МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ
Кафедра комп'ютерних наук

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: **«РОЗРОБКА ВІРТУАЛЬНОЇ НОВЕЛИ З ВИКОРИСТАННЯМ ІГРОВОГО РУШІЯ REN'PY»**

Виконала: студентка 4 курсу, групи 6.1221
спеціальності 122 Комп'ютерні науки
(шифр і назва спеціальності)
освітньої програми Комп'ютерні науки
(назва освітньої програми)
К.С. Самойленко
(ініціали та прізвище)
Керівник доцент кафедри комп'ютерних наук, PhD, Калюжняк А. В.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)
Рецензент доцент кафедри фундаментальної та прикладної математики, доцент, к.ф.-м.н. Д'яченко Н.М.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Запоріжжя 2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет математичний

Кафедра комп'ютерних наук

Рівень вищої освіти бакалавр

Спеціальність 122 Комп'ютерні науки
(шифр і назва)

Освітня програма Комп'ютерні науки

ЗАТВЕРДЖУЮ
Завідувач кафедри комп'ютерних наук,
д.т.н., професор

_____ Шило Г.М.
(підпис)

" ____ " _____ 2024 р.

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ

Самойленко Карині Сергіївні
(прізвище, ім'я та по-батькові)

1. Тема роботи Розробка віртуальної новели з використанням ігрового рушія Ren'Py

керівник роботи Калюжняк Анастасія Вікторівна, доцент, PhD
(прізвище, ім'я та по-батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від « 20 » грудня 2024 року № 2001-с

2. Строк подання студентом роботи 26.05.25

3. Вихідні дані до роботи

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Постановка задачі
2. Основні теоретичні відомості
3. Розробка віртуальної новели з використанням ігрового рушія Ren'Py

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
презентація

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 25.12.24

К А Л Е Н Д А Р Н И Й П Л А Н

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.	11.01.2025	Виконано
2.	Збір вихідних даних.	08.02.2025	Виконано
3.	Обробка методичних та теоретичних	25.02.2025	Виконано
	джерел.		
4.	Розробка першого та другого розділу.	20.03.2025	Виконано
5.	Розробка третього розділу.	25.04.2025	Виконано
6.	Оформлення та нормоконтроль	04.06.2025	Виконано
	кваліфікаційної роботи бакалавра.		
7.	Захист кваліфікаційної роботи.	18.06.2025	Виконано

Студент _____ К. С. Самойленко
(підпис) (ініціали та прізвище)

Керівник роботи _____ А. В. Калюжняк
(підпис) (ініціали та прізвище)

Нормоконтроль пройдено

Нормоконтролер _____ О. Г. Спиця
(підпис) (ініціали та прізвище)

РЕФЕРАТ

Кваліфікаційна робота бакалавра «Розробка віртуальної новели з використанням ігрового рушія Ren’Py»: 45 с., 36 рис., 2 табл., 13 джерел.

ВІРТУАЛЬНА НОВЕЛА, ГІЛКУВАННЯ СЮЖЕТУ, ГРАФІЧНИЙ ДИЗАЙН, ІГРОВИЙ РУШІЙ REN’PY, ІНТЕРАКТИВНИЙ СЮЖЕТ, НАРАТИВНІ ТЕХНОЛОГІЇ, ПРОГРАМУВАННЯ.

Об’єкт дослідження – процес розробки віртуальної новели з використанням рушія Ren’Py.

Мета роботи: розробити інтерактивну віртуальну новелу, що демонструє можливості рушія Ren’Py для створення нелінійного сюжету та максимального залучення гравця через інтерактивну взаємодію.

Метод дослідження – практичний, що включає аналіз, проєктування та експериментальну розробку.

Проаналізовано особливості жанру віртуальних новел, функціональні можливості рушія Ren’Py та принципи побудови інтерактивного наративу. Розглянуто технічні аспекти реалізації розгалуженого сюжету, анімації, звукового супроводу та інтерактивних елементів. В рамках дослідження створено віртуальну новелу, яка демонструє ефективність використання Ren’Py для створення захоплюючої інтерактивної історії.

Результатом роботи є розроблена віртуальна новела, що підтверджує потенціал Ren’Py як інструменту для створення якісних інтерактивних наративів.

SUMMARY

Bachelor’s Qualifying Theses «Virtual Novel Development Using Ren’Py Game Engine»: 45 pages, 36 figures, 2 tables, 13 references.

VISUAL NOVEL, PLOT BRANCHING, GRAPHIC DESIGN, REN’PY GAME ENGINE, INTERACTIVE PLOT, NARRATIVE TECHNOLOGIES, PROGRAMMING.

Object of the study – the process of developing a visual novel using the Ren’Py engine.

Aim of the study: to develop an interactive visual novel demonstrating the capabilities of the Ren’Py engine for creating a non-linear plot and maximizing player engagement through interactive interaction.

Research method: practical, including analysis, design, and experimental development.

The features of the visual novel genre, functional capabilities of the Ren’Py engine, and principles of interactive narrative construction have been analyzed. Technical aspects of implementing a branched plot, animation, sound accompaniment, and interactive elements have been examined. Within the framework of the study, a visual novel has been created, demonstrating the effectiveness of using Ren’Py for creating an engaging interactive story.

The result of the work is a developed visual novel that confirms the potential of Ren’Py as a tool for creating high-quality interactive narratives.

ЗМІСТ

Завдання на кваліфікаційну роботу 2
Реферат 4
Summary 5
Вступ 7
1 Теоретичні та технічні основи розробки візуальних новел за допомогою ігрового рушія Ren'py 8
1.1 Визначення та класифікація візуальних новел 8
1.2 Функціональність Ren'py 13
2 Проєктування інтерактивної візуальної новели 17
2.1 Опис сюжету 17
2.2 Концепція візуальної новели 17
2.3 Розробка персонажів 19
2.4 Розробка фонових зображень 22
2.5 Створення анімації 24
2.6 Створення іконки новели 25
3 Технічна реалізація та тестування візуальної новели 27
3.1 Структура проєкту в Ren'Py 27
3.2 Реалізація геймплейних елементів 28
3.3 Реалізація інтерфейсу користувача 32
3.4 Реалізація мініігор 36
3.5 Налаштування компіляції та збірка гри 40
Висновки 42
Перелік посилань 44

ВСТУП

Використання ігрового рушія Ren’Py для створення віртуальних новел є ефективним та доступним методом, який дозволяє створювати високоякісні інтерактивні наративи, які можуть бути застосовані як професіоналами, так і аматорами у сфері інтерактивного сторітелінгу. Програма Ren’Py надає широкий спектр інструментів та можливостей, що дозволяють реалізувати творчі ідеї та підвищити якість інтерактивних оповідей.

Метою кваліфікаційної роботи виступає розробка інтерактивної віртуальної новели, що демонструє можливості рушія Ren’Py для створення нелінійного сюжету та максимального залучення гравця через інтерактивну взаємодію.

Для досягнення поставленої мети необхідно виконати наступні задачі:

- провести аналіз літературних джерел та ресурсів, присвячених віртуальним новелам, особливостям роботи з Ren’Py та методам створення інтерактивних наративів;
- розкрити та застосувати різні техніки створення віртуальної новели в Ren’Py, такі як розробка розгалуженого сюжету, анімація, звуковий супровід, створення інтерактивних елементів для забезпечення глибокого занурення гравця в історію;
- провести розробку віртуальної новели засобами Ren’Py, яка включає в себе етапи побудови сюжетної лінії, створення графічного дизайну, програмування інтерактивних механік, додавання звукового супроводу та тестування;
- провести тестування розробки та ефективності використаних наративних технік.

1 ТЕОРЕТИЧНІ ТА ТЕХНІЧНІ ОСНОВИ РОЗРОБКИ ВІЗУАЛЬНИХ НОВЕЛ ЗА ДОПОМОГОЮ ІГРОВОГО РУШІЯ REN'PY

1.1 Визначення та класифікація візуальних новел

Візуальна новела – це інтерактивний жанр, що поєднує в собі елементи літератури, ілюстрації та ігрові механіки. Цей жанр характеризується представленням наративу через текст, який супроводжується статичними зображеннями персонажів, фоновими ілюстраціями, музичним та звуковим супроводом, а також включає елементи інтерактивності, що дозволяють гравцеві приймати рішення в ключові моменти сюжету, щоб впливати на хід історії.(Таблиця 1.1 – Порівняльна характеристика візуальних новел та традиційних комп'ютерних ігор)

Основні характеристики візуальних новел:

- наративна орієнтованість – сюжет та розвиток інформації має першочергове значення, на відміну від традиційних ігор, де механіка гемплею відіграють провідну роль;
- графічний інтерфейс – візуальні новели включають такі графічні елементи;
- статичні або анімовані зображення персонажів (спрайти);
- фонові зображення;
- екрани діалогів з текстом;
- елементи інтерфейсу з користувача.

Інтерактивність – користувач має можливість впливати на розвиток сюжету через:

- вибір діалогових опцій;
- прийняття рішень;

– розв’язання головоломок чи мініігор.

Таблиця 1.1 – Порівняльна характеристика візуальних новел та традиційних комп’ютерних ігор

Характеристика	Візуальна новела	Звичайні ігри
Основний фокус	Наратив, розвиток сюжету, емоційна залученість та розкриття персонажів; акцент на літературній складовій та міжособистісних взаєминах Геймплей, ігрові механіки, виконання завдань, досягнення цілей; акцент на інтерактивності та змагальності	Геймплей, ігрові механіки, виконання завдань, досягнення цілей; акцент на інтерактивності та змагальності
Інтерактивність	Обмежена: вибір діалогових опцій, прийняття ключових рішень у сюжеті, іноді прості головоломки або мініігри	Розширена: активне управління персонажем, різноманітні типи взаємодії з ігровим світом (бій, дослідження, будівництво, тощо)
Графічне представлення	Переважно двовимірні статичні або частково анімовані зображення персонажів, фонові ілюстрації, текстові блоки; графіка як ілюстративний елемент	Повноцінні тривимірні моделі з анімацією, динамічні ігрові середовища, складні візуальні ефекти; графіка як основний елемент ігрового процесу
Темп проходження	Розмірений, споглядальний, орієнтований на читання та роздуми; гравець визначає швидкість прогресу через текст	Варіативний, часто динамічний, з елементами напруги та часовими обмеженнями; темп часто задається розробником
Структура сюжету	Виражена нелінійність: множинні сюжетні гілки, альтернативні сценарії та декілька можливих закінчень залежно від рішень гравця	Переважно лінійний основний сюжет з можливими відгалуженнями на другорядні квести; часто має одне канонічне закінчення
Співвідношення тексту та графіки	Домінування текстової складової; графіка виконує допоміжну, ілюстративну функцію	Мінімізація текстового контенту; графічна складова є пріоритетною для передачі ігрового досвіду
Вимоги до гравця	Уважність до деталей сюжету, емоційна залученість, аналітичне мислення для прийняття рішень	Розвиток ігрових навичок, координація, швидкість реакції, стратегічне мислення
Емоційна складова	Акцент на емоційному зв’язку з персонажами, психологічних аспектах та моральних дилемах	Фокус на емоціях від досягнення цілей, подолання викликів та змагальному аспекті
Типова тривалість	Від 2-3 годин (короткі новели)	Від кількох годин до сотень

	до 50+ годин (епічні багатогілкові сюжети)	годин залежно від жанру та масштабу

Вивчення візуальних новел як інтерактивного жанру передбачає необхідність їхньої систематизації за різними критеріями. Отже, розглянемо основні підходи до класифікації візуальних новел.

