

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

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри _____ Є. О. Давиденко
підпис

«__» _____ 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА
Ігровий застосунок у жанрі візуальної новели на базі Ren'Py

Спеціальність «Інженерія програмного забезпечення»

121 – КРБ.1 – 408.22011011

Здобувач освіти

_____ М. О. Крик
підпис

«__» _____ 2024 р.

Керівник д-р тех. наук, професор

_____ А. В. Швед
підпис

«__» _____ 2024 р.

Консультант канд. техн. наук, доцент

_____ А. О. Алексєєва
підпис

«__» _____ 2024 р.

Миколаїв – 2024

Завдання на виконання кваліфікаційної роботи

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

ЗАТВЕРДЖУЮ

Зав. кафедри _____ Є. О. Давиденко

«_____» _____ 2024 р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи бакалавра

Видано здобувачу освіти групи 408 факультету комп'ютерних наук

Крику Максиму Олександровичу

(прізвище, ім'я, по батькові здобувача освіти)

1. Тема кваліфікаційної роботи

Ігровий застосунок у жанрі візуальної новели на базі Ren'Py

Затверджена наказом по ЧНУ від «22» грудня 2023 р. № 269

2. Строк представлення кваліфікаційної роботи «_____» _____ 2024 р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні

Очікуваним результатом є техно-демо версія гри на ігровому рушію Ren'Py

4. Перелік питань, що підлягають розробці:

- 1) аналіз існуючих застосунків для роботи з візуальними новелами;
- 2) моделювання концепції для застосунку;
- 3) створення графічних та аудіо матеріалів для гри;
- 4) впровадження механізмів вибору та нелінійного розвитку сюжету;
- 5) програмна реалізація ігрового застосунку;
- 6) розробка програмного забезпечення ігрового застосунку;
- 7) тестування та оцінка його ефективності.

5. Перелік графічних матеріалів:

презентація

6. Завдання до спеціальної частини

Аналіз охорони праці на робочих місцях фахівців з інформаційних технологій

7. Консультанти:

Консультант	Кафедра (організація)	Частина роботи
Алексеева А.О.	Кафедра екології	Спеціальна частина з охорони праці

Керівник роботи _____ д-р техн. наук, професор Швед Алена Володимирівна _____
(посада, прізвище, ім'я, по батькові)

(підпис)

Завдання прийнято до виконання _____ Крик Максим Олександрович _____
(прізвище, ім'я, по батькові здобувача освіти)

(підпис)

Дата видачі завдання «22» грудня 2023 р.

Календарний план виконання кваліфікаційної роботи

КАЛЕНДАРНИЙ ПЛАН виконання кваліфікаційної роботи

Тема: Ігровий застосунок у жанрі візуальної новели на базі Ren'Py

№	Найменування роботи	Початок	Закінчення	Примітки
1	Розробка та затвердження завдання на виконання КРБ	07.01.2023	07.01.2023	виконано
2	Огляд відеоматеріалів за темою роботи	07.01.2023	08.01.2023	виконано
3	Складання календарного плану КРБ	02.05.2024	02.05.2024	виконано
4	Аналіз предметної області	02.05.2024	02.05.2024	виконано
5	Розробка проектних рішень	02.05.2024	02.05.2024	виконано
6	Моделювання та конструювання ПЗ	03.05.2024	03.05.2024	виконано
7	Кодування, тестування та апробація розробленого ПЗ, аналіз результатів тестування	01.06.2024	03.06.2024	виконано
8	Розробка спеціальної частини з охорони праці	10.06.2024	12.06.2024	виконано
9	Відгук керівника КРБ	07.06.2024	07.06.2024	виконано
10	Оформлення КРБ та презентації	10.06.2024	12.06.2024	виконано
11	Попередній захист	03.06.2024	05.06.2024	виконано
12	Рецензування	14.06.2024	17.06.2024	виконано
13	Завершення оформлення КРБ та презентації	12.06.2024	14.06.2024	виконано
14	Захист кваліфікаційної роботи	24.06.2024	26.06.2024	виконано

Розробив здобувач освіти Крик Максим Олександрович

(прізвище, ім'я, по батькові)

(підпис)

«__» _____ 20__ р.

Керівник роботи д-р тех. наук, професор Швед Альона Володимирівна

(посада, прізвище, ім'я, по батькові)

(підпис)

«__» _____ 20__ р.

АНОТАЦІЯ

до кваліфікаційної роботи бакалавра

«Ігровий застосунок у жанрі візуальної новели на базі Ren'Py»

Здобувач освіти 408 гр.: Крик Максим Олександрович

Керівник: д-р тех. наук, професор Швед Альона Володимирівна

Актуальність теми кваліфікаційної роботи бакалавра зумовлена в контексті сучасних технологічних тенденцій та розвитку ігрової індустрії. Такий жанр дозволяє майже кожному реалізуватись не маючи видатних навичок програмування. Створення таких ігрових застосунків сприяє розвитку творчих та технічних навичок розробників, а також забезпечує користувачам можливість зануритися в різноманітні світи та історії.

Кваліфікаційна робота присвячена розробці ігрового застосунку в жанрі візуальної новели на базі Ren'Py, у результаті якої має бути готовий перший епізод сюжету. У цьому епізоді мають бути представлені ключові механіки гри, які користувач буде використовувати впродовж подальшого її проходження.

Об'єкт роботи: процес створення ігрового застосунку в жанрі візуальної новели.

Предмет роботи: технології та засоби розробки ігрового застосунку у жанрі візуальної новели.

Мета: розробка ігрового застосунку в жанрі візуальної новели на базі кросплатформенного рушія Ren'Py.

Кваліфікаційна робота бакалавра складається з вступу, чотирьох розділів, висновків та переліку джерел посилань.

У вступі викладені унікальність обраної теми, об'єкт і предмет дослідження, а також основні задачі, які необхідно вирішити для досягнення поставленої мети. А також короткий опис програмного забезпечення, на якому буде розроблений ігровий застосунок.

У першому розділі присвячений аналізу предметної області, будуть включені огляд існуючих ігрових застосунків в жанрі візуальної новели та порівняння їх особливостей, візуальних, сюжетних особливостей між собою.

У другому розділі описується процес моделювання системи згідно з зазначеними вимогами шляхом створення діаграм, які відображають різні аспекти роботи системи та їх взаємодію між собою.

У третьому розділі проводиться аналіз програмного забезпечення, яке було обрано для розробки ігрового застосунку. Будуть описані його особливості та можливості. Розгляд обраної мови реалізації та порівняння її з альтернативними варіантами. Розробка мокапів ігрового інтерфейсу гравця.

У четвертому розділі описується процес реалізації самого проекту та побудова користувацького інтерфейсу спираючись на розроблені мокапи. Реалізація тих чи інших ігрових механік за допомогою мови реалізації та можливостей самого ігрового рушія.

У висновках проведено аналіз виконаної роботи в кожному з розділів та підсумки розробки ігрового застосунку. Також будуть розглянуті можливі баги та недоліки при розробці та запуску ігрового застосунку.

Кваліфікаційна робота містить 60 сторінки, включаючи 4 розділи, 35 ілюстрацій, 12 таблиць та перелік із 16 джерел посилань.

Ключові слова: ігровий застосунок, візуальна новела, Ren'Py, інтерактивний досвід, розробка гри, Python.

ABSTRACT

to the qualification work of the bachelor

«Game application in the genre of visual novel based on Ren'Py»

Student 408 gr.: Kryk Maksym Oleksandrovysh

Head: Dr. Tech. of Sciences, Professor Shved A.V.

The relevance of the topic of the bachelor's qualification work is determined in the context of modern technological trends and the development of the gaming industry. This genre allows almost anyone to realize themselves without having outstanding programming skills. The creation of such game applications promotes the development of creative and technical skills of developers, and also provides users with the opportunity to immerse themselves in various worlds and stories.

The qualification work is devoted to the development of a game application in the genre of a visual novel based on Ren'Py, as a result of which the first episode of the plot should be ready. This episode should introduce the key mechanics of the game, which the user will use during the further passage of the game.

Object of work: the process of creating a game application in the visual novel genre.

The subject of the work: technologies and tools for developing a game application in the visual novel genre.

Goal: development of a game application in the visual novel genre based on the cross-platform Ren'Py engine.

The bachelor's thesis consists of an introduction, four chapters, conclusions and a list of reference sources.

The introduction describes the uniqueness of the chosen topic, the object and subject of research, as well as the main problems that must be solved to achieve the goal. As well as a brief description of the software on which the game application will be developed.

The first section is dedicated to the analysis of the subject area, an overview of existing game applications in the visual novel genre and a comparison of their features, visual, and plot features will be included.

The second section describes the process of modeling the system according to the specified requirements by creating diagrams that reflect various aspects of the system's operation and their interaction with each other.

The third section analyzes the software that was chosen for the development of the game application. Its features and capabilities will be described. Consideration of the chosen implementation language and its comparison with alternative options. Development of mockups of the player's game interface.

The fourth chapter describes the process of implementing the project itself and building the user interface based on the developed mockups. Implementation of certain game mechanics using the implementation language and capabilities of the game engine itself.

In the conclusions, an analysis of the work performed in each of the sections and the results of the game application development were carried out. Possible bugs and shortcomings in the development and launch of the game application will also be considered.

The thesis contains 60 pages, including 4 chapters, 35 figures, 12 tables and a list of 16 reference sources.

Keywords: game application, visual novel, Ren'Py, interactive experience, game development, Python.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	3
ВСТУП.....	4
1 АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ.....	5
1.1. Візуальна новела як ігровий жанр.....	5
1.2. Аналіз популярних ігор у жанрі візуальної новели.....	6
1.3. Аналіз програмного забезпечення, що розробляється.....	11
1.4. Специфікація вимог до розробляемого програмного забезпечення комп'ютерної гри.....	13
Висновки до розділу 1.....	16
2 МОДЕЛЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІГРОВОГО ЗАСТОСУНКУ.....	17
2.1. Варіанти використання системи.....	17
2.2. Діаграма взаємодії.....	19
2.3. Діаграма діяльності.....	20
2.4. Розробка мокапів інтерфейсу ігрового застосунку.....	22
Висновки до розділу 2.....	26
3 ВИБІР ЗАСОБІВ РОЗРОБКИ ІГРОВОГО ЗАСТОСУНКУ.....	27
3.1. Порівняння аналогів середовищ розробки ігрового застосунку.....	27
3.2. Вибір середовища розробки та мови програмування.....	33
3.3. Вибір редактора коду.....	36
Висновки до розділу 3.....	39
4 ПРОГРАМНА РЕАЛІЗАЦІЯ ІГРОВОГО ЗАСТОСУНКУ.....	40
4.1 Реалізація ігрового інтерфейсу.....	40
4.2 Створення системи збережень.....	45
4.3 Система вибору відповіді користувача.....	47
Висновки до розділу 4.....	50
ВИСНОВКИ.....	51
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	52

ПЕРЕЛІК СКОРОЧЕНЬ

ПЗ – програмне забезпечення

КРБ – кваліфікаційна робота бакалавра

ВН – візуальна новела

ІГ – ігровий застосунок

ІР – Ігровий рушій

UML – unified modeling language

UI – user interface

IDE – integrated Development Environment

ВСТУП

Унікальність жанру візуальної новели полягає в тому, що він дозволяє створювати розповіді з великим впливом гравця на сюжетний розвиток.

Ren'Py — це популярний інструмент для створення таких ігор, який надає розробникам доступ до широкого спектру можливостей, включаючи інтуїтивно зрозумілий скриптовий синтаксис, підтримку зображень, аудіо та відео, а також засоби для створення нелінійних сюжетів та варіантів вибору. А також при наявності достатньої кількості часу можна реалізувати багато інших механік, які дозволяють розширити звичний досвід гри.

Робота над застосунком у жанрі візуальної новели включає кілька ключових етапів: розробку сценарію, дизайн персонажів та фону, інтеграцію аудіо та візуальних ефектів, а також програмування різних гілок сюжету та виборів гравця. Але для даної роботи буде створена лише техно-демо версія без загального сюжету, для демонстрації певних механік та як вони будуть використовуватись.

Об'єкт роботи: процес створення ігрового застосунку в жанрі візуальної новели.

Предмет роботи: технології та засоби розробки ігрового застосунку у жанрі візуальної новели.

Метою: розробка ігрового застосунку в жанрі візуальної новели на базі кросплатформенного рушія Ren'Py.

