RPGChar_BASE, тому що при взаємодії головного героя з пляшкою із запискою має з'являтися діалогове вікно з текстом записки. Тобто механіка більше схожа до взаємодії з персонажами.

На рисунку 2.6 представлено ієрархію класів персонажів та об'єктів для RPG режиму.

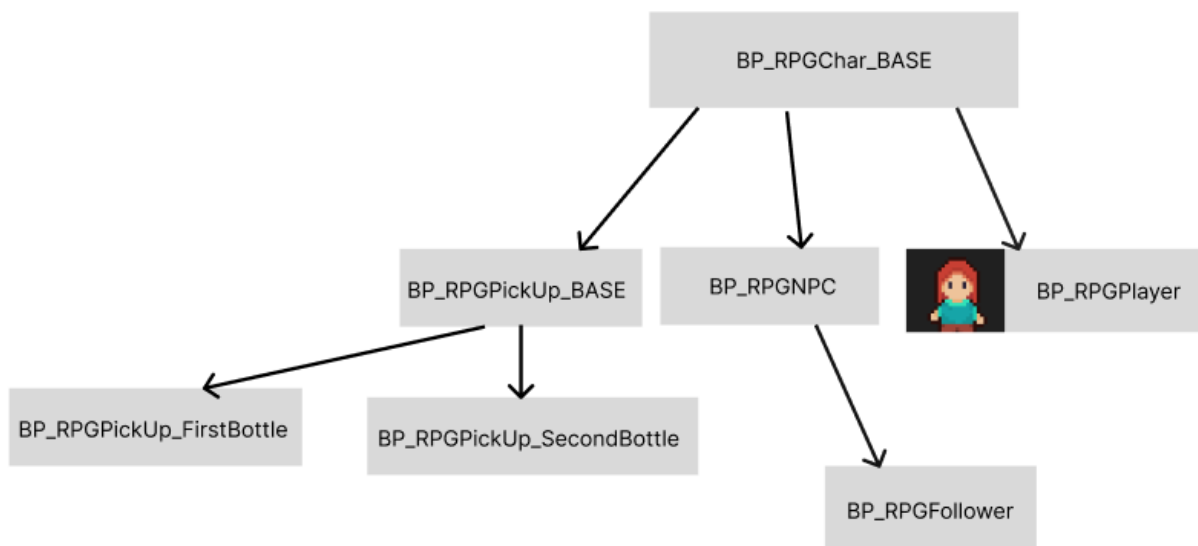


Рисунок 2.6 – Ієрархія класів персонажів та об’єктів для RPG режиму

2.3 Розробка схеми алгоритму роботи програмного модуля авторської 2D TopDown гри

Алгоритм – це процедура, яка використовується для вирішення проблеми або виконання обчислень. Алгоритми діють як точний список інструкцій, які крок за кроком виконують певні дії в апаратних або програмних процедурах.

Алгоритми широко використовуються в усіх сферах ІТ. У математиці, комп'ютерному програмуванні та інформатиці алгоритм зазвичай відноситься до невеликої процедури, яка розв'язує повторювану проблему. Алгоритми також використовуються як специфікації для виконання обробки даних та відіграють важливу роль в автоматизованих системах [20].

Схема алгоритму розроблена, щоб відобразити потік різних кроків у алгоритмі. Точне представлення алгоритму дозволяє ефективно оцінювати, оптимізувати та ділитися своїми процесами з командою [21].

Під час розробки відеоігри створюється багато функцій, подій, змінних. На рисунку 2.7 показано схему алгоритму події отримання будь-якої шкоди.



Рисунок 2.7 – Схема алгоритму події отримання будь-якої шкоди

Алгоритм події отримання будь-якої шкоди (рис. 2.7) складається з таких кроків:

Крок 1. Перевірка чи персонаж може бути уражений, це враховує сценарій, коли персонаж вже помер і більше неможливо завдати йому ніякої шкоди. Якщо так, то перехід до кроку 2.

Крок 2. Функція зменшення здоров'я, яка прописана в компоненті здоров'я та визивається з однойменного класу.

Крок 3. Функція відкидання ушкодженого опонента на певну відстань, щоб запобігти миттєву смерть і дати можливість персонажу прийти в себе.

Крок 4. Функція мерехтіння персонажа, що дає візуально зрозуміти, що персонажа поранено.

Крок 5. Функція обробки шкоди, яка більш детально представлена на рисунку 2.8.

Крок 6. Функція затримки удару, що заморожує атакуючого і того, хто атакує на невеликий проміжок часу. Ця функція потрібна, щоб дати гравцеві час, для того щоб зрозуміти, що відбувається та що робити потім.

На рисунку 2.8 представлено схему алгоритму роботи функції обробки шкоди для головного героя.

Алгоритм роботи функції обробки шкоди для головного героя (рис. 2.8) складається з таких кроків:

Крок 1. Оновлення поточного здоров'я.

Крок 2. Функція непереможності персонажа відбувається деякий час, тобто протягом визначеного часу щойно пораненому персонажу не можна нанести шкоду ще раз. Також від час цієї функції відключається колізія, що дає змогу пройти наскрізь інших персонажів. Це запобіжний захід на випадок коли персонаж оточений ворогами і не може відійти на безпечну дистанцію.

Крок 3. Виклик анімації поранення, вона присутня тільки у головного персонажу, всі інші вороги просто мерехтять та загоряються червоним на мить, щоб дати зрозуміти, що поранення відбулось.

Крок 4. Функція затримки удару.

Крок 5. Перевірка чи персонажа переможено. Якщо так, то наступний крок 6, якщо ні, то крок 7.

Крок 6. Функція анімації смерті.

Крок 7. Функція мерехтіння персонажа.



Рисунок 2.8 – Схема алгоритму роботи функції обробки шкоди для головного героя

На рисунку 2.9 представлено схему алгоритму події користування мечем.



Рисунок 2.9 – Схема алгоритму події користування мечем

Алгоритм події користування мечем (рис. 2.9) складатиметься з таких кроків:

Крок 1. Перевірка чи може діяти, вона потрібна для уникнення руху персонажа, якщо він вже помер.

Крок 2. Перевірка чи може використовувати меч, яка потрібна, щоб меч можна було використовувати тільки коли персонаж побачить його та підніме.

Крок 3. Чи атакує встановлюється на False, що забезпечує послідовне програвання анімації.

На рисунку 2.10 представлено схему алгоритму події взаємодії головного персонажа з іншими персонажами чи об'єктами.

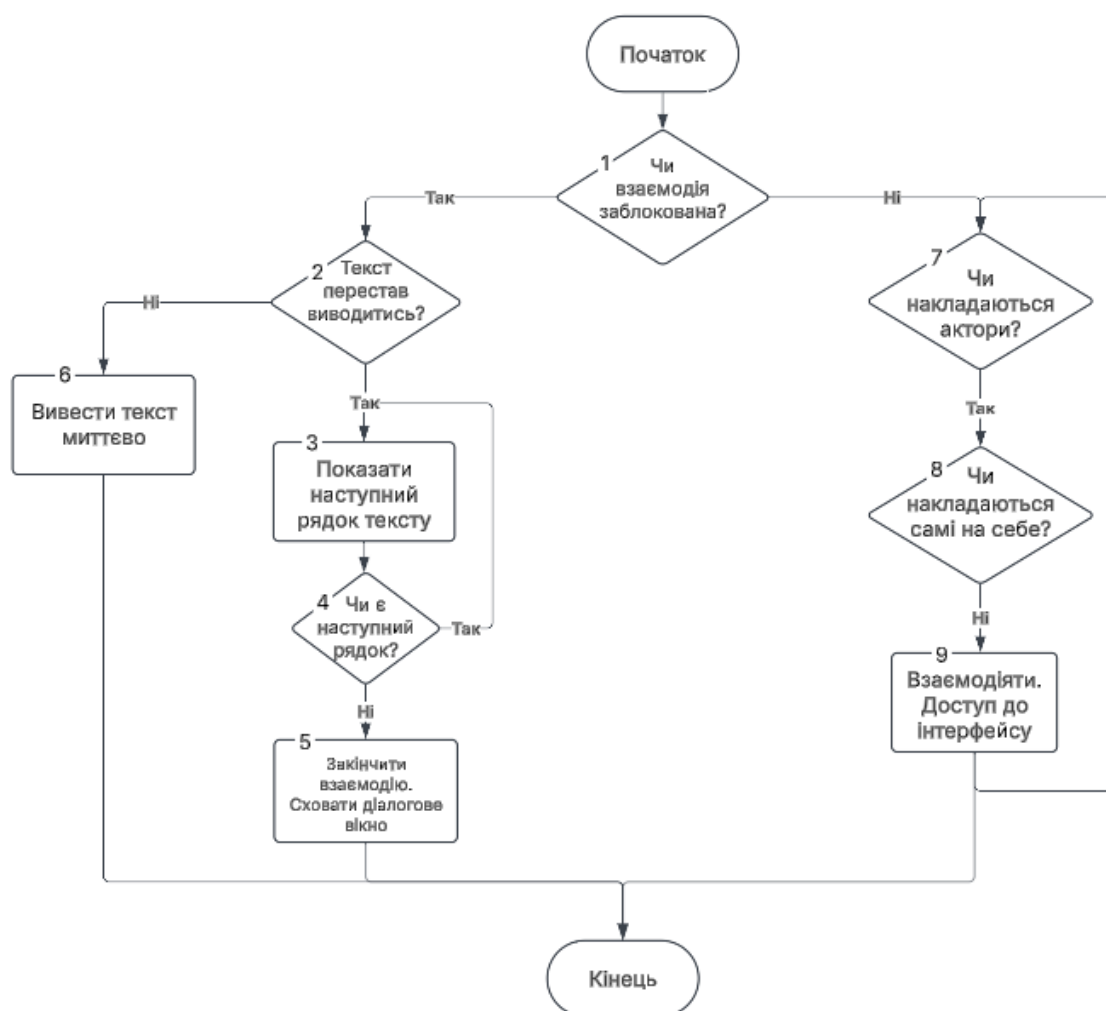


Рисунок 2.10 – Схема алгоритму події взаємодії головного персонажа з іншими персонажами чи об'єктами

Алгоритм події взаємодії головного персонажа з іншими персонажами чи об'єктами (рис. 2.10) складається з таких кроків:

Крок 1. Перевірка чи заблокована взаємодія, тобто чи головна героїня взаємодіє з кимось. Ця перевірка використовується в декількох інших місцях і

запобігає руху головного героя та інших персонажів під час діалогу, а також початок нового діалогу до закінчення активного діалогу. Якщо взаємодія заблокована перехід до кроку 2, якщо ні – крок 7.

Крок 2. Перевірка чи показався поступово весь текст, якщо ні перехід до кроку 6, якщо так – крок 3.

Крок 3. Показати наступний рядок тексту.

Крок 4. Перевірка чи є наступний рядок. Якщо так, то перехід до кроку 3, якщо ні – кроку 5.

Крок 5. Закінчення взаємодії. Сховати діалогове вікно.

Крок 6. При повторному натисканні кнопки «Е», то вивести текст миттєво.

Крок 7. Перевірка чи накладаються актори. Якщо так, то перехід до кроку 8.

Крок 8. Перевірка чи накладаються актори самі на себе. Якщо ні, то перехід до кроку 9.

2.4 Висновок

У цьому розділі було розроблено структуру програмного модуля авторської 2D TopDown гри, що дозволяє налагодити та полегшити організовану розробку програмного забезпечення. Визначено, що гра буде поділена на два ігрових режими, описано їх модулі та зв'язок між ними.

Було створено UML-діаграми основних класів, що описують механіки двох ігрових режимів. Описано змінні та методи, які будуть містити класи. Представлено загальну ієрархію класів для розуміння особливостей розробки.

Крім того, було розроблено схеми основних алгоритмів функціонування програмного модуля авторської 2D TopDown гри. Схеми алгоритмів дозволяють продумати деталі розробки та полегшити подальшу розробку. За допомогою схем алгоритмів було визначено як можна оптимізувати роботу функцій.