За ступенем інтерактивності:

За цим критерієм візуальні новели можуть бути розподілені на кілька типів. По-перше, виділяють кінетичні новели, які характеризуються повною відсутністю інтерактивних елементів, що передбачають вибір або прийняття рішень гравцем. У такому випадку користувач виконує лише функцію спостерігача за розгортанням фіксованого сюжету шляхом послідовного переходу між текстовими блоками та візуальними елементами. По-друге, існують класичні візуальні новели, які, на відміну від кінетичних, включають елементи вибору. Ці елементи мають безпосередній вплив на подальший розвиток сюжетної лінії та можуть призводити до різноманітних фіналів, залежно від прийнятих гравцем рішень. По-третє, окрему категорію становлять гібридні візуальні новели. Слід зазначити, що цей тип поєднує в собі характерні риси візуальних новел з елементами інших ігрових жанрів. Таким чином, можна виокремити такі комбінації, як: Visual Novel + RPG (рольова гра), що передбачає наявність елементів прокачки персонажів, виконання квестів та бойової системи у поєднанні з розгалуженим наративом.

- Visual Novel + Puzzle (головоломки), де для просування сюжетом гравець повинен розв’язувати різноманітні логічні завдання;
- Visual Novel + Dating Sim (симулятор побачень), що акцентує увагу на побудові відносин з персонажами через систему виборів та управління параметрами;
- Visual Novel + Adventure (пригодницька гра), де дослідження локацій, збір предметів та взаємодія з оточенням є важливими елементами ігрового процесу.

За тематикою та цільовою аудиторією:

Класифікація візуальних новел також може здійснюватися на основі їхньої тематичної спрямованості та орієнтації на певну аудиторію. У зв'язку з цим, можна виділити наступні категорії:

- Отоме-ігри, які переважно орієнтовані на жіночу аудиторію та характеризуються романтичними сюжетами з акцентом на взаємодії з чоловічими персонажами;
- Бісьодзьо-ігри, що, навпаки, спрямовані на чоловічу аудиторію та пропонують романтичні лінії з жіночими персонажами;
- BL-ігри (Boys' Love), основною тематикою яких є романтичні стосунки між чоловіками;
- GL-ігри (Girls' Love), що зосереджені на романтичних відносинах між жінками;
- нейтральні ігри, які не мають вираженого романтичного підтексту або розраховані на широке коло користувачів, незалежно від статі чи сексуальної орієнтації.

За жанровими особливостями:

- ще одним важливим критерієм для класифікації візуальних новел є їхні жанрові особливості. Отже, можна виділити такі жанри;
- романтичні візуальні новели, де ключовим елементом є розвиток романтичних відносин між персонажами;
- детективні візуальні новели, сюжет яких побудований навколо розслідування злочинів або розгадування складних загадок;
- фентезійні/науково-фантастичні візуальні новели, дія яких відбувається у вигаданих світах з елементами магії або футуристичних технологій;
- жахи/трилери, основною метою яких є створення атмосфери страху, напруження та тривоги у гравця;

– комедійні візуальні новели, що акцентують увагу на гумористичних ситуаціях та діалогах.

Таким чином, наведена класифікація дозволяє систематизувати різноманіття візуальних новел за ключовими характеристиками, що є важливим для подальшого дослідження їхніх особливостей та еволюції як інтерактивного жанру.

Приклади відомих візуальних новел створені за допомогою ігрового рушія Ren`Py (**Error! Reference source not found.**Рисунок 1.3 – Long Live the Queen).



Рисунок 1.1 – Doki Doki Literature Club!



Рисунок 1.2 – Katawa Shoujo



Рисунок 1.3 – Long Live the Queen

1.2 Функціональність REN'PY

Ren'Py – безкоштовний, вільний та відкритий ігровий рушій для створення візуальних новел у двовимірній графіці. Розроблений на базі мови програмування Python, рушій пропонує потужний і водночас доступний інструментарій для розробників різного рівня підготовки [1].

Розглянемо технічні особливості рушія Ren'Py, який вирізняється низкою унікальних технічних характеристик.

Для створення сценаріїв використовується мова програмування Python, що дозволяє розробникам створювати складні логічні конструкції та реалізовувати нестандартні механіки оповіді. Це надає надзвичайну гнучкість у розробці інтерактивних наративів.

Мультимедійні можливості рушія вражають своєю багатофункціональністю: він підтримує різні формати графічних файлів, має вбудовану систему управління звуком, забезпечує повноцінну інтеграцію відеоконтенту та дозволяє створювати анімовані переходи між сценами. Кросплатформеність є ще однією потужною перевагою Ren'Py.

Рушій підтримує експорт проектів на провідні операційні системи та мобільні платформи, включаючи Windows, macOS, Linux, Android та iOS. Це суттєво розширює потенційну аудиторію та доступність створених додатків. Функціональні переваги: (Рисунок 1.4 – Приклад розгалуженого діалогу з варіантами вибору в Ren’Py).

- наративні інструменти;
- розгалужені діалогові системи;
- механізм збереження та повернення до контрольних точок;
- створення складних діалогових flows;
- різноманітні варіанти інтерактивних меню.



Рисунок 1.4 – Приклад розгалуженого діалогу з варіантами вибору в Ren’Py

Інтерфейсні рішення:

- налаштування систем діалогів (Рисунок 1.5 – Приклад діалогового інтерфейсу в Ren’Py);
- створення користувацьких інтерфейсів;
- реалізація систем вибору та прийняття рішень;
- впровадження ігрових механік.



Рисунок 1.5 – Приклад діалогового інтерфейсу в Ren’Py

Переваги використання (Рисунок 1.6 – Інтерфейс Ren’Py Launcher).

– доступність: безкоштовний та відкритий вихідний код;
– простота вивчення: інтуїтивно зрозумілий синтаксис;
– гнучкість: широкі можливості кастомізац;
– спільнота: активна підтримка розробників.

Таким чином, можна дійти висновку, Ren’Py – це потужний та доступний інструмент для створення візуальних новел. Він пропонує широкий спектр функцій для розробки інтерактивних історій, включаючи розгалужені діалоги, анімацію, звуковий супровід та інтерактивні елементи. Ren’Py є безкоштовним і відкритим, що робить його доступним для всіх.

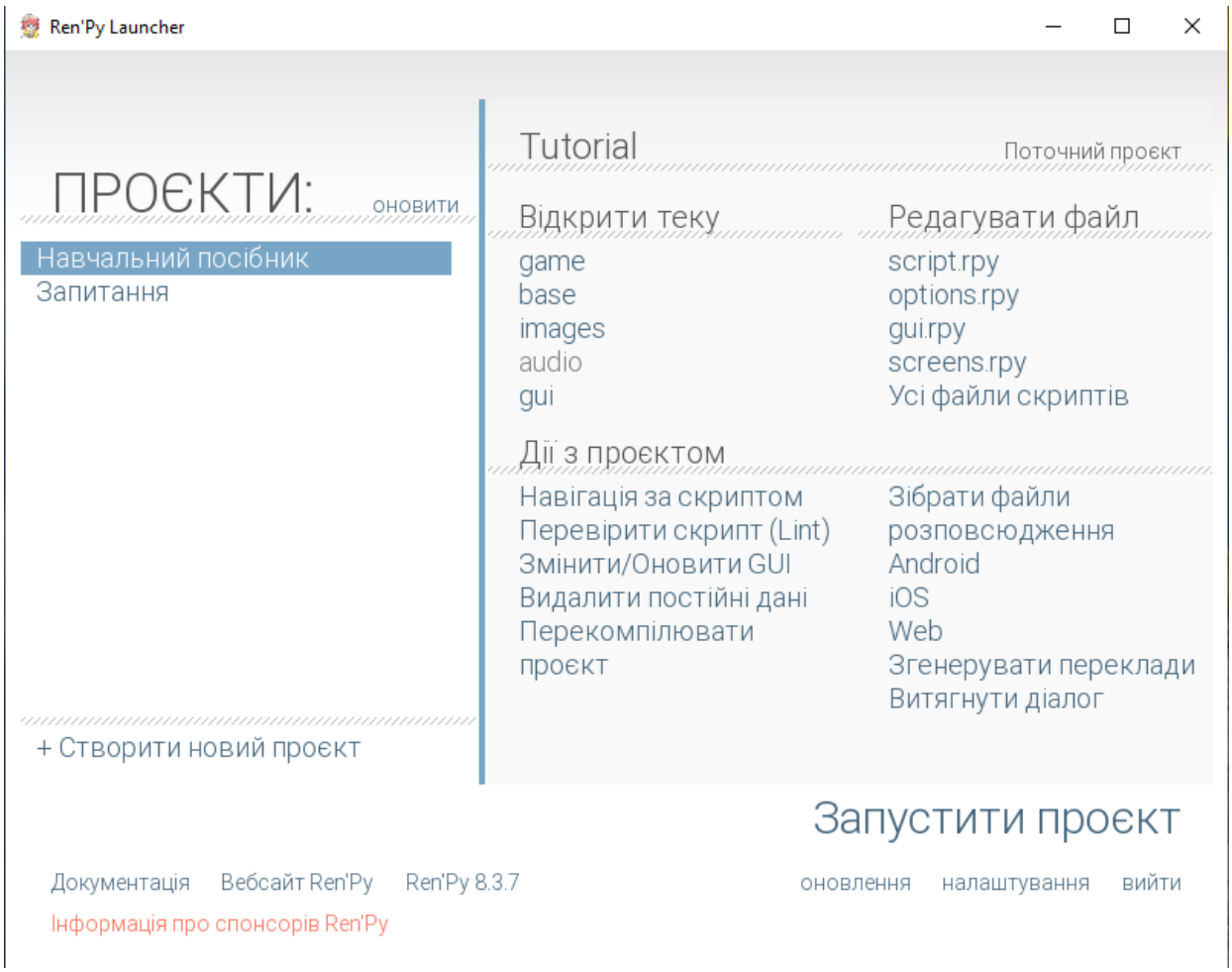


Рисунок 1.6 – Інтерфейс Ren’Py Launcher

2 ПРОЄКТУВАННЯ ІНТЕРАКТИВНОЇ ВІЗУАЛЬНОЇ НОВЕЛИ

2.1 Опис сюжету

Головна героїня, сімнадцятирічна Марія змушена провести літні канікули не в рідному місті, а в селі, де її тітка нещодавно придбала стару оселю. Замість звичних зустрічей з друзями на неї чекає монотонна допомога в облаштуванні нового дому – коробки, пил і нескінченне прибирання. Та невдовзі все змінюється.