Для досягнення мети необхідно вирішити такі завдання:

- 1) аналіз існуючих застосунків для роботи з візуальними новелами;
- 2) моделювання концепції для застосунку;
- 3) створення графічних та аудіо матеріалів для гри;
- 4) впровадження механізмів вибору та нелінійного розвитку сюжету;
- 5) програмна реалізація ігрового застосунку;
- 6) тестування застосунку та оцінка його ефективності;
- 7) проведення оптимізації та забезпечення якості роботи гри.

1 АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ

1.1. Візуальна новела як ігровий жанр

Візуальна новела — жанр відеоігор, в якому історія подається гравцеві з допомогою статичних зображень, текстових блоків і звуків.

Ступінь інтерактивності в таких іграх зазвичай низький, і гравцю лише зрідка потрібно зробити певний вибір, зокрема — вибрати варіант відповіді в діалозі. Персонажі цих ігор зазвичай виконані в стилі аніме, який, як і візуальні новели, виник в Японії. 70 % ігор для ПК, які видаються в Японії, відносяться саме до цього жанру[1].

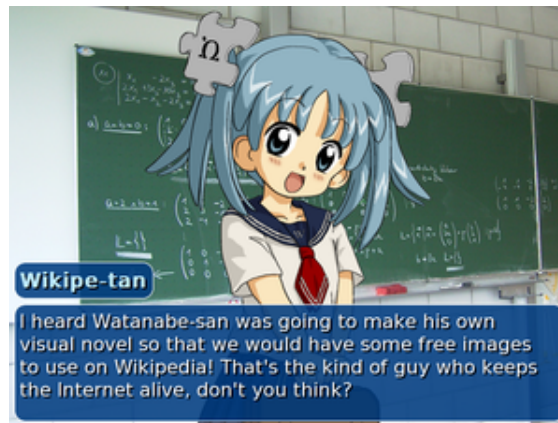


Рисунок 1.1 – Типове зображення гри у жанрі візуальна новела

Не зважаючи на простоту з зовнішньої, технічної та ігрової точки зору – цей жанр має свою популярність, а також збільшуються методи та стилі їх реалізації. В одних випадках візуальна новела має класичний свій вид й мінімум інтерактиву, в інших – гравцеві надається більше свободи та можливостей впливати на оточення або хід сюжету. Деякі люди бажають поринути у цікавий сюжет, але не мають бажання читати цілу книгу; а також гравці мають бажання прочитати цікавий сюжет, але без зайвих зусиль чи важких пазлів. Жанр «Візуальна новела» у першу чергу орієнтований на ці два групи.

Окрім повноцінних ігор цього жанру, елементи візуальної новели до сих пір використовуються у інших відеоігор із іншим жанром. Наприклад у паузах між рівнями або за проходження пазлів, гравцю дається можливість поговорити з персонажем, а також обрати варіант відповіді.

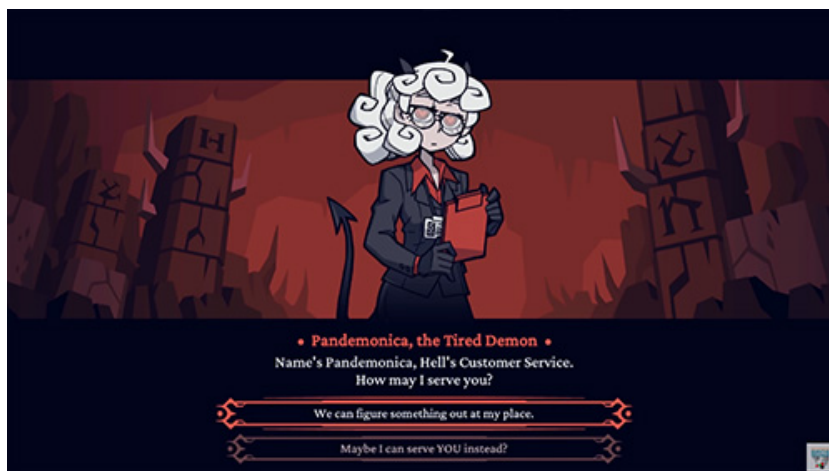


Рисунок 1.2 – Вибір відповіді у «Helltaker»

На прикладі рисунку 1.2 можна побачити елементи візуальної новели у грі “Helltaker” головний жанр якої – головоломка. Після кожного рівня можна поговорити із персонажем й обрати варіант відповіді, від якої буде залежати результат успішного проходження рівня.

1.2. Аналіз популярних ігор у жанрі візуальної новели

Незважаючи на те, що відеоігри у жанрі візуальна новела дуже схожі між собою, окрім сюжету є й інші елементи, якими автори привертають увагу до своїх творінь. Персонажі, стиль оформлення зовнішнього вигляду, додаткові механіки, а також вплив на сюжет. Однією з сильних сторін візуальної новели є сюжет, однак для більш обширного досвіду гравця можна додати нові механіки, які не притаманні жанру, але дозволяють більше зануритися у події та прийняти в них участь. У деяких випадках, ці механіки можуть бути занадто складними для спеціальних рушіїв, що вимагає переходу на інший ігровий рушій.

Таблиця 1.1 – Опис відеогри Phoenix Wright: Ace Attorney

Рік виходу	2001
Розробник	Capcom
Рушій	Unity
Платформи	Game Boy Advance, Nintendo DS, Windows, Wii, iOS

Phoenix Wright: Ace Attorney — це візуальна новела та пригодницька гра, розроблена та видана Capcom. Він був випущений у 2001 році для Game Boy Advance в Японії та був перенесений на кілька платформ. Це перший запис у серії Ace Attorney; Було створено кілька сиквелів і допоміжних програм, а цю гру було перенесено та оновлено для комп'ютерів, ігрових консолей і мобільних пристроїв[2].



Рисунок 1.3 – Phoenix Wright: Ace Attorney

Головною особливістю цієї гри є засідання суду, де гравець окрім опитування потерпілих, може займатися пошуком доказів та розкрити таємниці злочинів, щоб знайти винуватого. Крім базових елементів, як діалог із персонажами, гравець може знаходити недоліки у свідченнях та показувати докази, а також обирати варіант тлумачення того як розвивались ті чи інші події із участю цього доказу.

Таблиця 1.2 – Опис відеогри Tiny Bunny (Зайчик)

Рік виходу	2020
Розробник	Saikono, RUZURA Interactive, Saikono Games
Рушій	Ren'Py
Платформи	Windows

«Зайчик» (англ. Tiny Bunny) – комп'ютерна гра в жанрах візуальної новели і survival horror, створена за мотивами однойменного оповідання інтернет-письменника Дмитра Мордаса. У ході розвитку подій гравцю пропонують прочитувати текст знизу екрану, іноді даючи вибори, які впливають на кінцівку. Всього в грі планується від трьох до десяти кінцівок, кожна з яких кардинально відрізнятиметься один від одного. Усі діалоги озвучені[3].



Рисунок 1.4 – Tiny Bunny (Зайчик)

«Зайчик» як представник більш класичних візуальних новел виділяється за рахунок сеттингу, у якому відбуваються події й живуть персонажі, сюжету, стилю оформлення, а також стилю подачі сюжету. Усі репліки персонажів, окрім авторської мови, озвучені та переведені на інші мови. Стиль зовнішнього вигляду ілюстрацій новели досить не традиційний

для цього жанру, оскільки оформлений у чорно-білому стилі та більш реалістичним зображенням персонажів, з деякими змінами у аніме стилі. Із інтерактивних елементів по мінімуму, гравцю у певні моменти сюжету дається можливість зробити вибір, який потім впливає на розвиток подій у поточному та подальших епізодів.

Таблиця 1.3 – Опис відеогри Doki Doki Literature Club!

Рік виходу	2017
Розробник	Team Salvato
Рушій	Ren'Py
Платформи	Windows, Linux, OS X

Doki Doki Literature Club! (укр. Літературний клуб «Доки-докі!»; аббревіатура — DDLC) — психологічний хоррор та візуальна новела, розроблений командою «Team Salvato». Випущена 22 вересня 2017 року для Windows, Mac OS і Linux, а пізніше, 6 жовтня, і для Steam. Гра має загалом лінійний сюжет, альтернативні сцени та кінцівки, що залежать від вибору гравця[4].



Рисунок 1.5 – Doki Doki Literature Club!

На перший огляд хоч дана гра й вважається звичайною візуальною новелою, але більш к середині гри відбуваються неочікувані сюжетні повороти, а також візуальні обманки імітуючи комп'ютерні глюки. Також для більш несподіваного ефекту, гра зчитує назву персонального комп'ютеру користувача, щоб потім внести його як звертання напряду до гравця, а не до персонажу якого він відіграє. Крім цього як приклад, у ході проходження сюжету, гра можете сама змінювати, або видаляти свої файли, а у деяких випадках гравцю доведеться це робити самому для отримання секретних кінцівок.

Таблиця 1.4 – Опис відеогри Danganronpa: Trigger Happy Havoc

Рік виходу	2010
Розробник	Spike Chunsoft
Рушій	PhyreEngine
Платформи	PlayStation Portable, iOS, Android, PlayStation Vita, PC

Danganronpa: Trigger Happy Havoc – гра в жанрі візуальний роман, розроблена та опублікована компанією Spike Chunsoft, пізніше була адаптована в анімі та мангу. Назва Danganronpa походить від слів «куля» (яп. 弾丸) і «спростування, доказ» (яп. 論破), але так як слово написано катаканою, воно залишається неперекладним. Під час розслідування вбивства, коли учасники академії спростовують факти та припущення інших учнів, змальовується куля, яка розбиває невірні припущення[5].

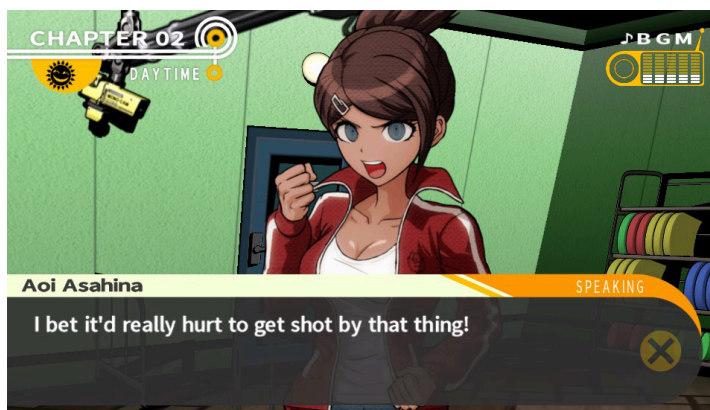


Рисунок 1.6 – Danganronpa: Trigger Happy Havoc

У даному приміру, елементи візуальної новели займають значний об'єм гри, але доповнюється іншими механіками, такими як: знаходження нестикувань у свідченнях, становлення цілої картини подій, а також розпізнавання ім'я підозрюваного. На відміну від Phoenix Wright: Ace Attorney, у даному випадку допити проходять у більш абстрактному стилі та окрім показу фізичних доказів, у хід йдуть документи та психологічні прийоми для вилучення брехні у підозрюваних. Також між етапами суду, йдуть розмови з персонажами, покращення взаємовідносин між ними, а також пошук доказів, якщо стався злочин.

1.3. Аналіз програмного забезпечення, що розробляється

Ігровий застосунок у жанрі візуальної новели розробляється для подання гравцю сюжет більш простим методом через візуальні елементи та взаємодію між ними. Гравець може читати діалог між персонажами, відповідати їм обираючи варіанти відповіді, а також взаємодіяти із інтерактивним оточенням. Попри цього, будуть додані деякі нові механіки, що розширять звичне сприйняття гри. Також гравець повинен мати можливість зберегти свій прогрес та завантажити його у будь-який момент, що надає надійності застосунку. Ці та інші задачі були виведені у таблицю 1.5, де більш розгорнуто було розглянуто вимоги.

Таблиця 1.5 – Опис системи, що розробляється

Основні задачі	<ol style="list-style-type: none">1) розробка складних сюжетів: Платформа надає можливість створювати складні гілки сюжету, засновані на виборі гравця, що впливає на розвиток історії;2) взаємодія з гравцем: Ren'Py дозволяє створювати механіку взаємодії, включаючи діалоги, вибір, інвентар та інші інтерактивні елементи;3) налаштування користувацького інтерфейсу: Платформа пропонує різноманітні засоби для створення та налаштування користувацького інтерфейсу, включаючи стилі, меню, кнопки тощо;4) підтримка мультимедіа: Ren'Py підтримує роботу з різними форматами графіки, анімації, аудіо та відео.
Сценарії роботи ігрового застосунку	<ol style="list-style-type: none">1) створення нової гри: Розробник створює нову гру, визначаючи основні елементи, такі як сюжет, персонажі, графіка та звук;2) редагування існуючої гри: Розробник може редагувати наявний проект, оновлюючи діалоги, графіку, звук чи механіку гри;3) перевірка гри: Під час розробки розробник може запускати гру для тестування механіки, сюжету та інших елементів;4) експорт гри: Після завершення розробки гри можна експортувати у формат для розповсюдження на різних платформах (Windows, macOS, Linux, Android, iOS).