Такі кроки є необхідними для ефективної розробки такого програмного продукту як відеогра, вони дозволяють притримуватись визначеної структури та виявити проблеми на ранніх етапах розробки. При розробці гри дуже важливо продумати ієрархію та зв'язки між модулями наперед, щоб зробити якісний продукт на завершенні та менше перероблювати в процесі розробки.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ПРОГРАМНОГО МОДУЛЯ АВТОРСЬКОЇ 2D TOPDOWN ГРИ

3.1 Обґрунтування вибору ігрового рушія та мови програмування

Для розробки програмного модуля авторської 2D TopDown гри обрано ігровий рушій Unreal Engine 5. Це нестандартний вибір рушія для 2D гри, загалом для таких проєктів прийнято обирати Unity або Godot, але Unreal Engine 5 має певні переваги, яких не мають ці два рушії.

Unreal Engine 5 – це сучасний потужний ігровий рушій, який доступний не тільки великим компаніям, а також прийнятний для розробки інді-ігор. Це доступне безкоштовне програмне забезпечення, яке підтримує багато сучасних потреб.

Unreal Engine 5 підтримує повноцінну розробку за допомогою C++ та систему візуального скриптингу Blueprints. Система візуальних сценаріїв Blueprints у Unreal Engine – це повна система сценаріїв ігрового процесу, заснована на концепції використання інтерфейсу на основі вузлів для створення елементів ігрового процесу в Unreal Editor. Як і в багатьох поширених мовах сценаріїв, він використовується для визначення об'єктно-орієнтованих (ОО) класів або об'єктів у механізмі.

Ця система є надзвичайно гнучкою та потужною, оскільки надає дизайнерам можливість використовувати практично повний спектр концепцій та інструментів, які зазвичай доступні лише програмістам. Крім того, спеціальна розмітка Blueprints, доступна в C++ реалізації Unreal Engine, дозволяє програмістам створювати базові системи, які можуть бути розширені дизайнерами. За допомогою Blueprints можна створити повноцінну гру.

Щоб компенсувати недостатню підтримку роботи з 2D іграми в Unreal Engine було використано безкоштовний плагін PaperZD. Цей плагін підтримує розробку двовимірних ігор та ізометричних ігор. За допомогою PaperZD можна

налаштовувати різнонаправлену анімацію та налаштування зміни анімації залежно від поставлених умов. Загалом для розробки ігор на ігровому рушії Unreal Engine 5 використовують поєднання Blueprints з C++, хоча можливо використання одного із способів для повноцінної розробки. Зазвичай C++ використовується для створення різних систем, а Blueprints – для надання інструкцій, визначення поведінки взаємодії між існуючими системами.

Blueprints не обмежуються лише класами акторів – їх можна використовувати з різними класами: ігровими режимами, ігровими станами, персонажами та пішаками, щоб назвати декілька. Можливо легко використовувати функціональні можливості за допомогою компонентів Blueprint або бібліотек макросів і функцій, що забезпечує ще швидший і гнучкіший дизайн. Blueprints – це візуальний спосіб кодування C++, що робить його більш зрозумілим. Також C++ має більший поріг входу для початку роботи, ніж Blueprints.

До переваг використання C++ відносять:

- Можливість зберігати частини коду для використання в інших проєктах;
- Легко будувати нові системи та контролювати їх, створювати інтерфейси із зовнішніми бібліотеками.

Для розробки програмного модуля авторської 2D TopDown гри обрано ігровий рушій Unreal Engine 5 за допомогою Blueprints через їх гнучкість, зрозумілість та минулий досвід роботи з Blueprints та Unreal Engine 5.

3.2 Обґрунтування вибору програмного забезпечення для створення графічної частини

Існує багато програмного забезпечення для 2D графіки, але не всі вони зручні для піксельної графіки.

Наприклад, безкоштовне програмне забезпечення Krita, яке прийнятне для 2D графіки, але для того, щоб розібратись з інтерфейсом потрібен час. Також

Adobe Photoshop – це мультизадачне програмне забезпечення для 2D графіки, але має велику складність при використанні. Також Adobe Photoshop платний та потребує щомісячної підписки для користування.

Також є декілька більш застарілих програм для роботи з піксель графікою, такі як Gimp, Pixieditor, Pyxeledit, Graphicgale та інші. Але однією з популярних програм є Aseprite. Це платне програмне забезпечення середньої складності, розроблено для піксельної графіки та створення анімацій. Aseprite має не складний інтерфейс з яким можна швидко розібратись. Також є багато різних уроків та порад через велику і дружню спільноту користувачів.

Порівняно з Adobe Photoshop та Krita, в Aseprite значно легше створювати анімації та тайлесети, які потрібні для розробки гри. Також це програмне забезпечення підтримує експорт файлів у форматі json. Це особлива функція та мало які програми мають її. Для пришвидшення створення фліпбуків найкраще імпортувати спрайти у Unreal Engine 5 у форматі json, що значно прискорює ділення спрайт файла на однакові кадри та створює правильні фліпбуки для кожної анімації. Такий метод прискорює імпорт файлів в рази. Зауважимо, якщо є тільки файл спрайта у форматі png, тоді потрібно розділи лист спрайтів на кадри. Для цього найкраще використати режим сітки, а не авто та розділити висоту на кількість рядків, а ширину на кількість стовпців. Таким чином лист спрайтів поділиться правильно і анімаційні фліпбуки будуть відображатись коректно.

Найкращим варіантом є використання Aseprite для розробки графічної частини авторської 2D TopDown гри.

Також для покращення графічної частини у Unreal Engine потрібно звернути увагу, що коли текстури імпортуються вони виглядають нечіткими. Для коректного відображення текстури спрайта потрібно встановити певні налаштування. За замовчуванням усі 2D-зображення, імпортовані в Unreal Engine, будуть імпортовані як текстурні ресурси. Обробка зображень за замовчуванням, яка призначена для обробки текстур для 3D-моделей і об'єктів, згладжує та пом'якшує файли з низькою роздільною здатністю, наприклад 2D-

персонажі та об'єкти. Тому для того, щоб зберегти вигляд спрайтів із низькою роздільною здатністю, потрібно застосувати до ресурсу налаштування двовимірної текстури паперу. Крім того, необхідно змінити фільтр текстури з Default на Nearest. На рисунку 3.1 подано порівняння текстур спрайта головного персонажа з правильними налаштуваннями та без них.

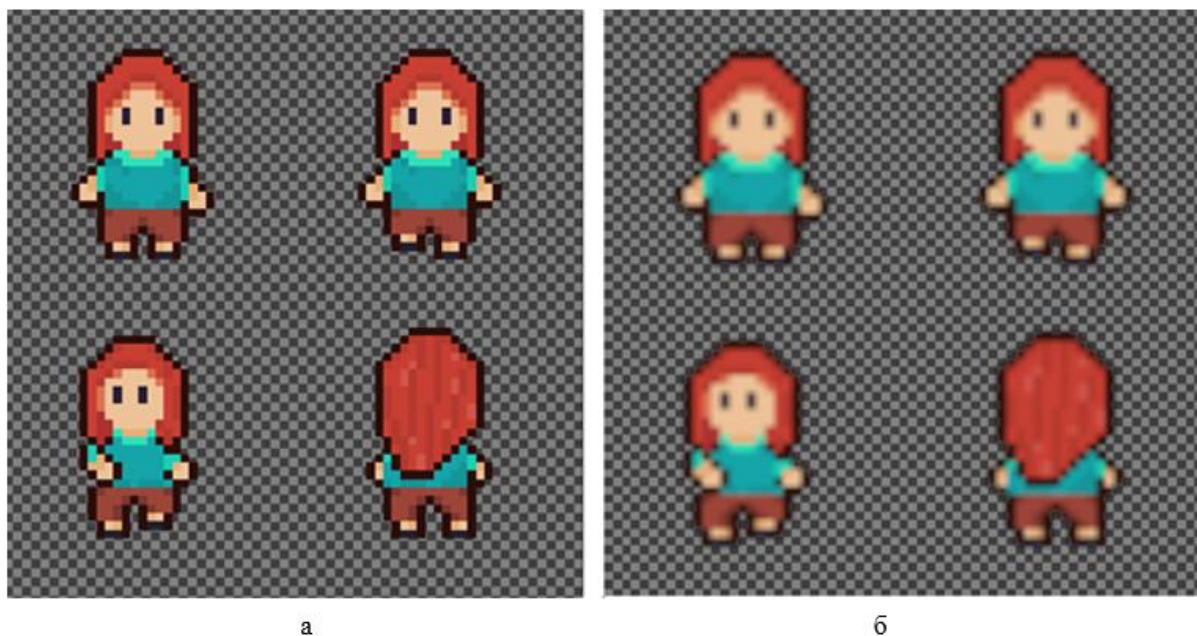


Рисунок 3.1 – Порівняння текстур спрайта головного персонажа: а – з правильними налаштуваннями; б – без правильних налаштувань

3.3 Розробка RPG gamemode

Для оптимальності гру розділено на 2 режими (gamemode), це також полегшує розробку ігрових механік.

GameMode визначає набір правил гри. Правила можуть включати спосіб приєднання гравців до гри, призупинення гри та перехід рівня, а також будь-яку специфічну поведінку гри, наприклад умови виграшу. GameMode встановлюється для кожного рівня, і GameModes можна повторно використовувати на кількох рівнях [22].

Наприклад ігровий режим RPG включає в себе розробку діалогів та взаємодії з іншими персонажами та предметами. Для цього створено основний

батьківський клас Blueprints PaperZDCharacter. Ієрархію класів Blueprints PaperZDCharacter наведено на рисунку 3.2.

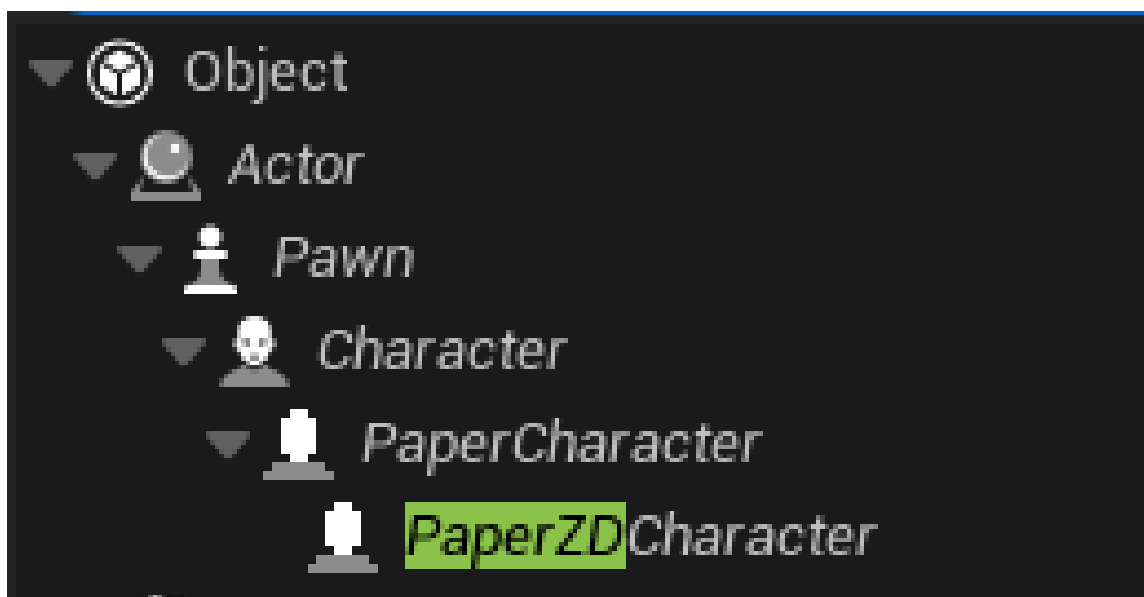


Рисунок 3.2 – Ієрархію класів Blueprints PaperZDCharacter

Для налаштування системи діалогів та взаємодії було використано інтерфейси. В Unreal Engine інтерфейсні класи корисні для того, щоб набір (потенційно) непов'язаних класів реалізовував загальний набір функцій. Це дуже корисно у випадках, коли деяка ігрова функціональність може бути спільною для великих, складних класів, які інакше не схожі.

Наприклад, у грі може бути система, за допомогою якої введення гучності тригера може активувати пастки, сповіщати ворогів або присуджувати бали гравцеві. Наприклад, пастки походять від AActor. Вороги походять від Пішки або Персонажа. Бали нараховуються з UDataAsset. Ці класи потребують спільної функціональності, але не мають спільного батька, крім базового UObject. У цьому випадку Unreal Interface може змусити всі ці об'єкти реалізувати потрібні функції [23].