Марія знайомиться з двома місцевими підлітками – енергійною Ненсі та стриманим та спокійним Сашком . Нові знайомства швидко переростають у міцну дружбу. Діти часто навідуються до Марії та разом досліджують околиці. Під час однієї з таких зустрічей вони вирішують піднятися на горище, де знаходять стару коробку з пожовклими речами. Серед них – карта і загадковий опис квесту під назвою «Таємниця старого саду». Як виявляється, колись ця гра була дуже популярною серед місцевої молоді, але так і не була завершена. Заінтриговані знахідкою, друзі вирішують відновити квест, поступово занурюючись у атмосферу таємниць, прихованих знаків і загадок минулого.

На них чекають не лише цікаві пригоди, а й важливі рішення, що можуть змінити не тільки хід гри, а й їх життя.

2.2 Концепція візуальної новели

Додаток є візуальною новелою з елементами детективу та трилеру, розроблений для на операційних системах Windows. Цільова аудиторія – підлітки віком від 12+ років.

Гра містить сцену з головним меню, інтерфейс вибору та понад 10 сюжетних сцен. Основна ігрова механіка полягає у виборі реплік і дій, що впливають на розвиток подій, розгалуження сюжету, рівень стосунків із персонажами та фінал історії.(Рисунок 2.1 – Блок-схема ігрового процесу).

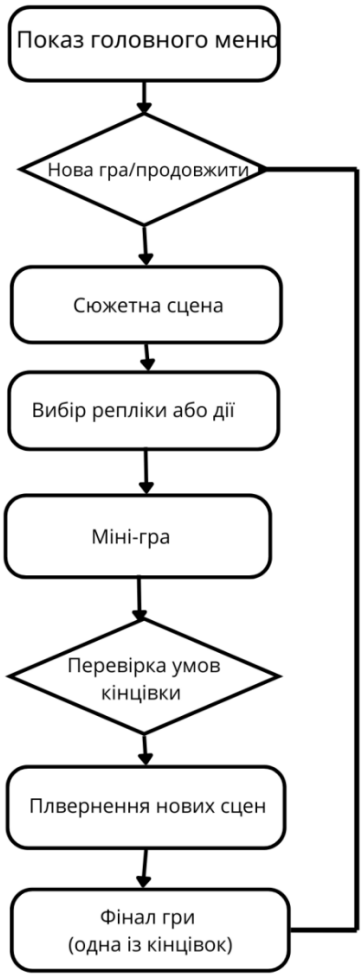


Рисунок 2.1 – Блок-схема ігрового процесу

Сюжет розгортається навколо сімнадцятирічної дівчини, яка приїжджає на літні канікули в село до тітки. У новій оселі на горищі вона з друзями випадково знаходить коробку зі старими речами. Разом із новими друзями героїня намагається розгадати старий квест, пов’язаний із “Таємницею старого саду”.

Під час проходження квесту гравці стикаються з мінііграми – простими логічними або інтерактивними завданнями, які потрібно розв’язати для

просування сюжетом. Це урізноманітнює геймплей і вимагає від гравця уваги, логіки та кмітливості.

Гравець має можливість приймати сюжетно важливі рішення, які змінюють перебіг подій.

У грі передбачено кілька кінцівок: хороша, нейтральна, погана та секретна.

Також реалізовано доступ до прихованих сцен та альтернативних гілок сюжету залежно від прийнятих рішень. У грі є система автозбереження.

Складність гри полягає в ухваленні зважених рішень, аналізі інформації та уважному дослідженні ігрового світу. Завдяки нелінійному сюжету новела має високу реіграбельність і мотивує проходити її кілька разів для розкриття всіх можливих варіантів історії.

2.3 Розробка персонажів

Для створення графічного образу персонажів використовується застосунок ibisPaint X. Він є популярним безкоштовним додатком для цифрового малювання та ілюстрації, особливо цінованим серед художників-аматорів та професіоналів, які працюють на мобільних пристроях Android та iOS (**Error! Reference source not found.**).

Таблиця 2.1 – Переваги ibisPaint X

Переваги ibisPaint X	Опис
Зручність мобільного малювання	Можливість малювати будь-де за допомогою планшета або смартфона.
Потужна підтримка шарів	Ідеально підходить для створення різних емоційних станів персонажів, малюючи базове тіло та різні вирази обличчя на окремих шарах.
Якісні та різноманітні пензлі	Дозволяють створювати як мультяшний стиль, популярний у візуальних новелах, так і більш реалістичний арт.

Продовження таблицы 2.1

Експорт у високій якості	Підтримує експорт у роздільності, придатній для використання в іграх.
Запис процесу малювання (timelapse)	Можливість записувати та ділитися процесом створення персонажів.
Підтримка планшетів з чутливістю до тиску	Дозволяє більш точно контролювати лінії та штрихи при використанні стилуса.
Зручний інтерфейс для сенсорних екранів	Інтерфейс адаптований для ефективної роботи на мобільних пристроях.

Отже, мною було створено серію персонажів (**Error! Reference source not found.Error! Reference source not found.**); у програмі ibisPaint X на прозорому фоні з різними видами обличчя:

– нейтральний;

– радісний;

– сумний.

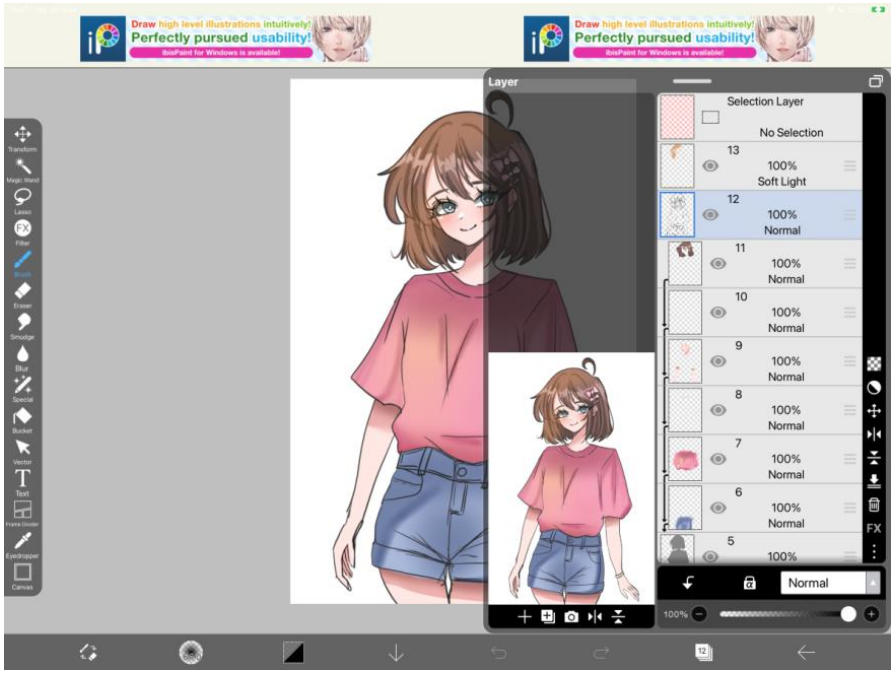


Рисунок 2.2 – Розробка персонажа Марії

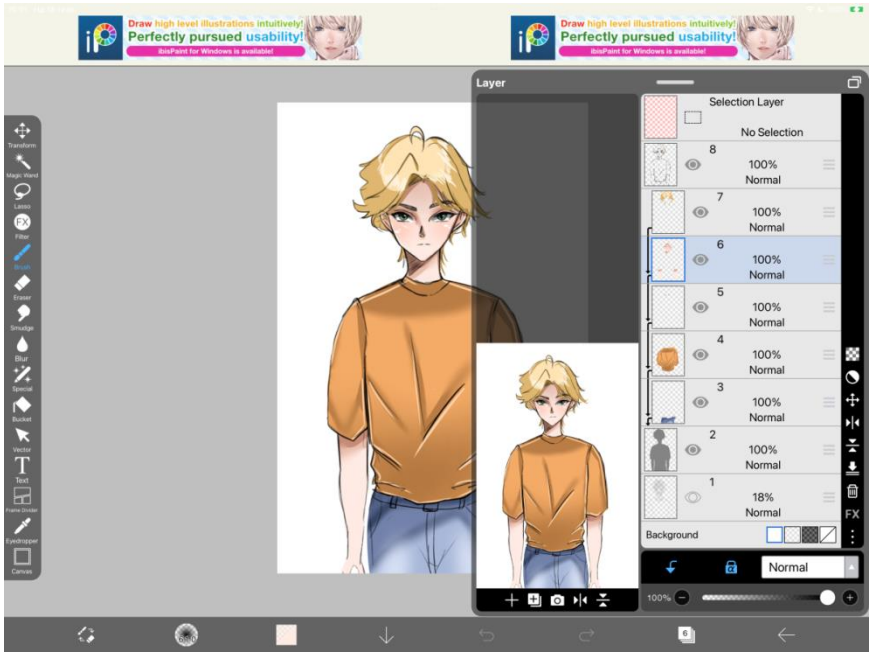


Рисунок 2.3 – Розробка персонажа Сашко

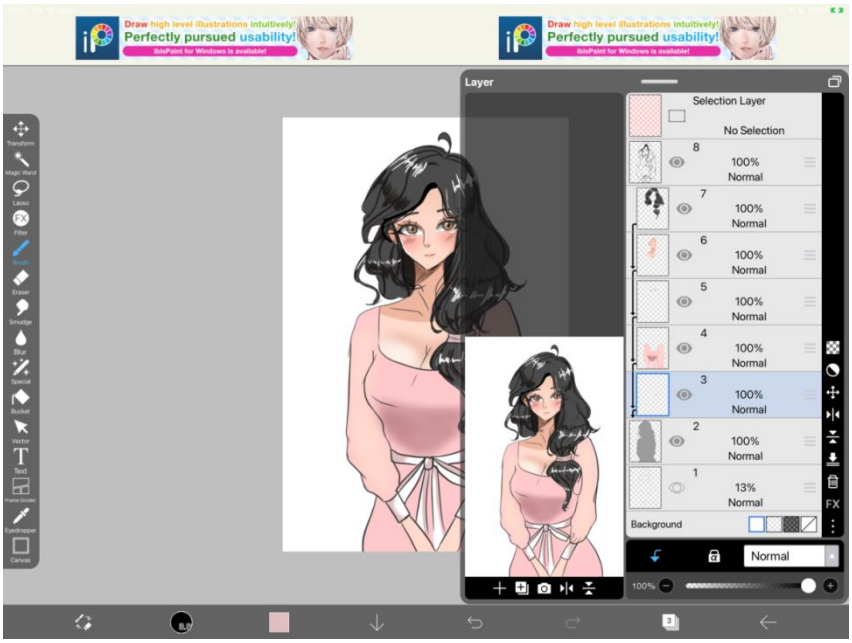


Рисунок 2.4 – Розробка персонажа тітка

Це дозволить безперешкодно інтегрувати їх у гру, розміщуючи поверх основного фону та змінюючи емоції залежно від розвитку сюжету

Отже, у програмі ibisPaint X на прозорому фоні було розроблено серію спрайтів персонажів з ключовими емоційними станами: нейтральним, радісним та сумним (Рисунок 2.5 – Спрайти).



