

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра інженерії програмного забезпечення

ДОПУЩЕНО ДО ЗАХИСТУ
Завідувач кафедри інженерії
програмного забезпечення
_____ Євген ДАВИДЕНКО
«20» червня 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА
КОМП'ЮТЕРНА ГРА В ЖАНРІ ВІЗУАЛЬНОЇ НОВЕЛИ
Спеціальність 121 Інженерія програмного забезпечення
Освітня програма «Інженерія програмного забезпечення»

Здобувач

Євгеній СКИВКА

«20» червня 2025 р.

Керівник роботи

канд. пед. наук,
доцентка

Катерина КІРЕЙ

«20» червня 2025 р.

Завдання на виконання кваліфікаційної роботи
Чорноморський національний університет імені Петра Могили

Факультет	Комп'ютерних наук
Кафедра	Інженерії програмного забезпечення
Рівень вищої освіти	Перший (бакалаврський)
Освітній ступінь	Бакалавр
Спеціальність	121 Інженерія програмного забезпечення
Освітня програма	Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри інженерії
програмного забезпечення

_____ Євген ДАВИДЕНКО

«14» листопада 2024 р.

ЗАВДАННЯ
на кваліфікаційну бакалаврську роботу здобувача
Скивки Євгенія

1. Тема кваліфікаційної роботи «Комп'ютерна гра в жанрі візуальної новели» затверджена наказом ректора ЧНУ ім. Петра Могили № __315__ від «13» листопада 2024 р.
2. Строк представлення кваліфікаційної роботи «18» червня 2025 р.
3. Очікуваний результат роботи та початкові дані якщо такі потрібні.
Розроблена комп'ютерна гра в жанрі візуальної новели
4. Перелік питань, що підлягають розробці:
 - провести аналіз жанру візуальної новели та інструментів для її створення;

- розробити сюжет, гілки вибору та логіку взаємодії користувача з грою;
- створити прототип гри на рушії Ren'Py з використанням графіки, фонової музики та анімацій;
- провести тестування гри;
- виявити та усунути помилки;

5. Перелік графічних матеріалів:

Слайди презентації

6. Консультанти:

Консультант	Кафедра (організація)	Частина роботи

Дата видачі завдання «13» листопада 2024 р.

КАЛЕНДАРНИЙ ПЛАН
виконання кваліфікаційної практики

Тема: **Комп'ютерна гра в жанрі візуальної новели**

№	Найменування роботи	Початок	Закінчення	Примітки
1.	Розробка та затвердження завдання на виконання КБР	24.10.2024	13.11.2024	Виконано
2.	Огляд літератури за темою роботи	14.11.2024	21.11.2024	Виконано
3.	Складання календарного плану КБР	22.11.2024	25.11.2024	Виконано
4.	Аналіз предметної області	26.11.2024	29.11.2024	Виконано
5.	Розробка проєктних рішень	02.12.2024	06.12.2024	Виконано
6.	Моделювання та конструювання ПЗ	09.12.2024	12.12.2024	Виконано
7.	Розробка структури гри та її сюжетних гілок	13.12.2024	30.12.2024	Виконано
8.	Розробка механізмів вибору та впливу на розвиток сюжету	31.12.2024	09.01.2025	Виконано
9.	Розробка макетів та прототипів користувацького інтерфейсу гри	10.01.2025	16.01.2025	Виконано
10.	Створення перших локацій та реалізація основного сюжету	17.01.2025	17.02.2025	Виконано
11.	Розробка додаткових сюжетних ліній і персонажів	18.02.2025	24.02.2025	Виконано
12.	Інтеграція музики та звукових ефектів	25.02.2025	06.03.2025	Виконано
13.	Розробка артів та спрайтів для персонажів та локацій	07.03.2025	20.03.2025	Виконано
14.	Відгук керівника КБР	20.06.2025	24.05.2025	Виконано
15.	Оформлення КБР та презентації	21.04.2025	17.05.2025	Виконано
16.	Попередній захист	25.05.2025	25.05.2025	Виконано
17.	Рецензування	17.06.2025	18.06.2025	Виконано
18.	Завершення оформлення КБР та презентації	18.06.2025	19.06.2025	Виконано
19.	Захист кваліфікаційної роботи	24.06.2025	24.06.2025	Виконано

Здобувач _____

Євгеній СКИВКА

«25» листопада 2024р.

Керівник роботи

канд. пед. наук,
доцентка

Катерина КІРЕЙ

«25» листопада 2024р.

АНОТАЦІЯ

до кваліфікаційної роботи

«Комп'ютерна гра в жанрі візуальної новели»

Здобувач 408 гр.: Сківка Євгеній

Керівник: канд. пед. наук, доцентка Катерина Кірей

У сучасному цифровому середовищі візуальні новели набувають дедалі більшої популярності завдяки своїй здатності поєднувати інтерактивність, глибокий сюжет та художнє оформлення. Багато наявних рішень мають обмежену гнучкість у сценарному дизайні або технічну складність у реалізації. Ця робота присвячена розробці інтуїтивно та наративно насиченої гри, що поєднує інтерактивний сюжет з художніми і технічними засобами реалізації.

Кваліфікаційна бакалаврська робота присвячена розробці комп'ютерної гри в жанрі візуальної новели, реалізованої на рушії Ren'Py з використанням мови програмування Python. У грі реалізовано нелінійний сюжет із фіксованим фіналом, що створює ілюзію вибору та глибоке занурення у наратив.

Метою роботи є розробка підходів щодо створення та оптимізації комп'ютерної гри в жанрі візуальної новели з використанням інтерактивних інтерфейсів, персоналізації користувацького досвіду та системи обміну повідомленнями в реальному часі.

Для досягнення зазначеної мети було виконано такі завдання:

- провів аналіз сучасних технологій, що застосовуються для створення інтерактивних та персоналізованих інтерфейсів у візуальних новелах;
- розробив алгоритми формування адаптивних інтерфейсів на основі виборів користувачів;
- реалізував систему обміну повідомленнями в реальному часі для моделювання діалогів між персонажами;
- створив сценарії з інтерактивними виборами, що впливають на хід гри, але не змінюють її фінальний результат;

– створено та адаптовано музичний супровід, що відповідає настрою та сценам гри;

– розроблено та стилізовано спрайти персонажів для відображення емоцій та взаємодії у сюжеті;

Об'єктом роботи є комп'ютерні ігри в жанрі візуальної новели.

Предметом кваліфікаційної роботи є розробка комп'ютерної гри в жанрі візуальної новели з використанням системи обміну повідомленнями в реальному часі, інтерактивних інтерфейсів та алгоритмів персоналізації.

У вступі обґрунтовано актуальність теми, мету, об'єкт і предмет дослідження.

Перший розділ містить аналіз існуючих рішень у сфері візуальних новел та огляд рушіїв для створення ігор. У другому розділі розглянуто загальну архітектуру гри, створено діаграми класів і розгортання, а також описано внутрішню структуру проєкту. Третій розділ присвячений розробці ключових компонентів гри: сценаріїв, інтерфейсу користувача, логіки вибору та роботи з файлами, створено діаграми класів, розгортання, послідовності, варіантів використання. У четвертому розділі наведено результати тестування, оптимізацію продуктивності та оцінку

КРБ викладена на 64 сторінки без додатків, вона містить 4 розділи, 22 ілюстрації, 3 таблиці, 21 джерело в переліку посилань.

Ключові слова: візуальна новела, комп'ютерна гра, Ren'Py, Python, сюжет, інтерфейс, інтерактивність, нелінійний сюжет, система вибору, тестування

ABSTRACT

to the qualifying bachelor`s thesis

«Computer game in the visual novel genre»

Student of 408 group: Skyvka Yevhenii

Supervisor: Candidate of Pedagogical Sciences, Associate Professor Kateryna Kirei

In today`s digital environment, visual novels are gaining increasing popularity due to their ability to combine interactivity, deep storytelling, and visual design. Most existing solutions either lack flexibility in scenario planning or are technically complex to implement. This thesis aims to create an intuitive and narratively rich game that combines interactive storytelling with visual and technical realization.

The bachelor's qualification thesis is dedicated to the development of a computer game in the visual novel genre, implemented using the Ren'Py engine and Python programming language. The game features a non-linear narrative with a fixed ending, creating an illusion of choice and deeper immersion in the story.

The purpose of the work is to develop approaches to creating and optimizing a computer game in the genre of a visual novel using interactive interfaces, personalization of user experience and a real-time messaging system.

To achieve this goal, the following tasks were performed:

- to develop algorithms for forming adaptive interfaces based on user choices;
- has developed algorithms for creating adaptive interfaces based on user choices;
- to implement a real-time messaging system for modeling dialogues between characters;
- has created scenarios with interactive elections that affect the course of the game but do not change its final outcome;
- created and adapted music that matches the mood and scenes of the game;
- developed and stylised character sprites to reflect emotions and interaction in the story;

The object of work is computer games in the genre of visual novel.

The subject of the qualification work is the development of a computer game in the genre of a visual novel using a real-time messaging system, interactive interfaces and personalization algorithms.

The introduction highlights the relevance of the topic and defines the aim, object, and subject of the research.

The first chapter analyzes existing solutions in the field of visual novels and compares game engines. The second chapter discusses the overall game architecture, presents a class diagram and a deployment diagram, and describes the internal structure of the project. The third chapter is devoted to the development of core components: story scripts, interface, choice logic, and file processing. The fourth chapter presents the testing results, performance optimization, and user experience evaluation. The conclusions summarize the completed work and outline the prospects for further project development. The thesis is presented on 64 pages, it contains 4 chapters, 22 illustrations, 3 tables, and 21 sources in the list of references.

Keywords: visual novel, computer game, Ren'Py, Python, storyline, interface, interactivity, nonlinear plot, choice system, testing.

ЗМІСТ

ВСТУП.....	4
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА АНАЛОГІЧНИХ ВЕБЗАСТОСУНКІВ...	6
1.1 Аналіз та характеристика предметної області	6
1.2 Огляд аналогічних існуючих застосунків	8
1.3 Поставлені завдання	14
Висновки до розділу 1	16
2 МОДЕЛЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	18
2.1 Вибір технологій та програмних засобів	18
2.2 Структура та логіка розроблюваного застосунку.....	21
2.3 Робота з графікою та музикою.....	22
2.4 Специфікація вимог	23
2.5 Опис інтерфейсу користувача.....	25
Висновки до розділу 2	27
3 ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	29
3.1 Створення діаграми варіантів використання	29
3.2 Створення діаграми послідовності.....	32
3.3 Створення діаграми класів.....	34
3.4 Створення діаграми станів	37
3.5 Створення діаграми компонентів	39
Висновки до розділу 3	42
4 ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАСТОСУНКУ	43
4.1 Робота з графічними матеріалами	43
4.1.1 Створення спрайтів персонажів	43
4.1.2 Створення фонів.....	46
4.2 Робота з музикою та звуком.....	48
4.3 Реалізація меню гри	50
4.3.1 Головне меню	50
4.3.2 Ігрове меню.....	53

4.4 Налаштування інтерфейсу гри.....	56
4.4.1 Основні параметри елементів керування	57
4.1.1 Параметри розташування інтерфейсних елементів.....	58
4.1.1 Визначення варіанту інтерфейсу "small"	60
Висновки до розділу 4	61
ВИСНОВКИ.....	62
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	63

ВСТУП

Актуальність. У сучасному світі з розвитком технологій та інтернет-платформ зростає потреба у створенні більш ефективних, зручних і персоналізованих систем для комунікації та взаємодії між користувачами. Це стосується як ігор так і візуальних новел, де особливу роль відіграють ефективні алгоритми для обробки даних та адаптивні інтерфейси. Технології, що використовуються для розробки таких платформ, постійно вдосконалюються, а вимоги до їх продуктивності, зручності та безпеки зростають. Важливим аспектом є також персоналізація досвіду користувачів, що дозволяє підвищити рівень взаємодії з платформою.

Мета кваліфікаційної роботи полягає у розробці підходів щодо створення та оптимізації комп'ютерної гри в жанрі візуальної новели з використанням інтерактивних інтерфейсів, персоналізації користувацького досвіду та системи обміну повідомленнями в реальному часі.

Для досягнення зазначеної мети необхідно виконати такі завдання:

- провести аналіз сучасних технологій, що застосовуються для створення інтерактивних та персоналізованих інтерфейсів у візуальних новелах;
- розробити алгоритми формування адаптивних інтерфейсів на основі виборів користувачів;
- реалізувати систему обміну повідомленнями в реальному часі для моделювання діалогів між персонажами;
- створити сценарії з інтерактивними виборами, що впливають на хід гри, але не змінюють її фінальний результат;
- розробити музичний супровід та звукові ефекти, що підсилюють емоційне сприйняття ігрового процесу;
- створити спрайти персонажів і фонові ілюстрації відповідно до стилістики гри.

Об'єктом кваліфікаційної роботи є методи розробки комп'ютерної гри в жанрі візуальної новели.

Предметом кваліфікаційної роботи є методи розробки комп'ютерної гри в жанрі візуальної новели з використанням системи обміну повідомленнями в реальному часі, інтерактивних інтерфейсів та алгоритмів персоналізації.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА АНАЛОГІЧНИХ ВЕБЗАСТОСУНКІВ

1.1 Аналіз та характеристика предметної області

Сфера розробки комп'ютерних ігор є однією з найдинамічніших галузей сучасної інформаційної індустрії. У межах цієї галузі особливе місце займає жанр візуальної новели – інтерактивного оповідання, що поєднує текстовий сюжет, графічні зображення, звукове оформлення та систему вибору, яка впливає на подальший розвиток подій.

Розробка комп'ютерних ігор є однією з найдинамічніших та інноваційних галузей сучасної інформаційної індустрії. Одним із найбільш популярних і перспективних напрямків у цій сфері є жанр візуальних новел. Візуальна новела – це інтерактивна форма оповідання, яка поєднує текстові елементи, графіку, анімацію, звукове оформлення та музичний супровід, створюючи багатогранний та емоційно насичений досвід для користувача. Цей жанр дозволяє не тільки розповідати захоплюючі історії, але й активно залучати гравця в процес прийняття рішень, що безпосередньо впливають на розвиток подій і результат гри.

Однією з ключових особливостей візуальних новел є їх нелінійна структура. У класичних відеоіграх сюжет часто є лінійним, де гравець слідує за визначеним сценарієм. Візуальні новели, натомість, дозволяють користувачеві впливати на хід подій, приймаючи важливі рішення на різних етапах гри. Це створює багатопланові наративи з численними можливими варіантами розвитку історії та різними кінцівками. Гравець може відчути себе справжнім учасником подій, що значно підвищує емоційну залученість і персоналізацію досвіду.

Візуальні новели часто поєднують текстові та графічні елементи, зображення персонажів, фонові малюнки та анімацію, що забезпечує візуальний супровід і покращує сприйняття історії. У сучасних візуальних новелах часто присутні не лише статичні картинки, а й анімації, звукові ефекти та музичне оформлення, що додає динамізму та атмосферності. Анімації, як правило,

використовуються для вираження емоцій персонажів, змін у навколишньому середовищі, а також для підсилення важливих моментів сюжету[4].

Для розвитку сюжету візуальні новели використовують систему вибору, яка дає гравцеві можливість взаємодіяти з персонажами і робити вибір, що впливає на подальший розвиток подій. Вибір гравця може стосуватися не лише діалогів, але й різних аспектів сюжету, таких як місце дії, завдання персонажів, або навіть ключові моральні дилеми. Така система дозволяє гравцеві відчувати, що його рішення важливі, а також забезпечує багатократне проходження гри з різними варіантами розвитку подій.

З технічного боку, створення візуальних новел вимагає використання спеціалізованих інструментів і фреймворків, що підтримують інтерактивні елементи, анімацію та мультимедійний контент. Одним з найбільш популярних інструментів є Ren'Py – фреймворк, який спеціалізується на розробці візуальних новел і забезпечує підтримку різноманітних форматів графіки, звуку, а також інтеграцію виборів та сценаріїв. Інші рушії, такі як Unity або Godot, також можуть використовуватися для створення візуальних новел, надаючи більше можливостей для інтеграції складних анімацій і міні-ігор.

Візуальні новели не обмежуються лише розважальними функціями. Вони можуть бути використані для навчання, розвитку критичного мислення та навіть у психологічній терапії. Завдяки своїй інтерактивній природі, цей жанр надає можливість створювати навчальні програми, які сприяють кращому запам'ятовуванню та залученню учасників до процесу навчання через інтерактивний досвід. Наприклад, можна використовувати візуальні новели для історичних реконструкцій, вивчення психології, вирішення соціальних проблем або навіть для розвитку м'яких навичок, таких як переговори або емоційний інтелект.

Особливо важливим аспектом у створенні візуальних новел є підтримка мультимедійного контенту, включаючи зображення, музику та анімації. Це створює емоційну атмосферу і підвищує враження від гри. Крім того, підтримка

мультимовності дає змогу локалізувати продукт для різних аудиторій, що значно розширює потенційний ринок гри.

З точки зору розробки, візуальна новела є багатокомпонентним продуктом, що включає як творчі, так і технічні аспекти. Розробник гри повинен володіти не тільки знаннями в галузі програмування, але й у сфері дизайну, сценаристики та інтеграції медіа-ресурсів. Крім того, необхідно мати глибоке розуміння психології гравців, щоб правильно налаштувати взаємодію між персонажами і користувачем, а також забезпечити правильну побудову нелінійного сюжету.

Завдяки своєму інтерактивному характеру, візуальні новели можуть створювати неповторні емоційні та інтелектуальні досвіди, що робить їх ідеальними для створення ігор, які не тільки розважають, але й навчають та сприяють розвитку різних навичок.

1.2 Огляд аналогічних існуючих застосунків

До складання специфікації вимог розроблюваного програмного забезпечення виникає необхідність у розгляді та аналізі вже наявних альтернативних ігор у жанрі візуальних новел, з виокремленням їхніх переваг, недоліків, основних функцій та характеристик. Ця інформація дозволяє сформулювати загальне уявлення про функціонал майбутньої гри та визначити, які саме елементи необхідно реалізувати для забезпечення повноцінної, емоційно насиченої та зручної взаємодії з користувачем. Серед найбільш відомих аналогів варто звернути увагу на такі продукти, як: Doki Doki Literature Club від розробника Team Salvato (табл. 1.1), Steins;Gate від 5pb. та Nitroplus (табл. 1.2), Clannad від компаній Key та Visual Arts (табл. 1.3).

