ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«УНІВЕРСИТЕТ ЕКОНОМІКИ ТА ПРАВА «КРОК»

КВАЛІФІКАЦІЙНА РОБОТА

Тема: «Комп'ютерна гра у жанрі візуальної новели з використанням алгоритмів генерації сюжетних ліній»

Ступінь вищої освіти - бакалавр
Спеціальність - 122 «Комп’ютерні науки»
Освітня програма «Комп’ютерні науки»

ПОЯСНЮВАЛЬНА ЗАПИСКА

Виконав: здобувач 4 курсу
групи КН-22
Михайло РАБЧЕВСЬКИЙ

Керівник: доцент кафедри інформаційного менеджменту, математики та статистики, к.ф.- м.н.
Віра ТКАЧЕНКО

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

м. Київ - 2026 рік

ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«УНІВЕРСИТЕТ ЕКОНОМІКИ ТА ПРАВА «КРОК»

ЗАТВЕРДЖУЮ:
завідувач кафедри
комп’ютерних наук
_____Сергій МІЧКІВСЬКИЙ
«__»____20___р

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Рабчевський Михайло Сергійович

Тема роботи	Комп'ютерна гра у жанрі візуальної новели з використанням алгоритмів генерації сюжетних ліній
Номер та дата наказу про затвердження теми	№ 133-7 від 22 грудня 2025 року
Коротка постановка завдання	Розробити гру візуальну новеллу з використанням двигуна Ren`Py. Реалізувати такі механіки: 1. Систему наративу з візуальними елементами 2. Рольовий вибір змінюючий сюжетний скрипт гри
Посилання на джерела інформації (не більше п’яти найменувань, які рекомендує науковий керівник)	1. Ren'Py Documentation. URL: https://www.renpy.org/doc/html/ 2. Плейліст Ren`Py Basics https://www.youtube.com/watch?v=u_cF3mXLTck&list=PL364lMeCMEsCA1a-1QiAMvHMFg-hdrnwl
Вимоги до кваліфікаційної роботи	Кваліфікаційна робота має передбачити теоретичне, системотехнічне або експериментальне дослідження складного спеціалізованого завдання або практичної проблеми в галузі комп’ютерних наук, яке характеризується комплексністю та невизначеністю умов і потребує застосування теорій і методів інформаційних технологій.

Дата видачі завдання 25 грудня 2025 р.

Керівник Віра ТКАЧЕНКО

Здобувач освітнього ступеня бакалавра Михайло РАБЧЕВСЬКИЙ

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання	Примітка
	Підготовчий етап		
1	Вибір напряму дослідження	01.12.2025 р.	*виконано*
2	Формування теми та призначення керівника	15.12.2025 р.	*виконано*
3	Затвердження теми кваліфікаційної роботи	22.12.2025 р.	*виконано*
4	Затвердження завдання на кваліфікаційну роботу	26.12.2025 р.	*виконано*
	Основний етап		
5	Розробка концепції кваліфікаційної роботи	12.01.2026 р.	*виконано*
6	Підбір та вивчення джерел інформації з напряму дослідження. Огляд існуючих аналогів	19.01.2026 р.	*виконано*
7	Затвердження розширеної постановки завдання. Підготовка та подання керівникові розділу 1 кваліфікаційної роботи	16.03.2026 р.	*виконано*
8	Проєктування. Підготовка та подання керівникові розділу 2 кваліфікаційної роботи	23.03.2026 р.	*виконано*
9	Підготовка доповіді для експертизи стану виконання кваліфікаційної роботи (проміжний контроль)	30.03-04.04.2026 р.	*виконано*
10	Реалізація. Підготовка та подання керівникові розділу 3 кваліфікаційної роботи	06.04.2026 р.	*виконано*
11	Підготовка та подання керівнику першого варіанту всієї кваліфікаційної роботи	13.04.2026 р.	*виконано*
12	Доопрацювання кваліфікаційної роботи з урахуванням зауважень керівника та представлення керівникові доопрацьованого варіанту кваліфікаційної роботи	20.04.2026 р.	*виконано*
	Завершальний етап		
13	Представлення рукопису для перевірки на плагіат	27.04-03.05.2026 р.	*виконано*
14	Підготовка презентації та доповіді на передзахист	04.05-10.05.2026 р.	*виконано*
15	Передзахист кваліфікаційної роботи	11.05-15.05.2026 р.	*виконано*
16	Доопрацювання роботи за результатами передзахисту	18.05-05.06.2026 р.	*виконано*
17	Експертиза роботи керівником та зовнішнім експертом	08.06-14.06.2026 р.	*виконано*
18	Доопрацювання доповіді та презентації для захисту	08.06-14.06.2026 р.	*виконано*
19	Захист кваліфікаційної роботи	15.06-21.06.2026 р.	*виконано*

Керівник Віра ТКАЧЕНКО

Здобувач освітнього ступеня бакалавра Михайло РАБЧЕВСЬКИЙ

Рабчевський М.С. Комп'ютерна гра у жанрі візуальної новели з використанням алгоритмів генерації сюжетних ліній.