Рисунок 2.5 – Спрайти

Створення персонажів у такому форматі значно спростить їхню подальшу інтеграцію в ігровий рушій, дозволяючи легко розміщувати їх поверх фонових зображень та динамічно змінювати вирази обличчя відповідно до сюжетних подій.

2.4 Розробка фонових зображень

Для візуального оформлення ігрових локацій застосовується техніка обробки реальних фотографій за допомогою художніх фільтрів у графічному редакторі ibisPaint X. Цей підхід вирізняється своєю практичністю та ефективністю, дозволяючи створювати естетично привабливі та рівномірні фонові зображення у стислі терміни. Завдяки широкому спектру фільтрів ibisPaint X забезпечується гнучкість у виборі художнього стилю, що гармонійно доповнює загальну атмосферу віртуальної новели (Рисунок 2.6 – Початкове зображення).

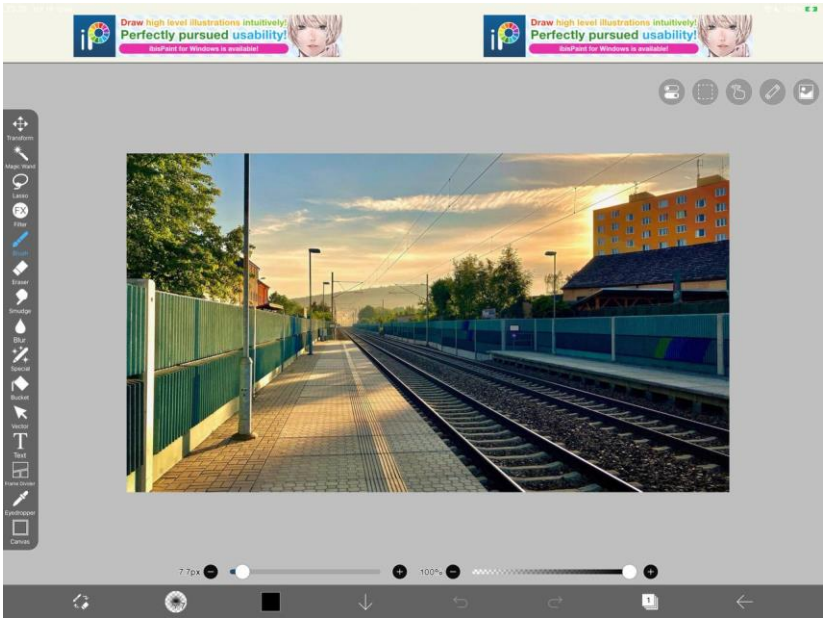


Рисунок 2.6 – Початкове зображення

Далі відбувається стилізація фотографії за допомогою художнього фільтра «Anime Background», що надає зображенню характерної естетики аніме. (Рисунок 2.7 – Фото з фільтром).

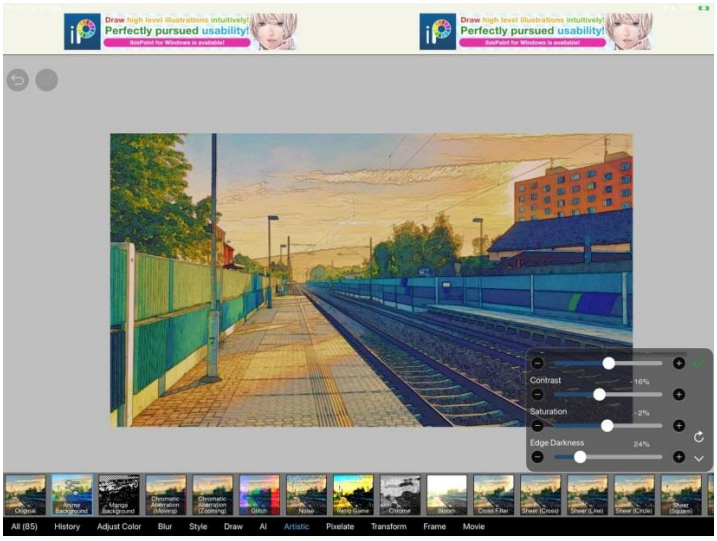


Рисунок 2.7 – Фото з фільтром

Після цього для підсилення візуальної глибини та створення більш м’якого освітлення додається шар із застосуванням ефекту «Hard Light» (Рисунок 2.8 – Результат обробки

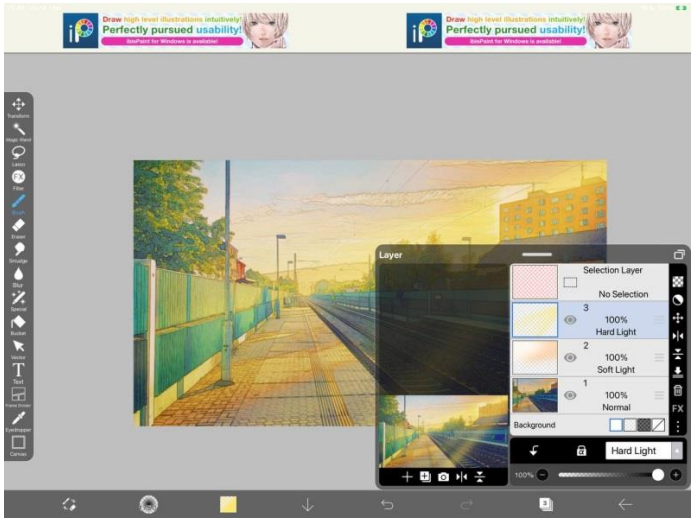


Рисунок 2.8 – Результат обробки

Застосування фільтрів у програмі ibisPaint X виявилося ефективним та зручним методом для створення фонових зображень. Цей підхід дозволив швидко досягти бажаного художнього стилю, зберігаючи при цьому чіткість та якість вхідних фотографій. Оброблені зображення гармонійно вписуються в загальну візуальну концепцію віртуальної новели, забезпечуючи необхідну атмосферу та естетичну привабливість.

2.5 Створення анімації

Для створення привабливого та динамічного вступу до гри було інтегровано анімаційний ролик. Ця анімація, що з’являється при запуску гри, містить назву новели та ім’я автора, створюючи унікальну візуальну візитівку проєкту.

Для розробки анімації було використано FlipaClip – мобільний додаток, призначений для створення 2D-анімації. Він надає широкий спектр інструментів для втілення анімаційних ідей, що робить його зручним як для початківців, так і для досвідчених аніматорів. FlipaClip дозволяє створювати анімаційні відео та GIF-файли, що було ідеально для реалізації вступного ролика (Рисунок 2.9 – Створення анімації



Рисунок 2.9 – Створення анімації

Завдяки цьому застосунку була створена анімація, яка ефективно представляє гру та її автора, занурюючи гравця в атмосферу проєкту ще до початку основної розповіді.

2.6 Створення іконки новели

Для створення власної іконки в Ren’Py грі процес є досить простим [4]: необхідно отримати файл window.icon з папки gui, імпортувати його в графічний редактор (наприклад, ibisPaint X) для створення унікального дизайну, а потім повернути змінений файл назад у ту ж папку gui (Рисунок 2.10 – Розробка іконки для гри), після чого оновлена іконка для гри буде успішно застосована (Рисунок 2.11 – Застосування іконки

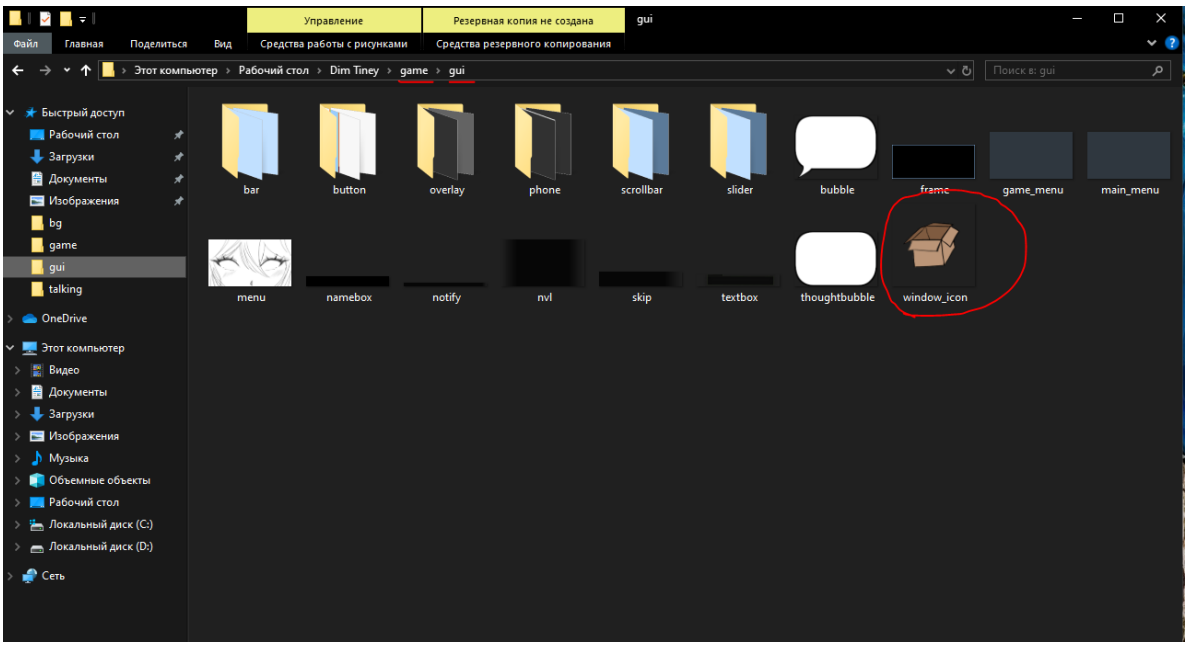


Рисунок 2.10 – Розробка іконки для гри

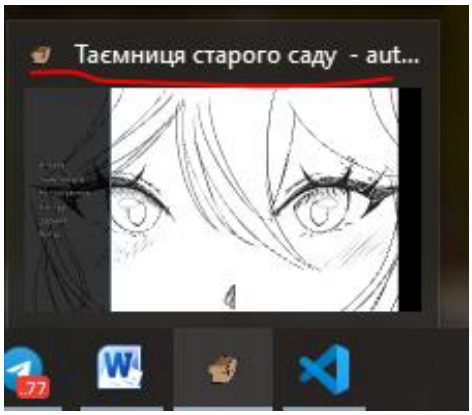


Рисунок 2.11 – Застосування іконки

Цей метод дозволяє легко персоналізувати візуальне представлення.