У цьому проєкті інтерфейс використано для налаштування діалогової системи, для показу текстових рядків реплік персонажів. Це і є основна механіка RPG режиму гри.

У BP_RPGChar_BASE запроваджено за допомогою події Interact підключення до інтерфейсу. На рисунку 3.3 показано EventGraph класу BP_RPGChar_BASE.

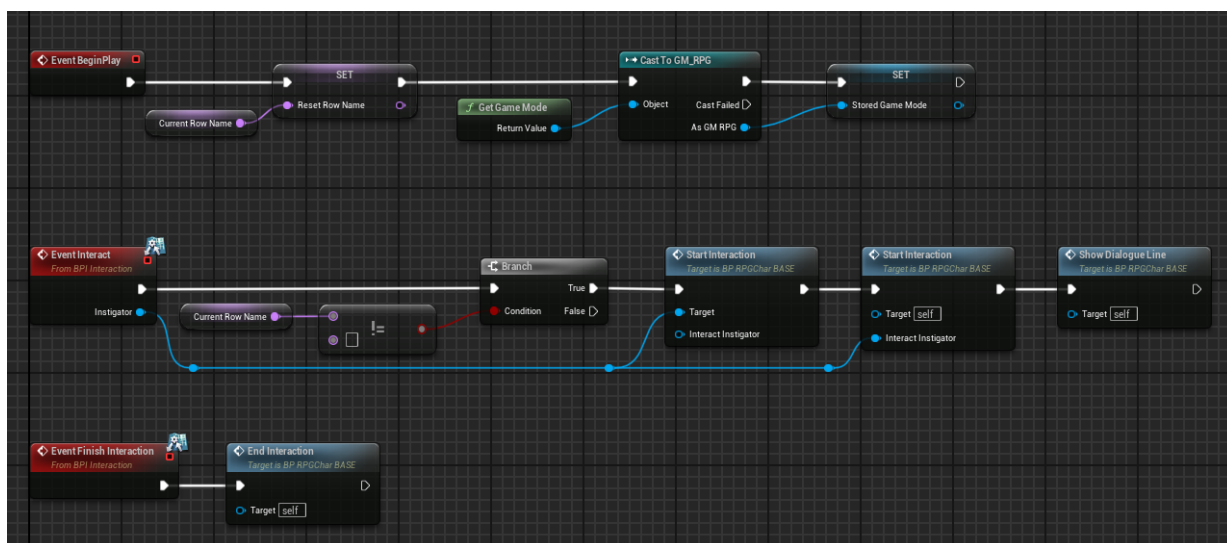


Рисунок 3.3 – EventGraph класу BP_RPGChar_BASE

Події – це вузли, які викликаються з коду ігрового процесу, щоб почати виконання окремої мережі в межах EventGraph. Вони дозволяють Blueprints виконувати серію дій у відповідь на певні події, що відбуваються в грі, наприклад, коли гра починається, коли рівень скидається або коли гравець отримує пошкодження [24].

Уся взаємодія відбувається між головним героєм та іншими персонажами чи предметами. Тому система показу діалогів і текстових рядків налаштована у BP_RPGPlayer. Також у RPG режимі присутні катсцени, для розробки яких потрібно врахувати, що весь ігровий світ завмирає і головним персонажем не можна керувати в цей момент. Для цього при налаштуванні керування є перевірка.

Подія IA_Interact прив'язана до дії введення, тобто виконується при натисненні на певну клавішу, в даному випадку клавішу E. При налаштуванні цієї події виконується перевірка чи взаємодія заблокована. Потім йде наступна перевірка чи закінчено показ всього тексту рядка. Якщо так, то тоді за наявності

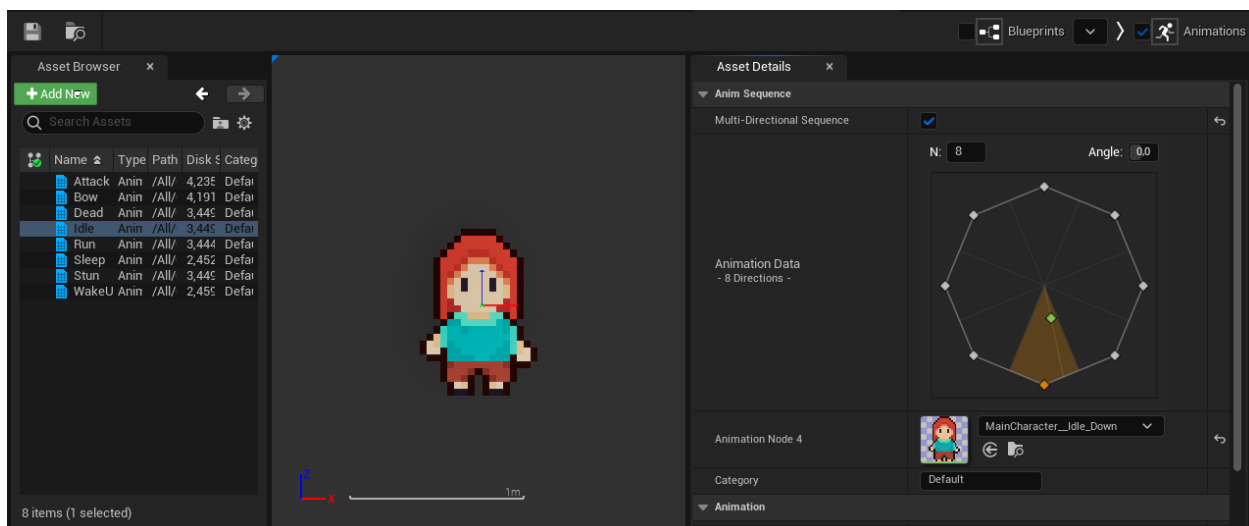


Рисунок 3.7 – AnimationSource головного персонажа

Для цього персонажа налаштовано 8 різних анімацій з 8 різними напрямленостями. Для налаштування руху та відображення правильних анімацій створено *AnimationSource Blueprints*. За допомогою нього можна створити State Machine, де налаштовують умови для відображення потрібних анімацій. Таким чином налаштування відображення певних анімацій значно оптимізованіше, ніж без AnimationSource Blueprints. Також налаштовано AnimNotify для анімації удару та використання лука, що допоможе з коректним налаштуванням отримання шкоди від зброї основного персонажа.

Крім того, для налаштування зміни анімації у AnimationInstance використано state machine. В залежності від швидкості персонажа, якщо вона дорівнює нулю, тоді вмикається анімація бездіяльності персонажа, якщо швидкість більше нуля, то вмикається анімація ходьби.

State Machines – це модульні системи, які можна створювати в Animation Blueprints, щоб визначати певні анімації, які можна відтворювати і коли їх можна відтворювати. У першу чергу цей тип системи використовується для співвіднесення анімації зі станами руху ваших персонажів, такими як бездіяльність, ходьба, біг та стрибки. За допомогою State Machines можливо створювати стани, визначати анімацію для відтворення в цих станах і створювати різні типи переходів, щоб контролювати час переходу в інші стани. Це полегшує

створення складного змішування анімації без використання надто складного Anim Graph. Таке налаштування більш гнучке та дозволяє пришвидшити та оптимізувати розробку. Також у більш масштабних проєктах таке налаштування оптимізує гру, тому що для налаштування потрібно менше нодів, ніж налаштування зміни анімацій з умовами без плагіну PaperZD. Використання State Machine у AnimationSource Blueprints економить час, адже налаштування анімацій з більшими напрямленнями та для більшої кількості акторів було б громіздким, якщо налаштовувати здійснювати в Actor Blueprints [26].

Для коректної реалізації Action режиму найкраще мати основний батьківський клас для всіх персонажів. У класі BP_ActionChar_BASE налаштовані функції прийняття шкоди, яка працює як і для головного героя так і для ворогів. На рисунках 3.8 та 3.9 наведено фрагмент коду події AnyDamage у класі BP_ActionChar_BASE.

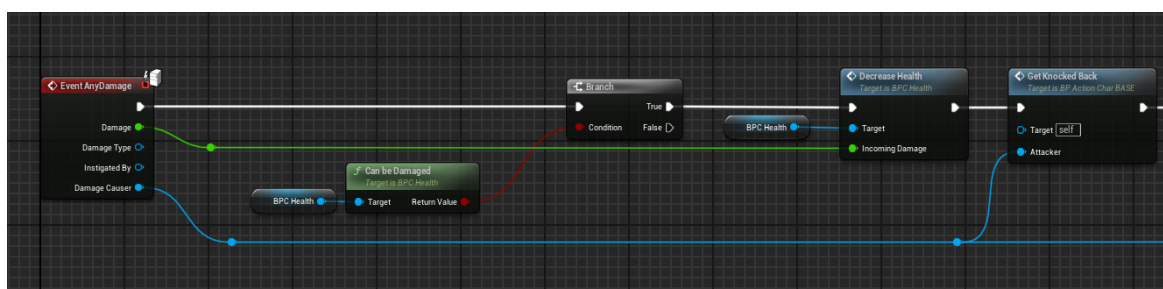


Рисунок 3.8 – Фрагмент коду події AnyDamage у класі BP_ActionChar_BASE

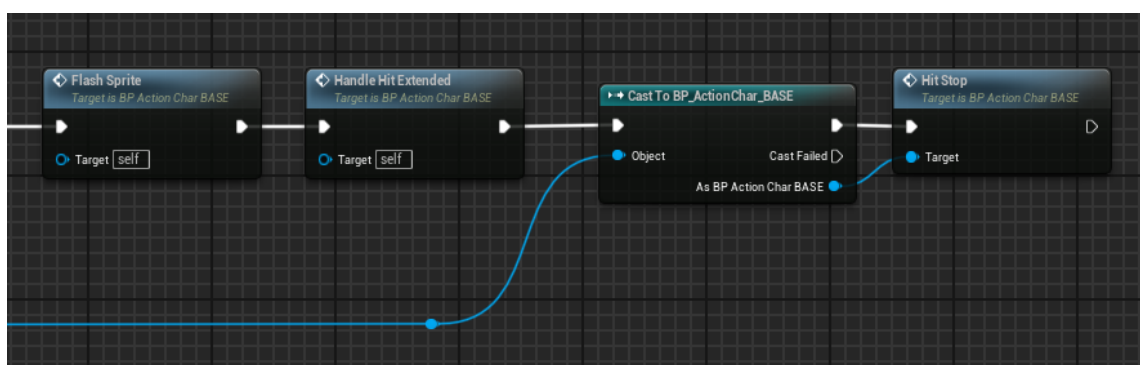


Рисунок 3.9 – Фрагмент коду події AnyDamage у класі BP_ActionChar_BASE

У цій події запроваджено зв'язок з системою здоров'я. Спочатку проводиться перевірка чи можна зашкодити персонажу, тобто чи персонаж ще живий. Потім слідує функція зменшення здоров'я, яка викликається з системи здоров'я. Наступною є функція відкидання персонажа, яка представлена на рисунку 3.10.

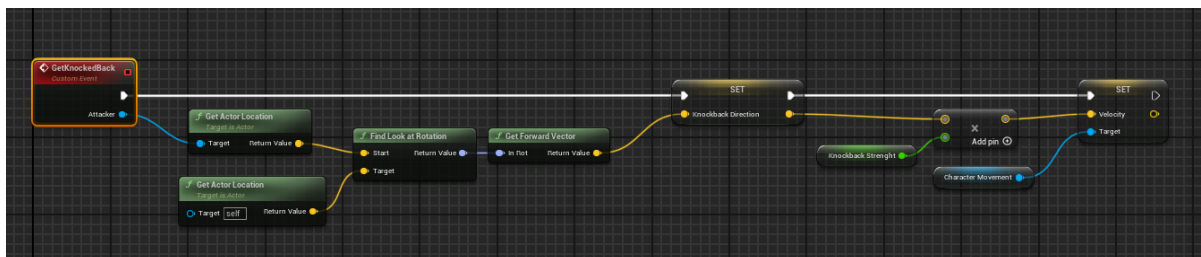


Рисунок 3.10 – Функція відкидання

Наступна функція Flash Sprite відповідає за мерехтіння персонажа червоним кольором після отримання шкоди. На рисунку 3.11 представлено функцію Flash Sprite.