Кінець таблиці 1.5

Засоби апаратної та програмної реалізації	<ol style="list-style-type: none">1) апаратна реалізація: Ren'Py працює на більшості сучасних апаратних платформ, включаючи настільні комп'ютери та мобільні пристрої;2) програмна реалізація: Розробка ігор у Ren'Py відбувається за допомогою програмування на Python, що надає гнучкість у розробці та налаштуванні гри.
Вихідні дані	<ol style="list-style-type: none">1) графічні ресурси: Зображення, ілюстрації та анімації для персонажів, фонів та інших графічних елементів;2) текстові ресурси: Діалоги, описи та інші текстові елементи для історії;3) мультимедійні ресурси: Аудіо (музика, звукові ефекти) та відео для використання у грі;4) сценарії: Структура сюжетних гілок, взаємодій та діалогів, написана на Python або спеціальній мові сценаріїв Ren'Py.

1.4. Специфікація вимог до розробляемого програмного забезпечення комп'ютерної гри

Призначення системи

Ігровий застосунок у жанрі візуальної новели на базі Ren'Py призначений для створення інтерактивного та захоплюючого досвіду для користувачів, поєднуючи елементи оповідання, графіки, музики та вибору користувача для управління сюжетним розвитком.

Погодження, ухвалені в програмній документації

Було вирішено використовувати ігровий рушій Ren'Py для розробки ігрового застосунку. Вона пропонує широкий спектр можливостей для

створення візуальних романів, включаючи підтримку графіки, звуку, тексту, анімації та взаємодії з користувачем. Також за роки свого існування вона має широку аудиторію із багатьма плагінами та відеоматеріалами.

Межі проєкту

Кінцевий термін завершення розробки застосунку передбачається на 25 червня 2024 року. Якщо нових виправлень чи перенесень не буде.

Сфера застосування

Застосунок буде спрямований на користувачів, які цікавляться візуальними новелами та інтерактивними історіями. Ігровий застосунок може бути використан для розважальних цілей.

Характеристики користувачів

Користувачі можуть використовувати застосунок на різних пристроях, таких як персональні комп'ютери, ноутбуки або планшети. Доступ до мережі Інтернет не є обов'язковим, оскільки застосунок може працювати офлайн.

Загальна структура та склад системи

Система містить наступні компоненти:

- 1) ігровий клієнт: Застосунок, розроблений на базі Ren'Py, що включає графіку, звук, текст та інтерактивні елементи для взаємодії з користувачем;
- 2) база даних: Використовується для збереження даних про прогрес користувачів та інших ігрових елементів такі як зроблений вибір який буде мати вплив на подальший розвиток.

Обмеження

Суворих обмежень не передбачається, оскільки ігровий застосунок може запуститись на майже кожному пристрої та не потребує доступу в Інтернет.

Функції застосунку

- 1) **взаємодія з користувачем:** Користувач здійснює вибір, що впливає на розвиток сюжету та результати подій. А також на інтерактивні об'єкти;
- 2) **збереження прогресу:** Система дозволяє користувачам зберігати прогрес у грі та продовжувати з тієї ж точки пізніше;
- 3) **графіка та звук:** Підтримка різних мультимедійних елементів та вивід їх на екран при певних умовах.

Джерела та зміст вхідної інформації

- 1) **вхідна інформація:** Дії користувача, його вибір та взаємодія з інтерфейсом;
- 2) **вихідна інформація:** Розвиток сюжету та візуальні/аудіо зміни в застосунку відповідно до вибору користувача.

Нормативно-довідкова інформація

Вимоги до даного пункту відсутні.

Вимоги до технічного забезпечення

Вимоги відсутні, оскільки ВН може запуститись на майже кожному пристрої.

Вимоги до ігрового застосунку

- 1) **архітектура:** Розробка застосунку на базі Ren'Py з інтеграцією тексту, графіки, звуку та сценарію;
- 2) **мови програмування:** Python, які є основною мовою для розробки на платформі Ren'Py.

Вимоги до зовнішніх інтерфейсів

- 1) **інтерфейс користувача:** Інтуїтивний, візуально привабливий та простий у використанні. Без великої кількості деталей та зайвого тексту;
- 2) **програмний інтерфейс:** Підтримка різноманітних платформ за допомогою Ren'Py.

Властивості ігрового застосунку

- 1) **доступність:** Застосунок буде доступний для широкого кола користувачів завдяки сумісності з різними платформами;
- 2) **безпека:** Оскільки застосунок не має доступу до Інтернету, то дані користувача будуть у безпеці;
- 3) **продуктивність:** Оптимізований код забезпечує швидку та стабільну роботу застосунку;
- 4) **простота використання:** Зрозумілий та інтуїтивний інтерфейс дозволить користувачам швидко ознайомитись з ігровим застосунком;
- 5) **функціональність:** Гнучкість у розробці та можливість розширення функціональності за потреби.

Висновки до розділу 1

У першому розділі проведено аналіз жанру «візуальна новела» та його популярних представників із різноманітними стилями, як візуального так й ігрового виконання. Відеоігри такого жанру не обмежені одним чи двома ігровими рушіями, але можуть бути створені на будь-якому з них та потребують мінімум потужностей платформи, на якій вони запускаються. Також було виявлено, що окрім повноцінних відеоігор цього жанру, розробники також використовують елементи цього жанру для розбавлення або доповнення своїх власних проєктів.

2 МОДЕЛЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІГРОВОГО ЗАСТОСУНКУ

2.1. Варіанти використання системи

Короткий Use Case

Користувач бажає почати тестування техно-демо версії ігрового застосунку. Він запускає ІЗ і у головному меню розпочинає нову гру. Після цього він обирає та тестує один чи декілька механік гри та здійснює вибір. Після тесту, він зберігає свій прогрес та виходить у головне меню, де може почати усе спочатку або завантажитись із збереженої точки прогресу.

Поверхневий Use Case

Головний сценарій (успішний):

Користувач бажає перевірити працездатність збережень ігрового процесу. Він з головного меню починає нову гру та робить декілька дій, що залишають свій слід. Починає діалог із персонажем та обирає варіант відповіді. Після цього він зберігає свій прогрес та виходить у головне меню. З головного меню він відкриває перелік збережень та обирає потрібне йому. Прогрес збережений ним був відновлений та усі зроблені ним дії та вибір були також відновлені.

Альтернативні сценарії:

- 1) користувач при завантаженні свого прогресу не зміг його успішно відновити, що означає його прогрес був втрачений або поламаний;
- 2) користувач при збереженні свого прогресу вимкнув гру чи пристрій, у ході чого процес збереження перервався;
- 3) при виконанні подальших дій користувачем, персонаж не запам'ятав зроблений ним вибір;
- 4) при спробі взаємодії із інтерактивними об'єктами нічого не відбувається.

Повний Use Case

Таблиця 2.1 – Повний Use Case

Scope	Ігровий застосунок у жанрі візуальна новела на базі Ren'Py
Level	Мета користувача (user-goal)
Primary Actor	Користувач
Preconditions	Відсутні
Stakeholders and interests	Користувач: зацікавлений у сприйнятті сюжету у більш легкий та доступній манері, з можливістю впливати на розвиток сюжету.
Main Success Scenario	<ol style="list-style-type: none"> 1) користувач заходить у гру; 2) обирає “Нова гра”; 3) починає діалог із персонажем; 4) обирає варіант відповіді; 5) зберігає свій прогрес.
Result	При повторному заході у головне меню та завантаженні збереження – успішне завантаження прогресу з точки збереження гравця
Extensions	<ol style="list-style-type: none"> 1) користувач при завантаженні свого прогресу не зміг його успішно відновити, що означає його прогрес був втрачений або поламаний; 2) користувач при збереженні свого прогресу вимкнув гру чи пристрій, у ході чого процес збереження перервався; 3) при виконанні подальших дій користувачем, персонаж не запам’ятав зроблений ним вибір;

Кінець таблиці 2.1

	4) при спробі взаємодії із інтерактивними об'єктами нічого не відбувається; 5) після вибору варіанту відповіді, нічого не відбувається.
Special Requirements	Будь який пристрій на базі Windows, Mac OS X, Ubuntu, Android, IOS.
Frequency of Occurrence	Кожний раз як користувач запустить ігровий застосунок.

У доповнення Use Case будуть створенні діаграми взаємодій та діяльності, для більш простої демонстрації роботи ігрового застосунку.

2.2. Діаграма взаємодії

Unified Modeling Language (UML) – це набір правил та визначень для специфікації системи програмного забезпечення, що керується Object Management Group. Дана система визначень надає набір графічних елементів для моделювання частин системи. В наступних підрозділах даного розділу використовуються UML-діаграми для моделювання розроблюваної системи[6].

У процесі аналізу розроблюваного ігрового застосунку необхідно продумувати варіанти взаємодії із користувачем та реакції на його вибір. Для демонстрації таких взаємодій була створена діаграма взаємодії (послідовності). Так, наприклад, на діаграмі 2.1 демонструється взаємодія користувача із ігровим застосунком при зберіганні його процесу.

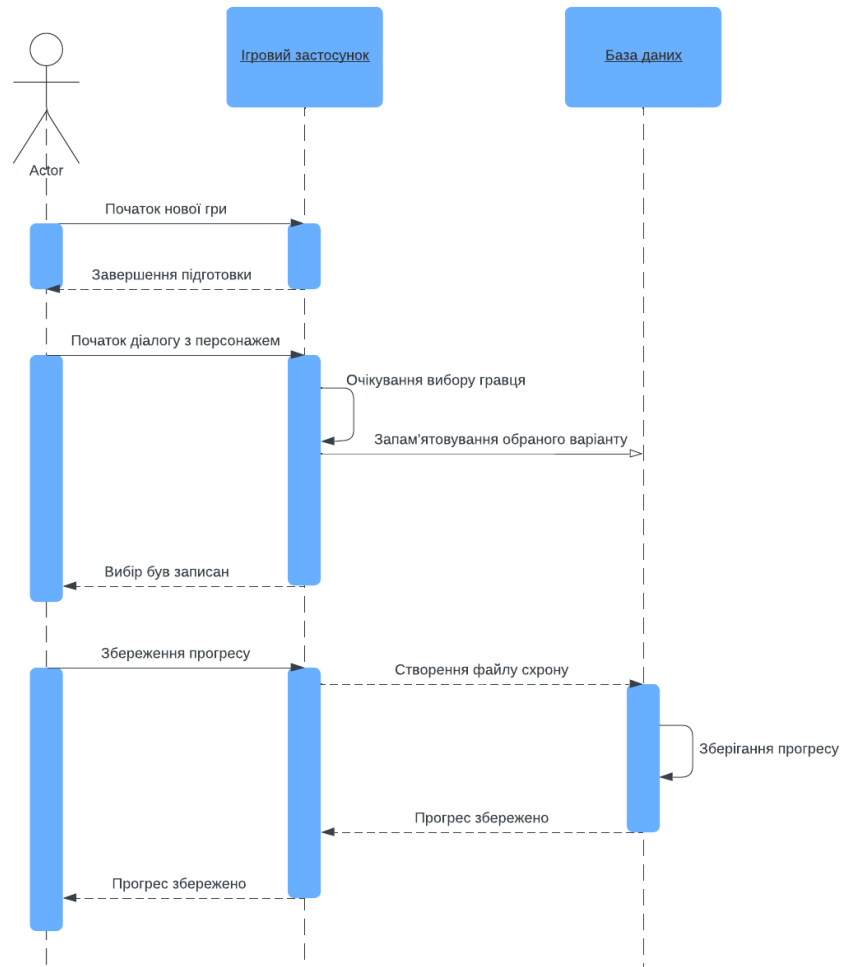


Рисунок 2.1 – Діаграма взаємодії користувача із схроном прогресу

На діаграмі 2.1 було продемонстровано успішний сценарій створення файлу схрону користувача куди був записан його поточний прогрес. При завантаженні цього файлу, прогрес буде відновлений з цієї точки. При необхідності, користувач може переписати файл на новий з іншим прогресом та прийнятими рішеннями.

2.3. Діаграма діяльності

Діаграма діяльності (англ. activity diagram) — в UML та SysML, візуальне представлення графу діяльностей. Граф діяльностей є різновидом графу станів скінченного автомату, вершинами якого є певні дії, а переходи відбуваються по завершенню дій.[7]

У даному випадку, сценарій взаємодії користувача із застосунком, буде продемонстровано на діаграмі діяльності. У ній ж буде продемонстровано працю застосунку виходячи з вибору користувача.