Doki Doki Literature Club

Першим аналізованим аналогом є Doki Doki Literature Club (Team Salvato) (рис. 1.1). Ця візуальна новела здобула популярність завдяки своєму незвичайному підходу до жанру та несподіваним поворотам сюжету. Основною перевагою Doki Doki Literature Club є інтерактивність і здатність грати з

очікуваннями гравця, де сюжет має декілька рівнів і зламів, що дозволяє створювати унікальний досвід. Гравці можуть взаємодіяти з персонажами, вибирати варіанти діалогів та навіть впливати на кінцівку гри[8].

Недоліком є те, що Doki Doki Literature Club іноді може здатися занадто психологічно важким, через що не кожен гравець готовий до таких експериментів у жанрі. Однак її здатність поєднувати класичні елементи візуальних новел з темною та драматичною атмосферою робить її важливим прикладом для розуміння, як можна адаптувати жанр під нестандартні наративні підходи.

Таблиця 1.1 – Опис існуючого аналогу Doki Doki Literature Club

Назва	<i>Doki Doki Literature Club</i>
Виробник	Team Salvato
Архітектура	Desktop application
Мова реалізації	Ren'Py (Python).
Основні функції, характеристики	1. Інтерактивний нелінійний сюжет; 2. Унікальна механіка зміни гри через злом файлів; 3. Візуальний стиль аніме; 4. Різні кінцівки залежно від вибору гравця; 5. Глибокі психологічні та драматичні елементи.
Переваги	1. Незвичайний підхід до жанру візуальних новел; 2. Захоплюючий сюжет із непередбачуваними поворотами; 3. Висока реіграбельність.
Недоліки	1. Висока емоційна напруга для гравців; 2. Відсутність класичних геймплейних механік.
Веб-сайт	https://ddlc.moe/

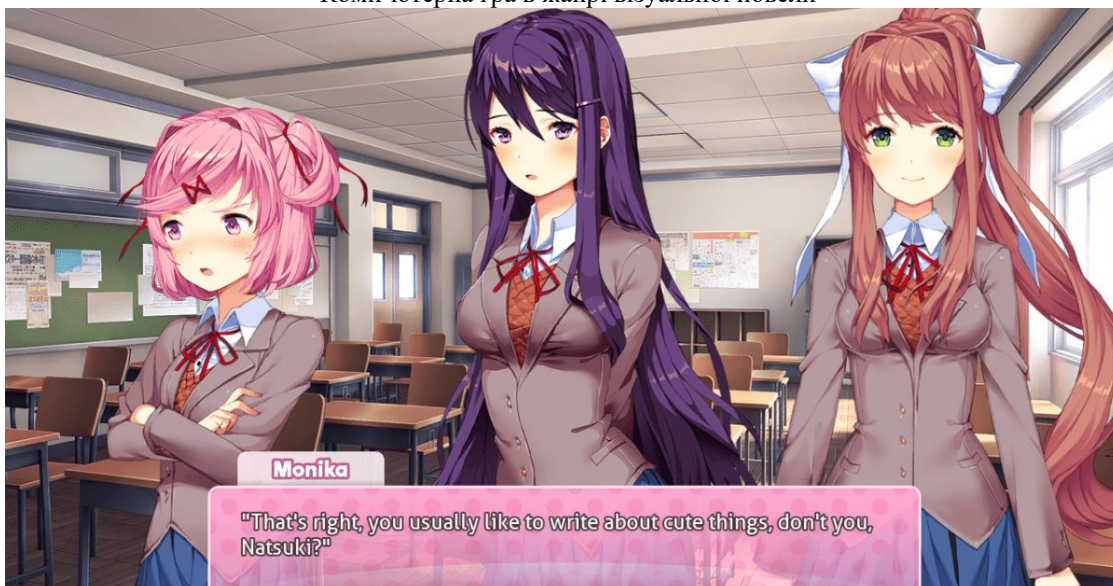


Рисунок 1.1 – Інтерфейс застосунку Doki Doki Literature Club

Steins;Gate

Другим аналізованим аналогом є Steins;Gate (5pb., Nitroplus) (рис 1.2). Ця візуальна новела є одним із кращих представників жанру "сай-фай" в контексті візуальних новел, де в центрі уваги – маніпуляції з часом і паралельними всесвітами. Перевагою Steins;Gate є глибокий, добре продуманий сюжет, який змішує елементи науки та філософії, створюючи неймовірно захоплюючу і важку для передбачення історію. Вибір гравця безпосередньо впливає на розвиток сюжету, що дозволяє кожному створювати унікальний досвід, враховуючи інтерпретації різних кінцівок[9].

Основним недоліком Steins;Gate є її важка для сприйняття структура, що іноді може заплутати гравців, особливо тих, хто не звик до складних наративних ходів. Однак, незважаючи на це, Steins;Gate заслужено займає важливе місце серед найкращих візуальних новел завдяки своїй глибині та непередбачуваним сюжетним поворотам[15].

Таблиця 1.2 – Опис існуючого аналогу **Steins;Gate**

Назва	<i>Steins;Gate</i>
Виробник	5pb., Nitroplus
Архітектура	Desktop application, Console application

Кінець таблиці 1.2

Мова реалізації	C++
Основні функції, характеристики	1. Глибокий науково-фантастичний сюжет; 2. Взаємодія через систему "телефонних викликів"; 3. Альтернативні сюжетні лінії; 4. Якісне аніме-візуальне оформлення; 5. Відмінний саундтрек.
Переваги	1. Прекрасно продуманий сюжет із науковою складовою; 2. Реалістичні та добре прописані персонажі; 3. Велика кількість альтернативних кінцівок.
Недоліки	1. Високий поріг входу для новачків; 2. Довгий розгін сюжету.
Веб-сайт	https://steins-gate.fandom.com/wiki/Steins;Gate

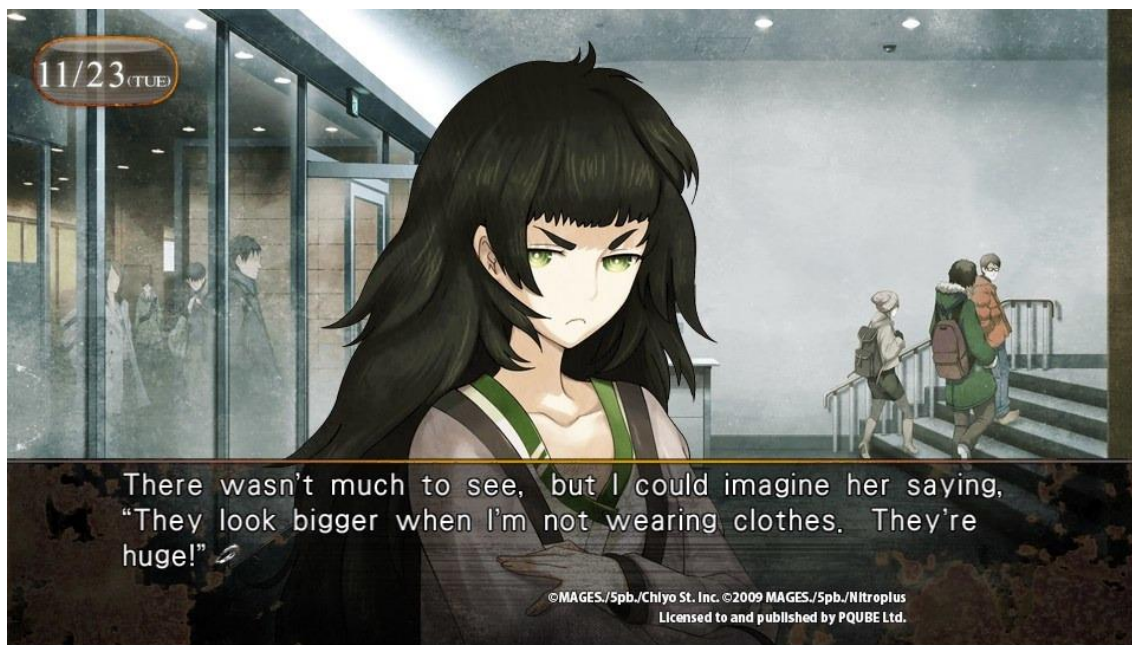


Рисунок 1.2 – Інтерфейс застосунку **Steins;Gate**

Clannad

Третім аналізованим аналогом є Clannad (розробники – Key та Visual Arts) (рис. 1.3). Ця візуальна новела є класичним представником жанру "гарем" з

виразними елементами романтики, драми та філософських роздумів про сімейні цінності, втрату й дорослішання. Clannad відзначається глибокими персонажами, ретельно прописаними сюжетними лініями та багатим емоційним наповненням, що дозволяє гравцям повністю зануритися в історію та сформувати тісний емоційний зв'язок з ключовими героями.

Особливою перевагою цієї новели є наявність великої кількості альтернативних сюжетних гілок і кінцівок, що залежать від вибору гравця. Це створює ефект інтерактивного роману, де кожне прийняте рішення має наслідки й впливає на розвиток подій. Clannad також використовує концепцію "After Story" – продовження основної сюжетної арки, яке розкриває наслідки виборів гравця у довгостроковій перспективі, що рідко зустрічається у подібних проектах[10].

Незважаючи на свою популярність, гра має деякі недоліки, зокрема – наявність стереотипних образів серед другорядних персонажів та іноді надмірно затягнуті діалоги. Однак її здатність гармонійно поєднувати романтичні, драматичні та психологічні елементи, робить її надзвичайно цінним прикладом для розробників, які прагнуть створити емоційно глибокий і життєвий сюжет у жанрі візуальної новели[18].

Таблиця 1.3 – Опис існуючого аналогу Clannad

Назва	<i>Clannad</i>
Виробник	Key, Visual Arts
Архітектура	Desktop application, Console application
Мова реалізації	C++
Основні функції, характеристики	1. Емоційний сюжет із глибокою соціальною тематикою, 2. Різні сюжетні маршрути та кінцівки, чудова анімація та художній стиль; 3. Озвучені персонажі, величезний вплив вибору гравця на фінал.
Переваги	1. Чуттєва та зворушлива історія; 2. Висока якість візуального та звукового оформлення; 3. Великий вплив вибору гравця.

Кінець таблиці 1.3

Недоліки	1. Повільний розвиток сюжету; 2. Деякі сюжетні лінії можуть бути передбачуваними.
Веб-сайт	https://key.visualarts.gr.jp/clannad/



Рисунок 1.3 – Інтерфейс застосунку Google Classroom

Узагальнюючи проведений аналіз, можна зробити висновок, що кожен із розглянутих аналогів – *Doki Doki Literature Club*, *Steins;Gate* та *Clannad* – демонструє унікальні підходи до реалізації візуальних новел. Вони відрізняються як за жанровою спрямованістю, так і за способом подачі сюжету, глибиною персонажів, наявністю альтернативних кінцівок та інтерактивністю. Отримана інформація дозволяє чіткіше окреслити ключові функціональні та художні елементи, які варто врахувати при розробці власного вебзастосунку. Зокрема, важливими аспектами є нелінійний наратив, психологічна глибина, атмосфера, а також зручність інтерфейсу для користувача. Аналіз аналогів став основою для формування цілісного бачення структури, функціональності та стилістики майбутньої гри у жанрі візуальної новели.

1.3 Поставлені завдання

Для досягнення поставленої мети в рамках розробки візуальної новели, важливо вирішити низку завдань, які охоплюють різні аспекти як технічної, так і художньої частини гри. Ось ключові завдання, які слід виконати:

1) Розробка адаптивних інтерфейсів на основі виборів користувачів.

Важливим етапом є розробка ефективних алгоритмів, які дозволяють створювати адаптивні інтерфейси, що змінюються залежно від вибору користувача. Це дозволить сформувати індивідуальний досвід для кожного гравця, який буде відчувати власний вплив на хід гри, зберігаючи при цьому загальну структуру сюжету. Необхідно створити систему, що враховує всі можливі варіанти розвитку подій, але не виходить за межі головної сюжетної лінії.

2) Розробка системи реального часу для обміну повідомленнями між персонажами.

Оскільки візуальна новела є інтерактивним продуктом, важливо реалізувати систему обміну повідомленнями в реальному часі, що моделює діалоги між персонажами. Це дозволить зробити спілкування з героями більш реалістичним та емоційним. Механізм діалогів, що реалізує цей процес, має бути гнучким, щоб дозволяти враховувати вибір гравця, зберігаючи плавність і природність комунікації.

3) Створення сценаріїв з інтерактивними виборами, що зберігають логіку сюжету.

Одним із основних завдань є створення сценаріїв, де вибори користувача впливають на розвиток подій, але не змінюють фінальний результат. Це дозволить гравцеві відчувати, що його рішення мають значення, і водночас зберігати логічну цілісність і структуру історії. Важливо збалансувати варіативність із необхідністю дотримання основного сюжету, щоб не втратити фокус на основній меті гри.

4) Розробка інтуїтивно зрозумілих інструментів для адміністрування платформи та профілів користувачів.

Важливим аспектом є створення зручних інтерфейсів для адміністрування гри, а також для персоналізації профілів користувачів. Це включає налаштування, які дозволяють гравцеві адаптувати гру під свої вподобання, а також збереження його прогресу і виборів. Такі інструменти підвищують рівень задоволення від гри та зручність взаємодії з платформою.

5) Комплексне тестування всіх компонентів гри.

Для забезпечення високої якості гри важливо провести комплексне тестування всіх її складових. Це включає тестування логіки сценаріїв, графічних і звукових ресурсів, а також взаємодії всіх елементів гри. Тестування має на меті виявлення технічних та дизайнерських помилок, а також оцінку впливу кожного компонента на загальний користувацький досвід.

6) Створення унікальних графічних ресурсів, що відповідають стилю гри.

Створення графічного контенту є основною частиною розробки візуальної новели. Важливо створити персонажів, фони та інші елементи, що відповідають художньому напрямку гри та взаємодіють між собою, забезпечуючи цілісність і гармонію всього візуального оформлення. Для цього необхідно ретельно продумати стиль графіки, вибір кольорів та деталей, що впливатимуть на емоційне сприйняття гравцем.

7) Створення оригінальної музики та звукових ефектів.

Музика в візуальних новелах має велике значення для створення атмосфери гри. Завданням є створення оригінальних музичних композицій, які відповідатимуть настрою кожної сцени. Також необхідно додати звукові ефекти, що підсилюють емоційну складову гри (наприклад, звуки кроків, дверей, чи вітру). Усі звуки повинні бути якісно відзвучені і гармонійно поєднуватися з візуальним контентом.

8) Розробка гнучкої логіки зміни станів персонажів та сцен.

Важливо забезпечити динамічну зміну станів персонажів та сцен у відповідь на вибори користувача. Це дозволить зробити сюжет більш інтерактивним та насиченим. Кожен емоційний стан, рух чи зміна виразу обличчя персонажа повинні відповідати вибору гравця та відображати його вплив на розвиток подій.

9) Інтуїтивно зрозумілий інтерфейс для користувача.

Оскільки візуальна новела орієнтована на широке коло користувачів, інтерфейс гри має бути простим, зрозумілим і доступним. Важливо створити таку навігацію, що дозволяє гравцеві без зайвих зусиль взаємодіяти з грою, не відволікаючись на складні елементи управління.

10) Оптимізація ресурсів гри для кращої роботи на різних пристроях.

Гра повинна працювати стабільно на різних пристроях, включаючи слабші комп'ютери та мобільні телефони. Задачею є оптимізувати використання ресурсів гри, щоб знизити навантаження на систему, що дозволить уникнути проблем з продуктивністю та забезпечити зручний досвід на будь-якому обладнанні.

11) Забезпечення надійної системи збереження прогресу.

Необхідно створити систему збереження прогресу гри, яка дозволить гравцеві повертатися до певних моментів і пробувати альтернативні варіанти розвитку подій. Це дозволить користувачеві насолоджуватися грою без страху втратити досягнуті результати при повторних спробах.

12) Можливість додавання нових сюжетних ліній та контенту.

Для забезпечення довговічності гри важливо передбачити можливість додавання нового контенту та сюжетних ліній без потреби в переписуванні всього проєкту. Це дозволить розширювати гру та забезпечити її гнучкість для майбутніх оновлень.

Висновки до розділу 1

У першому розділі було проведено детальний аналіз предметної області, яка охоплює процеси створення комп'ютерних ігор у жанрі візуальних новел.

Було визначено основні елементи структури таких ігор, зокрема сюжетну побудову, систему вибору, візуальне оформлення та механіки взаємодії з гравцем. Особливу увагу приділено ролі сценарію, типам персонажів, стилю візуалізації та особливостям реалізації нелінійного наративу, що є критично важливими для успішного сприйняття візуальної новели цільовою аудиторією.

Крім того, у розділі здійснено огляд існуючих аналогічних застосунків, таких як Doki Doki Literature Club, Steins;Gate та Clannad. Для кожної з розглянутих ігор проаналізовано основні функціональні можливості, художні та сценарні особливості, переваги та недоліки. Це дозволило сформувати загальне уявлення про сучасні підходи до реалізації візуальних новел і виявити як сильні сторони, що варто адаптувати у власному проєкті, так і проблемні моменти, яких слід уникати.

Проведений аналіз став основою для формування вимог до майбутнього програмного продукту. Завдяки вивченню існуючих рішень були окреслені ключові функціональні компоненти, які мають бути реалізовані у новій візуальній новелі для забезпечення її емоційної глибини, привабливості для користувача та загальної конкурентоспроможності на ринку. Отримані результати дозволяють перейти до етапу розробки технічної специфікації та архітектури програмного забезпечення.

2 МОДЕЛЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Вибір технологій та програмних засобів

Розробка комп'ютерної гри у жанрі візуальної новели передбачає використання спеціалізованих інструментів та мов програмування, які забезпечують інтерактивність, мультимедійність і зручність створення ігрового сценарію. До основних технологій, що використовуються у рамках даного проєкту, належать Ren'Py, Python, Paint Tool SAI та FL Studio.