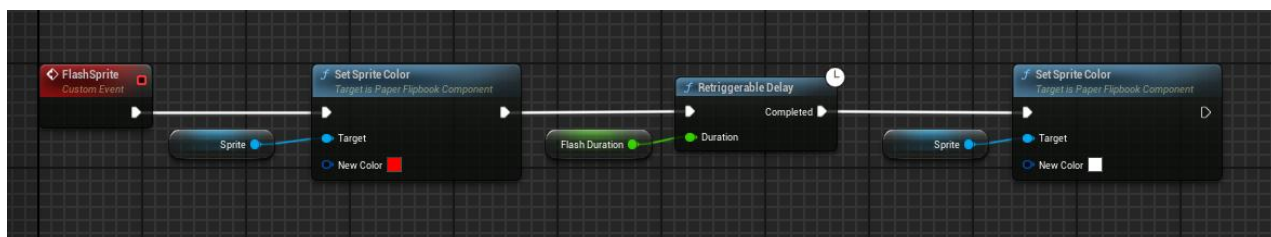


Рисунок 3.11 – Функція Flash Sprite

Також у цьому класі налаштовано функції, які використовуються чи перезаписуються у його дочірніх класах. Однією з таких функцій є Handle Hit Extended, яка створена у батьківському класі BP_ActionChar_BASE, але визначається тільки для головного героя у дочірньому класі BP_ActionPlayer для оновлення здоров'я, встановлення непереможності, зміни роботи колізії та зміни анімаційних фліпбуків.

Потім існує доступ до функції з дочірнього класу `BP_ActionPlayer` функції `HitStop`, яка відповідає за затримку анімації удару, це запроваджує більш натуральне відчуття від атак.

Для Action режиму налаштовано окремий рівень з іншою картою. Для цієї карти налаштовано катсцену, для запровадження якої використовуються налаштування персонажа з RPG режиму, тому що тільки та частина має всі коректні налаштування для правильної роботи катсцени. При потраплянні на нову карту запускається катсцена, яка показує гравцеві меч, яким неможливо користуватись поки персонаж не підніме його. Для налаштування катсцени використано `sequencer`.

`Sequencer` – це багатодоріжковий редактор `Unreal Engine`, який використовується для створення та попереднього перегляду кінематографічних послідовностей у реальному часі. `Unreal Engine` містить надійні інструменти `Cinematic`, які дозволяють створювати анімаційні та кінематографічні послідовності. Є можливість керувати камерами, щоб створювати прольоти рівнів, анімувати світло, переміщувати об'єкти, анімувати персонажів, рендерити вихідні послідовності тощо. В основі цих робочих процесів лежить `Sequencer`, потужний пакет нелінійного редагування [27].

Також на новому рівні налаштована поява ворогів як тільки персонаж потрапляє у певну зону. Після смерті кожного ворога вони зникають з певною анімацією. При вході в кожну нову «кімнату» з'являються нові вороги, яких потрібно вбити. Також з'являється їжа за допомогою якої можна відновити здоров'я. На третьому рівні з'являється лук, який також можна використати на клавішу «К». За допомогою тестування розроблено оптимальне значення здоров'я, а також персонажу видається їжа для лікування та нова зброя у вигляді лука. На рисунку 3.12 представлено рівень лісу у режимі розробки.



Рисунок 3.12 – Зовнішній вигляд рівня лісу у режимі розробки

3.5 Розробка компоненту системи здоров'я

Компоненти – це особливий тип об'єктів, які актори можуть приєднувати до себе як підоб'єкти. Компоненти корисні для спільного використання загальних способів поведінки, таких як здатність відображати візуальне представлення, відтворювати звуки. Вони також можуть представляти специфічні для проєкту концепції, наприклад, як транспортний засіб інтерпретує вхідні дані та змінює власну швидкість та орієнтацію. Наприклад, проєкт з керованими користувачами автомобілями, літаками та човнами може реалізувати відмінності в керуванні транспортним засобом і переміщенні шляхом зміни того, який компонент використовує актор транспортного засобу [28].

Для розробки авторської 2D TopDown гри було розроблено компонент системи здоров'я для Action режиму. Цей компонент підв'язано до класу BP_ActionChar_BASE, тому тепер всі дочірні класи також мають компонент системи здоров'я як свій підоб'єкт. На рисунку 3.13 наведено всі складові класу актора BP_ActionChar_BASE.

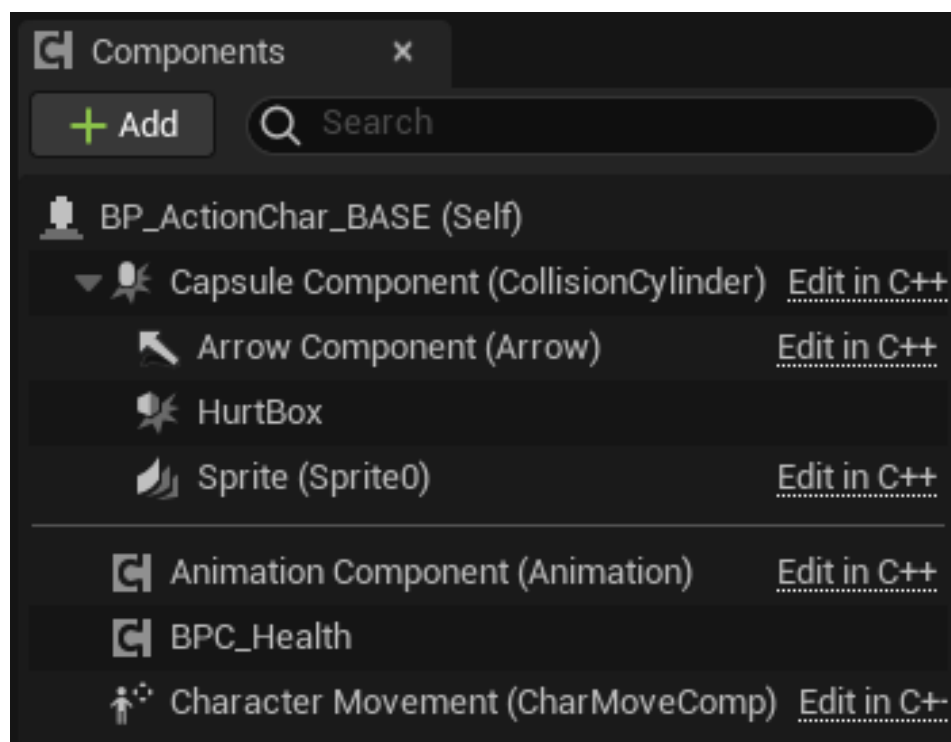


Рисунок 3.13 – Складові класу актора BP_ActionChar_BASE

За допомогою компонента системи здоров'я запроваджена робота події AnyDamage в якій посилаєся на функції компонента системи здоров'я. На рисунку 3.14 наведено функцію зменшення здоров'я.

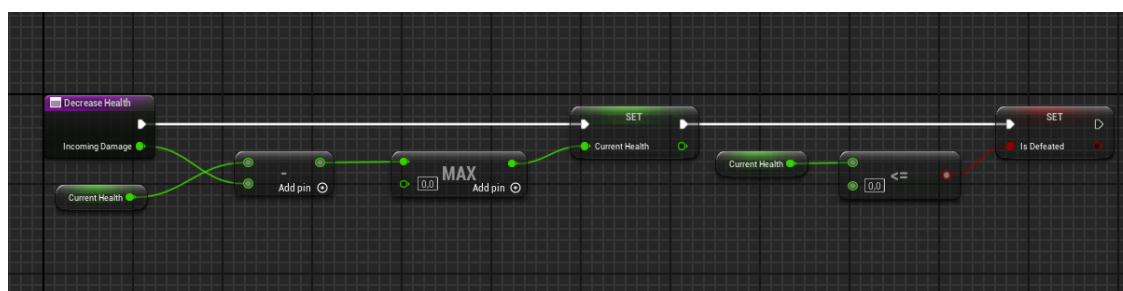


Рисунок 3.14 – Функція зменшення здоров'я

На рисунку 3.15 наведено функція непереможності, яка використовується коли персонаж не приймає жодної шкоди. Ця функція викликається одразу після отримання шкоди та потрібна, щоб попередити ситуацію якщо персонаж буде оточений ворогами, тоді неможливо буде миттєво вбити персонажа і для гравця був час зреагувати та прийняти будь-які дії.

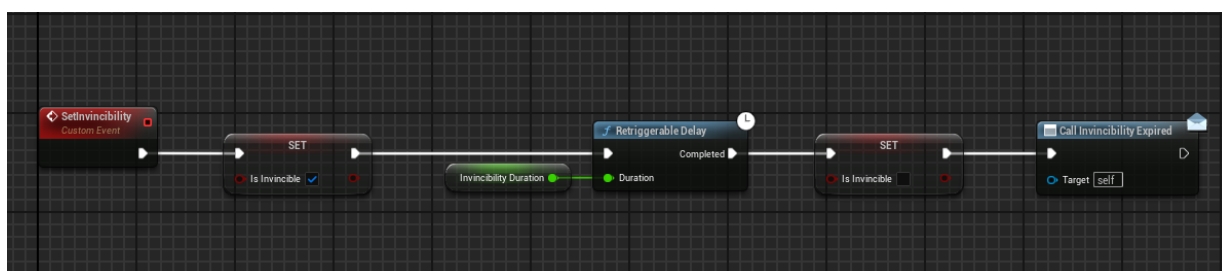


Рисунок 3.15 – Функція непереможності

3.6 Тестування роботи програми та аналіз результатів

Тестування роботи програмного модуля є заключною частиною розробки авторської 2D TopDown гри. Під час розробки було зроблено 100 тестових запусків гри, щоб перевірити як працюють ті чи інші впроваджені функції. Розробка гри потребує постійного тестування та перевірки, щоб гарантувати найкращий ігровий досвід для гравця.

При запуску гри починається катсцена, коли головна героїня прокидається на березі моря і її зустрічає мавпочка. На рисунку 3.16 зображено фрагмент гри «Початок гри».



Рисунок 3.16 – Фрагмент гри «Початок гри»

Після цього все керування персонажем надається гравцеві. Керування налаштовано кнопками «WASD». Крім того, керування у грі також працює якщо користуватись геймпадом серії Xbox, а не тільки клавіатурою та мишкою. Для керування персонажем на геймпаді потрібно використати лівий стік або хрестовину.

Також для того, щоб дізнатись що трапилось головна героїня може знайти на березі пляшку із запискою невідомо від кого. Для взаємодії з предметами потрібно натиснути кнопку «Е» або «Enter» на клавіатурі або «А» на геймпаді. На рисунку 3.17 зображено фрагмент гри «Взаємодію персонажа із знайденою запискою у пляшці».



Рисунок 3.17 – Фрагмент гри «Взаємодія персонажа із знайденою запискою у пляшці»

При дослідженні території головна героїня знаходить вхід до лісу, який охороняють дві мавпочки. При наближені до них запускається катсцена з діалогом, де мавпи попереджають головну героїню про небезпеку попереду. На рисунку 3.18 зображено фрагмент гри «Вхід у ліс».



Рисунок 3.18 – Фрагмент гри «Вхід у ліс»

Потім відбувається перехід між двома рівнями і персонаж потрапляє у ліс. На цьому рівні працює Action режим. На рисунку 3.19 представлено фрагмент гри «Початок рівня з лісом».