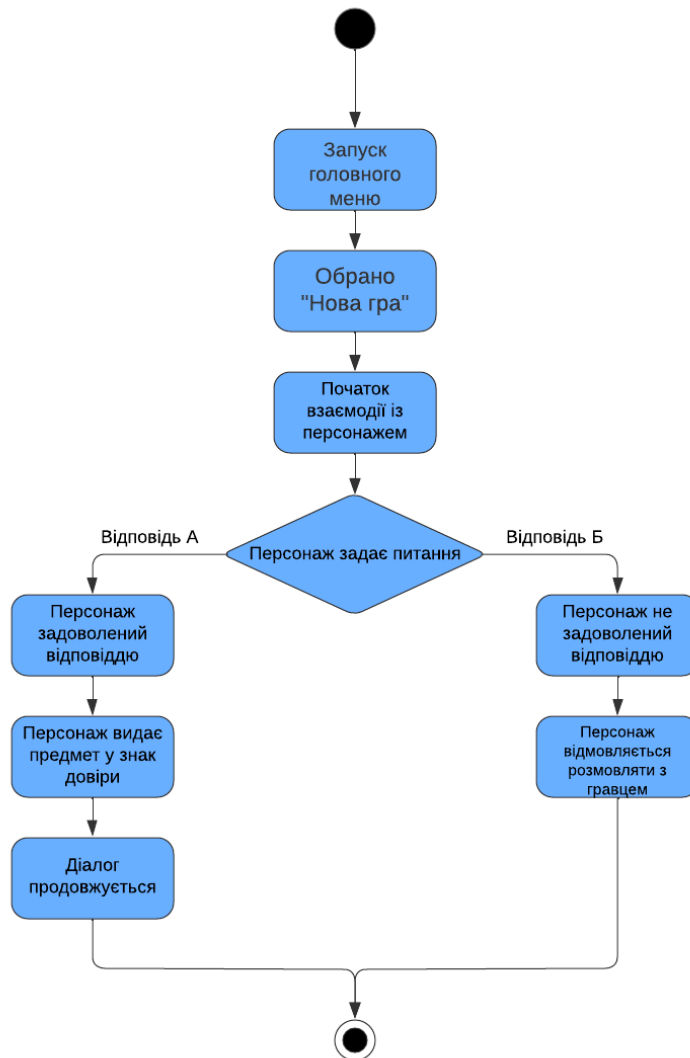


Рисунок 2.2 – Діаграма діяльності при обиранні відповіді користувачем

Виходячи з діаграми 2.2 можна зробити висновок, що при невдалій відповіді гравець нічого не отримує і персонаж відмовиться більше розмовляти із ним. Але при вдалій відповіді, гравець отримує певний предмет та діалог із персонажем продовжується. Також відповідь користувача буде запам'ятована у поточному процесі гри та збережний при наступному збереженні прогресу у файлі схрону. У подальшому сюжеті буде здійснена перевірка на цю відповідь і в залежності який варіант гравець обрав у

минулому – буде змінена поведінка інших персонажів, розвиток сюжету або декілька речень у наступному діалозі.

2.4. Розробка мокапів інтерфейсу ігрового застосунку

Перед розробкою ігрового застосунку, необхідно створити мокапи ігрового інтерфейсу гравця, на який буде зроблений опір у подальшій розробці. Інтерфейс гравця має бути мінімальним на деталі та простим у орієнтуванні для кожного гравця, а також бажано зробити його адаптивним, запобігаючи небажаних помилок при розгортанні гри на моніторі більшому чи меншому за розмірами, або на випадок ручного змінення розміру вікна гравцем.

На прикладі (рис. 2.3) головне меню буде мати мінімум деталей, що зроблено для демонстрації гравцю заднього зображення та без його перекривання розміщення елементів інтерфейсу. Серед елементів інтерфейсу головного меню є: головна назва ігрового застосунку, примітка головного меню, кнопка початку нової гри з початку, кнопка завантаження точки ігрового процесу з якої гравець зберігся, кнопка меню налаштування гри, кнопка для відображення титрів для перегляду розробника гри та кнопка виходу із ігрового застосунку. Усі кнопки розположені згідно звичного та зручного розташування елементів головного меню ігрового інтерфейсу, який є стандартом для більшості ігрових студій, а саме порядок зверху-вниз: нова гра, завантаження гри, налаштування, титри, вихід. Завдяки такому розташуванню гравцеві буде зручно та інтуїтивно зрозуміло орієнтуватися у головному меню.

Також для більш привабливого зовнішнього вигляду головного меню, можна додати анімації кнопок для переливання кольорами, а також параллакс ефект, що додасть динаміці головного меню.

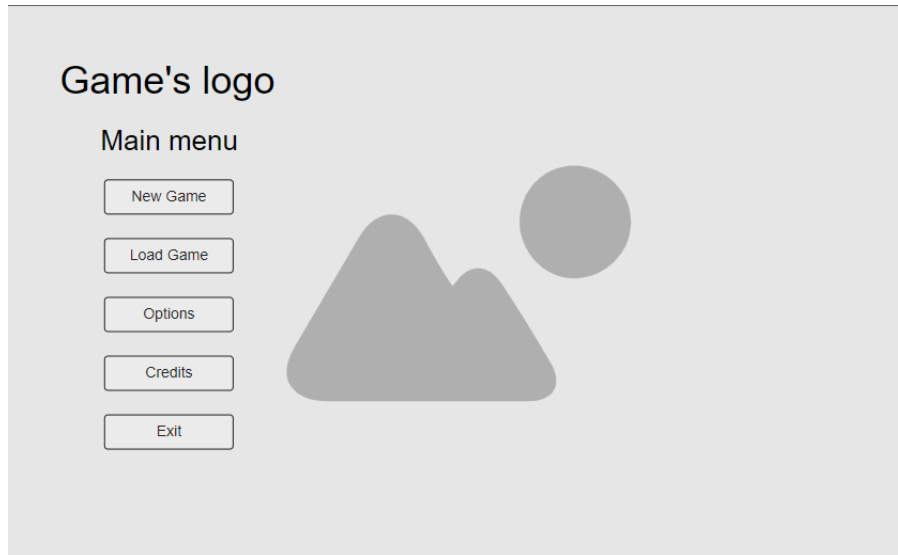


Рисунок 2.3 – Мокап головного меню

Рисунок 2.4 показує зовнішній вигляд інтерфейсу більшу частину гри. На ньому зображено вікно, де гравець читає текст, а вище ім'я персонажа, що говорить свою репліку. На задньому плані будуть показані локація, на якій відбуваються події, а також сам персонаж (або декілька персонажів) із ким гравець веде бесіду. Нижче вікна із текстом будуть розташовані кнопки: авто, яка буде у автоматичному режимі переключати текст, коли персонаж закінчить своє речення; лог, для перегляду реплік, що персонажі вже сказали до цього; приховати, що вимикає відображення кнопок та вікна тексту, для розгляду заднього плану та персонажа; пропустити, переключає на наступне вікно тексту; меню, відкриває меню, у якому гравець може зберегти та завантажити свій прогрес, а також відкрити налаштування чи вийти у головне меню. Таким чином у нижньої панелі будуть виведені функції, які гравець буде використовувати частіше всього під час проходження сюжету гри. Це зроблено для більшої зручності, аби гравцеві не доводилось розгортати та згорнути меню кожен раз, як він забажає зберегти свій прогрес або подивитися лог діалогів. За наявності великої кількості функцій, щоб уникнути переповнення нижньої частини екрану, решту менш часто використовуваних функцій можна перенести до меню.



Рисунок 2.4 – Мокап інтерфейсу ігрового процесу

Рисунок 2.5 показує як буде виглядати інтерфейс, якщо гравцю буде надана можливість зробити вибір. У цьому випадку приховується вікно тексту та ім'я персонажа, а замість них показуються кнопки із назвою варіанту відповіді, яку гравець має надати.

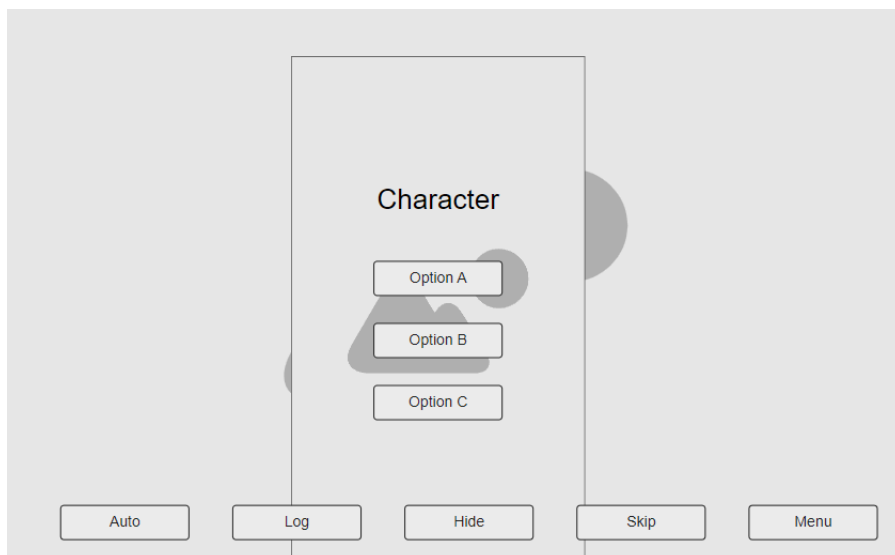


Рисунок 2.5 – Мокап вибору відповіді

Рисунок 2.6 демонструє меню збережень гравця, де у нього є 6 слотів у яких він може зберегти, завантажити чи видалити свій прогрес. У правому вікні буде відображено зображення сцени, на якій гравець зберіг свій прогрес, 2024 р.

ім'я збереження яке генерується випадково, але гравець може його змінити, а також реальна дата та час створення збереження. Також, для більшої зручності буде показано локацію, на якій гравець зберіг прогрес, а також можливість перезаписати збереження у тий ж слот.

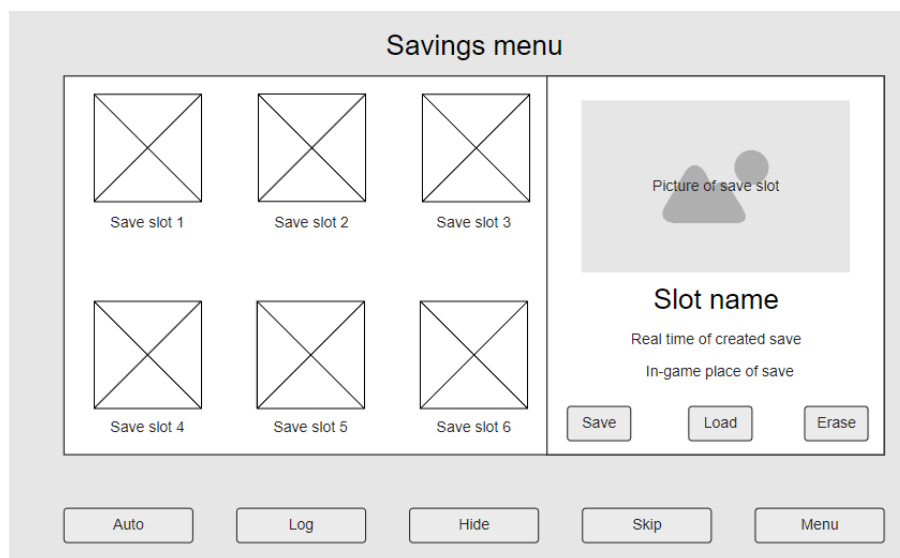


Рисунок 2.6 – Мокап меню збережень гравця

На рисунку 2.7 показан простий мокап екрану поразки гравця, якщо він вийшов на погану кінцівку. У такому випадку гравцю надається можливість завантажити свій прогрес, або вийти у головне меню.