Ren'Py – це спеціалізований двигун для створення візуальних новел, що базується на мові програмування Python. Він дозволяє легко інтегрувати діалоги, спрайти, фони, музику, а також реалізовувати системи вибору, які змінюють хід розповіді. Серед переваг Ren'Py – простота у вивченні, активна спільнота користувачів, підтримка багатьох платформ (Windows, macOS, Android). Ren'Py підтримує анімації, відтворення відео, логіку взаємодії між персонажами, а також автоматичне збереження прогресу[1] (рис 2.1).



Рисунок 2.1 – Ren'Py

Python використовується як внутрішня мова для розширення функціональності у Ren'Py. Завдяки своїй простоті та універсальності, Python дозволяє реалізовувати складні логічні гілки сюжету, працювати з умовами, змінними, масивами та іншими елементами програмування, необхідними для створення інтерактивного ігрового досвіду[3]. Крім того, Python дає змогу

інтегрувати додаткові модулі, розширити функціональність гри, наприклад, за допомогою збереження даних у зовнішні файли чи обліку статистики (рис 2.2).

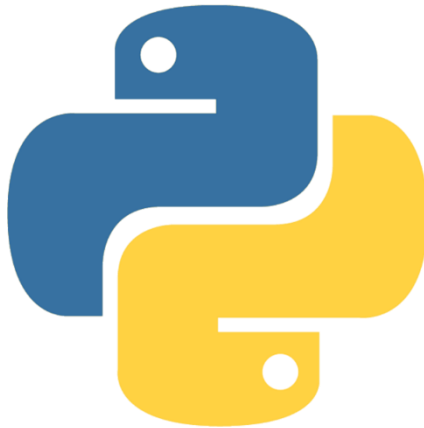


Рисунок 2.2 – Python

Paint Tool SAI – графічний редактор, що використовується для створення 2D-спрайтів, фонів, емоцій персонажів[5]. Він відзначається зручністю інтерфейсу, високою точністю малювання, підтримкою планшетів та шарами, що дає змогу створювати деталізовану і якісну графіку для гри. У середовищі SAI були промальовані основні елементи ігрової графіки: обличчя, емоції, тіла персонажів, а також фони для кожної сцени. Експорт відбувається у форматах .png або .jpg з високою роздільною здатністю (рис 2.3).



Рисунок 2.3 – Paint Tool SAI

FL Studio застосовується для створення музичного супроводу. Візуальні новели часто мають оригінальні саундтреки, які задають настрій сцені[8]. FL Studio дозволяє компонувати треки, використовувати синтезатори, семпли, ефекти, зводити музику до потрібного формату (наприклад, .ogg, .mp3), що сумісний із Ren'Py. У проєкті були створені тематичні музичні композиції для кожної сюжетної лінії: напружені сцени, романтичні моменти, сцени спокою тощо. Музика також впливає на загальний настрій користувача під час гри (рис 2.4).



Рисунок 2.4 – FL Studio

Для покращення інтеграції візуального та звукового контенту з програмною частиною, також застосовувались допоміжні утиліти та скрипти для автоматизації процесів імпорту, перевірки форматів, підготовки матеріалів до використання у рушії гри. Це дозволило зменшити час розробки та мінімізувати кількість технічних помилок[20].

2.2 Структура та логіка розроблюваного застосунку

Гра реалізується як послідовність сцен, які змінюються відповідно до виборів користувача. Кожна сцена містить діалог, ілюстрації та музичний супровід. Такий підхід дозволяє створити інтерактивне середовище, яке реагує на дії гравця, формуючи унікальний ігровий досвід. Структура гри умовно поділяється на:

- початкове меню з кнопками «Нова гра», «Продовжити», «Налаштування»;
- вступна сцена з коротким описом сюжету;
- розгалужена система діалогів з інтерактивними виборами;
- можливість повернення до попереднього вибору або пропуску сцен[17].

У грі реалізовано систему змінних, які впливають на розвиток діалогу. Наприклад, у залежності від вибору репліки, персонажі можуть змінювати емоції або реакції. Це створює ілюзію глибокої інтерактивності. Сценарій реалізовано у вигляді файлів з розширенням .gru, які підтримуються рушієм Ren'Py. Основні блоки коду відповідають за діалоги, сцени, переходи, звуковий супровід і логіку взаємодії з гравцем. Вся логіка побудована так, щоб у гравця виникало відчуття впливу на події, хоча загальна структура історії залишається керованою і логічною.

Кожен елемент сцени (персонаж, фон, звук) має свою змінну та управляється через прості команди (наприклад, show, hide, play music). Завдяки цьому гравець отримує динамічний ігровий досвід, де сцени швидко змінюються, відображаючи вибір користувача. Для реалізації деяких ігрових

механік були використані додаткові функції та умовні оператори на Python, що дало змогу створити складні сценарні гілки, які враховують попередній досвід гравця, його рішення та вибрані шляхи.

2.3 Робота з графікою та музикою

Створення візуального оформлення гри – важлива частина розробки. Усі персонажі та фони були промальовані вручну у програмі Paint Tool SAI. Кожен персонаж має кілька емоційних станів, які перемикаються у відповідь на репліки або вибори користувача[13]. Такий підхід дозволяє зробити спілкування з персонажами більш реалістичним та емоційним. Графіка експортується у форматі .png з прозорістю для подальшого імпорту в Ren'Py. Використання шарів дозволило створити анімаційні ефекти: моргання, зміну емоцій, легку анімацію рухів. Крім того, створювалися фони для кожної локації: класна кімната, вулиця, кімната головного героя тощо.

Окрему увагу було приділено узгодженості стилю між персонажами, фонами та загальним художнім напрямом гри. Підбір кольорів, деталізація та композиція кожної сцени ретельно перевірялися, щоб забезпечити цілісність візуального сприйняття. Задля цього застосовувались палітри кольорів, шаблони композиції, таблиці відповідності емоцій. Це дозволило досягти гармонії між різними елементами інтерфейсу, що, в свою чергу, позитивно впливає на занурення користувача у гру.

Для музичного оформлення було використано FL Studio. Створено кілька треків: вступна музика, тривожна сцена, романтичний фон. Музика експортується у форматі .ogg, що забезпечує хорошу якість при невеликому розмірі файлу. Вставка музики в Ren'Py відбувається через команди `play music` або `queue music`. Деякі звукові ефекти, наприклад, відкриття дверей або повідомлення, були створені вручну або адаптовані з відкритих бібліотек. Це дозволило зробити атмосферу гри більш живою та динамічною, забезпечуючи емоційне підсилення важливих сюжетних моментів.

Крім цього, використовувались MIDI-контролери та віртуальні інструменти, що допомогли створити якісні композиції з мінімальними витратами часу. Усі треки були зведені та протестовані у реальних умовах гри для досягнення гармонійного поєднання звуку та зображення. Музичний супровід був адаптований до ігрового ритму та інтенсивності сцен, щоб підтримати належний рівень залученості гравця.

2.4 Специфікація вимог

Розробка гри в жанрі візуальної новели передбачає створення інтерактивного програмного продукту, який поєднує у собі елементи художнього оповідання, графічного оформлення, звукового супроводу та взаємодії з користувачем через систему вибору. Щоб забезпечити якісну реалізацію ігрового проєкту, була сформована специфікація вимог до майбутнього застосунку. Ця специфікація складається з функціональних та нефункціональних вимог, які разом окреслюють повний обсяг очікуваного функціоналу та обмежень, що мають бути враховані під час розробки гри.

Функціональні вимоги

1) Можливість проходження сюжету з варіативною системою вибору, що впливає на подальший розвиток подій.

Ігровий процес повинен передбачати різноманітні гілки сюжету, що залежать від рішень, які приймає гравець у ключових моментах. Це забезпечує індивідуальний досвід проходження та підвищує реіграбельність гри.

2) Реалізація збереження та завантаження прогресу гравця.

Користувач повинен мати змогу вручну зберігати свій прогрес, а також скористатися функцією автозбереження. Завантаження даних повинно повертати гравця до відповідного моменту сюжету.

3) Відображення діалогів персонажів з динамічною зміною зображення.

Під час спілкування персонажів має змінюватися вираз обличчя, поза, фон чи інші візуальні елементи, що підсилюють атмосферу сцени.

4) Інтерактивний інтерфейс користувача.

Інтерфейс повинен бути зручним для навігації: кнопки для переходу вперед/назад, вибору реплік, відкриття меню тощо. Користувач має інтуїтивно розуміти, як здійснювати необхідні дії.

5) Наявність головного меню, меню налаштувань, журналу діалогів.

Меню повинні дозволяти користувачеві змінювати параметри гри (гучність, мова, швидкість тексту), переглядати попередні репліки та здійснювати інші дії, що покращують взаємодію з грою.

6) Підтримка мультимедійного контенту.

До складу гри входить графіка (фони, спрайти персонажів), звуковий супровід (музика, звукові ефекти), анімації (ефекти переходів, міміки). Все це створює повноцінну атмосферу візуальної новели.

7) Локалізація.

Передбачена можливість підтримки кількох мов інтерфейсу, принаймні української та англійської. Такий підхід дозволяє розширити цільову аудиторію гри.

Нефункціональні вимоги

1) Кросплатформеність.

Гра повинна працювати стабільно на різних операційних системах, зокрема Windows і Linux. Додатково розглядається можливість адаптації під Android-пристрої для розширення доступності.

2) Надійність збереження даних користувача.

Усі дані, пов'язані з прогресом гравця, мають зберігатися коректно та бути захищеними від випадкової втрати через помилки або раптове завершення роботи програми.

3) Масштабованість.

Структура гри має передбачати можливість розширення – додавання нових глав, персонажів, локацій без необхідності кардинального переписування коду.

4) Безпека даних.

Передбачаються механізми, які забезпечують цілісність файлів збереження та запобігають їх пошкодженню або несанкціонованому доступу.

5) Висока продуктивність.

Гра повинна ефективно функціонувати навіть на комп'ютерах із середніми характеристиками, не перевантажуючи системні ресурси та не викликаючи затримок.

6) Зручність користування інтерфейсом.

Інтерфейс гри повинен бути ергономічним, інтуїтивно зрозумілим для користувача, із можливістю зміни деяких елементів відповідно до особистих вподобань (наприклад, розмір тексту, колір фону тощо).

2.5 Опис інтерфейсу користувача

Інтерфейс користувача у візуальній новелі виконує ключову роль, адже саме через нього гравець взаємодіє з усіма елементами гри – читає текст, спостерігає за персонажами, здійснює вибір і керує ходом сюжету. Тому під час проєктування інтерфейсу було враховано не лише візуальну естетику, а й функціональність, інтуїтивність, а також відповідність жанровим очікуванням.

Головне меню

Після запуску гри користувач потрапляє до головного меню, яке оформлене в стилі гри – з фоновим зображенням, атмосферою музикою і плавною анімацією. Меню містить такі основні елементи:

- Start Game – початок нової гри. Переносить гравця до першої сцени з прологом або вступом.
- Load Game – відкриває екран з переліком збережених прогресів. Гравець може обрати потрібне збереження й продовжити гру з того моменту.
- Settings – розділ налаштувань, у якому можна відрегулювати:
 - 1) гучність музики, звукових ефектів та голосів персонажів;
 - 2) розмір та шрифт тексту;
 - 3) швидкість появи тексту;
 - 4) перемикання повноекранного режиму;
 - 5) вибір мови інтерфейсу.

- Gallery / Extras – необов'язковий розділ, де можна переглядати відкриті ілюстрації (CG), фонові зображення або саундтрек.
- Exit – вихід з гри.

Усі елементи меню оформлені у єдиному стилі, з урахуванням кольорової гами гри, а кнопки реагують на наведення або натискання зміненням кольору чи ефектом підсвічування.

Ігровий інтерфейс

Основний ігровий інтерфейс є центральним елементом гри. Його компоненти ретельно продумані, щоби не відволікати від сюжету, але водночас давати змогу зручно керувати грою:

- Текстове вікно розміщується в нижній третині екрана. У ньому виводиться весь наратив – репліки персонажів, внутрішні думки героя, описи подій або оточення. Зазвичай воно має напівпрозоре тло, щоби не повністю закривати ілюстрацію на фоні.

- Ім'я персонажа, що говорить, розміщується у верхній частині текстового вікна або в окремому фреймі поруч. Його може супроводжувати декоративне оформлення, наприклад кольоровий фон, індивідуальна рамка або іконка.

- Фонове зображення – це задній план кожної сцени, який змінюється відповідно до подій (наприклад: школа, ліс, кімната, дахьо).

- Спрайти персонажів розміщуються поверх фону – ліворуч, праворуч або по центру, залежно від того, хто говорить. Вони можуть мати кілька поз або емоцій, що змінюються відповідно до діалогу.

- Кнопки керування (інтерфейс навігації) зазвичай розташовуються в одному з нижніх кутів або викликаються натисканням:

- 6) Skip – пропуск уже прочитаного або всього тексту.
- 7) Auto – автоматичне прокручування тексту з заданою швидкістю.
- 8) Save – збереження поточного прогресу.
- 9) Load – відкриття меню збережень.
- 10) Backlog – перегляд нещодавніх діалогів.

11) Меню – пауза або повернення до головного меню.

Екран вибору

Під час ключових моментів гри на екрані з'являються опції вибору – зазвичай у вигляді кнопок із текстовими варіантами дій або відповідей. Гравець натискає на бажаний варіант, і сюжет змінюється (інколи лише злегка), в залежності від зробленого вибору. Дизайн кнопок відповідає загальному стилю гри – фон, шрифт, ефекти наведення.

Системне меню в грі

Натискання правої кнопки миші або клавіші *Esc* відкриває системне меню гри, яке дозволяє:

- продовжити гру;
- зберегти / завантажити;
- змінити налаштування;
- вийти в головне меню.

Графічний стиль інтерфейсу

Уся візуальна частина інтерфейсу оформлена відповідно до стилістики гри: шрифти, кольори, рамки, фони – все створено з метою підкреслити атмосферу візуальної новели, її жанр та настрій. Наприклад, для темної драми – темне напівпрозоре вікно, приглушені кольори та готичний шрифт, а для романтичної історії – світлі тони, округлі шрифти, плавні анімації.

Висновки до розділу 2

У другому розділі було розглянуто ключові етапи розробки комп'ютерної гри у жанрі візуальної новели. Проаналізовано специфіку жанру, його особливості та вимоги до побудови ігрового процесу. Розкрито основні підходи до написання сценарію, створення персонажів і побудови нелінійної структури оповіді. Представлено обґрунтування вибору інструментів для розробки гри, зокрема рушія Ren'Py, що забезпечує зручне створення інтерактивних сцен і гнучке управління сюжетними гілками.

Окрему увагу приділено створенню мокапів інтерфейсу, що демонструють основні екрани гри, включаючи головне меню, сцену діалогу та екран вибору. Також сформовано специфікацію вимог до майбутнього застосунку, яка охоплює функціональні, технічні та інтерфейсні аспекти. Завдяки детальному аналізу та підготовчій роботі, було створено чітке уявлення про кінцевий продукт, що дозволить реалізувати візуальну новелу з урахуванням вимог користувачів та стандартів сучасної розробки ігор.

3 ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Створення діаграми варіантів використання

Діаграма варіантів використання (Use Case Diagram) є одним із фундаментальних інструментів у методології моделювання UML (Unified Modeling Language), який дає змогу описати функціональні можливості системи через призму взаємодії зовнішніх суб'єктів (акторів) із самою системою. У рамках розробки гри у жанрі візуальної новели для персональних комп'ютерів така діаграма виконує ключову роль у визначенні функціональних вимог, структуруванні сценаріїв користувача та налагодженні логіки розробки. (рис 3.1).

Загалом діаграма варіантів використання дає змогу:

- **Візуально представити функціональні можливості програми, які** безпосередньо доступні кінцевому користувачу;
- **Виявити ролі користувачів та їх взаємодії з підсистемами, що** особливо корисно при роботі в команді;
- **Визначити взаємозв'язки між варіантами використання,** наприклад через включення (include) або розширення (extend);
- **Сформулювати технічне завдання на основі реального користувачького досвіду.**

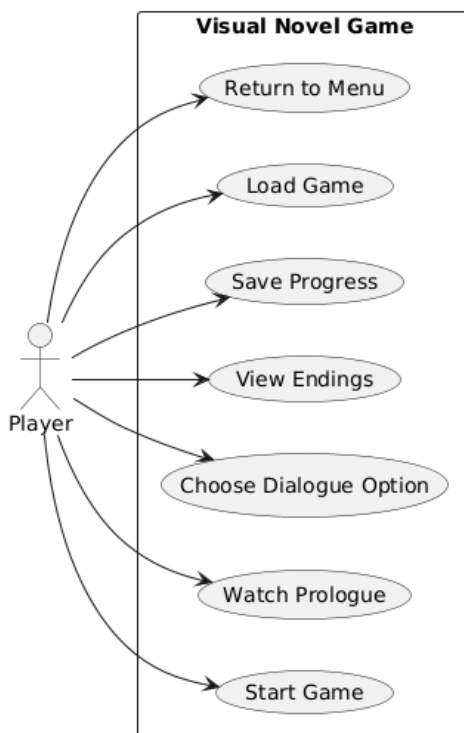


Рисунок 3.1 – Діаграма варіантів використання

У центрі діаграми для гри знаходиться основний актор – **гравець**, який взаємодіє з усіма основними елементами інтерфейсу та ігрового процесу. Гравець здійснює запити до системи (наприклад, «почати нову гру»), на які система реагує відповідними сценаріями.

Ключові варіанти використання, що деталізують життєвий цикл гри:

- **Запуск гри** – охоплює ініціалізацію рушія, завантаження ресурсів, перевірку сумісності пристрою, завантаження конфігурацій. Здійснюється автоматично, але може бути ініційований через системний лаунчер.
- **Перехід до головного меню** – доступ до навігаційних опцій: запуск, завантаження, налаштування, вихід.
- **Перегляд вступної сцени / прологу** – обов'язковий наративний блок, що налаштовує гравця на атмосферу гри. Часто супроводжується діалогами, озвученням або текстовим описом.
- **Початок нової гри** – запуск першого ігрового епізоду. Включає обробку сценарних гілок, запис у журнал подій, активацію внутрішніх тригерів.

– **Інтерактивний вибір** – кожна репліка чи діалоговий вибір, що пропонується гравцеві, фіксується в системі та може спричинити перехід до іншого варіанта розвитку подій.