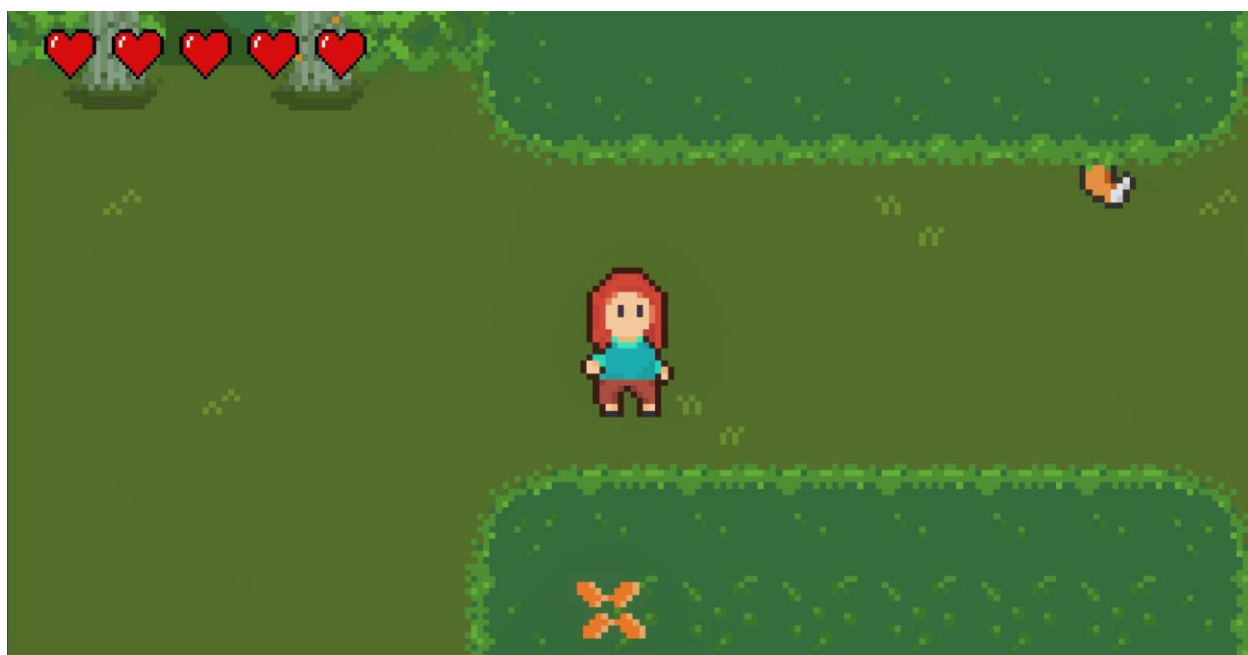


Рисунок 3.19 – Фрагмент гри «Початок рівня з лісом»

На інтерфейсі видно серця, які показують поточне здоров'я головного героя. На початку – це 5 сердечок. Через динамічне налаштування компоненту системи здоров'я та HUD системи, відображення сердець буде змінюватись в залежності скільки здоров'я має персонаж на даний момент. Також протягом подорожі лісом персонаж буде зустрічати хвіст лисиці, який з'являється з листя по бокам стін.

Також, на цьому рівні одразу після початкової катсцени персонаж знаходить меч, яким не може користуватись поки не візьме його. Керування мечем зроблено на кнопку «J». Для керування мечем на геймпаді потрібно натиснути кнопку «X». На рисунку 3.20 показано місце фрагмент гри «Розташування меча».



Рисунок 3.20 – Фрагмент гри «Місце розташування меча»

У кожній «кімнаті» лісу є тригер при проходженні через який з'являється різна кількість ворогів, а також у деяких «кімнатах» з'являється їжа, яка відновлює здоров'я на одне сердечко. Також при проходженні третьої «кімнати» з'являється додаткова зброя у вигляді лука. Керування цією зброєю відбувається за допомогою кнопок «K» та «Y» на клавіатурі та геймпаді відповідно. На рисунку 3.21 зображено ігровий процес у лісі.

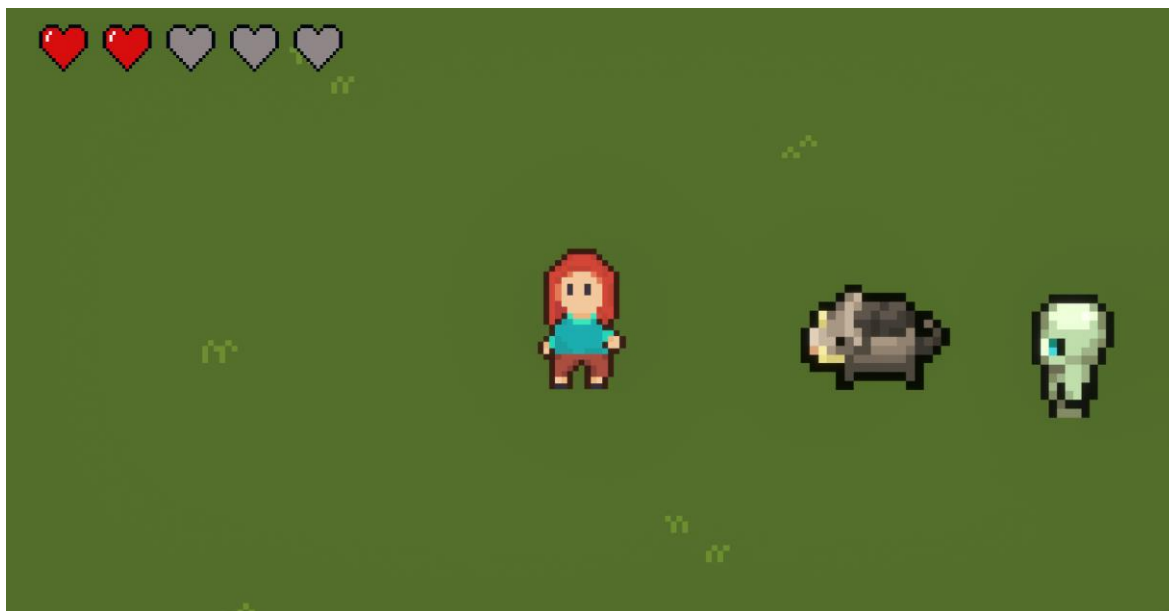


Рисунок 3.21 – Ігровий процес у лісі

У випадку, якщо вороги переможуть головну героїню, тоді вона помре. З'явиться анімація смерті персонажа та за декілька секунд гра перезапуститься на початок лісного рівня. На рисунку 3.22 зображено фрагмент гри «Смерть головної героїні від ворогів».

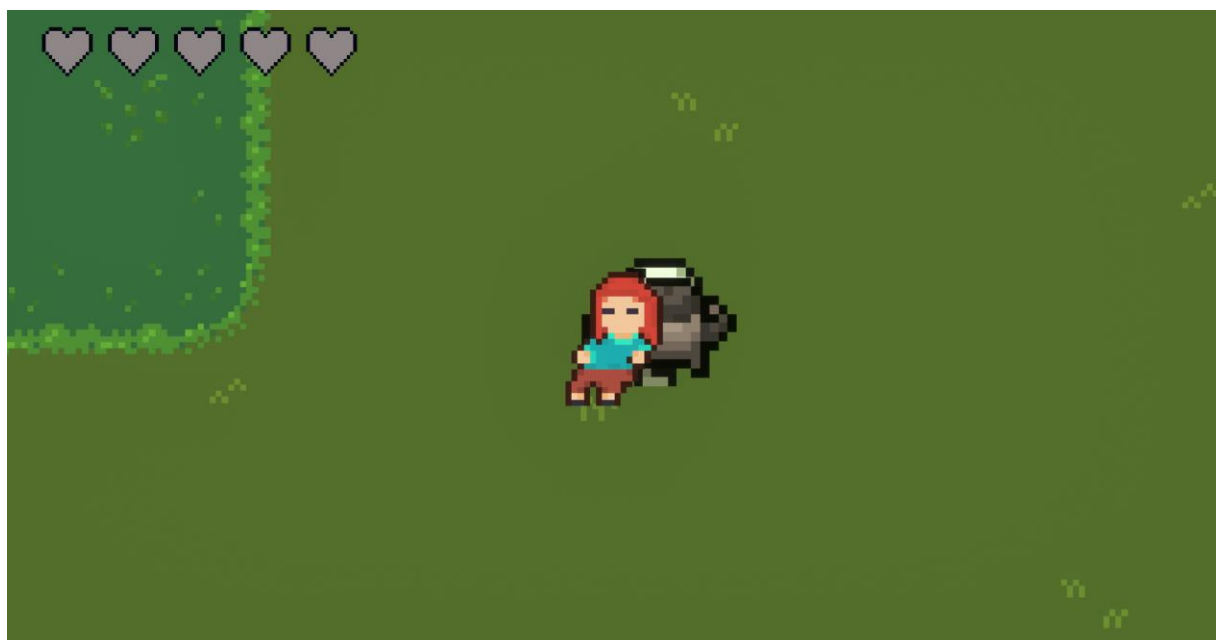


Рисунок 3.22 – Фрагмент гри «Смерть головної героїні від ворогів»

Після перемоги над всіма ворогами, посеред рівня з'являється риба, після підняття якої починається катсцена, де головна героїня зустрічає свого загубленого друга-лисицю. На рисунку 3.23 представлено фрагмент гри «Кінець лісного рівня».



Рисунок 3.23 – Фрагмент гри «Кінець лісного рівня»

Потім є перехід до окремого рівня, де всі жителі острова дякують головній героїні за те, що вона допомогла захистити ліс від монстрів. На рисунку 3.24 представлено фрагменти гри «Фінальна катсцена гри».



Рисунок 3.24 – Фрагмент гри «Фінальна катсцена гри»

Отже, розроблений програмний модуль авторської 2D TopDown гри успішно пройшов тестування та відповідає всім заданим вимогам.

Під час розробки було виконано такі задачі:

- Налаштовано середовище розробки ігровий рушій Unreal Engine 5 для роботи з 2D графікою;
- Розроблено та підготовлено графічну частину гри у програмному забезпеченні Aseprite. Графіка виконана у одному стилі: комфортна і релаксна;
- Розроблено два рівні гри з додатковою картою для фінальної катсцени;
- Розроблено дві ігрові механіки для гри, а саме RPG та Action режими;
- Розроблено інтерфейс для Action режиму гри;
- Здійснено тестування програмного модуля авторської 2D TopDown гри.

В результаті розробки було враховано, що гра має бути не важкою та доступною для слабких ПК. Також всі надлишкові налаштування Unreal Engine 5, які в більшості притаманні 3D іграм було змінено для оптимізації ігрового процесу. Реалізовано підтримку геймпадів, що дає можливість грати не тільки на клавіатурі з мишкою. Реалізація підтримки геймпадів дає також можливість підтримки спеціальних ігрових девайсів для людей з вадами кінцівок.

Порівнюючи цю інді-гру з іншими інді-іграми: 3D українським інді-проектом Titan Chaser та українською інді-грою Меліній встановлено, що Titan Chaser – це 3D гра у жанрі сюрреалістичний симулятор водіння та ходьби, де вам пропонується відганяти блукаючих по штату титанів прожектором, без агресії та насильства [29]. Ігровий процес цієї гри надто спокійний, навіть нудний. Цій грі бракує різноманіття.

Меліній це сюрреалістична, приправлена жорстокістю та гумором, візуальна новела з елементами RPG [30]. Серед розробки інді-ігор популярно розробляти сюрреалістичні ігри або ігри у жанрі хорор. Такий підхід не зовсім прийнятний до затишних ігор.

У таблиці 3.1 подано порівняння результатів тестування розробленого програмного продукту та аналогів.

Таблиця 3.1 – Результати тестування розробленого програмного продукту та аналогів

	Розроблена гра	Titan Chaser	Меліній
Поєднання двох механік	+	—	—
Безкоштовна гра	+	—	+
«Затишна гра»	+	+	—
Відсутність багів	+	—	—
Оптимізація для слабких ПК	+	—	+
Підтримка геймпаду	+	+	—
Розроблено на Unreal Engine 5	+	—	—

Отже, проаналізувавши та протестувавши програмний продукт, можна дійти висновку, що поставлених цілей було досягнуто.

З таблиці 3.1 можна побачити, що функціонал програмного модуля авторської 2D TopDown гри було покращено у порівнянні з аналогами. Розробка має покращений функціонал у порівнянні з системами-аналогами щонайменше на 4 можливості, а саме: поєднання двох механік у грі, виконано у жанрі «затишна гра», відсутність багів, підтримка геймпаду. Розширення функціоналу забезпечить покращення ігрового процесу гравцями, що і є метою дослідження.

3.7 Висновок

У цьому розділі було проаналізовано існуючі ігрові рушії для розробки ігор та програми для створення графічної частини гри. Було оглянуто та проаналізовано такі ігрові рушії: Unreal Engine 5, Unity, Godot та такі графічні програми: Aseprite, Adobe Photoshop, Gimp, Pixieditor, Pyxeledit, Graphicgale Krita. Розглянуто переваги та недоліки такого підходу, визначено особливості розробки та доцільність використання.