3 ТЕХНІЧНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ВІЗУАЛЬНОЇ НОВЕЛИ

3.1 Структура проєкту в Ren’Py

Для створення візуальної новели в Ren’Py було розроблено чітко структуровану файлову систему, яка забезпечує логічну організацію всіх компонентів гри [12].

Зокрема, у папці game/ або в Ren’py Launcher містяться основні скрипти проєкту: script.rpy є центральним файлом, що охоплює сюжет, діалоги, розгалуження та логіку переходів між сценами, а також визначає всі мітки та умови для альтернативних шляхів розвитку історії. Для налаштувань проєкту використовується options.rpy, який задає назву гри, версію, розмір вікна, параметри збережень та інші технічні показники, а також початкові значення змінних і режими роботи движка. Опис всіх екранів інтерфейсу користувача, таких як головне меню, налаштування та діалогові вікна, міститься у файлі screens.rpy, що відповідає за їхній зовнішній вигляд та поведінку. Натомість, візуальна концепція інтерфейсу, включаючи кольорову схему, шрифти, розміри кнопок та анімації переходів, налаштовується у файлі стилізації gui.rpy (Рисунок 3.1 – Файли скриптів).

У процесі розробки було реалізовано власноруч безліч унікальних елементів, що надають проєкту індивідуальності: це унікальні спрайти персонажів з їхніми емоційними станами, оригінальні фонові зображення локацій, кастомізований дизайн інтерфейсу та меню, а також авторський сценарій з розгалуженням сюжету, власна система діалогів з індивідуальним стилем для кожного персонажа та анімаційні переходи між сценами. Водночас, було ефективно використано стандартний функціонал Ren’Py, що дозволило оптимізувати процес розробки; до цього функціоналу належать базова система збереження/завантаження гри, стандартні переходи та ефекти

(dissolve, fade), вбудована система налаштувань (гучність, швидкість тексту), автоматичне масштабування під різні роздільності екрану, система історії діалогів та механізм пропуску прочитаного тексту.



Рисунок 3.1 – Файли скриптів

Таким чином, така архітектура проєкту забезпечує модульність коду, легкість внесення змін та ефективну організацію робочого процесу розробки.

3.2 Реалізація геймплейних елементів

У розробленій візуальній новелі система діалогів реалізована з використанням вбудованих можливостей Ren’Py [3]. Кожен персонаж має власні характеристики відображення тексту, що створює унікальну атмосферу спілкування (Рисунок 3.2 – Оголошення персонажу Марія

```
14  # Персонажі
15  define m = Character("Марія",
16      window_left_padding=-100,
17      who_ypos=15,
18      what_prefix="{cps=20}",
19      what_suffix="{/cps}",
20      who_prefix="{cps=15}",
21      who_suffix="{/cps}"
22  )
```

Рисунок 3.2 – Оголошення персонажу Марія

Для системи сцен, що побудована на принципі послідовного відображення фонових зображень, було використано різноманітні анімовані переходи для забезпечення плавності та динамічності [2]. Серед них – Fade, що є плавним затемненням і появою нової сцени, Dissolve, що забезпечує плавне розчинення одного зображення в іншому, а також Pan – горизонтальне переміщення камери. Прикладом такого переміщення є трансформація pan_right, яка переміщує зображення по осі X протягом 5 секунд, як показано в коді (Рисунок 3.3 – Приклад руху зображення за допомогою transform pan_right).

```
4  transform pan_right:
5      xpos 0
6      yalign 0.5
7      zoom 1.0
8      linear 5.0 xpos -717
```

Рисунок 3.3 – Приклад руху зображення за допомогою transform pan_right

Щоб персонажі з’являлися плавно та ефектно, застосовується метод трансформації, що змінюють їхні властивості з часом. Це може бути поступове проявлення (зміна прозорості), плавне переміщення в кадр, або комбінація цих ефектів [7].

Наприклад, трансформація titka_appear створює ефект поступової появи персонажа шляхом зміни його прозорості (alpha) (Рисунок 3.4 – Анімація появи персонажів).

```
10  transform titka_appear:
11      alpha 0.0 # Починаємо з повної прозорості (невидимий)
12      linear 0.5 alpha 1.0 # Протягом 0.5 секунди лінійно змінюємо прозорість до 1.0 (повністю видимий)
```

Рисунок 3.4 – Анімація появи персонажів

У цьому прикладі, titka_appear змушує персонажа («тітку») з'являтися на екрані, стаючи повністю видимим за півсекунди. Це створює м'який та непомітний перехід, що привертає увагу до нового елементу сцени без різких стрибків.

Для реалізації фонового зображення використовується команда show bg, яка відповідає за показ нового фону:

Також, було розроблено систему динамічної зміни спрайтів персонажів залежно від контексту діалогу [5]:

– спрайти з відкритим ротом для балакучих персонажів;
– спрайти з закритим ротом для мовчазних персонажів;
– плавні переходи між станами персонажів.

Цей механізм, який забезпечує плавні переходи між різними станами персонажів, детально показано у відповідному програмному коді:

Показуємо спрайт тітки з емоцією/позицією "t" (наприклад, коли вона говорить) з плавною появою (Рисунок 3.5 – Зміна спрайтів).

```
68      show titka_t with dissolve
69      # Відображає спрайт (зображення) персонажа "Тітка" (titka_t) на екрані коли вона говорить
70      t "Маріє! Привіт, дорога! Як доїхала?"
71      # Рядок діалогу, який вимовляє персонаж "Тітка" (t).
72
73      hide titka_t
74      # Приховує спрайт "Тітка", яка говорить
75      show titka with dissolve
76      # Відображає спрайт "Тітка" (titka) на екрані, у якої нейтральний вираз обличчя.
77
78      m "Привіт, тітко... Все добре. Тільки трохи незвично."
79      # Рядок діалогу, який вимовляє персонаж "Марія" (m).
```

Рисунок 3.5 – Зміна спрайтів

Було розроблено систему множинних кінцівок, що значно підвищує реіграбельність новели. До них належать: секретна кінцівка, яка досягається розкриттям секрету таємного саду; хороша кінцівка, де гравець успішно проходить квест, даючи всі правильні відповіді; нейтральна кінцівка, якщо гравець не відкриває коробку; і погана кінцівка, яка є результатом неправильних відповідей гравця в мінііграх, що перешкоджає друзям успішно завершити квест. Наприклад, у фрагменті, що відповідає за нейтральну кінцівку, рішення не відкривати коробку призводить до сцени завершення історії без розкриття головної таємниці:

У представленому коді реалізовано гілку діалогу, яка завершує сюжет нейтральним фіналом (Рисунок 3.6 – Нейтральна кіцівка гри.6).

```
# Початок сцени для варіанту "Не відкривати коробку"
label dont_open:
    m "Ні... мабуть, не варто. Це не наша річ. А й справді — тітка ж кликала на пиріг!"

    hide sashko_t
    show sashko at right
    hide nensi
    show nensi_t at left with dissolve
    n "Ой точно! Скоріше хочеться скуштувати!"

    hide nensi_t
    show nensi at left with dissolve
    hide sashko
    show sashko_t at right with dissolve
    s "Тоді гайда!"

    scene bg scene6 at pan_right with fade

    show titka_t with dissolve
    t "О, ви якраз вчасно! Пиріг тільки-но з печі. Сідайте!"

    hide titka_t
    show titka with dissolve

    m "{i}(Може, воно й на краще... Але чомусь у мене досі в голові крутиться образ тієї дивної коробки.)"


    # ! Кінцівка гри
    scene black with fade
    centered "{size=+10}КІНЕЦЬ{/size}"
    pause 2
    scene black
    return
```

Рисунок 3.6 – Нейтральна кіцівка гри

Ця кінцівка створює спокійну, проте не повністю задовільну атмосферу, залишаючи гравця з відчуттям невідомого. Такий підхід не лише дозволяє переосмислити пройдений сюжет, але й стимулює до повторного проходження гри, спонукаючи досліджувати альтернативні вибори для досягнення інших фіналів [10].

Також було реалізовано режим автозбереження гри, що значно підвищує зручність для користувача. Ця функція автоматично зберігає прогрес гравця у визначені моменти, наприклад, після завершення ключових сцен або розділів, використовуючи команду.

Це забезпечує надійний захист від втрати даних і гарантує, що гравці завжди зможуть продовжити з того місця, де зупинились, навіть у випадку непередбачених обставин. Про успішне збереження гравців інформує повідомлення: "Гру збережено автоматично".

3.3 Реалізація інтерфейсу користувача

Для створення живого враження від головного меню було реалізовано динамічну анімацію очей персонажа [8]. Очі слідкують за рухом курсора за допомогою функції обчислення зміщення eye_offset, яка оновлює положення зображень з частотою 100 разів на секунду. Це створює ефект взаємодії персонажа з гравцем уже на стартовому екрані.