– **Збереження прогресу гри** – реалізується як у ручному, так і в автоматичному режимі. Система створює файл збереження, що містить дані про поточну сцену, змінні стану, таймери, контекст діалогу тощо.

– **Завантаження гри** – відновлення гри з конкретної точки, використовуючи обране збереження.

– **Перегляд фіналу / кінцівок** – у візуальній новелі це важливий аспект наративу. Реалізується як стан із декількома можливими варіаціями завершення історії.

– **Доступ до налаштувань** – включає зміну параметрів звуку, мови, інтерфейсу, швидкості тексту.

– **Вихід з гри** – завершення сесії з попередньою пропозицією збереження прогресу (опційно), звільненням ресурсів і закриттям вікна програми.

Крім основного актора – **гравця**, на діаграму також можуть бути додані другорядні системні актори:

– **Система збереження** – взаємодіє з базою збережених станів гри, відповідає за коректне зчитування та запис файлів.

– **Модуль обробки налаштувань** – реагує на запити гравця щодо зміни параметрів гри, може бути реалізований як окрема підсистема.

– **Служба відтворення медіа** – забезпечує програвання музики, відео та анімацій у потрібні моменти.

Зв'язки між варіантами використання можуть мати форму:

– `<<include>>` – наприклад, кожне завантаження гри включає ініціалізацію медіа та відтворення сцени.

– `<<extend>>` – наприклад, під час вибору певної репліки може розширитися додатковий діалог, що залежить від попереднього вибору.

У результаті діаграма варіантів використання дозволяє розробникам:

- Спроекувати логіку ігрового рушія;
- Організувати ігровий інтерфейс відповідно до очікувань гравця;
- Підготувати основу для модульного програмування (розбиття гри на підсистеми).

Таким чином, діаграма варіантів використання виконує важливу роль на етапі проектування ПЗ, дозволяючи ефективно спланувати архітектуру системи, забезпечити її розширюваність у майбутньому, а також донести логіку роботи продукту до не технічної аудиторії, такої як потенційні інвестори або замовники.

3.2 Створення діаграми послідовності

Діаграма послідовності (Sequence Diagram) є одним із ключових інструментів моделювання в UML, який відображає хронологічну взаємодію між об'єктами системи у вигляді послідовного обміну повідомленнями. У контексті розробки гри в жанрі візуальної новели ця діаграма допомагає візуалізувати, як саме здійснюється обробка дій користувача на рівні програмної логіки та які внутрішні об'єкти системи беруть участь у цьому процесі. (рис 3.2).

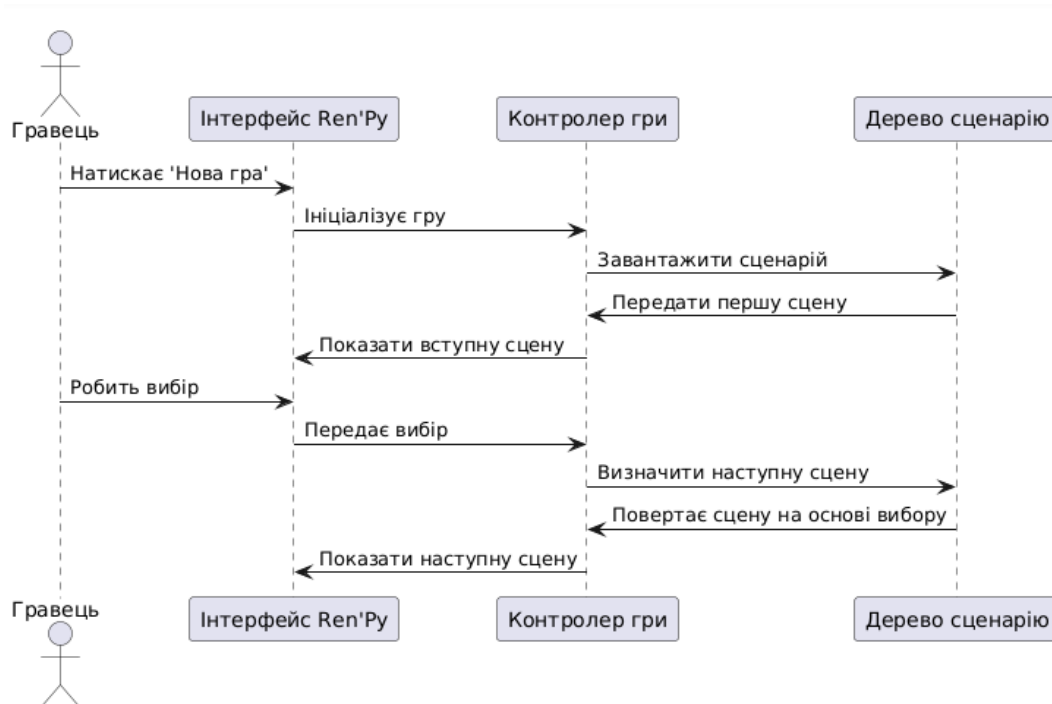


Рисунок 3.2 – Діаграма послідовності

У рамках розробки візуальної новели на платформі Ren'Py було змодельовано сценарій запуску нового епізоду гри, що є типовим прикладом користувацької взаємодії. На діаграмі відображено такі основні учасники процесу:

- користувач – ініціює взаємодію, наприклад, натисканням кнопки «Нова гра» або вибором дії.
- інтерфейс Ren'Py – обробляє введення користувача та виводить сцену.
- контролер гри – аналізує дії користувача, керує сценарною логікою та взаємодіє з ресурсами.
- сценарій (Script Engine) – зберігає структуру діалогів, варіанти вибору та переходи між сценами.

Основна послідовність дій:

- користувач натискає кнопку «Нова гра» в головному меню;
- інтерфейс фіксує дію та передає її контролеру гри;
- контролер гри зчитує зі сценарію необхідні дані для першої сцени;
- завантажується фонове зображення, музичний супровід і перша репліка;
- сцена виводиться на екран – гравець починає читати текст;
- після закінчення репліки з'являються варіанти відповіді (реплік або дій). Контролер визначає наступну сцену на основі вибору гравця;
- користувач обирає один із запропонованих варіантів;
- вибір передається назад до контролера, який аналізує сценарій;
- визначається наступна сцена, яка знову завантажується та відображається;

Незважаючи на те, що вибір користувача не змінює кінцівку історії (відповідно до особливостей ігрового дизайну даної візуальної новели), система все одно моделює різні переходи між сценами для створення ефекту варіативності та глибшого занурення.

Додаткова цінність діаграми послідовності полягає в тому, що вона дозволяє:

- виявити надлишкові або дубльовані виклики між об'єктами;
- перевірити логічну послідовність переходів;
- уніфікувати обробку введення та виходу;
- відстежити затримки або асинхронні дії (наприклад, завантаження ресурсів);
- спростити подальшу реалізацію логіки в коді.

Ця діаграма є особливо важливою при співпраці між програмістом, дизайнером інтерфейсу та сценаристом, оскільки дозволяє узгодити логіку розгортання подій у грі ще до етапу програмної реалізації.

3.3 Створення діаграми класів

Class-діаграма (діаграма класів) є одним із ключових структурних інструментів моделювання в нотації UML (Unified Modeling Language). Вона використовується для формалізації та візуалізації внутрішньої структури програмного забезпечення. Зокрема, вона демонструє класи системи, їхні атрибути (властивості), методи (функції), а також асоціації, спадкування, композицію чи агрегацію між класами. (рис 3.3).

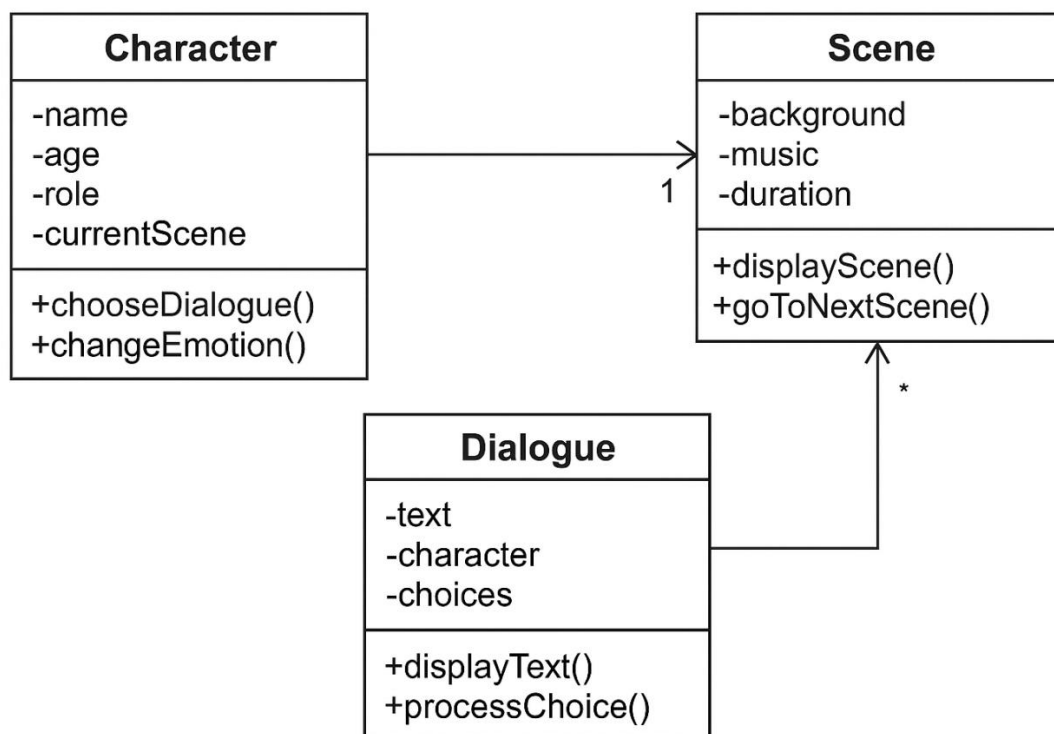


Рисунок 3.3 – Діаграма класів

Ця діаграма описує основну логіку розроблюваної комп'ютерної гри в жанрі візуальної новели та є важливою частиною моделювання, оскільки дозволяє формалізувати сутності гри ще до написання коду, що знижує ризик логічних помилок і покращує якість архітектури.

У моделі представлені такі основні класи:

Character (Персонаж)

Відповідає за опис героїв гри.

Атрибути:

- ім'я (`name`): ім'я персонажа;
- вік (`age`): числове значення віку;
- роль (`role`): тип персонажа (головний, другорядний, антагоніст тощо);
- поточна сцена (`currentScene`): сцена, у якій персонаж зараз перебуває.

Методи:

- `вибір діалогу()` – дозволяє персонажу реагувати на вибір гравця;

– зміна емоцій() – змінює вираз обличчя або позу персонажа відповідно до сценарію.

Scene (Сцена)

Клас, що репрезентує окремий фрагмент ігрового процесу.

Атрибути:

- фон (background): зображення сцени;
- музика (music): аудіосупровід;
- тривалість (duration): тривалість сцени в секундах або умовних одиницях.

Методи:

- відображення сцени() – виводить сцену на екран;
- перехід до наступної сцени() – виконує логічний перехід відповідно до сценарію.

Dialogue (Діалог)

Клас, що містить репліки персонажів, варіанти вибору та зв'язок із поточним персонажем.

Атрибути:

- текст (text): репліка, що буде відображена;
- персонаж (character): персонаж, якому належить репліка;
- вибір (choice): масив або список можливих відповідей гравця.

Методи:

- відображення тексту() – виводить діалог на екран;
- обробка вибору() – виконує дію залежно від вибраного варіанта.

Зв'язки між цими класами відображають, що Scene містить один або кілька Dialogue, а кожен Dialogue пов'язаний із конкретним Character. Це дозволяє реалізувати логіку, коли персонажі взаємодіють між собою в межах сцени, а сценарій розвивається відповідно до вибору гравця.

Deployment-діаграма використовується для візуального представлення архітектури розгортання програмного забезпечення. Вона демонструє, як

розподіляються основні компоненти системи між фізичними або віртуальними серверами, які вузли використовуються, їхні зв'язки та типи з'єднань.

3.4 Створення діаграми станів

Діаграма станів відображає загальну логіку поведінки гри в жанрі візуальної новели, що створюється для персональних комп'ютерів із використанням рушія Ren'Py. Вона демонструє послідовність переходів між різними станами гри, починаючи з моменту запуску програми й закінчуючи її завершенням. Такий підхід дозволяє формалізувати всі можливі варіанти поведінки системи в реакції на дії користувача, що є критично важливим на етапі проєктування сценаріїв взаємодії та програмної логіки.

Життєвий цикл гри починається з початкового стану, коли виконується запуск додатку. У цьому стані здійснюється ініціалізація ресурсів – завантаження необхідних компонентів інтерфейсу, налаштування графічного середовища, підготовка сценарного рушія тощо. Після цього користувач потрапляє до Головного меню, яке слугує центральною навігаційною точкою програми(рис 3.4).

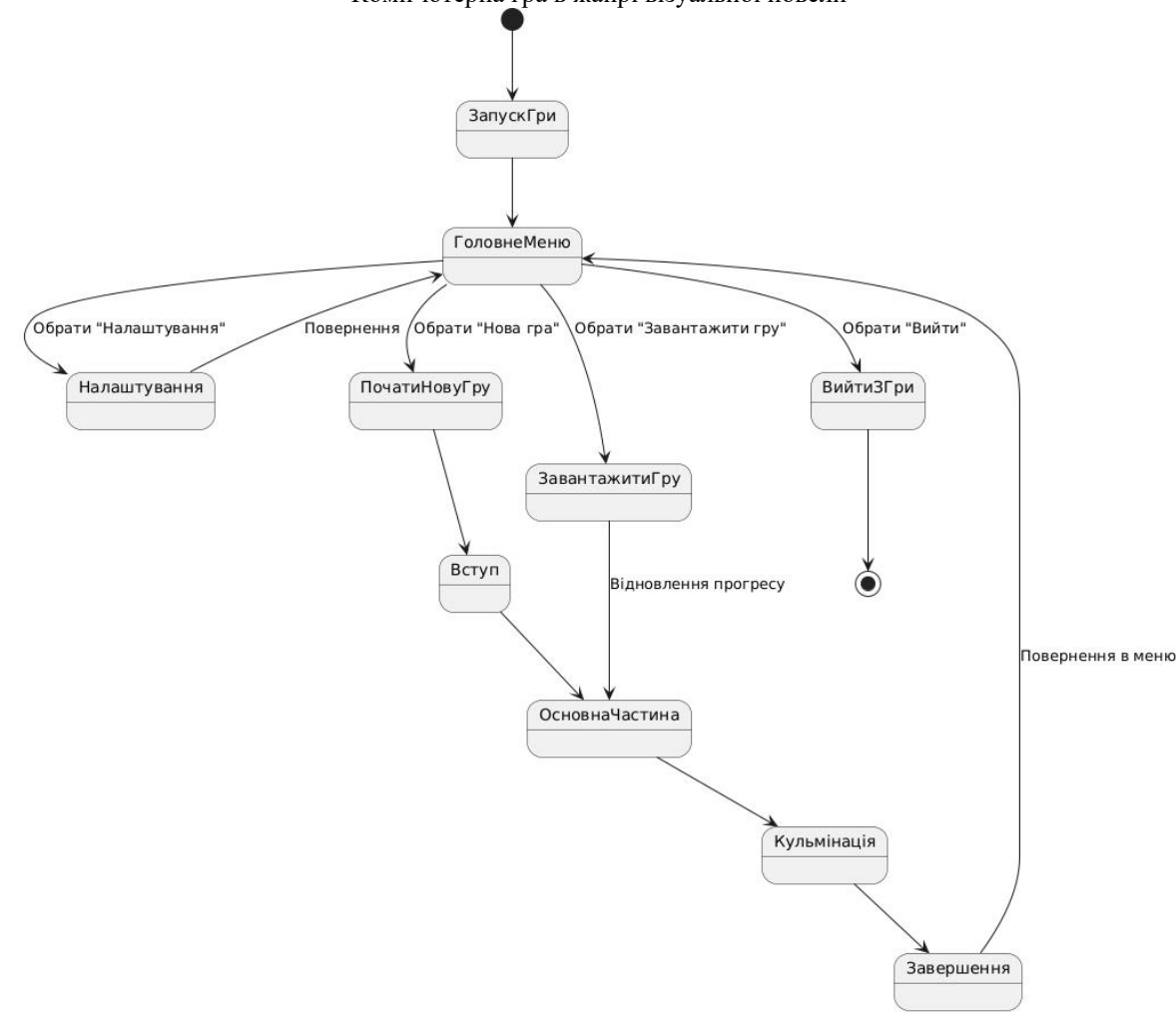


Рисунок 3.4 – Діаграма станів

У межах Головного меню доступні кілька ключових опцій:

- Почати нову гру – при виборі цього варіанта запускається ігровий процес з самого початку. Гравець поступово проходить через послідовність етапів: вступна сцена, розвиток сюжету, кульмінаційні події, розв’язка. Кожен із цих етапів представлений окремим станом, із чітко визначеними переходами на основі взаємодії користувача з діалогами, виборами або подієвими тригерами.
- Завантажити гру – ця опція дозволяє відновити гру з останньої точки збереження. Збереження містить інформацію про поточну сцену, стан персонажів, зроблені вибори тощо. Завантаження відбувається шляхом переходу до відповідного стану, який відображає ситуацію, зафіксовану в збереженні.
- Налаштування – користувач може змінити параметри гри, такі як гучність звуку, мова інтерфейсу, якість графіки, режим повноекранного

відображення тощо. Вихід із цього стану повертає гравця назад до головного меню.

– Вийти з гри – завершує сеанс, ініціюючи очищення ресурсів та коректне завершення роботи програми.

Під час ігрового процесу реалізуються складні сценарії переходів, які залежать від виборів користувача. Наприклад, вибір певної репліки в діалозі може призвести до зміни ходу подій, виклику нової сцени або переходу до альтернативної сюжетної гілки. Ці події також відображаються як переходи між станами, що підтримує гнучкість та нелінійність ігрового наративу.

Фінальний стан гри – це завершення сюжетної арки, після чого користувач автоматично повертається до головного меню. Він може розпочати нове проходження, переглянути інші варіанти розвитку історії або вийти з гри.