У результаті аналізу було обрано ігровий рушій Unreal Engine 5 у поєднанні з безкоштовним плагіном PaperZD та використанням системи візуальних сценаріїв Blueprints, тому що Unreal Engine 5 один з найсучасніших і доступних ігрових рушіїв. Unreal Engine 5 – це один з провідних ігрових рушіїв який прийнятний для розробки тривимірних та двовимірних ігор різної складності. Також цей ігровий рушій зручно використовувати для розробки ігор без знань програмування.

Для графічної частини обрано Aseprite, тому що це програмне забезпечення має не складний інтерфейс та всі потрібні функції для розробки анімацій, тайлесетів, можливість збереження вихідних файлів у форматі json, що зручно для імпорту файлів у Unreal Engine 5.

Також було розроблено механіки для двох режимів гри Action та RPG. Для Action режиму розроблена система боїв, налаштовано зміни анімацій в залежності від потрібної ситуації. Для режиму RPG розроблено систему діалогів та взаємодії з іншими персонажами та предметами.

Розроблено компонент системи здоров'я, який пов'язаний з Action режимом. Цей компонент відповідає за зміну здоров'я у головної героїні та її ворогів.

Було проведено тестування програмного модуля авторської 2D TopDown гри. У результаті зроблено висновок, що всі основні механіки та функції було реалізовано. Розроблений програмний продукт має покращений функціонал порівняно з аналогами на 4 можливості, а саме: поєднання двох механік у грі, виконано у жанрі «затишна гра», відсутність багів, підтримка геймпаду.

ВИСНОВКИ

Всі задачі, поставлені для реалізації програмного модуля авторської 2D TopDown гри були виконані у повному обсязі, а саме: проведено аналіз існуючих ігор у жанрах 2D та TopDown, методів їх реалізації та обґрунтована доцільність розробки; спроектовано програмний модуль авторської 2D TopDown гри; програмно реалізовано модуль авторської 2D TopDown гри на ігровому рушії Unreal Engine 5; виконано тестування програми та проведено аналіз отриманих результатів; розроблено інструкцію користувача.

Для розробки був використаний ігровий рушій Unreal Engine 5 та плагін PaperZD разом із використанням системи візуальних сценаріїв Blueprints від UE5. Базовий Unreal Engine 5 обмежений для розробки двовимірних ігор, тому для покращення та прискорення розробки було використано поєднання різних технік та додаткових плагінів.

Для створення графічної частини було обрано програмне забезпечення Aseprite. Це програмне забезпечення підтримує збереження спрайтів у форматі json, що значно прискорює імпорт файлів до Unreal Engine 5. Також Aseprite нескладне програмне забезпечення, яке має багато зручних функцій для створення спрайтів, тайлсетів та анімацій.

Розроблений програмний продукт успішно пройшов тестування. Робота виконана згідно вимог та методичних рекомендацій щодо підготовки бакалаврських кваліфікаційних робіт [31].

Мета роботи, яка полягала в розширенні функціональних можливостей програмного модуля авторської 2D TopDown гри була досягнута шляхом покращення ігрового процесу 2D TopDown гри. Порівняно з аналогами, розробка має покращений функціонал щонайменше на 4 можливості, а саме: поєднання двох механік у грі – Action та RPG; виконано у жанрі «затишна гра»; відсутність багів; підтримка геймпаду. Отже, розроблений програмний модуль авторської 2D TopDown гри відповідає заданим вимогам.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Єрмолаєва М. А., Крилик Л. В. Особливості розробки програмного модуля авторської 2D TopDown гри на ігровому рушії Unreal Engine 5. *LIV Всеукраїнська науково-технічна конференція факультету інтелектуальних інформаційних технологій та автоматизації*. URL: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2025/paper/view/23096> (дата звернення: 07.05.2025).
2. Крилик Л. В., Єрмолаєва М. А. Особливості розробки авторської 2D гри з використанням ігрового рушія UNREAL ENGINE 5. Вісник Хмельницького національного університету. Серія «Технічні науки». 2025. №3(351). Т.1. С. 230–236.
3. Назарій Семенюк. 100 найкращих ігор всіх часів за версією GQ. URL: <https://gameverse.com.ua/newsarticle/78-gq-top-100-games-of-all-time/> (дата звернення: 07.05.2025).
4. Video Game Genres: Everything You Need to Know. URL: www.hp.com/us-en/shop/tech-takes/video-game-genres (дата звернення: 07.05.2025).
5. Resident Evil: Defining the Survival Horror Genre. URL: <https://capcomasia.com.hk/resident-evil-defining-the-survival-horror-genre/> (дата звернення: 07.05.2025).
6. 'It Takes Two' tests your ability to save a marriage and fly a fidget spinner. URL: <https://www.washingtonpost.com/arts-entertainment/2021/04/02/it-takes-two-game-review/> (дата звернення: 07.05.2025).
7. We Were Here Forever Review. URL: <https://www.ign.com/articles/we-were-here-forever-review> (дата звернення: 07.05.2025).
8. Minecraft. URL: <https://www.minecraft.net/en-us/article/what-minecraft> (дата звернення: 07.05.2025).
9. The History of Minecraft. URL: <https://www.redbull.com/se-en/history-of-minecraft> (дата звернення: 07.05.2025).

10. Chris Plante. Stardew Valley, Harvest Moon, and how wholesome games taught me a terrible lesson. URL: <https://www.theverge.com/2016/3/8/11179044/stardew-valley-harvest-moon-video-games-life-guide> (дата звернення: 07.05.2025).
11. Tubagus Zufri, Dodi Hilman, Octavianus Frans. Research on the Application of Pixel Art in Game Character Design. Visual Communication Design School of Design Bina Nusantara Jakarta, Indonesia, 2002. 31с.
12. Ross Bramble. The Ultimate Guide To 2D Game Genres. URL: <https://gamemaker.io/en/blog/2d-game-genres> (дата звернення: 07.05.2025).
13. Unreal Engine: Features. URL: <https://www.unrealengine.com/en-US/features> (дата звернення: 07.05.2025).
14. Jesse Schell. The art of game design. USA: MK, 2008. 518 с.
15. Blueprints Visual Scripting. URL: <https://dev.epicgames.com/documentation/en-us/unreal-engine/blueprints-visual-scripting-in-unreal-engine> (дата звернення: 07.05.2025).
16. Aseprite. URL: <https://www.aseprite.org> (дата звернення: 07.05.2025).
17. Itch.io. URL: <https://itch.io> (дата звернення: 07.05.2025).
18. Alexander Brazie. Video Game Mechanics: A Beginner's Guide (with Examples). URL: <https://gamedesignskills.com/game-design/video-game-mechanics/> (дата звернення: 07.05.2025).
19. Ion Corarcea, Prashant Dwivedi. How can you use OOP to create video games? URL: <https://www.linkedin.com/advice/1/how-can-you-use-oop-create-video-games-skills-programming-ntv1e> (дата звернення: 07.05.2025).
20. Alexander S. Gillis. What is an algorithm? URL: <https://www.techtarget.com/whatis/definition/algorithm> (дата звернення: 07.05.2025).
21. Algorithm flowchart example. URL: <https://www.lucidchart.com/pages/templates/algorithm-flowchart-example> (дата звернення: 07.05.2025).

22. Setting Up a Game Mode. URL: <https://dev.epicgames.com/documentation/en-us/unreal-engine/setting-up-a-game-mode-in-unreal-engine> (дата звернення: 07.05.2025).

23. Interfaces. URL: https://dev.epicgames.com/documentation/en-us/unreal-engine/interfaces?application_version=4.27 (дата звернення: 07.05.2025).

24. Events. URL: <https://dev.epicgames.com/documentation/en-us/unreal-engine/events-in-unreal-engine> (дата звернення: 07.05.2025).

25. Best 2D Settings for Unreal Engine 5. URL: <https://cobracode.notion.site/Best-2D-Settings-for-Unreal-Engine-5-0895133a56924f1ca7f2149cf111983b> (дата звернення: 07.05.2025).

26. State Machines. URL: <https://dev.epicgames.com/documentation/en-us/unreal-engine/state-machines-in-unreal-engine> (дата звернення: 07.05.2025).

27. Cinematics and Sequencer. URL: https://dev.epicgames.com/documentation/en-us/unreal-engine/cinematics-and-movie-making-in-unreal-engine?application_version=5.4 (дата звернення: 07.05.2025).

28. Components. URL: <https://dev.epicgames.com/documentation/en-us/unreal-engine/components-in-unreal-engine> (дата звернення: 07.05.2025).

29. Titan Chaser. URL: https://store.steampowered.com/app/1290170/Titan_Chaser/ (дата звернення: 07.05.2025).

30. Меліній. URL: <https://store.steampowered.com/app/2726320/Melinij/> (дата звернення: 07.05.2025).

31. Методичні вказівки до виконання бакалаврських кваліфікаційних робіт для студентів спеціальності 122 «Комп'ютерні науки» [Електронний ресурс] / уклад.: А. А. Яровий, О. В. Сілагін, І. В. Богач. Вінниця : ВНТУ, 2023. (PDF, 54 с.).

ДОДАТКИ

ДОДАТОК А (ОБОВ'ЯЗКОВИЙ)

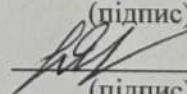
ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Назва роботи: Розробка програмного модуля авторської 2D TopDown гриТип роботи: бакалаврська кваліфікаційна робота
(бакалаврська кваліфікаційна робота / магістерська кваліфікаційна робота)Підрозділ кафедра комп'ютерних наук
(кафедра, факультет, навчальна група)Коефіцієнт подібності текстових запозичень, виявлених у роботі
системою StrikePlagiarism 2,00 %

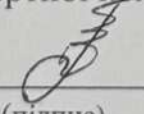
Висновок щодо перевірки кваліфікаційної роботи (відмітити потрібне)

☒ Запозичення, виявлені у роботі, є законними і не містять ознак плагіату, фабрикації, фальсифікації. Роботу прийняти до захисту☐ У роботі не виявлено ознак плагіату, фабрикації, фальсифікації, але надмірна кількість текстових запозичень та/або наявність типових розрахунків не дозволяють прийняти рішення про оригінальність та самостійність її виконання. Роботу направити на доопрацювання.☐ У роботі виявлено ознаки плагіату та/або текстових маніпуляцій як спроб укриття плагіату, фабрикації, фальсифікації, що суперечить вимогам законодавства та нормам академічної доброчесності. Робота до захисту не приймається.

Експертна комісія:

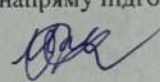
Яровий А. А., зав. каф. КН
(прізвище, ініціали, посада)Колесницький О. К., проф. каф КН
(прізвище, ініціали, посада)
(підпис)
(підпис)Особа, відповідальна за перевірку 
(підпис)Озеранський В. С.
(прізвище, ініціали)

З висновком експертної комісії ознайомлений(-на)

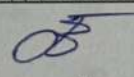
Керівник 
(підпис)Крилик Л. В.
(прізвище, ініціали, посада)Здобувач 
(підпис)Ермолаєва М. А.
(прізвище, ініціали)

ДОДАТОК Б (ОБОВ'ЯЗКОВИЙ)**ГРАФІЧНА ЧАСТИНА****«РОЗРОБКА ПРОГРАМНОГО МОДУЛЯ АВТОРСЬКОЇ
2D TOPDOWN ГРИ»**

Виконала: студентка 4 курсу, групи 4КН-216
спеціальності 122 – Комп'ютерні науки
(шифр і назва напрямку підготовки, спеціальності)

 Єрмолаєва М.А.
(прізвище та ініціали)

Керівник: к.т.н., доцент каф. КН

 Крилик Л. В.
(прізвище та ініціали)

«03» серпня 2025 р.