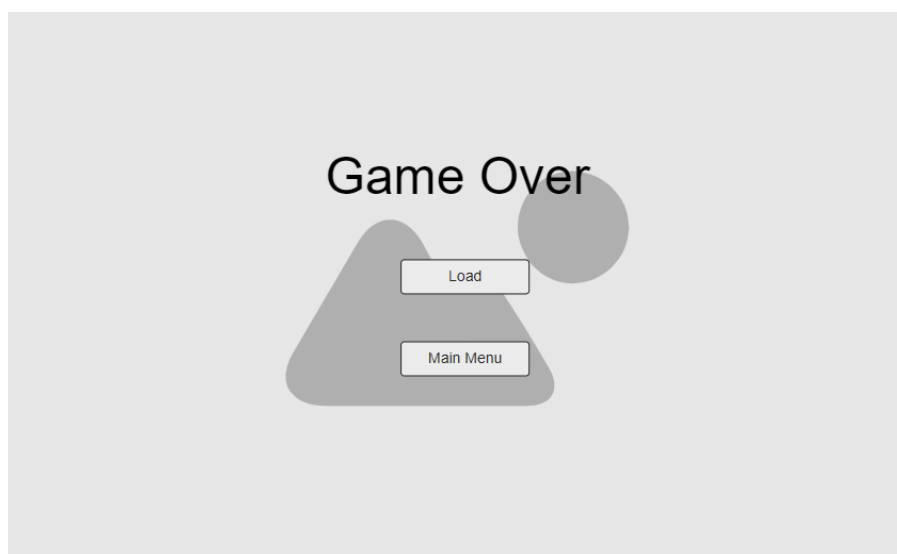


Рисунок 2.7 – Мокап екрану поразки гравця

З представлених ключових мокапів, можна скласти загальне уявлення, яким буде приблизний ігровий інтерфейс та які функції буде мати.

Висновки до розділу 2

Другий розділ присвячено моделюванню та проєктуванню поведінки ігрового застосунку та можливі вдалі та хибні сценарії його роботи. Для більш простої демонстрації цього процесу й були розроблені діаграми взаємодії та діяльності. За допомогою цих діаграм не лише розробнику, але й користувачу буде зрозуміло як працює ігровий застосунок й які наслідки матимуть ті чи інші дії користувача, а також як вони будуть впливати на подальше проходження сюжету. Було спроектовано дизайн інтерфейсу основних екранів ігрового застосунку та розроблені мокапи їх зовнішнього вигляду.

3 ВИБІР ЗАСОБІВ РОЗРОБКИ ІГРОВОГО ЗАСТОСУНКУ

3.1. Порівняння аналогів середовищ розробки ігрового застосунку

Перед тим як розпочати розробку проєкту, необхідно впевнитися які є аналоги серед застосунків для створення ВН та їх відмінності. На основі проведеного аналізу можна виділити переваги та недоліки того чи іншого аналогу, а також чи відповідають вони вимогам. Серед аналогів Ren'Py були виділені такі застосунки:

- 1) Twine (табл. 3.1 та рис. 3.1);
- 2) Adventure Game Studio (табл. 3.2 та рис. 3.2);
- 3) Visual Novel Maker (табл. 3.3 та рис 3.3).

Twine

Twine — це платформа з відкритим кодом, яка дозволяє розробникам створювати інтерактивні історії та візуальні новели. Вона пропонує зручний візуальний інтерфейс, що дозволяє розробникам створювати складні гілки сюжету та діалоги.[1] Однак у Twine обмежені можливості для графічних ефектів і анімацій, що може бути недоліком порівняно з Ren'Py. Перевагою Twine є його здатність експортувати ігри у веб-формат, що робить його більш доступним для користувачів.

Таблиця 3.1 – Аналіз застосунку Twine

Виробник	Chris Klimas
Архітектура	Ігровий рушій
Мова реалізації	v2.*, JavaScript v1.*, Python

Кінець таблиці 3.1

Основні функції	<ol style="list-style-type: none">1) створення інтерактивних історій та візуальних новел;2) зручний візуальний інтерфейс, який дозволяє створювати складні гілки сюжету та діалоги;3) підтримка експорту ігор у веб-формат, що робить їх доступними для користувачів через інтернет.
Переваги	<ol style="list-style-type: none">1) відкритий код та активне співтовариство розробників;2) простий у використанні, особливо для новачків;3) легкість експорту історій у різні формати, що робить їх доступними для різних платформ.
Недоліки	<ol style="list-style-type: none">1) обмежені можливості для графічних ефектів та анімацій порівняно з іншими платформами;2) може бути не придатний для розробки ігор із складною ігровою логікою.
Веб сайт	https://twinery.org

У випадку із Twine, єдиною мовою реалізації був Python, але потім у версії v2.* розробник додав можливість реалізації через JavaScript, що розширює можливості ігрового рушія для реалізації проекту. Якщо варіант розробки через Python більш підходящий для платформ настільних комп'ютерів та ноутбуків, то варіант за допомогою JavaScript підходить для розробки під веб платформи, що значно допомагає оптимізувати ігровий застосунок під різні браузері.

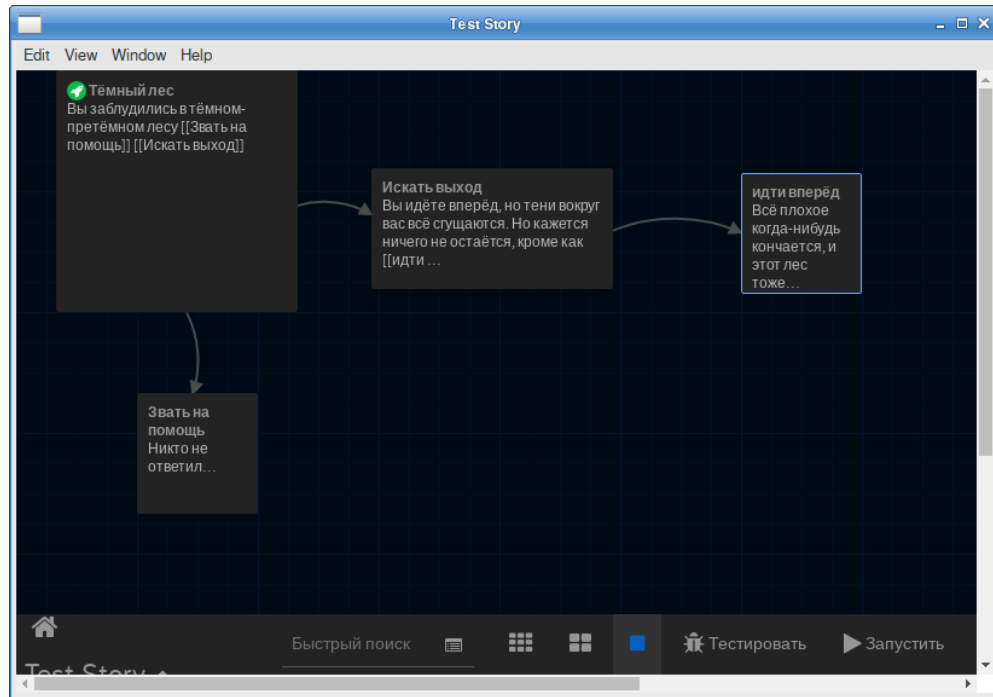


Рисунок 3.1 – Вигляд інтерфейсу застосунку Twine

На цьому рисунку (рис. 3.1) видно інтерфейс ПЗ, що має подобу візуальної демонстрації зв'язку фраз та дій між собою.

Adventure Game Studio

Adventure Game Studio (AGS) — це платформа для створення пригодницьких ігор, але також підтримує створення візуальних новел. AGS має власний скриптовий мову для програмування ігрової логіки та надає розробникам більше контролю над процесом розробки.[2] Переваги AGS включають потужні інструменти для створення графіки та анімацій, але розробка ігор на AGS може бути більш складною через необхідність знань у програмуванні.

Таблиця 3.2 – Аналіз застосунку Adventure Game Studio

Виробник	Крис Джонс
Архітектура	Ігровий рушій

Кінець таблиці 3.2

Мова реалізації	C++
Основні функції	<ol style="list-style-type: none">1) створення пригодницьких ігор та візуальних новел;2) потужні інструменти для створення графіки, анімацій та звуку;3) власна скриптова мова програмування для створення ігрової логіки.
Переваги	<ol style="list-style-type: none">1) велика кількість інструментів для розробки ігор з багатим сюжетом та детальним світом;2) велика спільнота розробників, яка надає підтримку та ресурси.
Недоліки	<ol style="list-style-type: none">1) більша складність використання, особливо для новачків, через необхідність знань у програмуванні;2) орієнтація на пригодницькі ігри може бути обмеженням для інших жанрів.
Веб сайт	https://www.adventuregamestudio.co.uk

На прикладі Adventure Game Studio можна побачити, що подібні ігрові застосунки можуть бути написані на C++, що може покращити працездатність ігрового застосунку через більш тонку роботу із пам'яттю. Але це може ускладнити процес розробки, оскільки код буде більш складно упорядкувати та читати.

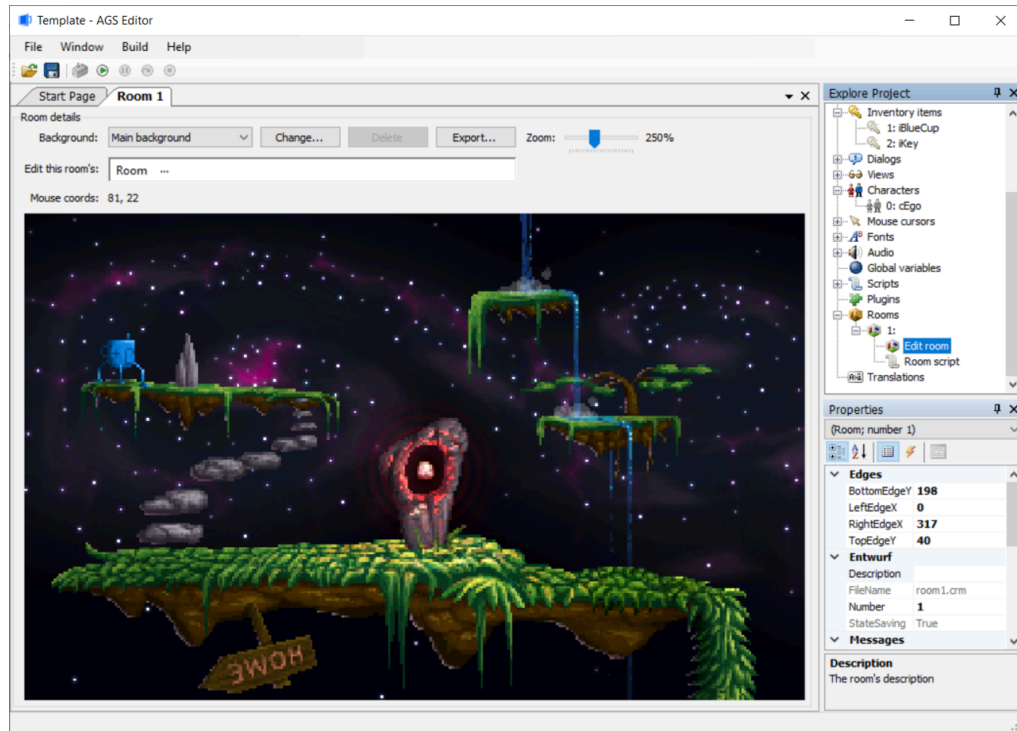


Рисунок 3.2 – Видгляд інтерфейсу застосунку Adventure Game Studio

На цьому рисунку (рис. 3.2) представлений інтерфейс ПЗ, який у даному випадку застарілий і у деяких аспектах програє конкурентам.

Visual Novel Maker

Visual Novel Maker — це комерційний інструмент для створення візуальних новел. Він пропонує широкий спектр інструментів для розробки, включаючи шаблони для створення сюжетних гілок, діалогів, анімацій та музичного супроводу.[3] Одна з переваг Visual Novel Maker полягає в тому, що він підтримує експорт ігор для різних платформ, включаючи комп'ютери та мобільні пристрої. Однак комерційний характер платформи може бути обмежувальним для розробників, які шукають безкоштовні рішення.

Таблиця 3.3 – Аналіз застосунку Visual Novel Maker

Виробник	Degica Co., Ltd
Архітектура	Ігровий рушій

Кінець таблиці 3.3

Мова реалізації	CoffeeScript, JavaScript
Основні функції	<ol style="list-style-type: none">1) створення візуальних новел з використанням різноманітних інструментів для розробки;2) шаблони для створення сюжетних гілок, діалогів, анімацій та музичного супроводу;3) підтримка експорту ігор для різних платформ, включаючи комп'ютери та мобільні пристрої.
Переваги	<ol style="list-style-type: none">1) широкий спектр інструментів для розробки візуальних новел;2) легкість інтеграції різних елементів (графіка, звук, текст) у гру;3) підтримка багатьох платформ, що розширює охоплення аудиторії.
Недоліки	<ol style="list-style-type: none">1) комерційний характер платформи може бути обмежувальним для розробників, які шукають безкоштовні рішення;2) може бути складним для новачків, особливо через різноманітність можливостей та інструментів.
Веб сайт	https://visualnovelmaker.com

У випадку Visual Novel Maker більш зрозумілий для нового користувача інтерфейс, багато зручних функцій таких як «Автопереклад» на інші мови. Але ігровий застосунок значно обмежений у редагуванні та додаванні нових

механік, що робить даний рушій підходящим лише для простих візуальних новел, які не планують додавання нових ігрових механік.