Таким чином, діаграма станів виступає інструментом високого рівня абстракції, який дозволяє зрозуміти загальну динаміку функціонування гри. Вона є важливою складовою технічної документації, оскільки допомагає розробникам, тестувальникам і дизайнерам краще орієнтуватися в логіці переходів, своєчасно виявляти помилки або непередбачені ситуації. Завдяки чіткому візуальному поданню діаграма полегшує реалізацію програмного коду, забезпечує кращу підтримку проєкту та сприяє покращенню взаємодії користувача з готовим продуктом.

3.5 Створення діаграми компонентів

У процесі проєктування програмного забезпечення важливою складовою є побудова діаграми компонентів, яка дозволяє представити архітектуру системи у вигляді взаємопов'язаних функціональних модулів. Такий підхід сприяє кращому розумінню структури програми, розподілу відповідальностей між компонентами та забезпечує зручність у подальшій модифікації та супроводі коду.

Діаграма компонентів (англ. *Component Diagram*) належить до структурних діаграм мови моделювання UML. Вона описує фізичні компоненти системи

(файли, бібліотеки, модулі) та їхні зв'язки, залежності й інтерфейси. У межах розробки комп'ютерної гри у жанрі візуальної новели на рушії **Ren'Py**, ця діаграма використовується для формалізації ключових модулів і взаємозв'язків між ними(рис 3.5).

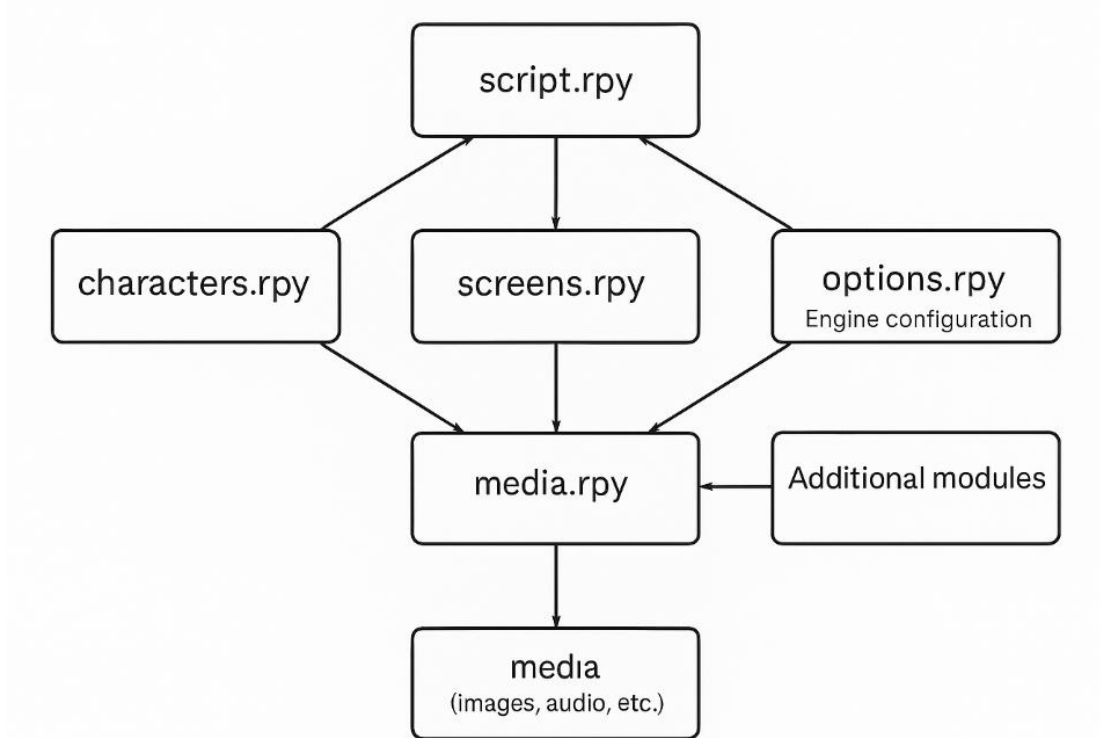


Рисунок 3.5 – Діаграма компонентів

Компоненти, що входять до складу системи:

- script.rpy – Основний сценарний модуль
Містить реалізацію основної логіки гри: сюжетні сцени, репліки персонажів, гілки вибору, переходи між сценами. Є центральним компонентом, що координує взаємодію з іншими модулями.
- characters.rpy – Модуль опису персонажів
Описує характеристики персонажів: імена, кольори діалогових реплік, зовнішній вигляд, портрети. Визначає візуальну та логічну ідентифікацію кожного героя гри.
- screens.rpy – Модуль графічного інтерфейсу користувача
Містить визначення внутрішньоігрових екранів, таких як меню гри, вікна вибору, інтерфейс налаштувань. Забезпечує взаємодію користувача з грою.

– `gui.rpy` – Конфігурація зовнішнього вигляду інтерфейсу
Налаштовує стилі GUI-елементів: розміри, шрифти, кольори, розміщення на екрані. Визначає візуальну естетику гри.

– `options.rpy` – Файл глобальних налаштувань рушія
Містить конфігурацію рушія Ren'Py, зокрема назву гри, параметри автозбереження, мову інтерфейсу, іконки та інші загальні параметри.

– Система збереження (`persistent`, `save`)
Відповідає за збереження стану гри, підтримку автозбереження та ручного збереження, завантаження ігрового прогресу. Є невід'ємною частиною рушія, але може бути розширена або змінена розробником.

– Медіаресурси (`images`, `audio`, `movies`)
Включають графічні, звукові та відеофайли, що використовуються в грі: фони, спрайти, CG-сцени, музика, звукові ефекти. Зберігаються у відповідних підкаталогах проєкту.

– Додаткові компоненти (модулі Python або сторонні плагіни)
Можуть використовуватися для розширення функціоналу гри, наприклад реалізації системи досягнень, журналу подій, мініігор тощо.

Зв'язки між компонентами:

– Компонент `script.rpy` викликає всі інші модулі: `characters.rpy`, `screens.rpy`, а також звертається до графіки, музики та збережень.

– `screens.rpy` залежить від `gui.rpy` для формування зовнішнього вигляду інтерфейсу.

– Компонент `options.rpy` встановлює глобальні параметри, які впливають на поведінку інших частин системи.

– Медіаресурси використовуються всіма основними модулями (`script.rpy`, `characters.rpy`, `screens.rpy`) у вигляді вставок у діалоги та сцени.

Висновки до розділу 3

У цьому розділі було здійснено проєктування програмного забезпечення для гри в жанрі візуальної новели. Метою проєктування стало створення формалізованої моделі, яка б відображала основні функціональні та структурні характеристики майбутнього застосунку, а також полегшувала процес реалізації та супроводу системи. Було побудовано низку UML-діаграм, кожна з яких відіграє важливу роль у розумінні та візуалізації різних аспектів роботи гри:

- Діаграма варіантів використання дала змогу визначити основні сценарії взаємодії користувача з програмою – запуск гри, вибір варіанту дій, перегляд сюжетних сцен, збереження та завантаження прогресу. Ці варіанти допомагають зосередитись на ключовому функціоналі, що необхідний для користувача.

- Діаграма послідовності дозволила змодельовати порядок виклику методів і обмін повідомленнями між об'єктами в процесі виконання типового сценарію – наприклад, запуск нової гри. Це допомогло з'ясувати логіку виконання дій у часі та визначити необхідні компоненти й функції.

- Діаграма класів надала структуроване уявлення про основні класи, які будуть реалізовані в коді. Зв'язки між класами, їх атрибути та методи дозволили сформулювати чітке уявлення про архітектуру застосунку з точки зору об'єктно-орієнтованого програмування.

- Діаграма станів описала поведінку застосунку залежно від дій користувача та внутрішніх умов – від початкового завантаження і до завершення гри або виходу в головне меню. Це особливо корисно для моделювання ігрової логіки та переходів між сценами.

Узагальнюючи, варто зазначити, що створення проєктної документації у вигляді UML-діаграм є критично важливим етапом у процесі розробки будь-якого складного застосунку, особливо у сфері ігор. Такі моделі забезпечують узгоджене бачення системи, підвищують якість розробки та зменшують імовірність помилок на етапах реалізації та тестування.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАСТОСУНКУ

4.1 Робота з графічними матеріалами

У процесі розробки комп'ютерної гри у жанрі візуальної новели важливу роль відіграють графічні матеріали, які формують візуальне сприйняття проєкту та забезпечують занурення користувача в ігрове середовище. До таких матеріалів належать фонові зображення, спрайти персонажів, інтерфейсні елементи, анімації тощо. У даному розділі розглядається підхід до підготовки та використання графічного контенту, а також його інтеграція в рушій гри.

4.1.1 Створення спрайтів персонажів

Для створення спрайтів персонажів було використано графічний редактор **Paint Tool SAI**, який забезпечує широкі можливості для цифрового малювання з використанням пензлів різної товщини та текстури. На початковому етапі були створені ескізи персонажів за допомогою графічного планшета. Далі виконувалося накладання базових кольорів, після чого – деталізація з використанням світлотіні, змін виразів обличчя та елементів одягу. Для кожного персонажа створено кілька варіантів емоційних станів, зокрема: радість, сум, здивування, злість тощо. Фінальні зображення зберігалися у форматі **PNG з прозорим фоном**, що дозволяє коректно відображати персонажів у середовищі ігрового рушія(рис 4.1).

- Спочатку було зроблено ескізи персонажів олівцем на планшеті.
- Потім наклав основні кольори.
- Наступним кроком було додавання деталей: тіні, світла, виразів обличчя, складок на одязі.
- Окремо були створені варіанти емоційних станів для кожного персонажа (наприклад, радість, сум, здивування).
- Фінальні спрайти зберігались у форматі PNG з прозорим фоном.

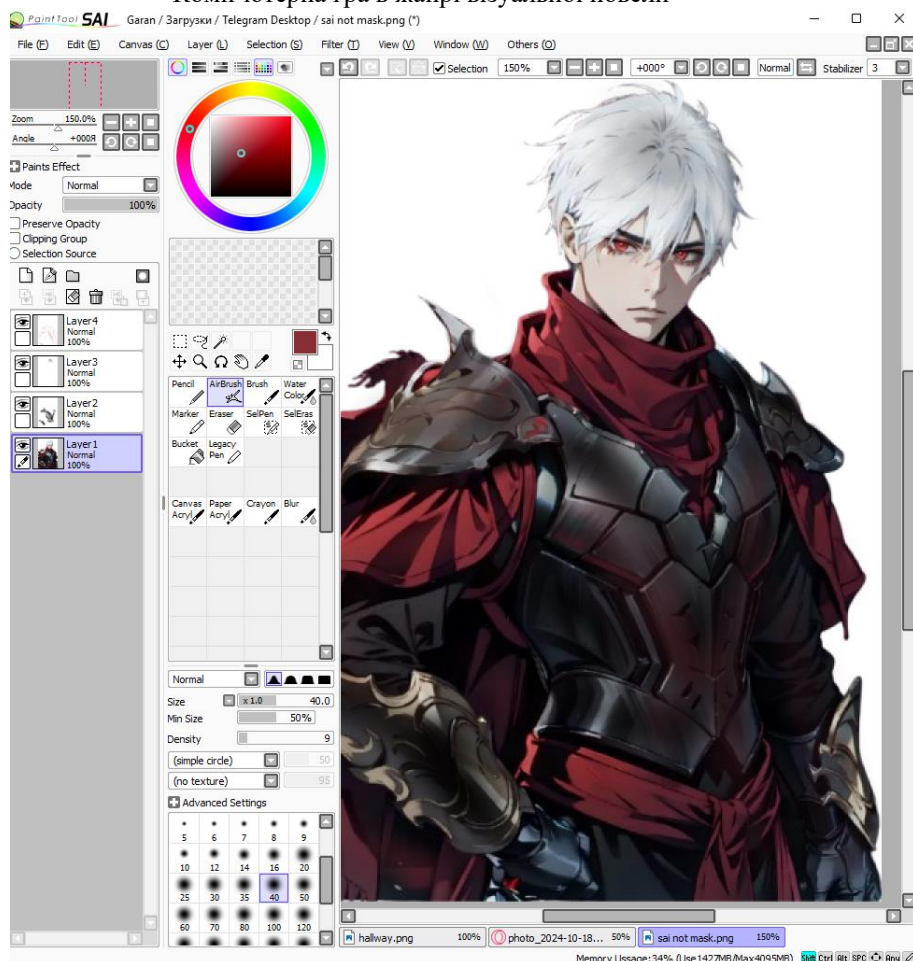


Рисунок 4.1 – Розробка спрайту головного героя

Після створення основного спрайта ретельно перевіряв кожен елемент на відповідність стилю проекту та загальній атмосфері гри. Особливу увагу було приділено виразам обличчя, оскільки вони відіграють важливу роль у передачі емоційної насиченості сюжету. Для досягнення максимальної деталізації використовував багатошарові пензлі та градієнти, що надали малюнкам глибину і живість. Всі спрайти були збережені у форматі PNG із прозорим фоном, що забезпечує їх легку інтеграцію у движок гри без додаткової обробки. Завдяки цьому процес створення спрайтів зайняв кілька ітерацій редагування, що в результаті дозволило отримати якісний та гармонійний візуальний ряд персонажів.

Для відображення персонажів у грі використовуються так звані спрайти – окремі зображення, що відображають зовнішній вигляд персонажа. У моїй грі кожен персонаж має власний набір спрайтів. Наприклад, спрайт персонажа на

ім'я **Сай** (Sai) був створений як окреме зображення, яке підключив до гри через оголошення змінної image.

Лістинг коду оголошення спрайта персонажа з масштабуванням:

```
image sai:  
    "sai not mask"  
    zoom 1.5
```

У цьому фрагменті "sai not mask" – це шлях до графічного файлу, тоді як zoom 1.5 задає масштабування. Це означає, що спеціально збільшив спрайт у 1.5 **раза**, щоб персонаж виглядав більш виразно на тлі сцени. Така маніпуляція дозволяє адаптувати оригінальний розмір зображення до загального візуального стилю гри(рис 4.2).

Лістинг коду виведення та приховування спрайта персонажа з ефектом:

```
show sai at left  
with dissolve  
  
hide sai  
with dissolve
```



Рисунок 4.2 – Спрайт головного героя в грі

Використання ефекту dissolve створює плавні переходи, що надає сценам більш професійного вигляду.

4.1.2 Створення фонів

Фони локацій було розроблено у тому ж графічному редакторі – **Paint Tool SAI**. Під час створення дотримано єдиного стилістичного напрямку, який відповідає художній концепції проєкту. Основна увага приділялася м'яким кольоровим переходам, атмосферному освітленню та деталізації об'єктів на передньому і задньому плані. Для досягнення ефекту глибини було використано методи перспективного малювання та накладання розмиття на дальні об'єкти. Готові фонові зображення збережено у форматі **JPG** або **PNG**, залежно від вимог до розміру та якості сцени(рис 4.3).

- Спершу було створено композицію фону – розміщення основних об'єктів.
- Потім були намальовані основні кольори.
- Далі додані тіні, світло, дрібні деталі (рослинність, предмети).
- Для ефекту глибини деякі елементи були зроблені більш розмитими.
- Фон експортувався у формат PNG з високою роздільністю.

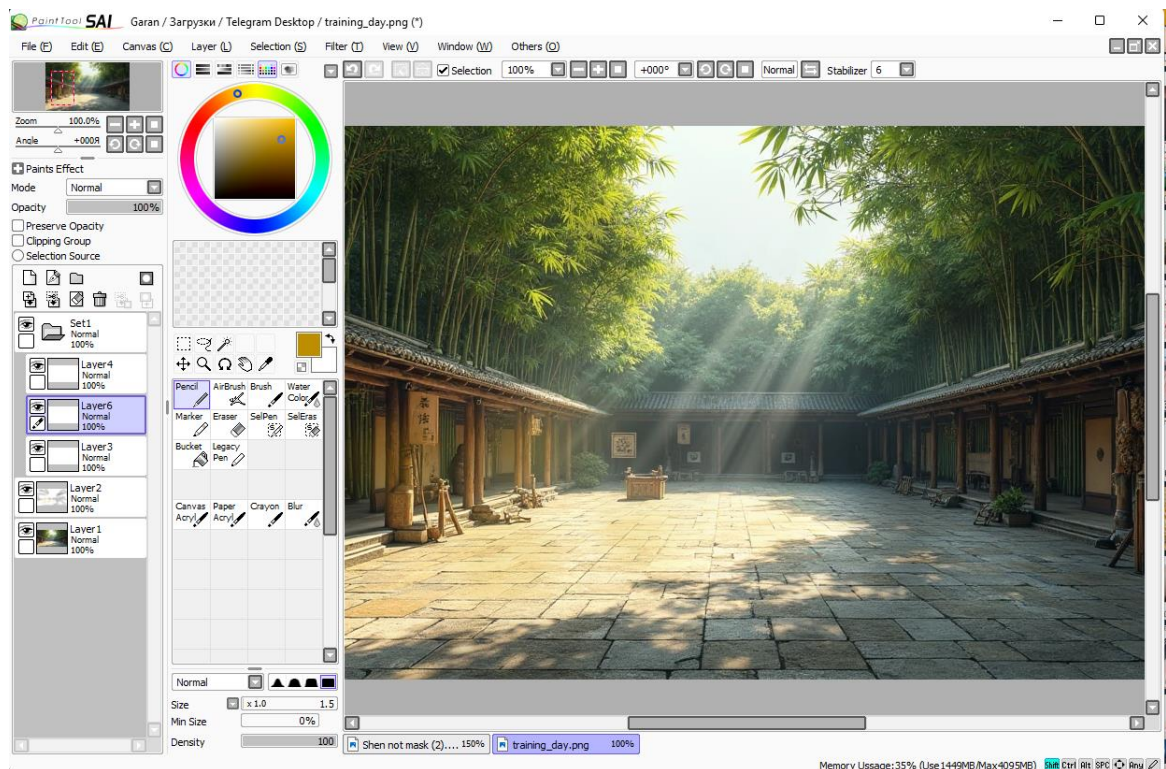


Рисунок 4.3 – Розробка фону тренувального майданчика

Процес створення фонів вимагав особливої уваги до колірної палітри та композиції, щоб кожна локація передавала потрібний настрій і водночас гармонійно поєднувалася з іншими графічними елементами новели. Особливу увагу було приділено м'яким переходам кольорів і текстурам, що нагадують акварельний живопис, що додає роботам легкості та затишку. Розмиті ділянки допомогли створити ефект глибини простору, завдяки чому фон не відволікає увагу від персонажів, а підкреслює основні об'єкти сцени. Використання PNG формату з високою роздільністю забезпечує чіткість і деталізацію зображень на різних екранах, що є важливим для комфортного сприйняття гравцями. Такий підхід значно підсилює загальний візуальний стиль гри та покращує занурення у сюжет.