Рисунок Б.1 – Игровой процесс The Legend of Zelda: Breath of the Wild



Рисунок Б.2 – Игровой процесс Stardew Valley



Рисунок Б.3 – Приклад використання простої піксельної графіки у грі Undertale



Рисунок Б.4 – Приклад використання більш деталізованої піксельної графіки у грі The Siege and the Sandfox



Рисунок Б.5 – Приклад гри з боковим видом і піксельною графікою

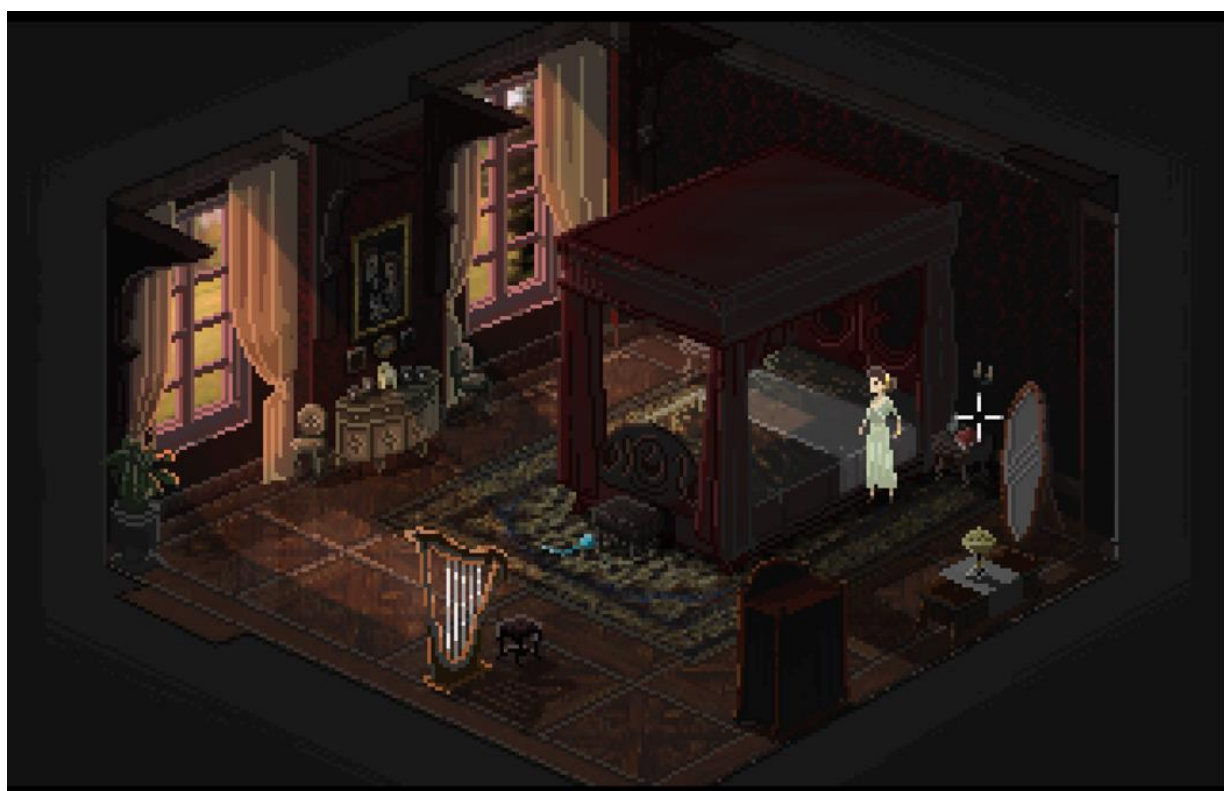


Рисунок Б.6 – Приклад гри з ізометричною піксельною графікою

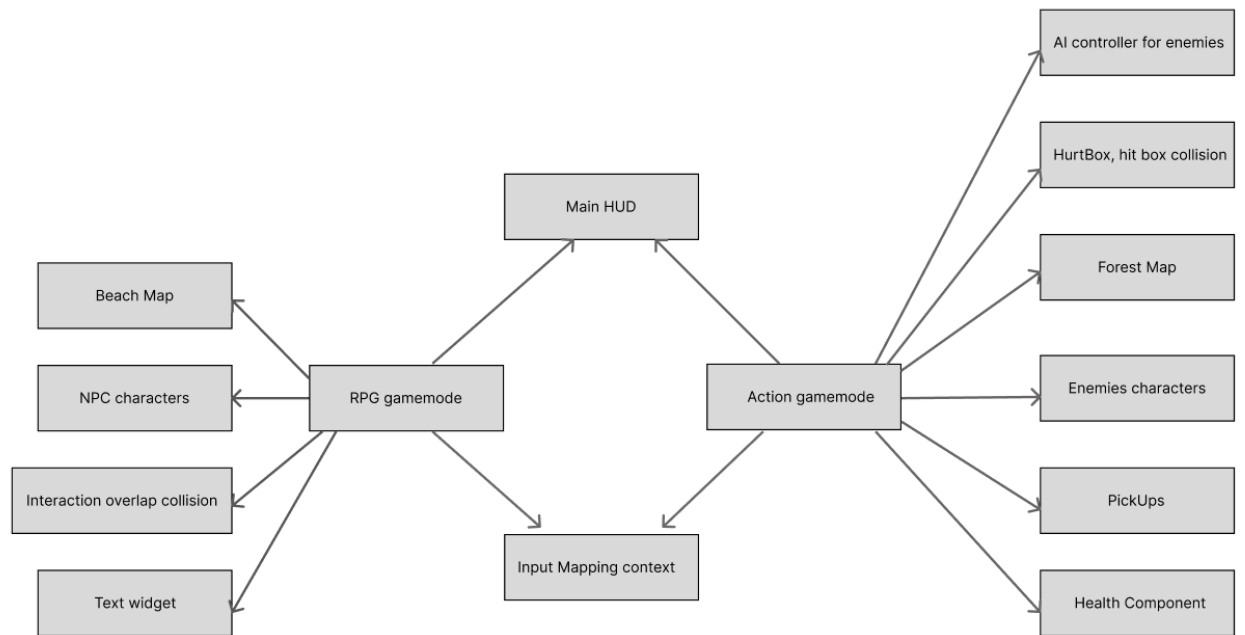


Рисунок Б.7 – Roadmap авторської 2D TopDown гри з 2 ігровими режимами

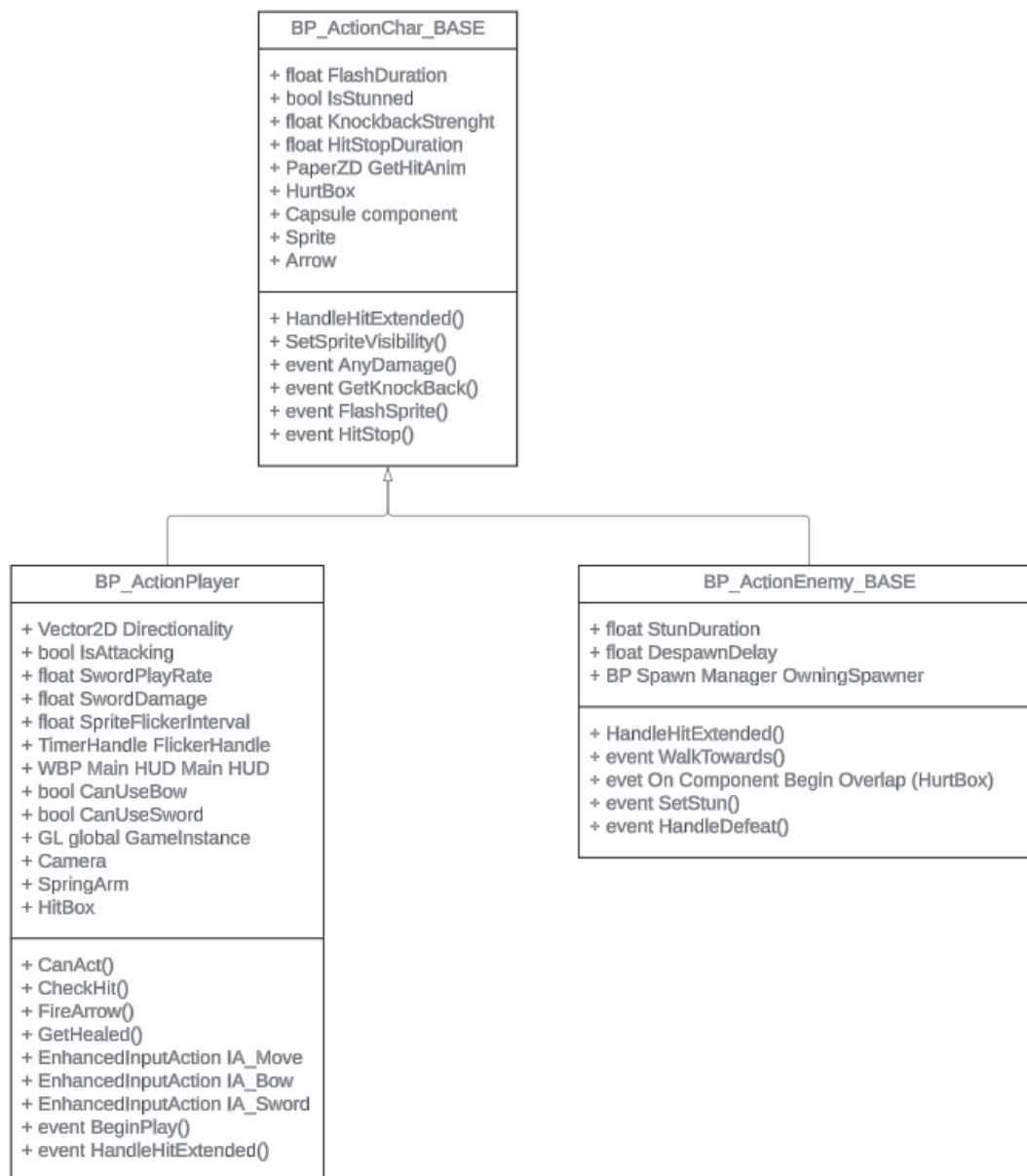


Рисунок Б.8 – UML-діаграма класів для персонажів Action gamemode

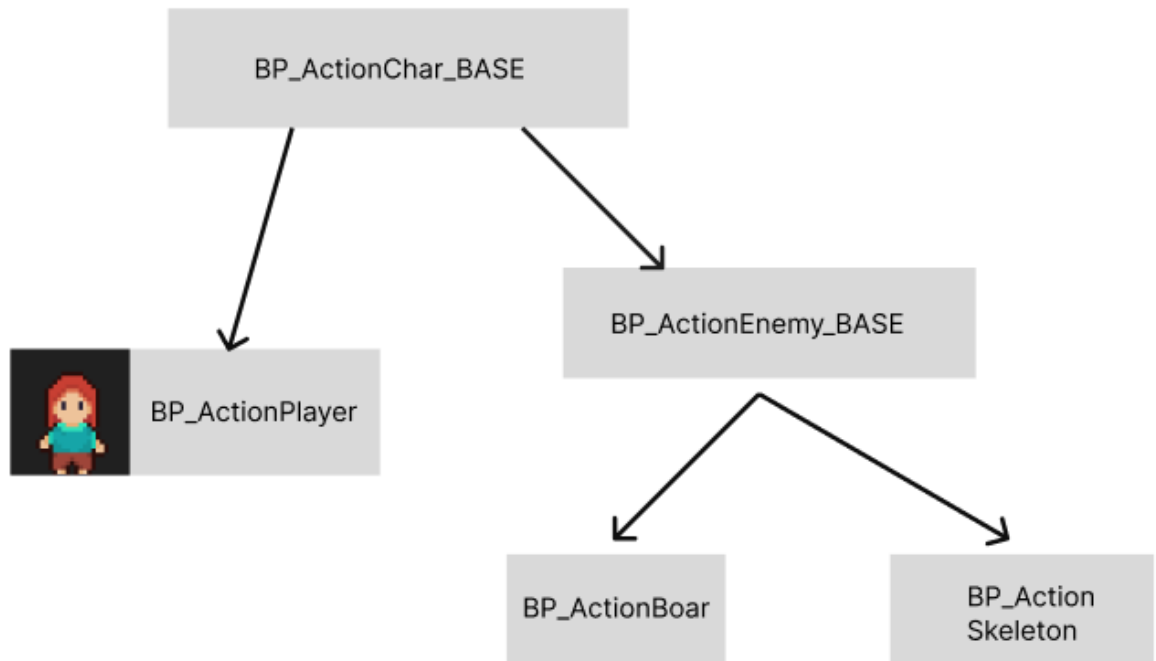


Рисунок Б.9 – Ієрархія класів персонажів для Action режиму

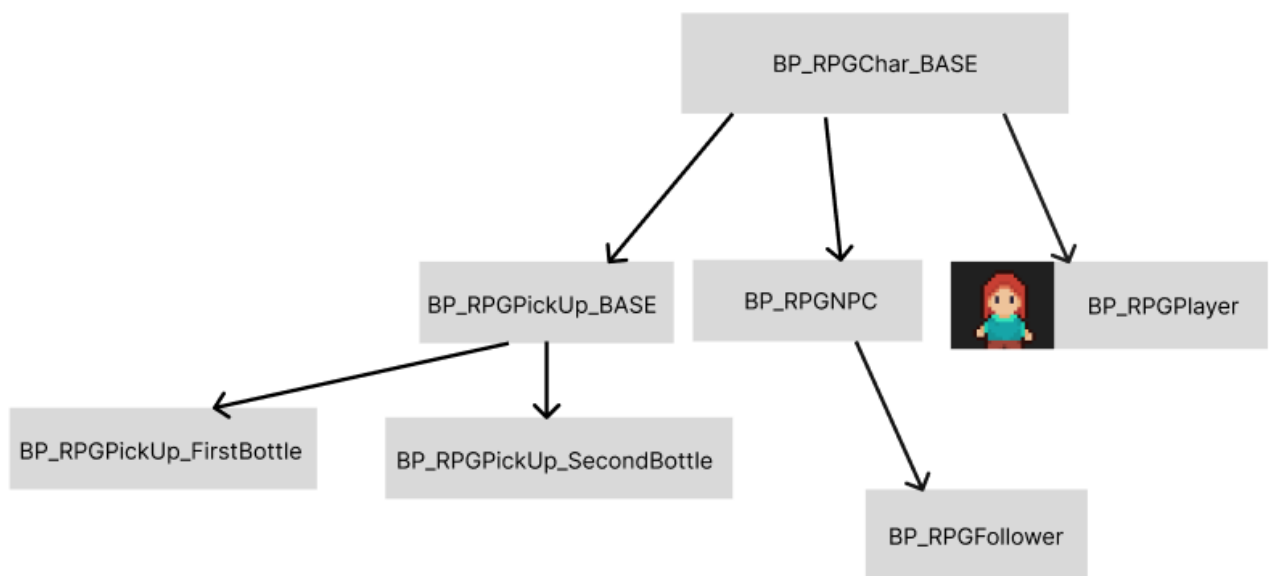


Рисунок Б.10 – Ієрархія класів персонажів та об'єктів для RPG режиму

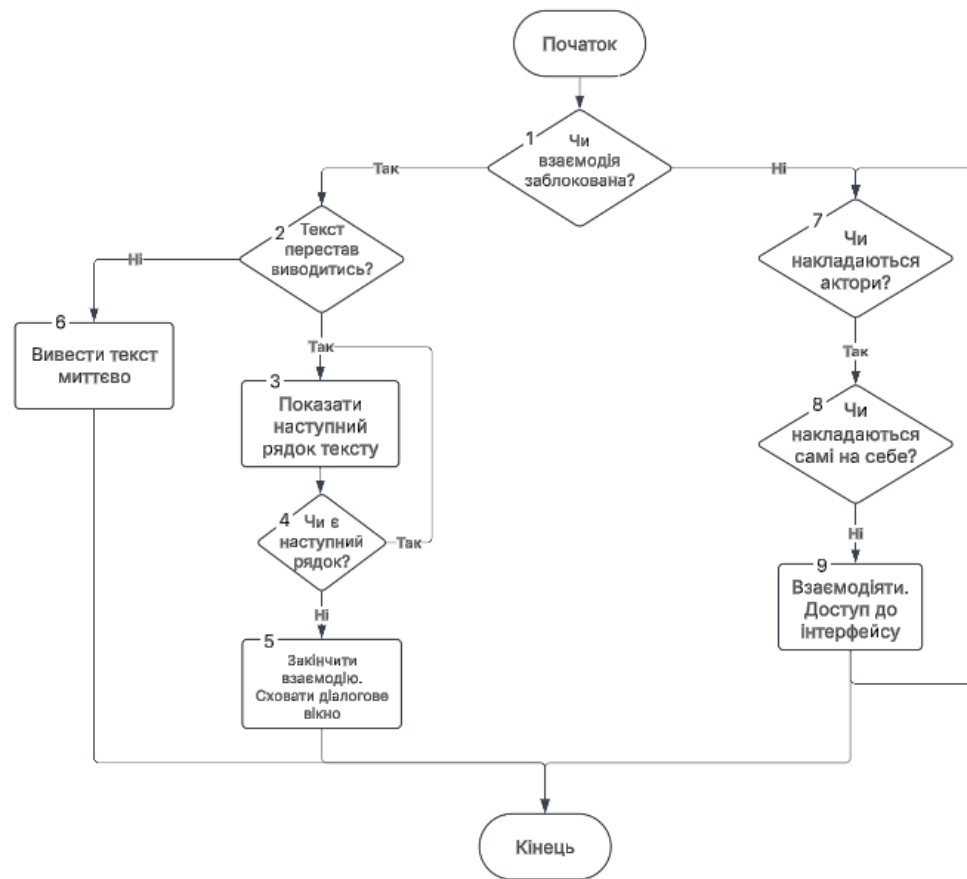


Рисунок Б.11 – Схема алгоритму події взаємодії головного персонажа з іншими персонажами чи об'єктами



Рисунок Б.12 – Фрагмент гри «Початок гри»



Рисунок Б.13 – Фрагмент гри «Місце розташування меча»

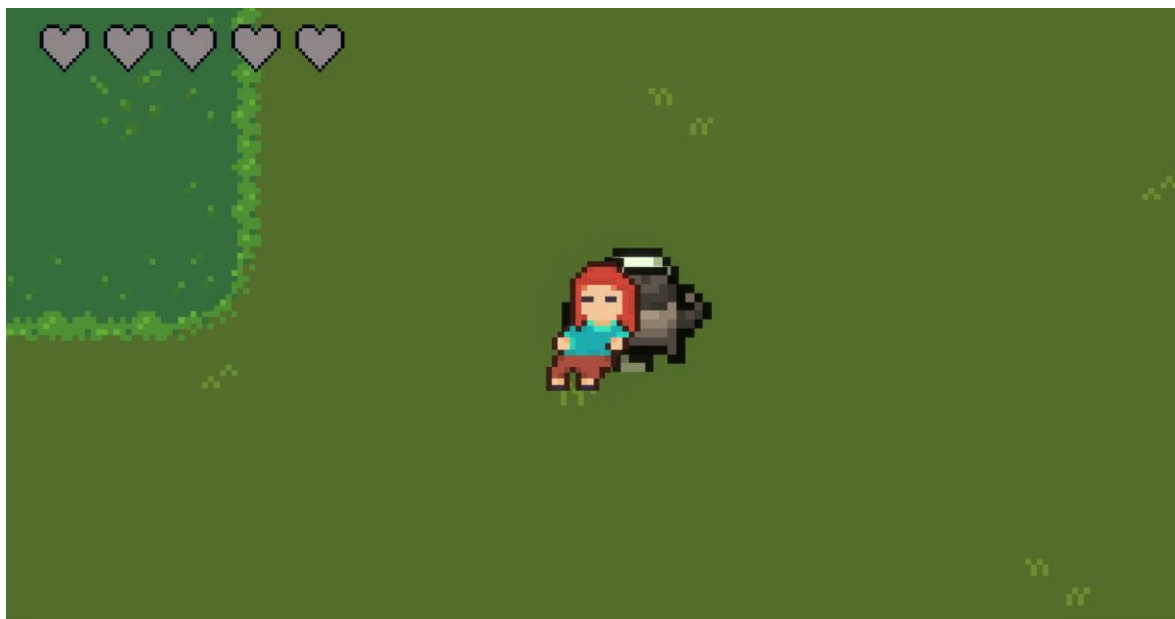