Для реалізації такого ефекту було попередньо створено окремі зображення обличчя та окремі зображення для кожного ока (правого та лівого) на прозорому фоні. (Рисунок 3.7 – Анімація очей–Рисунок 3.8 – Створення очей).

```
383     # Python-блок для обчислення зміщення очей у напрямку курсору
384     python:
385         def eye_offset(pos, center, max_offset=10):
386             dx = min(max(pos[0] - center[0], -max_offset), max_offset)
387             dy = min(max(pos[1] - center[1], -max_offset), max_offset)
388             return dx, dy
389
390         # Отримуємо позицію курсору
391         mx, my = renpy.get_mouse_pos()
392         dx_l, dy_l = eye_offset((mx, my), eye_center_left)
393         dx_r, dy_r = eye_offset((mx, my), eye_center_right)
394
395     # Відображення зображення лівого ока з обчисленим зміщенням
396     add "eye_left.png" xpos eye_center_left[0] + dx_l ypos eye_center_left[1] + dy_l anchor (0.5, 0.5)
397     # Відображення зображення правого ока з обчисленим зміщенням
398     add "eye_right.png" xpos eye_center_right[0] + dx_r ypos eye_center_right[1] + dy_r anchor (0.5, 0.5)
```

Рисунок 3.7 – Анімація очей

Для підсилення занурення у гру було додано музичний супровід у головному меню [11]. Мелодія автоматично запускається під час відображення екрана головного меню, що підкреслює емоційний настрій новели (Рисунок 3.9 – Звукове оформлення).

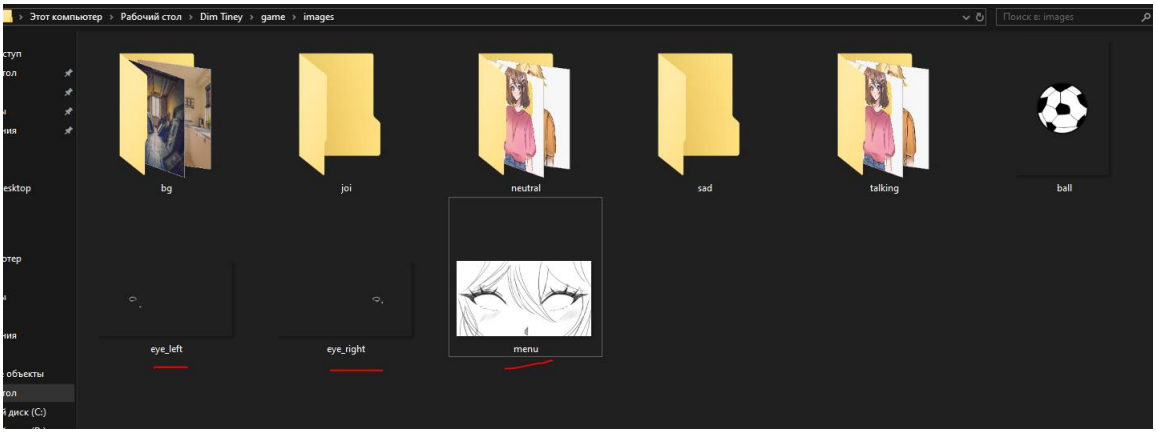


Рисунок 3.8 – Створення очей

```
367    # Музика меню
368    on "show" action Play("music", "audio/music.mp3")
```

Рисунок 3.9 – Звукове оформлення

Назву гри «Таємниця старого саду» було встановлено через змінну config.name у файлі options.rpy. Рядок огорнуто у функцію перекладу _(), що забезпечує підтримку мультимовності в майбутньому (Рисунок 3.10 – Назва гри).

```
15    define config.name = _("Таємниця старого саду")
```

Рисунок 3.10 – Назва гри

У файлі screens.rpy було створено екран навігаційного меню, що відповідає за основні кнопки (початок гри, збереження, завантаження, налаштування тощо). Крім того, для кнопок задано індивідуальні стилі та ефект наведення курсора (Рисунок 3.11 – Стиль кнопок).

```
style navigation_button:
    size_group "navigation"
    properties gui.button_properties("navigation_button")
    hover_background "#ff69b430"
    hover_sound "audio/sfx/hover.ogg"

style navigation_button_text:
    properties gui.text_properties("navigation_button")
    hover_color "#ff69b4"
    hover_size 30
    outlines [(2, "#000000", 0, 0)]
```

Рисунок 3.11 – Стиль кнопок

Це дозволяє візуально виділити меню та зробити взаємодію більш динамічною. Основні стилі:

hover_background – напівпрозорий рожевий фон при наведенні.

hover_color – колір тексту при наведенні.

hover_size – збільшення розміру шрифту.

outlines – чорна обводка тексту для кращої читабельності.

Цей ефект створює плавне збільшення кнопки при наведенні, що покращує візуальне сприйняття інтерфейсу (Рисунок 3.12 – Ефект збільшення кнопки).

```
transform button_hover_effect:
    on hover:
        linear 0.1 zoom 1.05
    on idle:
        linear 0.1 zoom 1.0
```

Рисунок 3.12 – Ефект збільшення кнопки

Усі кнопки в інтерфейсі розміщуються вертикально (vbox) та автоматично вирівнюються по центру екрана, забезпечуючи естетичний вигляд. Для динамічного відображення елементів, команда if main_menu: дозволяє адаптувати варіанти кнопок залежно від того, чи перебуває гравець у головному меню, чи вже занурений у гру (Рисунок 3.13 – Екран навігації).

```
## Екран навігації ############################################################
screen navigation():
    vbox:
        style_prefix "navigation"
        xpos gui.navigation_xpos
        yalign 0.5
        spacing gui.navigation_spacing

        if main_menu:
            textbutton _("Почати"):
                action Start()
                at button_hover_effect
```

Рисунок 3.13 – Екран навігації

Кожна кнопка має чітко визначену дію: наприклад, Start() починає гру, ShowMenu("save") відкриває меню збереження, а Quit() завершує ігровий сеанс.

У проєкті візуальної новели було індивідуально налаштовано шрифти та розміри тексту для різних елементів інтерфейсу [9]. Це дозволяє зробити візуальну подачу гри цілісною, читабельною та естетично привабливою (Рисунок 3.14 – Налаштування шрифтів).

```
## Шрифт, який використовується для тексту в грі.
define gui.text_font = "PTSerif-BoldItalic.ttf"

## Шрифт, який використовується для імен персонажів.
define gui.name_text_font = "PTSerif-Bold.ttf"

## Шрифт, який використовується для тексту поза грою.
define gui.interface_text_font = "PTSerif-Bold.ttf"
```

Рисунок 3.14 – Налаштування шрифтів

Шрифти, необхідні для коректного відображення тексту, поміщаються безпосередньо у папку game (Рисунок 3.15 – Шрифти).

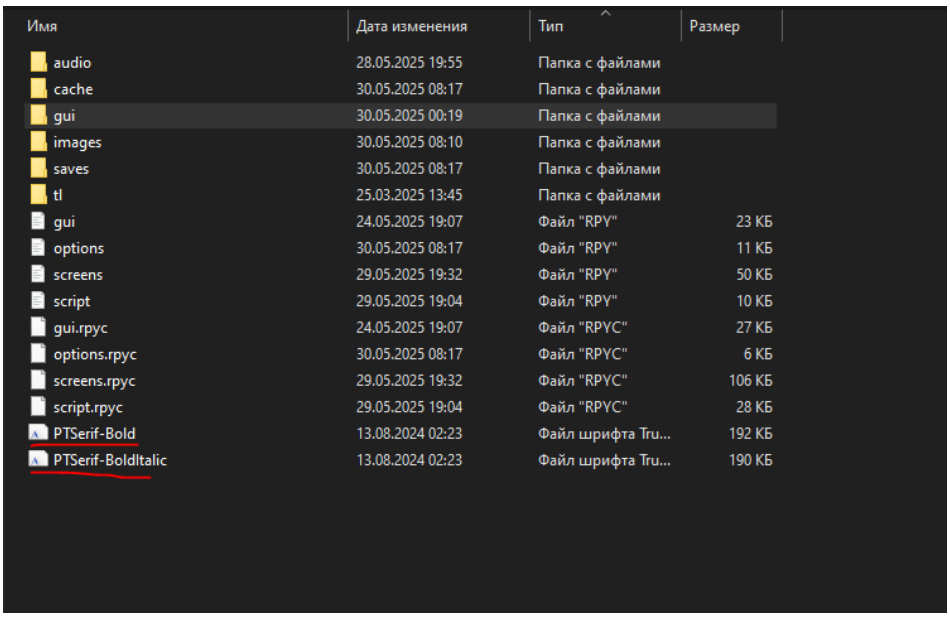


Рисунок 3.15 – Шрифти

Це дозволяє зробити візуальну подачу гри цілісною, читабельною та естетично привабливою.

3.4 Реалізація мініігор

Для збагачення ігрового досвіду та підвищення інтерактивності, було розроблено серію мініігор [13]. Ці інтегровані елементи не лише урізноманітнюють ігровий процес, але й відіграють ключову роль у розвитку сюжету та впливають на кінцівки. Кожна мінігра створювалася з метою перевірки логічних здібностей гравця, його уважності або вміння швидко приймати рішення, що додає глибини наративу.

У рамках проєкту було розроблено та інтегровано мінігру «Вікторина», що не лише урізноманітнює ігровий процес, а й може слугувати для перевірки знань гравця або занурення його в тематичний контекст. Ця міні-гра побудована на системі запитань із варіантами відповідей і включає механізми підрахунку балів та відстеження "життів" (спроб).