Аналогічний підхід застосовувався і для роботи з фонами у рушії Ren'Py. Для прикладу, фонове зображення сцени тренування було оголошено та масштабовано

Лістинг коду оголошення фону з масштабуванням:

```
image training_daybg:  
    "training_day"  
    zoom 2.0
```

Застосування параметра zoom 2.0 дозволило збільшити фон удвічі, що допомогло уникнути порожніх країв і зробити сцену більш об'ємною (рис 4.4).

Лістинг коду виведення фону на сцену з ефектом переходу:

```
scene training_daybg  
with dissolve
```

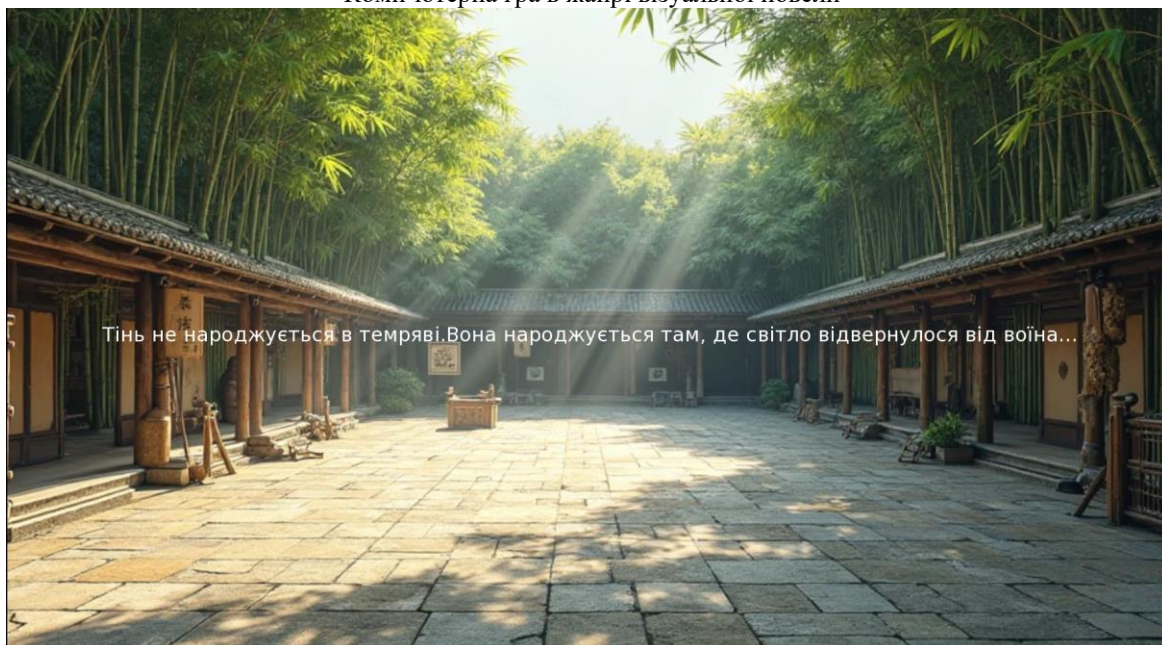



Рисунок 4.4 – Фон тренувального майданчика в грі

Цей код демонструє відображення фону на сцені з м'яким переходом, що забезпечує приємну зміну локації для користувача.

4.2 Робота з музикою та звуком

Для створення музичного супроводу використовував програму FL Studio – потужний секвенсор і редактор звуку, який надає широкі можливості для створення музики різних жанрів та настроїв (рис 4.5 - рис 4.6). Процес створення музики можна розділити на кілька основних етапів:

- **Визначення стилю музики:** На цьому етапі підбирав музичний стиль, що максимально відповідає настрою конкретної сцени. Для спокійних діалогів це була ніжна інструментальна мелодія, а для драматичних моментів – напружена і динамічна композиція.

- **Створення окремих треків:** Кожен музичний елемент – мелодія, бас, ударні та ефекти – створювався окремо, що дозволяло тонко налаштовувати звучання кожного інструмента.

- **Використання віртуальних інструментів:** Для реалізації композиції застосовувалися синтезатори, семпли та інші аудіобібліотеки, що значно розширювало можливості створення унікального звучання.

- **Адаптація музики під сюжет:** Деякі композиції писалися з нуля, а інші редагувалися під ритм та атмосферу конкретної сцени, що дозволяло краще підкреслити емоційний фон.
- **Експорт музичних файлів:** Після завершення створення музики композиції експортувалися у форматах WAV або MP3 для подальшого використання у візуальній новелі.
- **Інтеграція у Ren'Py:** За допомогою спеціальних команд музика підключалась до новели, забезпечуючи плавний перехід між треками та підтримуючи цілісність атмосфери гри.

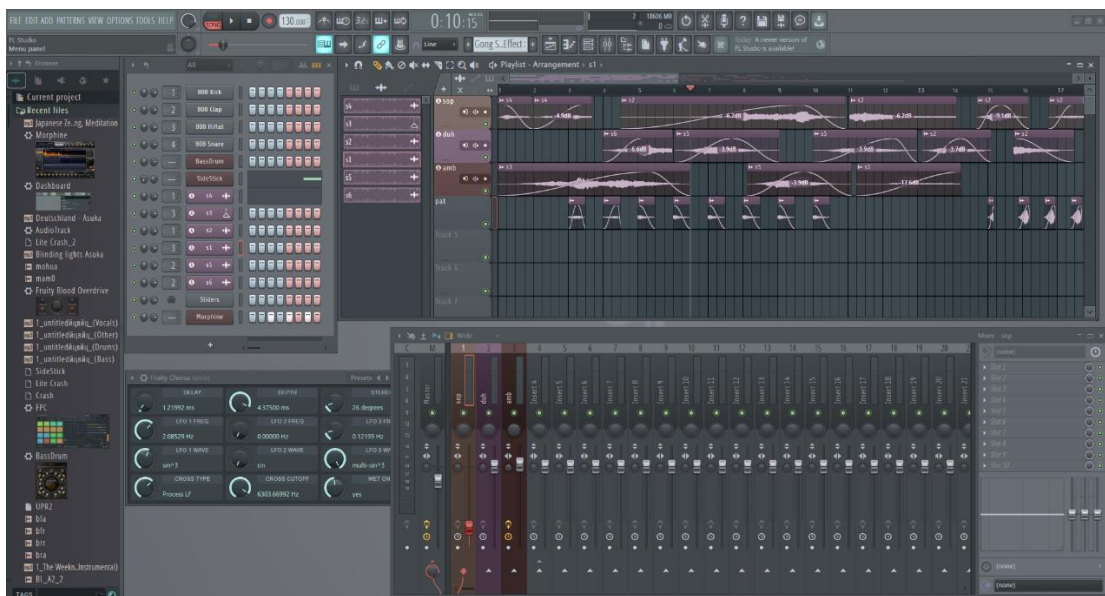


Рисунок 4.5 – Розробка фонової музики для гри



Рисунок 4.6 – Розробка звуку гонгу для гри

Такий поетапний підхід до створення музичного супроводу дозволив отримати якісний і відповідний настрою звук, який підсилює емоційний вплив сюжету та сприяє кращому зануренню гравця у світ новели.

4.3 Реалізація меню гри

У процесі розробки візуальної новели було створено два основні типи меню: головне меню та ігрове меню. Вони відповідають за навігацію користувача та забезпечують зручний інтерфейс для керування грою. Для реалізації меню використано можливості системи Ren'Py з налаштуванням екранних елементів і стилів.

4.3.1 Головне меню

Головне меню відповідає за початковий екран гри, де гравець може почати нову гру, завантажити збереження або вийти з гри. В його реалізації використано екран з тегом `menu`, що гарантує заміну будь-яких інших меню. Основний фон задається через графічний ресурс `gui.main_menu_background` (рис 4.7).

1. Створення екрану головного меню

- Для головного меню було створено спеціальний екран `main_menu()` у Ren'Py.
- Використовується тег `menu`, що гарантує заміну будь-якого іншого екрану меню цим екраном.
- Фон меню задається через додавання графічного ресурсу `gui.main_menu_background`.
- Додатково застосовується затемнювальна рамка (`frame` зі стилем `main_menu_frame`), яка робить фон більш приглушеним і фокусує увагу на меню.
- Вміст меню (кнопки навігації) імплементований через оператор `use navigation`, що підключає екран навігації Ren'Py.

2. Відображення назви гри і версії

- Якщо у налаштуваннях (`gui.show_name`) дозволено показ, зверху виводиться назва гри (`config.name!`) і номер версії (`config.version`).

– Відповідний текст форматовано за допомогою стилів `main_menu_title` і `main_menu_version`.

3. Стилiзацiя меню

– Основна рамка меню (`main_menu_frame`) має фонове зображення `gui/overlay/main_menu.png`, розмір по ширині 420 пікселів, а висота динамічно підлаштовується.

– Текстові елементи оформлені через набір стилів, що беруться з глобальних налаштувань GUI.

– Для вертикального розташування тексту і елементів використовується `vbox` з власним стилем `main_menu_vbox`.

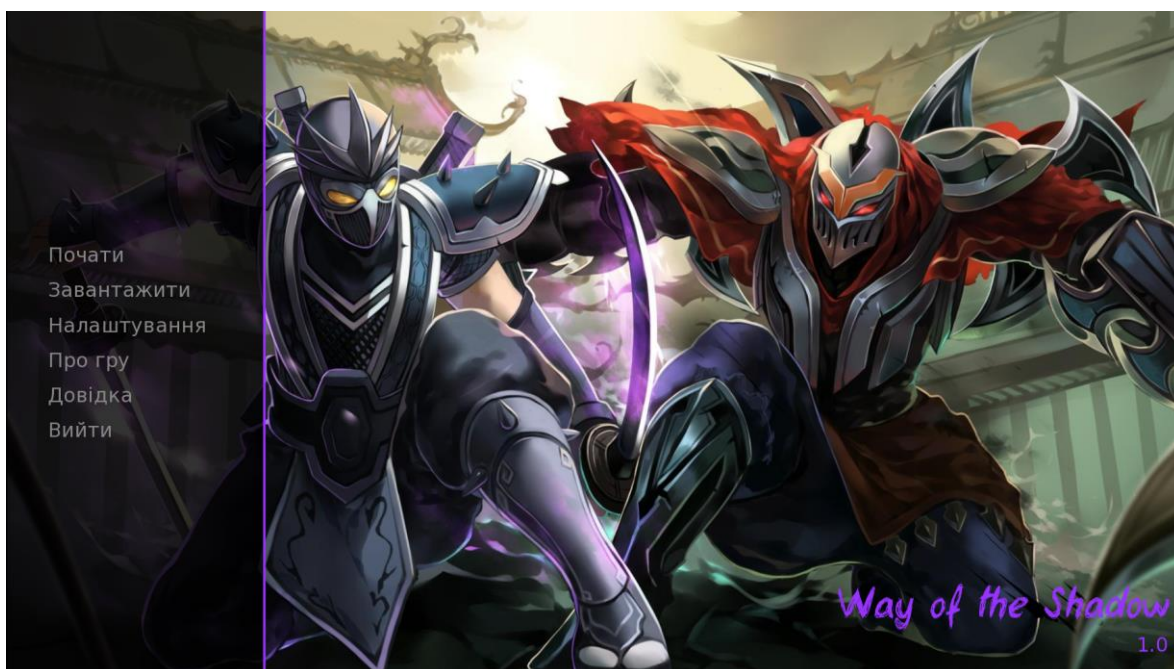


Рисунок 4.7 – Скріншот головного меню

Щоб зробити фон більш приглушеним і акцентувати увагу гравця на меню, додано рамку (`frame`) із стилем `main_menu_frame`, яка накладає затемнювальний ефект.

Вміст головного меню – кнопки навігації – реалізовані через оператор `use navigation`, що включає окремий екран навігації `Ren'Py` і забезпечує консистентність інтерфейсу.

Лістинг коду основних елементів екрану головного меню:

```
screen main_menu():  
    tag menu  
    add gui.main_menu_background  
  
    frame:  
        style "main_menu_frame"  
  
    use navigation
```

Відображення назви гри і версії

Важливим UX-елементом є показ назви гри та її версії у верхньому правому куті. Це робиться лише за умови, що у налаштуваннях (gui.show_name) дозволено їх показ.

Текст назви гри береться з конфігурації через config.name!t, а версія – з config.version.

Для оформлення тексту застосовуються стилі main_menu_title та main_menu_version, що відповідають загальному дизайну інтерфейсу.

Лістинг коду відображення назви та версії у меню:

```
if gui.show_name:  
    vbox:  
        style "main_menu_vbox"  
        text "[config.name!t]":  
            style "main_menu_title"  
        text "[config.version]":  
            style "main_menu_version"
```

Стилізація меню

Рамка меню (main_menu_frame) має ширину 420 пікселів, а висота динамічно підлаштовується під вміст за допомогою ufill True. Як фон для рамки використовується зображення gui/overlay/main_menu.png.

Вертикальний контейнер (vbox) для текстових елементів позиціонується у верхньому правому куті за допомогою стилю main_menu_vbox. Цей стиль встановлює відступи та вирівнювання, щоб текст виглядав охайно.

Текстові стилі використовують глобальні налаштування GUI, що забезпечує уніфікований вигляд шрифтів і кольорів по всій грі.

Лістинг коду прикладу стилів для головного меню:

```
style main_menu_frame:
    xsize 420
    yfill True
    background "gui/overlay/main_menu.png"

style main_menu_vbox:
    xalign 1.0
    xoffset -30
    xmaximum 1200
    yalign 1.0
    yoffset -30
```

4.3.2 Ігрове меню

Створення екрану ігрового меню

Ігрове меню реалізоване через екран `game_menu()`, що приймає параметри: заголовок меню (`title`), спосіб прокрутки вмісту (`scroll`), початкову позицію прокрутки (`yinitial`) і відступи між елементами (`spacing`).

Фон меню змінюється залежно від того, чи меню є головним: для головного меню використовується `gui.main_menu_background`, для інших випадків – `gui.game_menu_background`.

Лістинг коду визначення фону і параметрів екрану `game_menu()`:

```
screen game_menu(title, scroll=None, yinitial=0.0, spacing=0):
    style_prefix "game_menu"

    if main_menu:
        add gui.main_menu_background
    else:
        add gui.game_menu_background
```

Рамка меню і розміщення вмісту

Ігрове меню обгорнуте у рамку зі стилем `game_menu_outer_frame`, яка задає відступи зверху (180 пікселів) та знизу (45 пікселів), а також фонове зображення.

Вміст меню розміщується у горизонтальному контейнері (`hbox`), що поділяється на дві основні частини:

– Ліва – навігаційна панель (`game_menu_navigation_frame`), зарезервована для кнопок та інших елементів керування.

– Права – основний вміст меню (`game_menu_content_frame`), де відображаються елементи меню, список опцій, тощо.

Лістинг коду розміщення вмісту у рамці з двома секціями:

```
frame:
    style "game_menu_outer_frame"
hbox:
    frame:
        style "game_menu_navigation_frame"
    frame:
        style "game_menu_content_frame"
```

Прокрутка вмісту меню

Екран підтримує різні типи прокрутки вмісту, залежно від параметру `scroll`:

– `"viewport"` – вертикальна прокрутка через елемент `viewport`, що підтримує мишине колесо, перетягування та клавіші сторінок.

– `"vpgrid"` – вертикальний скролінг списку з фіксованою кількістю колонок (`vpgrid`).

– Якщо прокрутка не вказана, вміст відображається без скролінгу.

Вміст, який потрібно прокрутити, задається оператором `transclude`, що дозволяє вміщати дочірні елементи.

Лістинг коду варіанти прокрутки вмісту:

```
if scroll == "viewport":
    viewport:
        yinitial yinitial
        scrollbars "vertical"
        mousewheel True
        draggable True
        pagekeys True
        side_yfill True
        vbox:
            spacing spacing
            transclude

elif scroll == "vpgrid":
    vpgrid:
        cols 1
        yinitial yinitial
        scrollbars "vertical"
        mousewheel True
        draggable True
        pagekeys True
        side_yfill True
        spacing spacing
```

```
else:  
    transclude
```

Навігація та керування

Для зручності користувача в меню присутня кнопка "Повернутися", яка виконує дію `Return()`, повертаючи назад до попереднього екрану.

Угорі відображається заголовок меню, який задається параметром `title`.

Якщо меню викликається з головного меню, клавіша `game_menu` повертає користувача назад на головний екран.

Лістинг коду кнопки повернення і заголовка:

```
use navigation  
  
textbutton _("Повернутися"):  
    style "return_button"  
    action Return()  
  
label title  
  
if main_menu:  
    key "game_menu" action ShowMenu("main_menu")
```

Стилізація ігрового меню

Стиль `game_menu_outer_frame` визначає верхній відступ 180 пікселів і нижній 45 пікселів, а також встановлює фон меню.

Навігаційна панель (`game_menu_navigation_frame`) має ширину 420 пікселів і займає всю висоту меню.

Основний вміст (`game_menu_content_frame`) має лівий і правий відступи (60 та 30 пікселів відповідно).

Прокрутка реалізована з урахуванням зручності користування: підтримується мишине колесо, перетягування та клавіші навігації сторінок.

Текстові елементи оформлені відповідно до загального стилю гри, з урахуванням кольорів, розмірів та вирівнювання (рис 4.8).