Рисунок Б.14 – Фрагмент гри «Смерть головної героїні від ворогів»



Рисунок Б.15 – Фрагмент гри «Кінець лісного рівня»



Рисунок Б.16 – Фрагмент гри «Фінальна катсцена гри»

ДОДАТОК В (довідниковий)

Інструкція користувача

Щоб відкрити гру потрібно відкрити файл `diploma_v1.exe` у теці файлів гри (рис. В.1).

Ім'я	Дата змінення	Тип	Розмір
diploma_v1	08.05.2025 14:03	Папка файлів	
Engine	08.05.2025 14:03	Папка файлів	
diploma_v1.exe	08.05.2025 11:01	Застосунок	150 КБ
Manifest_NonUFSFiles_Win64.txt	08.05.2025 11:01	Текстовий докум...	3 КБ
Manifest_UFSFiles_Win64.txt	08.05.2025 11:01	Текстовий докум...	371 КБ

Рисунок В.1 – Тека файла гри

Після запуску одразу запускається гра. Керування персонажем можливе за допомогою клавіш WASD, взаємодія з предметами та персонажами на клавішу Е, удар мечем клавішу J, користування луком на клавішу К (рис. В.2).



Рисунок В.2 – Взаємодія з предметами

Щоб перейти на наступну локацію потрібно підійти до входу в ліс (рис. В.3).



Рисунок В.3 – Вхід в ліс

На наступному рівні потрібно знешкодити всіх ворогів, щоб отримати нагороду та повернутись назад на пляж (рис. В.4).

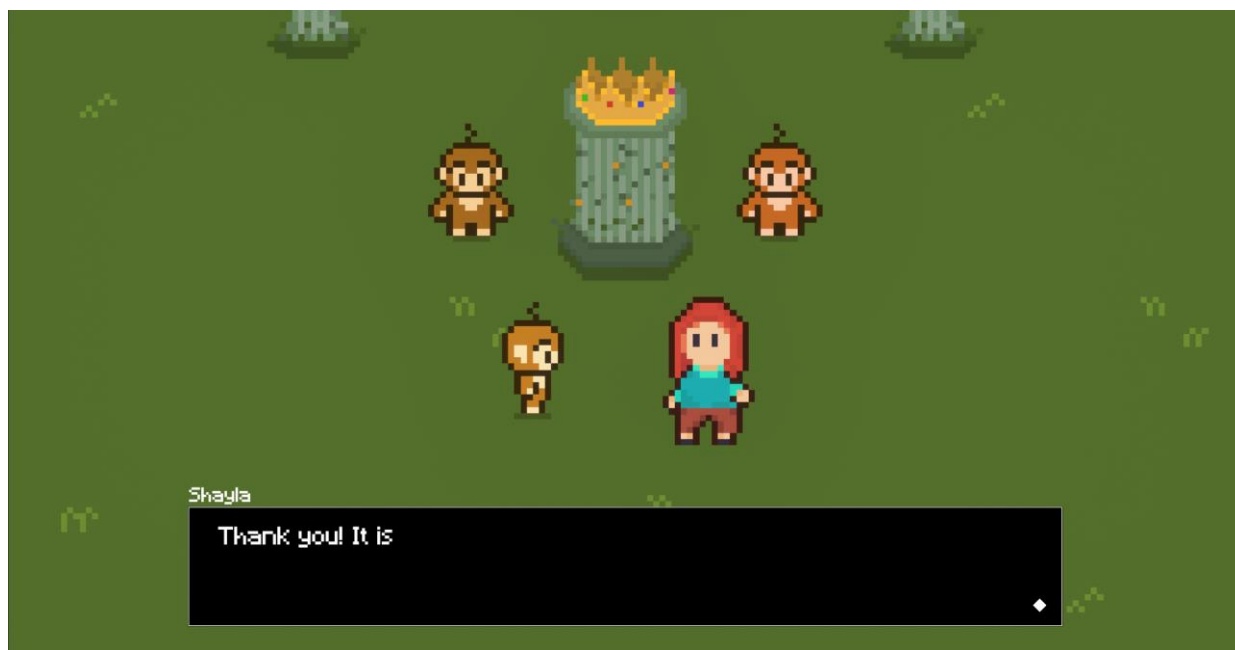


Рисунок В.4 – Сцена нагороди за вбивство всіх ворогів