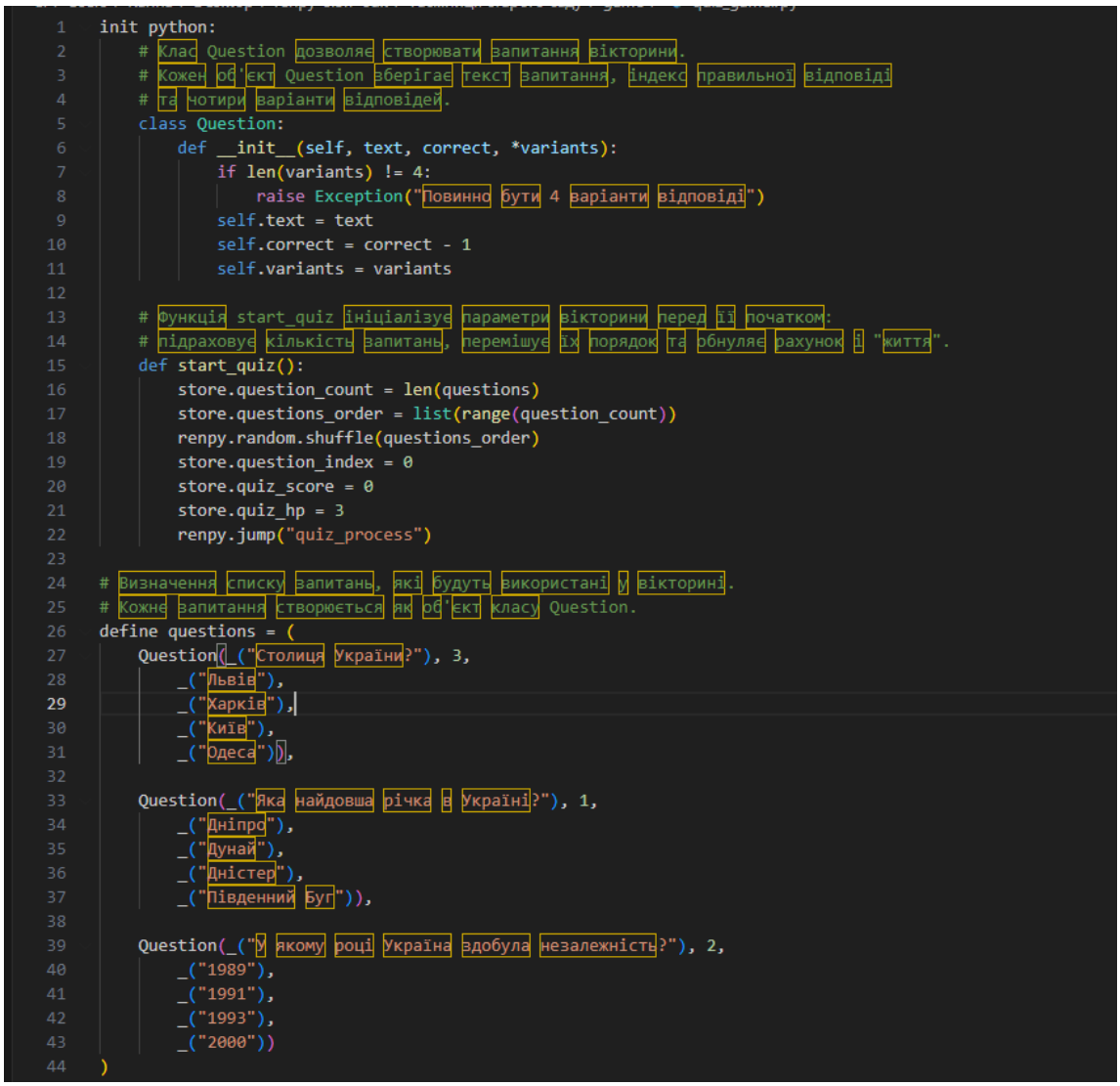
Основою вікторини є клас Question, визначений у блоці init python. Цей клас дозволяє легко створювати нові запитання, вказуючи текст запитання,

номер правильної відповіді та чотири варіанти відповідей. Такий підхід робить додавання нових запитань швидким і зручним, підтримуючи масштабованість гри.

Перед початком вікторини функція start_quiz() ініціалізує необхідні змінні: загальну кількість запитань, випадковий порядок їх відображення (щоб вікторина не була одноманітною при повторних проходженнях), поточний індекс запитання, поточний рахунок гравця та кількість доступних спроб (quiz_hp).

Візуальна частина вікторини реалізована за допомогою екранів Ren’Py. Екран quiz_screen відповідає за відображення самого запитання та кнопок з варіантами відповідей. Кожна кнопка пов’язана з відповідним варіантом і повертає його індекс при виборі. Екран quiz_result використовується для відображення миттєвого фідбеку гравцю – чи була відповідь правильною, чи ні.

Основний ігровий цикл вікторини знаходиться в мітці quiz_process. Тут послідовно відображаються запитання у випадковому порядку. Після кожної відповіді гравця система перевіряє її правильність, оновлює рахунок або зменшує кількість «життів». Якщо кількість «життів» падає до нуля, гравець програє, і вікторина завершена з відповідним повідомленням. Після проходження всіх запитань, гравцю відображається фінальний результат, який залежить від його успішності (Рисунок 3.16 – фрагмент коду мінігри «Вікторина).



```
init python:
    # Клас Question дозволяє створювати запитання вікторини.
    # Кожен об'єкт Question зберігає текст запитання, індекс правильної відповіді
    # та чотири варіанти відповідей.
    class Question:
        def __init__(self, text, correct, *variants):
            if len(variants) != 4:
                raise Exception("Повинно бути 4 варіанти відповіді")
            self.text = text
            self.correct = correct - 1
            self.variants = variants

    # Функція start_quiz ініціалізує параметри вікторини перед її початком:
    # підраховує кількість запитань, перемішує їх порядок та обнуляє рахунок і "життя".
    def start_quiz():
        store.question_count = len(questions)
        store.questions_order = list(range(question_count))
        renpy.random.shuffle(questions_order)
        store.question_index = 0
        store.quiz_score = 0
        store.quiz_hp = 3
        renpy.jump("quiz_process")

# Визначення списку запитань, які будуть використані у вікторині.
# Кожне запитання створюється як об'єкт класу Question.
define questions = (
    Question(_("Столиця України?"), 3,
        _("Львів"),
        _("Харків"),
        _("Київ"),
        _("Одеса")),

    Question(_("Яка найдовша річка в Україні?"), 1,
        _("Дніпро"),
        _("Дунай"),
        _("Дністер"),
        _("Південний Буг")),

    Question(_("У якому році Україна здобула незалежність?"), 2,
        _("1989"),
        _("1991"),
        _("1993"),
        _("2000"))
)
```

Рисунок 3.16 – фрагмент коду мінігри «Вікторина»

Для збагачення ігрового досвіду та додавання інтерактивних елементів до візуальної новели, було розроблено та інтегровано мінігру «Клікер». Ця гра проста в освоєнні, але водночас пропонує динамічну механіку, що вимагає від гравця швидкої реакції. Вона може використовуватися для винагороди гравця, активації певних сюжетних подій або як елемент для перевірки його спритності.

Основна ідея гри полягає в натисканні на кнопку якомога швидше, щоб накопичувати очки. Інтерфейс мінігри, реалізований за допомогою екрану clicker, відображає поточний рахунок гравця. Центральним елементом є велика кнопка «Натисни!», яка візуально анімується (transform pulse) при кожному кліку, створюючи ефект "пульсації" або "зменшення-збільшення".

Гра також включає індикатор "сили кліка" (click_power), що відображається у вигляді прогрес-бару. Цей показник розраховується на основі швидкості кліків гравця: чим швидше гравець клікає, тим вище значення "сили кліка". Це додає елемент стратегії та виклику, оскільки гравцю потрібно підтримувати високий темп натискань.

Логіка гри обробляється Python-функцією handle_click(), яка збільшує рахунок, перемикає анімацію кнопки та оновлює «силу кліка» на основі часу між кліками. Ігровий цикл (label clicker_loop) постійно оновлює "силу кліка", поступово зменшуючи її, якщо гравець не клікає, і перевіряє умови завершення гри (наприклад, досягнення певного рахунку або вихід гравця) (Рисунок 3.17 – Фрагмент коду мінігри «Клікер).

Також було розроблено та інтегровано мінігру «Розшифруй слово», що є різновидом словесної головоломки. Ця мінігра збагачує ігровий процес, пропонуючи гравцеві активне інтелектуальне завдання, яке може бути використане для перевірки словникового запасу, логічного мислення або просто для відпочинку від основного сюжету.

Основна ідея гри полягає у відновленні оригінального слова з його перемішаних літер. Ініціалізація гри відбувається в мітці scramble_game, де визначається список пар слів: оригінальне слово та його зашифрована версія. Змінна score відстежує успішність гравця.

```
    transform pulse:
    # Анімація кнопки при кліку.
        linear 0.05 zoom 0.9
        linear 0.1 zoom 1.0

    python:
    # Функція handle_click обробляє кліки, оновлює рахунок та силу.
        def handle_click():
            global score, pulse_toggle, click_power, last_click_time
            score += 1
            pulse_toggle = not pulse_toggle

            current_time = renpy.get_game_runtime()
            if last_click_time > 0:
                time_diff = current_time - last_click_time
                click_power = min(100, int(100 * (0.5 / max(0.01, time_diff))))
            else:
                click_power = min(100, click_power + 10)

            last_click_time = current_time

    show screen clicker

    # 🔁 Цикл через call-in-label з паузою
    label clicker_loop:
     # Цикл гри, що постійно оновлює стан та перевіряє умови завершення.
        if exit_game or score >= 100:
            jump clicker_done

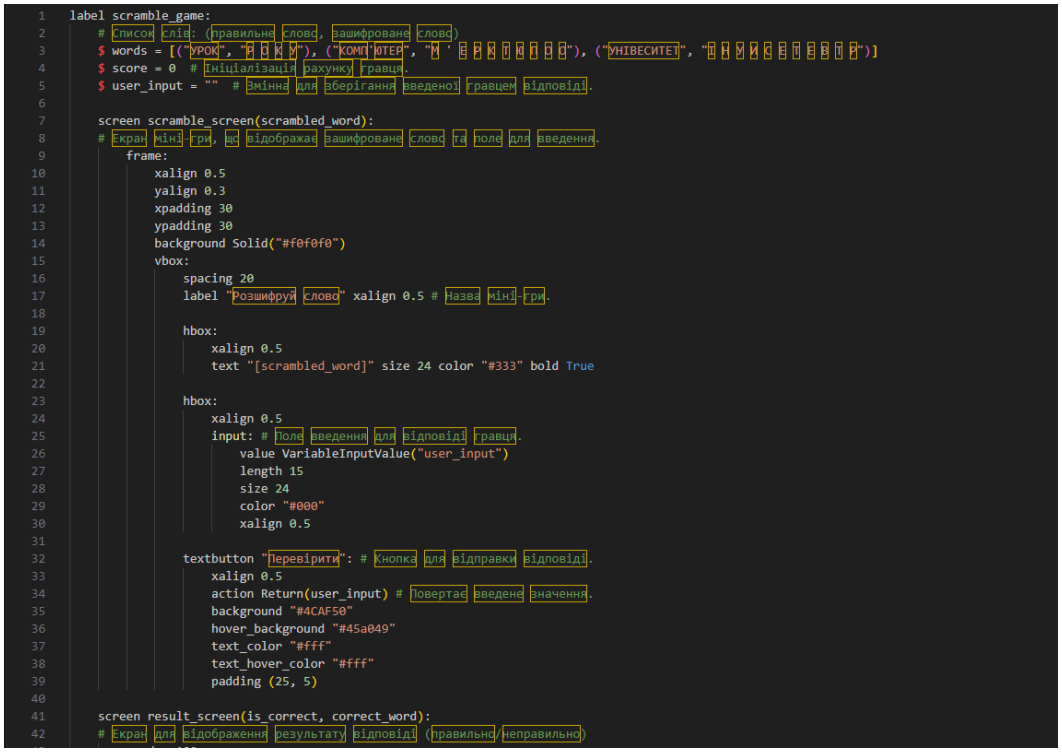
        $ click_power = max(0, click_power - 0.5)
        $ renpy.pause(0.1)
        jump clicker_loop
```

Рисунок 3.17 – Фрагмент коду мінігри «Клікер»

Візуальна частина гри реалізована за допомогою екранів Ren’Py. Екран scramble_screen є центральним елементом, який відображає переплутане слово і надає гравцеві поле для введення своєї відповіді. Після введення відповіді гравець натискає кнопку «Перевірити».