Лістинг коду стилів ігрового меню:

```
style game_menu_outer_frame:  
    bottom_padding 45  
    top_padding 180  
    background "gui/overlay/game_menu.png"  
  
style game_menu_navigation_frame:  
    xsize 420  
    yfill True  
  
style game_menu_content_frame:  
    left_margin 60  
    right_margin 30  
    top_margin 15
```

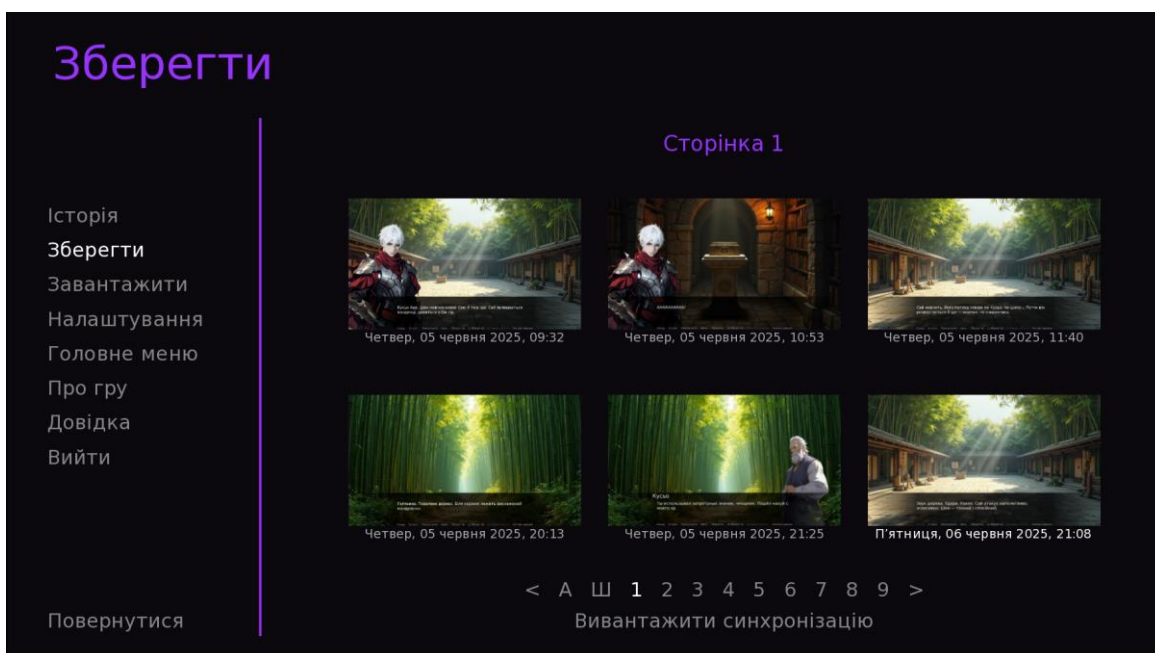


Рисунок 4.8 – Скріншот ігрового меню

Таким чином, екран ігрового меню у Ren'Py реалізований як гнучка та адаптивна структура, яка забезпечує зручність навігації, естетичний вигляд та повну інтеграцію з загальним інтерфейсом гри. Його модульність дозволяє легко змінювати компоненти або додавати нові функції відповідно до потреб візуальної новели.

4.4 Налаштування інтерфейсу гри

Графічний інтерфейс – це «обличчя» візуальної новели, через яке гравець сприймає сюжет, взаємодіє з персонажами, здійснює вибори та керує грою. Саме

тому правильно продуманий дизайн інтерфейсу може суттєво підвищити залученість і комфорт під час проходження.

В Ren'Py інтерфейс можна налаштовувати дуже гнучко: змінювати вигляд кнопок, розмір і стиль шрифтів, позицію панелей і навіть анімації їх появи чи зникнення. Це дозволяє не тільки створити унікальний стиль, що відображає атмосферу гри, але й адаптувати інтерфейс під різні платформи – ПК, ноутбуки, планшети чи мобільні телефони.

У цьому розділі ми докладно розглянемо основні параметри, які відповідають за вигляд та розташування інтерфейсних елементів. Розуміння цих налаштувань допоможе тобі зробити гру більш професійною, зручною і приємною для гравця.

4.4.1 Основні параметри елементів керування

Всі інтерактивні елементи інтерфейсу – кнопки, радіокнопки, чекбокси, підтвердження вибору – мають базові властивості, які можна гнучко налаштовувати(рис 4.7). Це:

- Розмір і колір тексту
- Внутрішні відступи (padding)
- Вирівнювання тексту всередині кнопок
- Колір та стиль у різних станах (звичайний, активний, наведений)

Лістинг коду стилів ігрового меню:

```
define gui.radio_button_borders = Borders(27, 6, 6, 6)
define gui.check_button_borders = Borders(27, 6, 6, 6)

define gui.confirm_button_text_xalign = 0.5

define gui.page_button_borders = Borders(15, 6, 15, 6)

define gui.quick_button_borders = Borders(15, 6, 15, 0)
define gui.quick_button_text_size = 21
define gui.quick_button_text_idle_color = gui.idle_small_color
define gui.quick_button_text_selected_color = gui.accent_color
```

- Borders(27, 6, 6, 6) – це параметр, який визначає відступи всередині кнопок або інших елементів: 27 пікселів зліва, 6 зверху, справа і знизу. Відступи

впливають на те, наскільки «вільним» буде текст або графіка всередині елемента – це впливає на зручність натискання, сприйняття дизайну та загальну естетику.

- `confirm_button_text_xalign = 0.5` – текст на кнопці підтвердження вирівнюється по центру. Центрування тексту особливо важливе для таких кнопок, щоб гравець легко міг його прочитати і не було відчуття "переїзду" тексту.

- `quick_button_text_idle_color` і `quick_button_text_selected_color` – це кольори тексту в кнопках швидкого доступу залежно від їх стану. Колір робить елемент помітним і допомагає гравцеві зрозуміти, коли кнопка активна чи натиснута(рис 4.9).

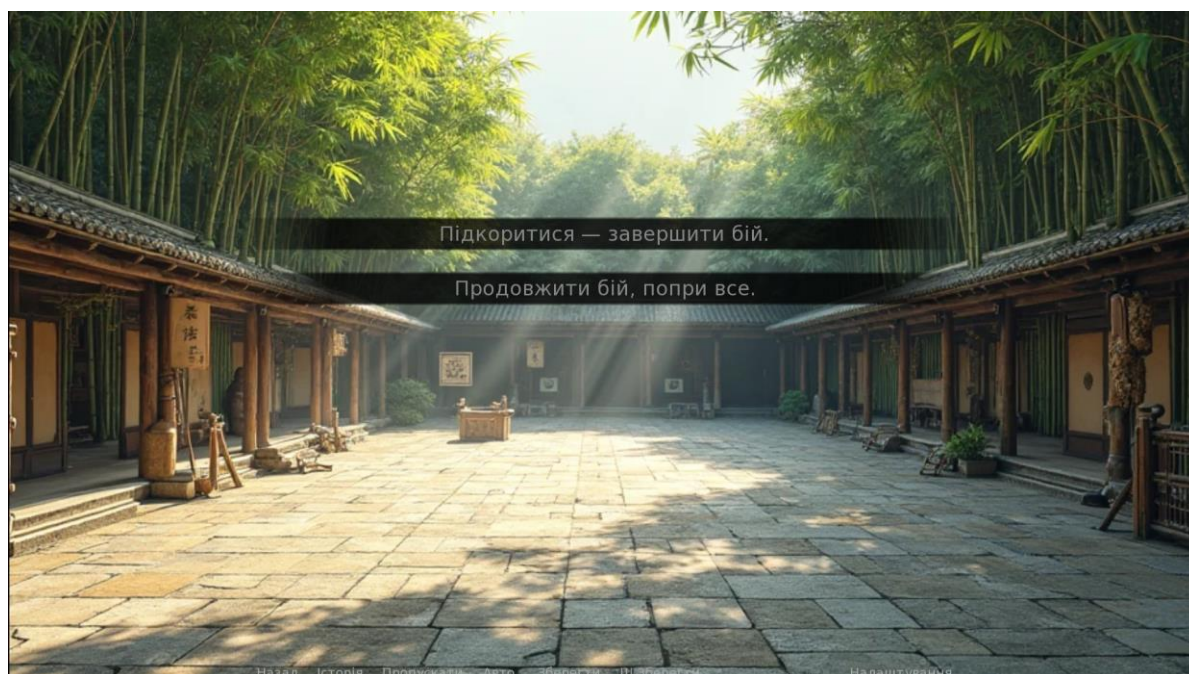


Рисунок 4.9 – Скріншот інтерфейсу вибору діалогів

Такі налаштування допомагають створити єдиний стиль для інтерфейсу, що посилює цілісність візуального образу гри і робить її більш професійною.

4.1.1 Параметри розташування інтерфейсних елементів

Крім зовнішнього вигляду, дуже важливим є розташування елементів на екрані – вони не повинні заважати перегляду сюжету, перекривати текст або бути важкодоступними(рис 4.10).

Лістинг коду налаштування позиціонування навігації, вибору, меню та сповіщень:

```
define gui.navigation_xpos = 60
define gui.skip_ypos = 15
define gui.notify_ypos = 68
define gui.choice_spacing = 33
define gui.navigation_spacing = 6
define gui.pref_spacing = 15
define gui.pref_button_spacing = 0
define gui.page_spacing = 0
define gui.slot_spacing = 15
define gui.main_menu_text_xalign = 1.0
```

– navigation_xpos = 60 – задає горизонтальне положення панелі навігації. Якщо зробити це значення більшим, панель зміститься праворуч, меншим – ліворуч. Важливо підібрати так, щоб кнопки не були надто далеко від очей гравця, але й не заважали перегляду.

– choice_spacing = 33 – відстань між варіантами вибору в діалогах. Добре підібраний інтервал підвищує читабельність і допомагає гравцю не плутати варіанти.

– main_menu_text_xalign = 1.0 – означає, що текст у головному меню вирівнюється по правому краю, що може підкреслювати стиль оформлення меню або слугувати елементом дизайну.

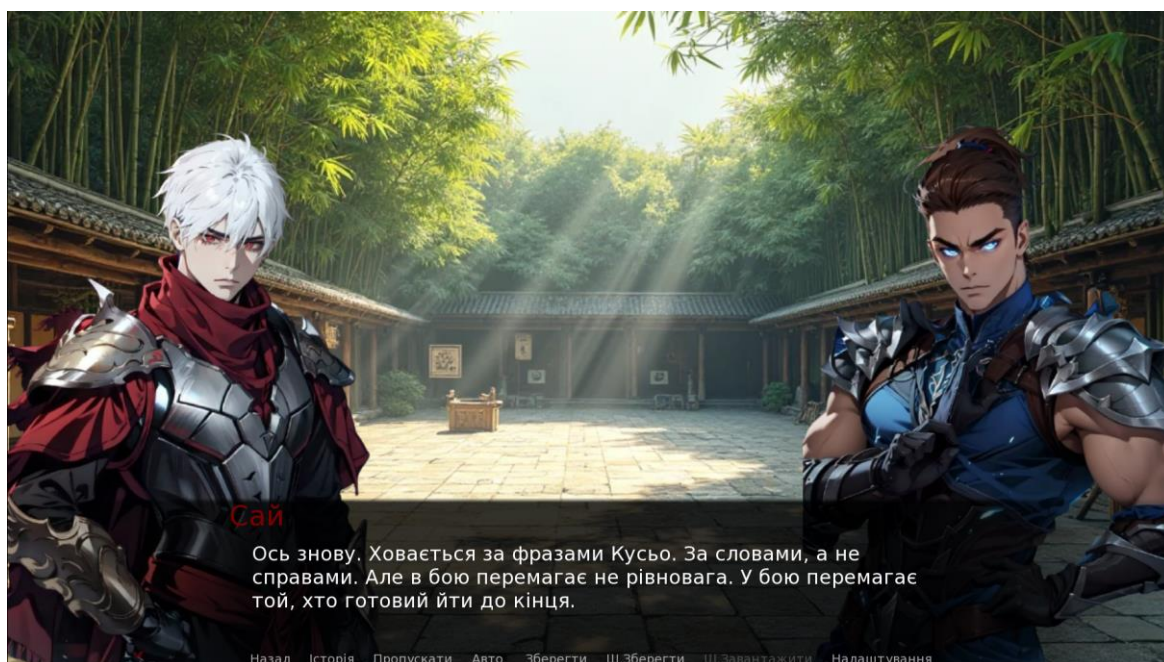


Рисунок 4.10 – Скріншот інтерфейсу гри

Ці налаштування дозволяють розробнику керувати просторовою організацією інтерфейсу, що особливо важливо для збереження балансу між зручністю та естетикою. Враховуй, що на мобільних пристроях це критично – кнопки мають бути достатньо великими і розташованими так, щоб їх легко можна було натиснути пальцем.

4.1.1 Визначення варіанту інтерфейсу "small"

Однією з сильних сторін Ren'Py є можливість створення різних варіантів оформлення інтерфейсу, які називаються "variants". Це дозволяє автоматично адаптувати гру під різні розміри екрану чи типи пристроїв.

Варіант small призначений для компактних екранів, наприклад, смартфонів або невеликих планшетів. У цьому режимі всі елементи інтерфейсу, включно з текстом і кнопками, будуть зменшені, щоб оптимально використовувати простір.

Лістинг коду інтерфейсу small:

```
@gui.variant
def small():
    gui.text_size = 45
    gui.name_text_size = 54
    gui.notify_text_size = 38
    gui.interface_text_size = 45
    gui.button_text_size = 45
    gui.label_text_size = 51
    gui.textbox_height = 360
    gui.name_xpos = 120
    gui.dialogue_xpos = 135
    gui.dialogue_width = 1650
    gui.slider_size = 54
    gui.choice_button_width = 1860
    gui.choice_button_text_size = 45
    gui.navigation_spacing = 30
    gui.pref_button_spacing = 15
    gui.history_height = 285
    gui.history_text_width = 1035
    gui.quick_button_text_size = 30
    gui.file_slot_cols = 2
    gui.file_slot_rows = 2
    gui.nvl_height = 255
    gui.nvl_name_width = 458
    gui.nvl_name_xpos = 488
    gui.nvl_text_width = 1373
    gui.nvl_text_xpos = 518
    gui.nvl_text_ypos = 8
    gui.nvl_thought_width = 1860
```

```
gui.nvl_thought_xpos = 30  
gui.nvl_button_width = 1860  
gui.nvl_button_xpos = 30
```

Варіант `small` є прикладом адаптивного інтерфейсу, що дозволяє забезпечити зручність гри на пристроях із невеликими екранами. У Ren'Py це реалізується шляхом визначення окремої функції, яка виконується, коли система виявляє малий розмір екрана або коли користувач примусово вибирає цей варіант.

Цей підхід особливо корисний для мобільних платформ, адже він не потребує створення окремої версії гри. Замість цього – використовується гнучке масштабування інтерфейсу відповідно до поточного середовища, що дозволяє розробнику підтримувати єдиний кодовий базис для різних платформ.

Висновки до розділу 4

У четвертому розділі здійснено детальний аналіз програмної реалізації ключових компонентів візуальної новели, що охоплює графічне оформлення, фонове середовище, а також звуковий супровід. Наведено приклади створення та інтеграції спрайтів персонажів із різними емоційними станами, підкреслено значущість художньої цілісності та відповідності стилістичній концепції проєкту. Процес розробки фонів продемонстрував важливість глибини сцени, м'яких колірних переходів та текстур для забезпечення атмосферності візуального простору. Також охарактеризовано етапи створення музичного супроводу за допомогою FL Studio, акцентовано на адаптації музичних тем до драматургії сюжету. Всі технічні засоби та прийоми спрямовано на досягнення естетичної гармонії, емоційної виразності та глибшого занурення користувача у світ гри.

ВИСНОВКИ

У межах виконання кваліфікаційної роботи реалізовано комплекс заходів, спрямованих на створення комп'ютерної гри в жанрі візуальної новели, що поєднує інтерактивні інтерфейси, персоналізований користувацький досвід та систему обміну повідомленнями в реальному часі. Всі поставлені завдання були виконані в повному обсязі, що дозволило досягти основної мети дослідження.

Зокрема, було проведено глибокий аналіз сучасних технологій, що застосовуються у розробці візуальних новел, включаючи системи адаптивного інтерфейсу, механізми прийняття рішень та сценарні структури. Результати аналізу стали основою для обґрунтованого вибору інструментів і підходів до реалізації проєкту.

Розроблено алгоритми побудови адаптивного інтерфейсу, здатного змінюватися залежно від виборів користувача, що дозволило значно підвищити рівень занурення в ігровий процес. Створена система обміну повідомленнями в реальному часі дозволяє моделювати живу комунікацію між персонажами, що надає грі додаткової динамічності та емоційної глибини.

Також розроблено сценарії з розгалуженням виборів, які впливають на перебіг сюжету, проте не змінюють загальну кінцівку, що забезпечує баланс між свободою гравця та цілісністю історії. Водночас створено художнє оформлення гри: спрайти персонажів, анімовані сцени, фонові ілюстрації та звуковий супровід, які гармонійно поєднуються із загальною стилістикою проєкту.

1. Ren'Py Documentation – The Visual Novel Engine. URL: <https://www.renpy.org/doc/html/>, (date of access: 04.04.2025).
2. TyranoBuilder Documentation – Visual Novel Maker. URL: <https://tyranobuilder.com/>, (date of access: 04.04.2025).
3. Python Documentation URL: https://learn.microsoft.com/ru-ru/training/browse/?utm_source=chatgpt.com, (date of access: 04.04.2025).
4. Visual Novel Maker Documentation – Official Help Guide. URL: <https://support.degica.com/en/support/solutions/24000021105>, (date of access: 04.04.2025).
5. Paint Tool SAI Documentation URL: <https://saipainttool.com/manual/about/>, (date of access: 04.04.2025).
6. Ink Documentation – Scripting Language for Interactive Fiction. URL: <https://www.inklestudios.com/ink/>, (date of access: 04.04.2025).
7. Twine Documentation – Interactive Fiction Tool. URL: <https://twinery.org/>, (date of access: 04.04.2025).
8. Team Salvato. (2017). *Doki Doki Literature Club!* [Visual novel]. Available on Steam. URL: https://store.steampowered.com/app/698780/Doki_Doki_Literature_Club/, (date of access: 04.04.2025).
9. Chiromaru Shikura. (2010). *Steins;Gate* [Visual novel]. Available on Steam. URL: <https://store.steampowered.com/app/412830/STEINSGATE/> (date of access: 04.05.2025).
10. Key. (2006). *Clannad* [Visual novel]. Available on Steam. URL: <https://store.steampowered.com/app/324160/CLANNAD/> (date of access: 04.05.2025).
11. FL Studio Documentation URL: <https://www.image-line.com/fl-studio-learning/fl-studio-online-manual/>, (date of access: 04.04.2025).
12. Godot Engine Documentation – Free and Open Source Game Engine. URL: <https://docs.godotengine.org/>, (date of access: 04.04.2025).