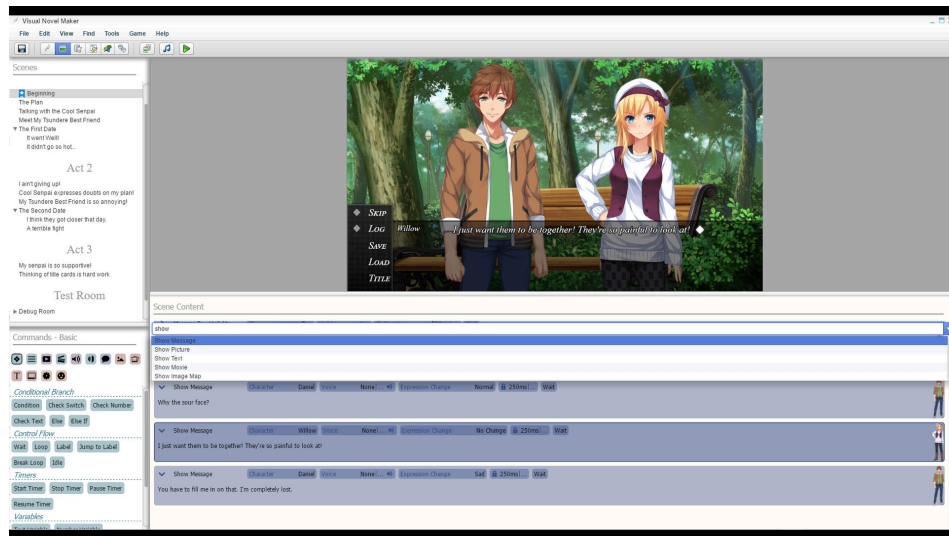


Рисунок 3.3 – Вигляд інтерфейсу застосунку Visual Novel Maker

У випадку аналогу на цьому рисунку (рис. 1.3), інтерфейс набагато приємніший та інтуїтивно зрозумілий. Але функціонал значно обмежений та ціна продукту занадто висока у порівнянні із безкоштовними аналогами із більшою кількістю можливостей.

3.2. Вибір середовища розробки та мови програмування

Серед усіх аналогів, був обран ігровий рушій Ren'Py, який відомий своєю популярністю серед інді-розробників візуальних новел. Більшість обирає саме його через простоту та широкий простір для реалізації різноманітних механік, які не притаманні для класичного жанру візуальної новели, а також розбавляють досвід гравця, надаючи йому більше можливостей для взаємодії із ігровим застосунком. Також, Ren'Py виділяється простим інтерфейсом та зрозумілою довідкою користувача, що допомагають легше орієнтуватися у середовищі розробки.

У нижче приведеній таблиці 3.4 був проведений аналіз даного програмного забезпечення, для виявлення його можливостей та виділення

його, як підходящого середовища розробки для ігрового застосунку. Головна його перевага – це широкий спектр можливостей при безкоштовній ліцензії.

Таблиця 3.4 – Аналіз ігрового рушія Ren'Py

Розробник	Ren'Py розроблений та підтримується спільнотою розробників
Архітектура	Ігровий рушій
Мова програмування	Python
Можливості	<ol style="list-style-type: none">1) створення розгалужених історій;2) системи збереження файлів;3) відкату до попередніх точок в історії;4) різних переходів між сценами;5) додавання нестандартних механік;6) внесення DLC (додатковий контент).
Платформи	Windows, Linux, macOS, Android, iOS та браузерів із HTML 5
Переваги	<ol style="list-style-type: none">1) Ren'Py безкоштовний;2) крос-платформний ;3) має велику спільноту, яка може надати підтримку;4) дуже гнучкий і редагуємий, що дозволяє розробникам додавати додаткові функції.

Кінець таблиці 3.4

Недоліки	<p>1) складність управління спрайтами персонажів за допомогою стрілок, тобто призначення кнопок;</p> <p>2) складність реалізації нестандартних механік, таких як покрокові битви.</p>
Ціна	Ren'Py є безкоштовним ігровим рушієм
Офіційний сайт	https://www.renpy.org

У якості мови програмування був обраний Python через його простоту та багатофункціональність, однак у якості альтернативи Ren'Py також має підтримку Cython, який є надбудовою над Python. Але для більш коректного вибору було проведено порівняння цих двох мов.

Таблиця 3.5 – Порівняння доступних мов реалізації

Характеристика	Python	Cython
Клас мови	Високорівнева мова програмування	Надбудова над Python, що дозволяє використовувати C-подібний синтаксис
Переваги	<p>1) простота синтаксису та читабельність;</p> <p>2) широка підтримка бібліотек.</p>	<p>1) збільшення продуктивності;</p> <p>2) дозволяє писати код, схожий на Python, але зі швидкістю C.</p>

Кінець таблиці 3.5

Недоліки	Може бути повільним та неефективним для складних завдань	Потребує вивчення C/C++
Найкраще підходить для	1) створення прототипів та незалежних проєктів; 2) створення 2D-ігор.	Вирішення проблем продуктивності

Як було зазначено у таблиці 3.5 більш пріоритетною мовою реалізації є Python, через його простоту та широкий спектр підтримуваних бібліотек, завдяки яким більшість можливостей можуть бути реалізовані більш оптимальним та легким шляхом. Однак у самому редакторі коду використовується .гру синтаксисі, що не є кардинальною відмінністю від Python. Відмінності тільки в синтаксисі, для більш простої реалізації базових функцій.

3.3. Вибір редактора коду

Оскільки в Ren'Py виникає потреба редагувати код та бачити зміни у реальному часі, то для більшого доступу до проєкту необхідно обрати IDE та підключити його до програмного застосунку. Серед запропонованих редакторів коду (рис. 3.4) було обрано Visual Studio Code.

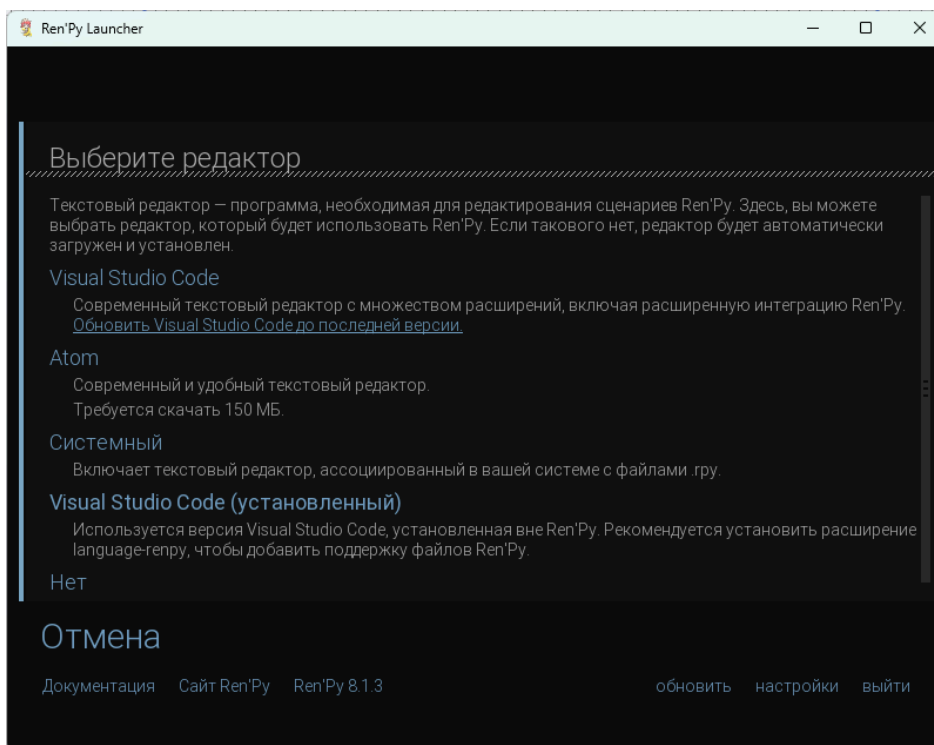


Рисунок 3.4 – Налаштування редактора коду у програмному забезпеченні

Після підключення обраного IDE, необхідно встановити додаткові бібліотеки та плагіни, для більш комфортної та швидкої роботи з проектом. Для цього Visual Studio Code необхідно оновити до останньої актуальної версії та упевнитися, що Ren'Py реєструє його як встановлене програмне забезпечення у системі.

Також, у самому IDE необхідно встановити певні розширення, для повної синхронізації редактора коду із Ren'Py. Перш за все необхідно встановити розширення для підтримки Python (рис. 3.5), оскільки на цій мові більшу частину розробки й буде присвячений проект.

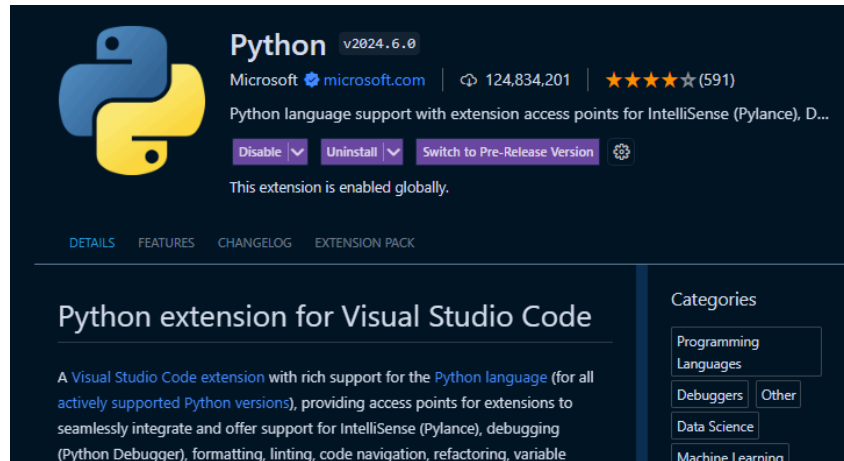


Рисунок 3.5 – Розширення Python

Далі, оскільки у Ren'Py є власна надбудова над Python, то також необхідно встановити відповідне розширення (рис. 3.6), яке надасть доступ до синтаксису та більш зручного редагування деяких аспектів ігрового застосунку (такі як колір тексту), а також доступ до сніпетів, що дозволить легко прописувати цілі блоки коду.

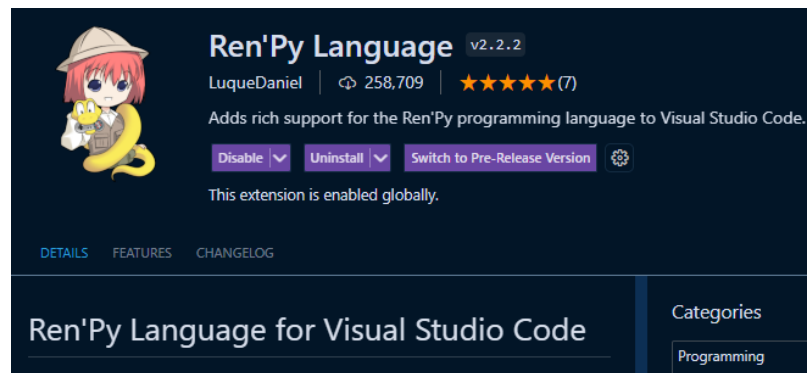


Рисунок 3.6 – Розширення Ren'Py Language

А також необхідно встановити розширення «Ren'Py Launch and Sync» (рис. 3.7), яке дозволить без запуску через ігровий рушій підключитись до проєкту. Завдяки чому, достатньо відкрити папку з проєктом і автоматично буде підключено до нього.

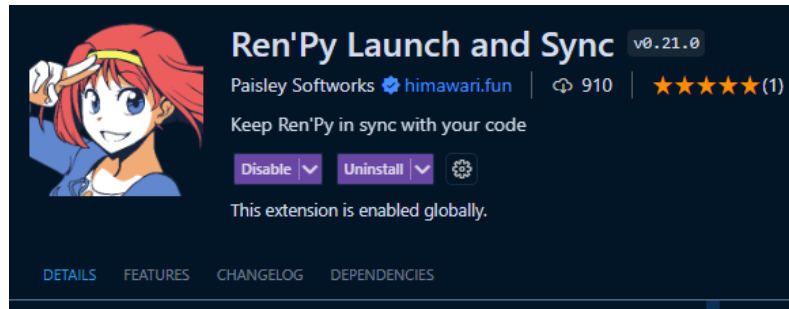


Рисунок 3.7 – Розширення Ren'Py Launch and Sync

Після встановлення розширень, йде перевірка їх працездатності. Як показано на (рис. 3.8) розширення та плагіни встановлені коректно та успішно функціонують.