Логіка перевірки відповіді знаходиться в мітці next_word. Система порівнює введену гравцем відповідь з правильним словом. Залежно від результату, на короткий час відображається екран result_screen, який повідомляє, чи була відповідь правильною, і, у випадку помилки, показує правильне слово. Після цього рахунок гравця оновлюється (Рисунок 4 – Фрагмент коду мінігри «Розшифруй слово).

Гра триває доти, доки не закінчаться всі слова у списку. Після проходження всіх завдань, екран end_game_screen відображає підсумковий рахунок гравця, завершуючи мінігру.



```
label scramble_game:
    # Список слів: (правильне слово, зашифроване слово)
    $ words = [("УРОК", "Р О К У"), ("КОМП'ЮТЕР", "К ' Е Р К Т О П Ю"), ("УНІВЕРСИТЕТ", "Т Е Р И В У Н С І Т Е")]
    $ score = 0  # Ініціалізація рахунку гравця.
    $ user_input = ""  # Змінна для зберігання введеної гравцем відповіді.

    screen scramble_screen(scrambled_word):
    # Екран міні-гри, що відображає зашифроване слово та поле для введення.
        frame:
            xalign 0.5
            yalign 0.3
            xpadding 30
            ypadding 30
            background Solid("#f0f0f0")
            vbox:
                spacing 20
                label "Розшифруй слово" xalign 0.5 # Назва міні-гри.

                hbox:
                    xalign 0.5
                    text "[scrambled_word]" size 24 color "#333" bold True

                hbox:
                    xalign 0.5
                    input: # Поле введення для відповіді гравця.
                        value VariableInputValue("user_input")
                        length 15
                        size 24
                        color "#000"
                        xalign 0.5

                textbutton "Перевірити": # Кнопка для відправки відповіді.
                    xalign 0.5
                    action Return(user_input) # Повертає введене значення.
                    background "#4CAF50"
                    hover_background "#45a049"
                    text_color "#fff"
                    text_hover_color "#fff"
                    padding (25, 5)

    screen result_screen(is_correct, correct_word):
    # Екран для відображення результату відповіді (правильно/неправильно)
```

Рисунок 4 – Фрагмент коду мінігри «Розшифруй слово»

3.5 Налаштування компіляції та збірка гри

Для збірки фінальної версії гри на Ren’Py було використано розширений інструментарій платформи [6]. Це дозволило скомпілювати проєкт у формат, призначений для Windows (64-біт), створивши автономний виконуваний файл (.exe). Зібрана версія пройшла обов’язкове тестування на цільових пристроях, що підтвердило її коректну роботу та відповідність заявленим характеристикам (Рисунок 3.19 – Збірка гри).

Отже, фінальна версія візуальної новели тепер доступна як Windows executable (.exe), що забезпечує її максимальну доступність для користувачів цієї операційної системи.

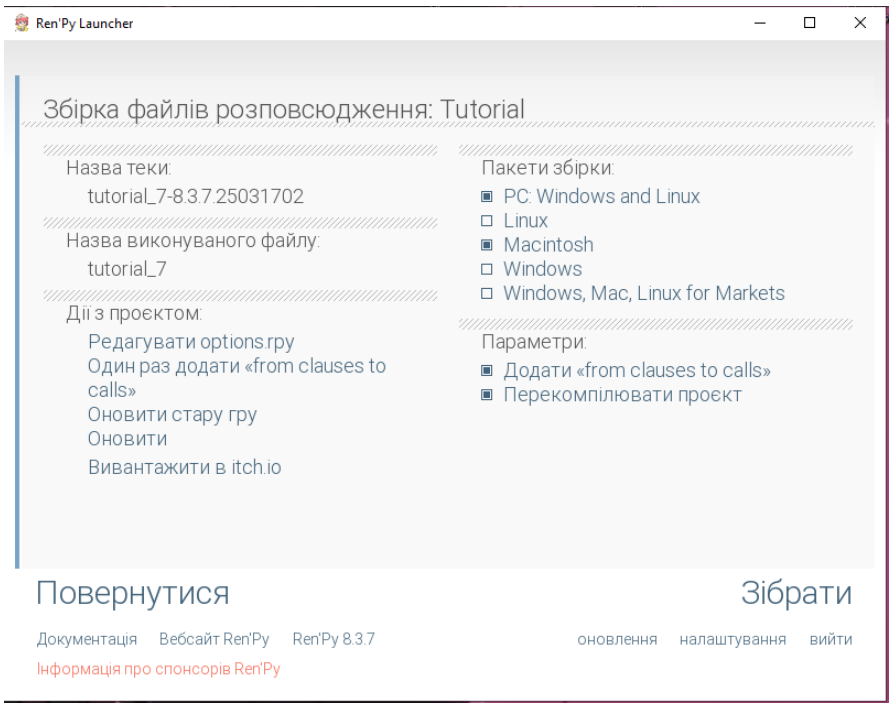


Рисунок 3.19 – Збірка гри

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи «Розробка візуальної новели за допомогою ігрового рушія Ren’Py» досягнуто низки важливих результатів, які переконливо підтверджують ефективність використання Ren’Py для створення інтерактивних та нелінійних сюжетних ігор. Основні висновки роботи є такими:

- проведений аналіз літературних джерел та програмних інструментів засвідчив, що Ren’Py становить собою потужний рушій для розробки візуальних новел. Він забезпечує підтримку сценарного програмування, інтерактивних виборів, інтеграцію мультимедійних елементів та кросплатформенну збірку. Його безкоштовна доступність та активна спільнота істотно сприяють швидкому освоєнню та успішній реалізації власних проєктів;
- у межах даної роботи розроблено повноцінну візуальну новелу з оригінальним сценарієм, нелінійним розвитком подій, численними кінцівками та комплексною системою вибору. Особливу увагу приділено деталізації дизайну персонажів, фонових зображень та створенню аудіовізуального супроводу, що значно підвищує рівень занурення гравця в сюжетну канву;
- реалізовано низку інтерактивних механік, серед яких вибір реплік, зміна гілок сюжету та наявність кількох варіантів завершення історії, що є характерними рисами якісних візуальних новел. Крім того, геймплей гри урізноманітнено шляхом додавання мініігор (вікторина, пазл, гра на пам’ять);
- у результаті компіляції проєкт було успішно зібрано у формати для Windows та Android. Тестування на різних пристроях підтвердило стабільну роботу гри, а всі виявлені помилки були оперативно

усунені. Отримані відгуки користувачів та власна оцінка розробника дозволяють дійти висновку про повну відповідність гри поставленим цілям та технічним вимогам;

– досвід створення візуальної новели в середовищі Ren’Py продемонстрував, що цей рушій є ефективним інструментом для реалізації творчих задумів. Сама розробка також володіє значним потенціалом для подальшого вдосконалення та розширення, включаючи локалізацію (переклад), покращення графічного оформлення та додавання нових сюжетних відгалужень.

ПЕРЕЛІК ПОСИЛАНЬ

1. Ren’Py: сайт. URL : https://www.renpy.org/wiki/renpy/rus/doc/tutorials/Руководство_для_начинающих (дата звернення : 20.03.2025).

2. Youtube. “ 3D камера в РенПай – Уроки RenPy #40 | Космо”: сайт. URL : https://www.youtube.com/watch?v=ibrA4OBx1bg (дата звернення : 20.04.2024).

3. Youtube. “ RenPy | Урок 2. Необхідний мінімум: виклик спрайтів та фонів, робота з аудіо” : сайт. URL : https://www.youtube.com/watch?v=w_QzJZ76cWA&t=98s (дата звернення : 28.03.2024).

4. Youtube. “ Как изменить иконку игры и текст в разделе "Об игре"? Уроки RenPy #17 | Космо” : сайт. URL : https://www.youtube.com/watch?v=GUpsgnY7thU&list=PLYU2dsNxZ7IQ-7eFJrB-Wrn3TZFZUjtdN&index=19 (дата звернення : 20.05.2024).

5. Youtube. “ Как легко менять эмоции персонажей в РенПай? Уроки RenPy #9 | Космо” : сайт. URL : https://www.youtube.com/watch?v=LW9NiGHaGHo&list=PLYU2dsNxZ7IQ-7eFJrB-Wrn3TZFZUjtdN&index=10 (дата звернення : 28.03.2024).

6. Youtube. “ Сборка новеллы для ПК (Windows / Mac / Linux). Уроки RenPy #38 | Космо” : сайт. URL : https://www.youtube.com/watch?v=IeGJuQRwOzo (дата звернення : 30.05.2024).

7. Youtube. “ Урок 20. Переходи в RenPy” : сайт. URL : https://www.youtube.com/watch?v=vrS8JoMZGPo&list=PLyNL5FPKoKetgrPH6bg5-7K4Q3-TWqPDM&index=20 (дата звернення : 20.04.2024).

8. Youtube. “ Урок 5. Найпростіші анімації в RenPy : SnowBlossom, іскри, анімація об’єкту на фоні, зсув спрайтівУрок.” : сайт. URL : https://www. youtube.com/watch?v=uh7pDuJJQaM (дата звернення : 05.04.2024).

9. Youtube. “ Урок 6. Кастомізація головного меню новели. Заміна шрифтів.” : сайт. URL : https://www.youtube.com/watch?v=9UwSv5y5UA0&t=5s (дата звернення : 10.04.2024).

10. Youtube. “ Урок 7. Як створити меню виборів у візуальній новелі” : сайт. URL : https://www.youtube.com/watch?v=JVMZI2OoOtM&list=PLyNL5FPKoKetgrPH6bg5-7K4Q3-TWqPDM&index=7 (дата звернення : 20.04.2024).

11. Youtube. “ Урок 8. Музика в головному меню та анімований курсорф” : сайт. URL : https://www.youtube.com/watch?v=TMuIIpCPeiU&list=PLyNL5FPKoKetgrPH6bg5-7K4Q3-TWqPDM&index=8 (дата звернення : 20.04.2024).

12. Youtube. “ Як створити візуальну новелу в RenPy? Урок 1. Створення проєкту” : сайт. URL : https://www.youtube.com/watch?v=qs0wygx5k0A (дата звернення : 28.03.2024).

13. Документація Ren’Py : сайт. URL : https://www.renpy.org/doc/html/ (дата звернення : 25.05.2025).