13. ChoiceScript Documentation – Choice-based Interactive Fiction. URL: <https://www.choiceofgames.com/make-your-own-games/choicescript-intro/>, (date of access: 04.04.2025)
14. Unity Documentation – Real-time development platform. URL: <https://docs.unity3d.com/>, (date of access: 04.04.2025).
15. GameMaker Studio 2 Documentation. URL: <https://manual.yoyogames.com/>, (date of access: 04.04.2025).
16. Shikura Chiyomaru. Steins;Gate Official Document. – 5pb, 2010. – 256 с.
17. Mangold M. What is a Visual Novel? – ResearchGate Publishing, 2019. – 112 с.
18. Rogers S. Level Up! The Guide to Great Video Game Design. – Wiley, 2014. – 552 p.
19. Key. Clannad Official Complete Guide. – Enterbrain, 2006. – 320 с.
20. Schell J. The Art of Game Design: A Book of Lenses. – CRC Press, 2019. – 600 p.
21. Nystrom R. Game Programming Patterns. – Genever Benning, 2014. – 354 p.

ДОДАТОК А

Лістинг коду

```
image master = im.Flip("images/Sprites/Master.png", horizontal="True")

define Сай = Character("Сай", what_text_markup=True, color="#800000")
define Шен = Character("Шен", color="#00008B")
define Кусьо = Character("Кусьо", color="#FFFAFA")
define Мандрівник = Character("Мандрівник", color="#FFFFFF")
define Ніндзя = Character("Ніндзя", color="#696969")
define Шепіт = Character("Шепіт", color="#696969")
define Невідомий = Character("Невідомий")

image training_daybg:
    "training_day"
    zoom 2.0

image training_nightbg:
    "training_night"
    zoom 2.0

image casketbg:
    "casket"
    zoom 2.0

image bamboo_forestbg:
    "bamboo_forest"
    zoom 2.0

image hallwaybg:
    "hallway"
    zoom 2.0

image lakebg:
    "lake"
    zoom 2.0

image bamboo_nightbg:
    "bamboo_night"
    zoom 2.0

image temple_entrancebg:
    "temple_entrance"
    zoom 2.0

image master1:
    "master"
    zoom 1.5

image sai:
    "sai not mask"
    zoom 1.5

image shen:
```



```
"shen not mask"
zoom 1.5

image traveler_s:
    "traveler"
    zoom 0.6

image shadow_ninja:
    "shadow ninja"
    zoom 1.2

image sai_mask:
    "sai mask"
    zoom 1.2

image shen_mask:
    "shen mask"
    zoom 1.2

label start:

label chapter1:

    window hide

    play music "/music/Japanese Zen Music - Japanese Flute Music For
Healing, Soothing, Meditation.mp3" fadein (2.0)

    scene training_daybg

    with dissolve

    show text "Тінь не народжується в темряві. Вона народжується там, де
світло відвернулося від воїна..."

    with dissolve
    pause 20

    hide text
    with fade

    show text "Десь у самому серці Іонії, в храмі, захищеному серед гір і
тиші, існує Орден Кінку – воїни, що стережуть рівновагу між світлом і тінню."

    with dissolve
    pause 20

    hide text

    show text "Серед них жив хлопчик. Він не знав свого минулого. Його
знайшли на краю попелястого лісу після однієї з численних воєн. Він був сам.
Голодний. Мовчазний."

    with dissolve
    pause 20
```

hide text

show text "Майстер Кусьо прихистив його. Дав йому ім'я, якого той не мав. Дав йому шлях, який ніхто не міг пройти замість нього. Його звали Сай."

with dissolve
pause 20

hide text

show text "З юності він був впертим і прагнув сили. Тренувався більше за інших. Шукав не визнання, а переваги."

with dissolve
pause 20

hide text

show text "Але поряд із ним завжди був Шен – син Майстра, спокійний і врівноважений, ідеальний учень... ідеальний у всьому. Вони зростали разом. Але разом із цим росло і те, що ховалося в серці Сая."

with dissolve
pause 20

hide text
with fade

stop music fadeout 2.0

play sound "/sound/Gong Sound Effect.mp3"

show text "{b}Глава 1: Відкритий конфлікт{/b}"

with dissolve
pause 5

hide text
with fade

window show

play music "/sound/Sword Fight Sound Effect.mp3"

"Звук дерева. Удари. Ухили. Сай атакує наполегливо, агресивно. Шен – точний і спокійний."

show sai at left
with dissolve

Сай "{i}Кожен рух – випробування.{/i}"

Сай "{i}Щоранку я прокидаюся раніше, лягаю пізніше.
Я ріжу своє тіло тренуваннями, щоб стати найкращим...{/i}"

Сай "{i}А він – просто усміхається. Завжди стриманий. Завжди
врівноважений.

І все ж саме він – спадкоємець.

Шен. Син Майстра.

Чому він? Чому не я?{/i}"

hide sai

with dissolve

"Сай різко атакує, але Шен ухиляється від удару й ставить свій меч до
його шиї."

show shen at right

with dissolve

Шен "Ти став швидшим. Але досі дозволяєш люті керувати собою."

show sai at left

with dissolve

Сай "Може, ти просто боїшся, що одного дня я буду сильнішим за тебе?"

Шен "Сила без рівноваги – це шлях до падіння, Сай."

Сай "Ось знову. Ховається за фразами Кусьо. За словами, а не справами.
Але в бою перемагає не рівновага. У бою перемагає той, хто готовий йти до
кінця."

hide sai

with dissolve

hide shen

with dissolve

stop music

window hide

menu:

"Підкоритися – завершити бій.":

\$ player_choice = "good"

jump fight_result

"Продовжити бій, попри все.":

\$ player_choice = "bad"

jump fight_result

label fight_result:

if player_choice == "good":

"Сай відходить на крок назад, опускаючи меч. Шен киває."

show sai at left

with dissolve

Сай "{i}Ти виграв. Але наступного разу... я не зупинюся.{/i}"

else:

play music "/sound/Sword Fight Sound Effect.mp3"

"(Сай із ривком кидається вперед, змушуючи Шена відступити. Бій
стає агресивнішим, але зрештою – нічия.)"

show shen at right

with dissolve

stop music

Шен "Це не змагання, Сай. Ми – брати по зброї. А не вороги."

hide shen

with dissolve

jump next

label next:

Кусьо. "За хвилину. Обоє стоять на краю майданчика. З'являється Майстер

show master1 at right

with dissolve

Кусьо "Гарно. Сила й техніка. Стриманість і рішучість. Ви вдвох – гордість Кін Ку."

"Сай опускає голову, стиснувши кулаки."

show sai at left

with dissolve

Сай "{i}Знову «ви вдвох». Він не бачить, скільки я віддав. Не бачить, як я рвуся вперед, щоб довести – я вартий більшого. Він ніколи не скаже: «Ти кращий, Сай.»{/i}"

Сай "{i}Ніколи.{/i}"

Кусьо "Завтра на світанку – продовжимо. Справжній шлях пізнається не в перемогах... а в терпінні."

hide master1

with dissolve

hide shen

with dissolve

"Кусьо йде. Шен мовчки киває Саю й теж іде. Сай залишається наодинці, дивиться в бік гір."

Сай "{i}Завтра і подивимось хто що скаже Майстер.{/i}"

hide sai

with dissolve

scene black

with dissolve

jump chapter2

label chapter2:

window hide

scene training_nightbg

with dissolve

```
play sound "/sound/Gong Sound Effect.mp3"
```

```
show text "{b}Глава 2: Таємниці Тіней{/b}"
```

```
with dissolve  
pause 5
```

```
hide text  
with fade
```

```
window show
```

"Ніч була глибокою й холодною. Місяць ховався за хмарами, лишаючи храм КінКу в майже повній темряві. В повітрі – тиша, в якій було чутно лише далекий подих вітру."

"Сай сидів у внутрішньому дворі, дивлячись на порожній тренувальний майданчик."

```
show sai at left  
with dissolve
```

Сай "{i}Стільки років навчання... але де межа? Що ще приховує цей храм? Я відчуваю – тут є щось більше, ніж нам дозволяють бачити.{/i}"

Сай "{i}Чому деякі двері завжди замкнені? Чому навіть наставники не говорять про це? Я не хочу порушити правил... Але чому вони мовчать? Що вони приховують?..{/i}"

```
hide sai
```

"Його погляд падає на темний прохід у західному крилі храму – той, куди не веде жоден з навчальних маршрутів. Він відчуває дивне тягнення. Не страх, а цікавість."

"Він піднімається. Повільно, майже несвідомо, йде туди, де його ще ніколи не було. Його кроки обережні, але впевнені. Стіни стають вужчими, підлога – старішою. Тут давно ніхто не ходив. Повітря важке, застійне, наче зберігає дихання минулих століть."

"Нарешті перед ним – масивні дерев'яні двері, обвиті темними візерунками, які ніби пульсують у напівтемряві. Вони не замкнені. Лише трохи прочинені, неначе запрошують... або застерігають."

```
play sound "/sound/Secret Pathway Door Open (Stone Wall) - Sound Effect  
for editing.mp3"
```

"Сай тягнеться рукою, торкається холодного дерева й повільно відчиняє двері ще трохи. Відкривається вузька щілина – і з неї війнуло прохолодою, мов з глибини печери."

```
scene hallwaybg  
with dissolve
```

"Темрява за дверима поглинає його, але очі швидко звикають. Зала не велика, проте кожна її стіна покрита давніми символами, ніби хтось залишав попередження... або молитви."

"У самому центрі, на п'єдесталі з чорного каменю, стоїть скринька. Вона оздоблена тонкою, майже живою різьбою: тіні воїнів, звивисті лінії, що переплітаються, мов серпанок."

"Вона виглядає мов витвір мистецтва, але... від неї виходить щось недобре. Сама її присутність тисне на груди."

"Це не просто предмет. Це – осередок сили. Сай відчуває, як його тіло реагує: тремтіння в руках, пульс у скронях. Його серце б'ється гучніше, ніж будь-коли."

scene casketbg
with dissolve

show sai at left
with dissolve

Сай "Це вона... Але ж я знаю, що всередині можуть бути заборонені знання. Те, що приховували не просто так. Може, я не готовий? Чи... не маю права"

hide sai
with dissolve

"Він завмирає, погляд прикований до різьбленої шкатулки, а всередині борються два голоси – один шепоче про обережність, інший – про силу."

"Він завмирає. Один крок – і він перетне межу. Але сумніви стискають груди."

window hide

menu:
"Спробувати піти":
jump cowardice
"Йти далі й відкрити двері":
jump determination

label cowardice:

window show

scene hallwaybg

with dissolve

"Сай відступає, серце б'ється швидше. Він повертається в коридор, але думки не дають спокою. Цікавість сильніша за страх."

scene casketbg

with dissolve

"Минає кілька секунд, і внутрішній голос Сая переконує його повернутися. Він знову підходить до дверей, цього разу – з рішучістю в очах."

jump next1

label determination:

window show

"Сай збирає всю свою волю, штовхає двері й заходить у заборонене приміщення."

jump next1

label next1:

"Сай обережно відкриває двері. У центрі залу, на п'єдесталі, стоїть давня розмальована скринька. Від неї йде зловісна, але приваблива енергія."

"Без жодної тіні сумніву Сай піднімає кришку. У ту ж мить тіні вириваються зі скриньки, мов живі істоти, й обплутують його розум і тіло."

Шепіт "Ти шукав нас. Ми – твої знання. Увійди в темряву – і сила стане твоєю."

show sai at left
with dissolve

Сай "AAAAAAAAAAAA!"

hide sai

scene black
with dissolve

jump chapter3

label chapter3:

window hide

scene training_daybg

play sound "/sound/Gong Sound Effect.mp3"

show text "{b}Глава 3: Випробування сили{/b}"

with dissolve
pause 5

hide text
with fade

window show

"Ранок освітив тренувальний майданчик м'яким світлом. У повітрі витало напруження – сьогодні Сай прийшов не просто на спаринг. Озброєний давно забутими знаннями, він був готовий кинути виклик Шену."

show sai at left

with dissolve

Сай "{i}Тепер я сильний. Я доведу всім, що гідний бути тут. Що саме я – майбутнє ордену.{/i}"

hide sai
with dissolve

"Бій почався. Удари були швидкі й точні, але тепер Сай відчував у собі нову міць, даровану тінями. Він перехитрив Шена й переміг."

"Шен упав на коліна, а Сай стояв над ним, серце билося від очікування."

show sai at left
with dissolve

Сай "Я переміг. Тепер вони повинні визнати мене..."

hide sai
with dissolve

"У цю мить двері храму розчинилися. Увійшов Майстер Кусьо, і його суворий погляд одразу впав на Сая."

show sai at left
with dissolve

show master1 at right
with dissolve

Кусьо "Саю, я дізнався про твої дії. Ти насмілюся використати заборонені знання – темні техніки, які ми відкинули заради рівноваги.»"

Сай "Але я зробив це заради вас, майстру, щоб ви мною пишалися."

Кусьо "Ти зрадив шлях, який я тобі дав. За це на тебе чекає вигнання."

hide sai
with dissolve
hide master1
with dissolve

window hide

menu:
 "Підкоритися й піти.":
 jump saiubmissiveness
 "Посперечатися з Кусьо.":
 jump hubris

label saiubmissiveness:

window show

"Сай схиляє голову, стримуючи гнів. Він повільно відходить у тінь, мовчки приймаючи вигнання. Але в його погляді – не покора, а обіцянка колись довести свою правоту."

jump chapter4

label hubris:

window show

"Сай залишається на місці. Його голос тремтить, але не від страху."

show sai at left
with dissolve

Сай "Я не намагався зрадити вчення... Я лише хотів, щоб ви мною пишалися. Я йшов цим шляхом заради Ордену. Заради вас."

show master1 at right
with dissolve

Кусьо "Я бачу, ким ти став, Саю. Це вже не мій учень. Це хтось інший. І якщо ти й далі підеш цим шляхом... нам більше не по дорозі."

hide sai
with dissolve
hide master1
with dissolve

"Сай мовчить. Його погляд ковзає по Кусьо, по Шену... Потім він розвертається й іде – мовчки, не озираючись."

scene black
with dissolve

jump chapter4

label chapter4:

scene bamboo_forestbg

window hide

play sound "/sound/Gong Sound Effect.mp3"

show text "{b}Глава 4: Скитання{/b}"

with dissolve
pause 5

hide text
with fade

window show

scene bamboo_forestbg

with dissolve

"Бамбуковий ліс. Після довгих годин шляху Сай виходить до тихого озера, схованого між деревами. Вода спокійна, віддзеркалює вечірнє небо."

scene lakebg

with dissolve

show sai at left

with dissolve

Сай "Інколи... природа мовчить так само, як і я. Без слів. Без відповідей. Але в цій тиші щось живе."

"Він дістає саморобну ліску, знаходить гілку і сідає на березі."

Сай "Тишу складніше зловити, ніж рибу... але обидві речі потрібні, щоб вижити."

window hide

scene bamboo_nightbg

with dissolve

window show

"Пізніше. Сутінки. Він розпалює невелике вогнище біля води. Чистить рибу, нанизує її на гілку й смажить над полум'ям."

"Озеро відображає зорі. Легкий вітер шелестить траву. Вогонь потіскує."

show sai at left

with dissolve

Сай "Не пам'ятаю, коли востаннє їв спокійно. Без тривоги. Без крові на руках."

hide sai

with dissolve

"Сай їсть, потім відкидається на спину й дивиться на небо. Очі заплющуються, дихання вирівнюється."

scene black

with dissolve

Невідомий "Допоможіть..."

"Сай різко прокидається, хапає зброю. Вогонь майже згас."

scene bamboo_forestbg

with dissolve

Невідомий "Будь ласка... хто-небудь..."

"Він встає, прямує на звук. Пробирається крізь зарості."

"Галявина. Повалене дерево. Біля коріння лежить виснажений мандрівник."

show sai at left
with dissolve

Сай "Ти кликав на допомогу."

show traveler_s at right
with dissolve

Мандрівник "Я... мене поранили. Хотів дістатися села, але... не встиг.
Ти... хто ти?..."

window hide

menu:
"Розповісти все":

 window show

Сай "Я навчався в Кінку. Я знав, що таке честь... поки не побачив, як її використовують проти тих, хто не боїться сили."

Мандрівник "Ти з ордену? Неймовірно... Дозволь іти з тобою. Мені нікуди більше."

 "Приховати минуле.":

 window show

Сай "Я мандрівник. Мої кроки не ведуть назад."

Мандрівник "Тоді ми схожі. Дозволь просто йти поруч. Мені не потрібно багато."

window hide

menu:
"Взяти мандрівника з собою.":

 window show

Сай "Ти вижив у лісі – і не помер. Це вже щось."

Мандрівник "Дякую. Ти не пошкодуєш."

 "Відмовити.":

 window show

Мандрівник "Іноді самотність – гірше за смерть. Будь ласка."

window hide

pause(1.0)

window show

Сай "Гаразд. Але запам'ятай – довіру треба заслужити."

Мандрівник "Я її здобуду."

scene black

jump chapter5

label final:

show text "Після цього Сай поклав в основу свого вчення лише один принцип..."

with dissolve

pause 5

hide text

with fade

scene training_daybg

with dissolve

show sai_mask at left

with dissolve

Сай "Удосконалюй техніку та вбивай усіх ніндзя, які відмовляються прийняти Тінь. Рівновага – брехня. Справжній ніндзя йде нашим шляхом."

scene black

with dissolve

show text "{b}КІНЕЦЬ{/b}{w=5.0}{b}?{/b} "

with dissolve

pause 5

hide text

with fade

show text "Дякую за проходження моєї гри. XD"

with dissolve

pause 5

hide text

with fade

return