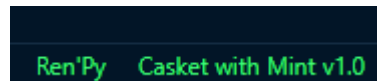


Рисунок 3.8 – Підключення до проєкту

Тепер редактор коду має змогу редагувати код і у реальному часі бачити зміни.

Висновки до розділу 3

У третьому розділі на основі проведеного порівняльного аналізу обрано засоби для розробки ігрового застосунку. Було проведено аналіз ігрового рушія, на якому буде йти розробка, а також мови програмування. Для більш коректного вибору, було також розглянуто аналоги програмного забезпечення із схожим напрямком роботи, а також мови програмування які вони підтримують. Для більш комфортної та оптимальної роботи з проєктом було підключено редактор коду та встановлено кілька розширень, що полегшують процес розробки.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ ІГРОВОГО ЗАСТОСУНКУ

4.1 Реалізація ігрового інтерфейсу

Перед тим як зануритися в технічні деталі, важливо зрозуміти основні принципи дизайну ігрового інтерфейсу:

- 1) **зручність використання:** Інтерфейс має бути інтуїтивно зрозумілим для користувача. Гравець не повинен витратити багато часу на навчання використанню меню та інших елементів управління;
- 2) **естетичність:** Візуальна привабливість інтерфейсу допомагає зацікавити гравця у грі. Елементи дизайну повинні відповідати загальній стилістиці гри;
- 3) **функціональність:** Кожен елемент інтерфейсу повинен мати своє призначення та допомагати гравцеві взаємодіяти з грою.

Ren'Py використовує сценарії на мові програмування Python з додатковими можливостями Ren'Py для створення діалогів, анімацій та інтерактивних елементів. Основні файли проекту включають:

- 1) **script.rpy:** Основний сценарій гри, що містить діалоги, вибори, персонажів та сценарії;
- 2) **gui.rpy:** Файл, в якому визначаються елементи користувацького інтерфейсу, такі як розмір шрифту, колір тексту, стилі шрифту та зображення;
- 3) **options.rpy:** Загальні налаштування гри, де визначається робоча назва, зображення іконки ігрового застосунку, переходи тощо;
- 4) **screens.rpy:** Зовнішній вигляд головних діючих екранів ігрового застосунку та реалізація медіа файлів та шрифтів, визначених у файлі `gui.rpy`.

Згідно мокапам, які були розроблені до цього, виконується реалізація ігрового інтерфейсу гри. Завдяки надбудовою над Python у вигляді Ren'Py мови, процес реалізації та редагування ігрового інтерфейсу буде значно

полегшений. Усі шрифти, стилі, зображення тощо ініціалізуються у файлі `gui.rpy`, де для більшої зручності ініціалізуються усі стилі, кольора, зображення та інші медіа, які потім будуть реалізовані у файлі `screens.rpy` та інших. Це зроблено для оптимізації та більш легкого налаштування, як стилів, так і їх розташування та розмір на екрані.

Першим чином, треба змінити головне меню, тому для цього знаходиться шлях до файлів головного меню. За цим шляхом автоматично згенероване зображення фону замінюється на бажане, а також змінюється його розмір для адаптування під розширення екрану.

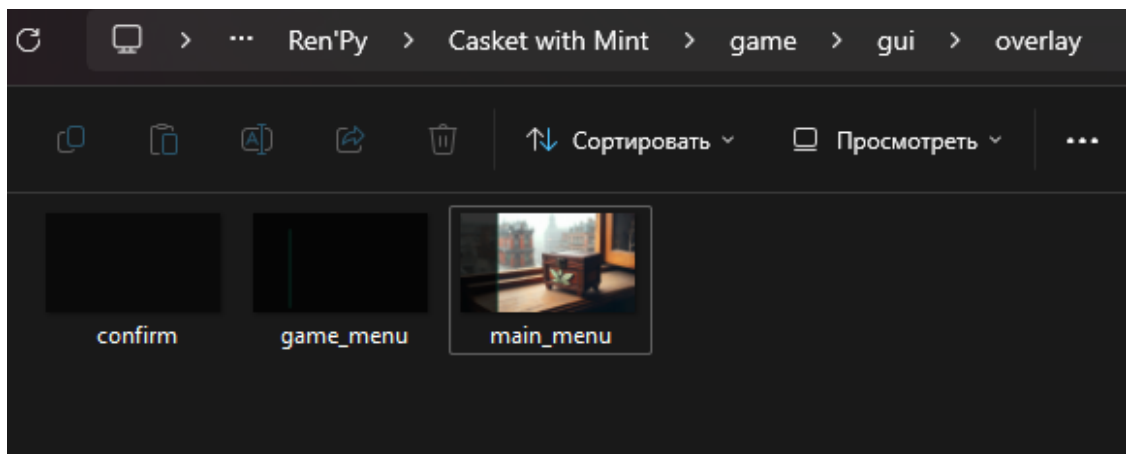


Рисунок 4.1 – Папка із зображеннями фону головного меню

Замінене зображення потім треба ініціалізувати у коді `gui.rpy`, яке потім буде реалізовано у файлі `screens.rpy` де потім буде змінено за стилями.

```
## Изображения, используемые в главном и игровом меню.  
define gui.main_menu_background = "gui/main_menu.png"  
define gui.game_menu_background = "gui/game_menu.png"
```

Рисунок 4.2 – Ініціалізоване зображення фону головного меню

Також разом із реалізацією фону, було додано назву та версію проєкту.


```
tag menu

add gui.main_menu_background

## Эта пустая рамка затемняет главное меню.
frame:
    style "main_menu_frame"

if gui.show_name:
    vbox:
        style "main_menu_vbox"

        text "[config.name!t]":
            style "main_menu_title"

        text "[config.version]":
            style "main_menu_version"
```

Рисунок 4.3 – Додавання фону, назву та версію проєкту

У результаті цих та нових рядків коду, було додано ігровий інтерфейс та скореговано для більшої зручності. Також головне меню повинно мати необхідні для користувача кнопки навігації: почати нову гру, завантажити збереження, відкрити вікно налаштувань ігрового застосунку, побачити короткий опис «О грі», невеличка пояснення управління для гравця, а також кнопка виходу з гри. Усі ці кнопки будуть додані та відредаговані згідно стилям, а також через файл `navigation.json` налагоджена навігація між цими екранами. При необхідності, кількість кнопок у головному меню можна змінити або перейменувати їх.

За замовчуванням, уся навігація між екранами у Ren'Py вже прописана у файлі `navigation.json`, тому основна робота з функціоналом та зовнішнім виглядом цих екранів здійснюється через `screens.rpy` та `gui.rpy`. Можливий варіант додати свої екрани, але потім слід їх занести до `navigation.json` та ініціалізувати у `screens.rpy`.

```
screen navigation():  
  
    vbox:  
        style_prefix "navigation"  
  
        xpos gui.navigation_xpos  
        yalign 0.5  
  
        spacing gui.navigation_spacing  
  
        if main_menu:  
            textbutton _("Начать") action Start()  
  
        else:  
            textbutton _("История") action ShowMenu("history")  
  
            textbutton _("Сохранить") action ShowMenu("save")  
  
            textbutton _("Загрузить") action ShowMenu("load")  
  
            textbutton _("Настройки") action ShowMenu("preferences")
```

Рисунок 4.4 – Додані кнопки головного меню

Тепер користувач має змогу перейти до обраного ним екрану при натисканні на кнопку. Після реалізації кнопок, екранів, навігації між ними, а також зовнішній вигляд головного меню, результатом є функціональний екран головного меню (рис. 4.5).

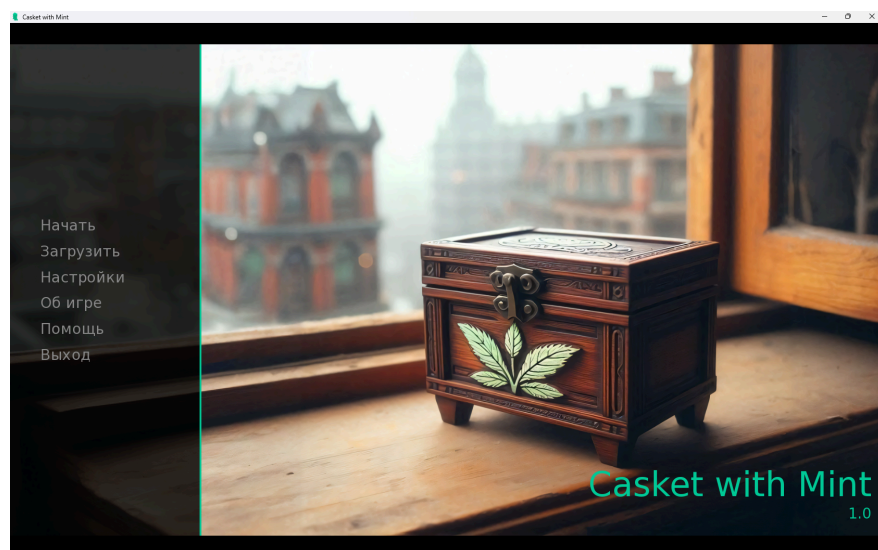


Рисунок 4.5 – Головне меню проекту

Після створення інтерфейсу головного меню, також необхідно реалізувати інтерфейс основної частини візуальної новели – діалогові вікна. Діалогові вікна є основною частиною візуальних новел. Вони повинні бути добре оформленими та зручними для читання.

```
style namebox:
    xpos gui.name_xpos
    xanchor gui.name_xalign
    xsize gui.namebox_width
    ypos gui.name_ypos
    ysize gui.namebox_height

    background Frame("gui/namebox.png", gui.namebox_borders, tile=gui.namebox_tile, xalign
padding gui.namebox_borders.padding

style say_label:
    properties gui.text_properties("name", accent=True)
    xalign gui.name_xalign
    yalign 0.5

style say_dialogue:
    properties gui.text_properties("dialogue")

    xpos gui.dialogue_xpos
    xsize gui.dialogue_width
    ypos gui.dialogue_ypos

adjust_spacing False
```

Рисунок 4.6 – Реалізація інтерфейсу діалогових вікон

Завдяки синтаксису Ren'Py, код не займає багато місця та просто читається. У результаті створення діалогового вікна, а також додавання зображень персонажа та фону за ним, отримується результат як на (рис. 4.7).



Рисунок 4.7 – Інтерфейс діалогового вікна

Також, на той випадок, якщо гравець пропустить речення, яке він не впівав прочитати, то для зручності було додано меню логу. Завдяки цьому меню, гравець може переглянути який персонаж, що сказав у минулому. У будь-який момент можна буде перечитати чи пригадати попередні репліки персонажей, при натисканні на кнопку відкриття логу.

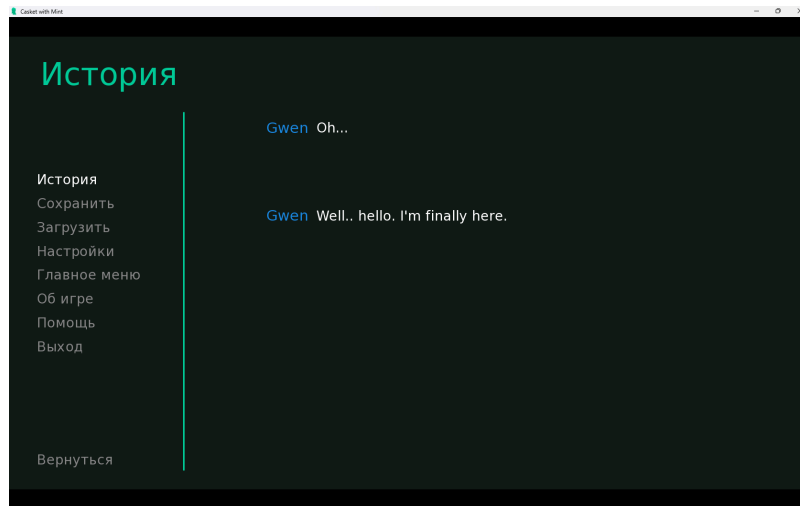


Рисунок 4.8 – Меню логу реплік персонажу

Таким чином, був створений інтерфейс екрану діалогів із персонажами, на якому гравець буде проводити більшу частину свого ігрового часу. Але якщо гравець захоче повернутись до минулого моменту гри, або зберегти свій прогрес – для цього була створена система збережень, яка дозволяє це реалізувати.

4.2 Створення системи збережень

Ren'Py автоматично підтримує збереження та завантаження ігрового стану. Під час збереження свого прогресу, створюється файл .save у якому зберігаються необхідна інформація про прогрес гравця та його зроблений вибір відповіді. За замовчуванням відведених слотів для збережень - безмежне. Також крім ручного збереження, Ren'Py підтримує систему автозбереження, яке автоматично створює файли збереження гравця кожний відповідний період часу. Також є функція швидкого збереження, яке без

введення назв або вибору слоту, при натисканні на відповідну кнопку створює швидке збереження поточного прогресу гравця. Однак, для покращення користувацького досвіду, можна налаштувати всі ці функції під свої потреби.

Меню збережень дозволяє гравцям зберігати поточний стан гри. Його можна викликати через основне меню або за допомогою спеціальної кнопки в інтерфейсі гри.

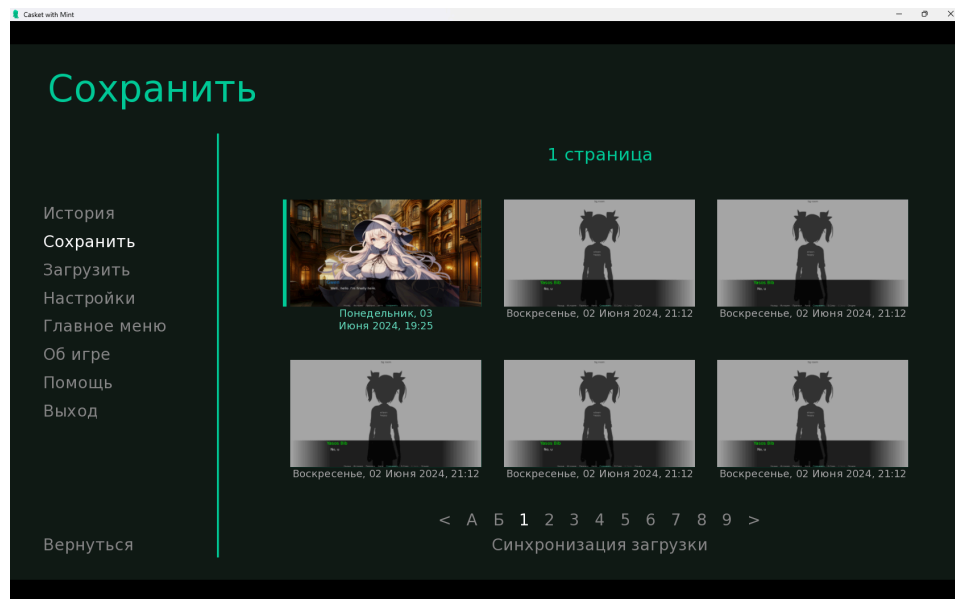


Рисунок 4.9 – Меню збереження прогресу гравця

Меню завантаження дозволяє гравцям вибирати раніше збережені ігри із автоматичних, швидких або ручних слотів для продовження. Це зручно для гравців, оскільки дозволяє не турбуватися про те, що вони забудуть зберегти прогрес.

Налаштування системи збережень включає в себе визначення кількості слотів для збережень, налаштування автоматичного збереження, а також налаштування інтерфейсу. Для створення більш естетичного та зручного інтерфейсу збережень, можна налаштовувати вигляд елементів користувацького інтерфейсу. Також слід обмежити кількість доступних слотів збережень для гравця до 6, для більшої зручності та додання трохи

складності для гравця, щоб він розпоряджався доступними слотами більш свідомо.

4.3 Система вибору відповіді користувача

Слід виділити основні принципи системи вибору, на яких вона базується:

- 1) **інтуїтивність**: Вибори повинні бути чіткими та зрозумілими для гравця;
- 2) **вплив на сюжет**: Вибори повинні мати реальний вплив на розвиток сюжету;
- 3) **збалансованість**: Вибори повинні бути добре збалансованими, не роблячи один вибір очевидно кращим за інший.

Ці три пункта важливі при розробці системи вибору, що дозволить допомогти гравцеві поглибитись у сюжет гри, а також робити зважені рішення, які потім приведуть його до однієї з декількох кінцівок гри. Ren'Py надає простий і зручний спосіб створення виборів за допомогою структури «menu», як показано на (рис. 4.10).

```
Label start:
    e "Що ти хочеш робити далі?"

    menu:
        "Піти на прогулянку":
            jump walk

        "Залишитися вдома":
            jump stay_home

Label walk:
    e "Ти вирішив піти на прогулянку. Було приємно."
    # Продовження сюжету
    return

Label stay_home:
    e "Ти вирішив залишитися вдома. Було спокійно."
    # Продовження сюжету
    return
```

Рисунок 4.10 – Код вибору гравця

Важливо, щоб вибір відповідей мали наслідки, які впливають на розвиток сюжету, інакше гравець не буде приділяти їм доцільної уваги. Це можна зробити за допомогою змінних, які зберігають стан гри та визначають подальші події, де в ході ланцюга зроблених виборів виводить на відповідну кінцівку. Слід зазначити, якщо кінцівок наприклад 3, то слід зробити вибір відповідей логічним для здобуття однієї з них.

Для покращення користувацького досвіду можна налаштувати вигляд та розташування елементів вибору. Це можна зробити за допомогою стилів у Ren'Py (рис. 4.11).

```
screen choice(items):
    window:
        style "menu_window"
    vbox:
        for i, (label, action) in enumerate(items):
            textbutton label action action:
                style "menu_choice_button"
                text_style "menu_choice_text"
```

Рисунок 4.11 – Код налаштування стилів вибору

У файлі options.rpy можна налаштувати стилі кнопок та тексту (рис. 4.12).

```
init -2:
    style.menu_window.background = Frame("gui/button.png", 10, 10)
    style.menu_choice_button.background = Frame("gui/button_hover.png", 10, 10)
    style.menu_choice_text.color = "#FFFFFF"
    style.menu_choice_text.font = "DejaVuSans.ttf"
```

Рисунок 4.12 – Код стилю кнопок вибору

Іноді потрібно, щоб певні вибори були доступні тільки за певних умов. Наприклад, якщо гравець раніше обрав підібрати певний предмет, то пізніше у діалозі в нього з'явиться новий варіант відповіді, який при інших умовах не з'явився би. Це можна реалізувати за допомогою умовних конструкцій, як показано на (рис. 4.13).

```
Label start:  
    $ special_item = False  
  
    е "Що ти хочеш робити далі?"  
  
    меню:  
        "Піти на прогулянку":  
            jump walk  
  
        "Залишитися вдома":  
            jump stay_home  
  
        "Використати спеціальний предмет" if special_item:  
            jump use_item
```

Рисунок 4.13 – Варіант відповіді при наявності спеціального предмету

Таким чином, можна реалізувати будь-яку кількість варіантів відповідей та пов'язати їх із продовженням реплік. За необхідністю, у файлі `gui.rpy` та `screens.rpy` можна змінити зовнішній вигляд варіантів вибору відповіді, або навіть додати нові анімації.



Рисунок 4.14 – Вибір відповіді гравця

Як продемонстровано на (рис. 4.14), система вибору відповіді гравця працює справно та не створює зайвих багів у системі.

Висновки до розділу 4

У четвертому розділі розглянуто три важливі аспекти розробки візуальної новели за допомогою Ren'Py: створення ігрового інтерфейсу, розробка системи збережень та реалізація системи вибору. Кожен з цих аспектів відіграє ключову роль у забезпеченні захоплюючого та комфортного ігрового досвіду для гравця.

Було розглянуто основні принципи дизайну ігрового інтерфейсу, включаючи зручність використання, естетичність та функціональність. Було детально описано процес розробки основних елементів інтерфейсу, таких як головне меню, діалогові вікна та меню налаштувань. Особлива увага була приділена налаштуванню інтерфейсу для забезпечення його відповідності загальній стилістиці гри та підвищення зручності для гравця. Основний опір був зроблений на зовнішньому вигляді інтерфейсу гравця.

Система збережень є важливим компонентом будь-якої гри, дозволяючи гравцям зберегти свій прогрес та повертатися до гри в будь-який час. Було розглянуто базові принципи роботи системи збережень у Ren'Py, включаючи ручне та автоматичне збереження. Крім того, було описано процес налаштування інтерфейсу збережень та тестування системи для забезпечення її надійності та зручності використання. Слід зазначити, що файли збережень мають розширення `.save` замість більш звичних та простих `.json` та його аналогів, що підвищує захист збережень від зовнішнього втручання гравця для нечесної гри.

Система вибору є ключовою для інтерактивних історій, дозволяючи гравцям впливати на розвиток сюжету. Було розглянуто основні пункти створення системи вибору в Ren'Py, обробку їх наслідків та налаштування інтерфейсу виборів. Особливу увагу було приділено розширеним можливостям системи вибору, таким як умовні вибори, що дозволяють зробити ігровий процес гравця більш динамічним, а також логічність виборів для досягнення однієї з доступних кінцівок сюжету.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи бакалавра було створено робочий прототип ігрового застосунку у жанрі візуальна новела. Для досягнення поставленої мети виконано наступні завдання:

- 1) проаналізовано існуючі застосунки для роботи з візуальними новелами;
- 2) здійснено моделювання концепції для застосунку;
- 3) створено графічні та аудіо матеріали для гри;
- 4) впроваджено механізми вибору та нелінійного розвитку сюжету;
- 5) здійснена програмна реалізація ігрового застосунку;
- 6) проведено тестування застосунку та оцінку його ефективності;
- 7) виконано оптимізацію та забезпечення якості роботи гри.

У першому розділі КРБ були досліджені питання жанру "візуальна новела" та його популярні представники з різними стилями візуальної та ігрової реалізації. Був проведений аналіз вимог до програмного забезпечення для розробки ігрового застосунку та визначено основні функції, що необхідні процесу розробки.

У другому розділі було проведено моделювання зовнішнього вигляду інтерфейсу та проектування поведінки ігрового застосунку та можливі сценарії його роботи. Були розроблені мокапи та приведені таблиці.

У третьому розділі були розглянуті та обрані засоби для розробки ігрового застосунку. Було проведено порівняння декількох ігрових рушіїв між собою, їх функції та недоліки. Проведено детальний аналіз ігрового рушія Ren'Py та доступних мов реалізації.

У четвертому розділі було проведено програмну реалізацію ігрового застосунку, його зовнішній вигляд та функції. Додані основні ключові елементи та механіки, а також внесення медіа матеріалів для візуального та звукового представлення сюжету гравцеві.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

- 1) Візуальна_Новела. URL: https://drama.kropyva.ch/Візуальна_новела
(дата звернення 08.05.2024)
- 2) Phoenix_Wright:_Ace_Attorney. URL:
<https://www.ace-attorney.com/trilogy/uk/> (дата звернення 08.05.2024)
- 3) Зайчик_(гра). URL: <https://bunny-horror.tumblr.com> (дата звернення
08.05.2024)
- 4) Doki_Doki_Literature_Club!. URL: <https://ddlc.moe> (дата звернення
08.05.2024)
- 5) Danganronpa:_Trigger_Happy_Havoc. URL:
https://store.steampowered.com/app/413410/Danganronpa_Trigger_Happy_Havoc/
(дата звернення 08.05.2024)
- 6) Unified_Modeling_Language. URL:
<https://miro.com/diagramming/what-is-a-uml-diagram/> (дата звернення
08.05.2024)
- 7) UML_Sequence_Diagram. URL:
<https://www.lucidchart.com/pages/uml-sequence-diagram> (дата звернення
08.05.2024)
- 8) UML_Activity_Diagram. URL:
<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-a-ctivity-diagram/> (дата звернення 08.05.2024)
- 9) Ren'Py_Documentation. URL: <https://www.renpy.org/doc/html/> (дата
звернення 10.06.2024)
- 10) Visual_Studio_Code. URL: <https://code.visualstudio.com> (дата звернення
10.06.2024)
- 11) Twine. URL: <https://twinery.org> (дата звернення 10.06.2024)
- 12) Adventure_Game_Studio. URL: <https://www.adventuregamestudio.co.uk>
(дата звернення 10.06.2024)

- 13) Visual_Novel_Maker. URL: <https://visualnovelmaker.com> (дата звернення 10.06.2024)
- 14) Ren'Py. URL: <https://www.renpy.org> (дата звернення 10.06.2024)
- 15) Canva. URL: <https://www.canva.com/create/mockup-generator/> (дата звернення 10.06.2024)
- 16) Photopea. URL: https://fotoo.org/edit_photo/ (дата звернення 10.06.2024)