Пояснювальна записка кваліфікаційної роботи за спеціальністю 122 - Комп’ютерні науки (освітня програма - Комп’ютерні науки) СО Бакалавр. - ВНЗ «Університет економіки та права «КРОК», Навчально-науковий інститут інформаційних та комунікаційних технологій, кафедра комп’ютерних наук, Київ, 2026.

Розглянуто підходи до проєктування та реалізації інтерактивної наративної системи з алгоритмічною обробкою розвитку подій на основі рішень користувача. Розроблено програмний продукт, що реалізує механізми морального вибору, взаємодії з персонажами та алгоритмічної обробки наслідків користувацьких рішень.

Ключові слова: комп’ютерна гра, Ren`Py, візуальна новелла.

Кількість ілюстрацій - 27, таблиць - 1, використаних джерел - 13.

Rabchevskyi M.S. Computer game in the visual novel genre using algorithms for generating storylines.

Project explanatory note by specialty 122 - Computer science. - «KROK» University, Educational and Scientific Institute of information and communication technologies, Department of Computer Science, Kyiv, 2026.

Approaches to the design and implementation of an interactive narrative system with algorithmic processing of events based on user decisions. A software product has been developed that implements mechanisms of moral choice, interaction with characters, and algorithmic processing of the consequences of user decisions.

Keywords: computer game, Ren`Py, visual novel.

Number of figures - 27, tables - 1, references - 13.

ЗМІСТ

ВСТУП

Актуальність теми. Інтерактивні програмні продукти, орієнтовані на сюжет та користувацький вибір, набувають дедалі більшої популярності у сфері цифрових розваг і навчальних систем [1]. Особливе місце серед них займають наративні ігрові рішення, в яких поведінка системи адаптується до рішень користувача. Водночас більшість існуючих продуктів обмежуються поверхневими варіантами розгалуження сюжету без глибокого алгоритмічного моделювання моральних наслідків, довгострокової репутації або психологічних характеристик персонажів.

З технічної точки зору створення подібних систем потребує поєднання методів системного аналізу, моделювання поведінки, проєктування архітектури програмного забезпечення та реалізації механізмів збереження станів. Це зумовлює актуальність дослідження підходів до розробки програмного продукту, який поєднує сюжетну логіку, змінні стану (довіра, моральний профіль, репутація), алгоритмічну обробку наслідків та багаторівневу структуру системи.

Розробка інтерактивної системи «Магічна аптека» як програмного продукту дозволяє продемонструвати застосування сучасних методів комп’ютерних наук у контексті створення адаптивної наративної моделі з відкладеними наслідками рішень користувача. Таким чином, обрана тема є актуальною як у прикладному, так і в навчально-дослідницькому аспекті.

Мета дослідження: проєктування та реалізація програмного продукту - інтерактивної наративної системи з механізмом морального вибору та адаптивної сюжетної логіки із застосуванням методів системного аналізу, UML-моделювання та алгоритмічного проєктування, зокрема алгоритмів формування сценарної траєкторії та фінального результату залежно від дій користувача і поточного стану системи.

У межах даної кваліфікаційної роботи поняття алгоритмів генерації сюжетних ліній трактується як алгоритмічне формування індивідуальної траєкторії проходження в межах наперед визначеної системи сценарних вузлів,

переходів і кінцевих станів. У цьому випадку генерація реалізується не як автоматичний синтез нових сюжетів, а як визначення доступних варіантів розвитку подій, переходів між сценами та фінального результату на основі вибору користувача і поточних значень внутрішніх параметрів системи.

Завдання дослідження. Для досягнення поставленої мети необхідно:

- проаналізувати предметну область, здійснити порівняльний аналіз аналогів та сформувати характеристику цільової аудиторії програмного продукту;
- сформулювати концепцію системи, визначити функціональні та нефункціональні вимоги, побудувати UML-моделі поведінки та структури;
- спроєктувати архітектуру програмного продукту та логіку обробки змінних стану (рівень довіри, моральний профіль, репутація);
- реалізувати програмний продукт із використанням обраної платформи та технологічних засобів, абезпечивши логічну організацію основних складових системи;
- провести функціональне та модульне тестування, оцінити відповідність реалізації визначеним вимогам.

Об’єктом дослідження є процес створення програмного продукту з адаптивною сюжетною логікою та механізмом збереження станів користувацьких рішень.

Предметом дослідження є методи проєктування та реалізації програмного забезпечення для наративної системи з алгоритмічним керуванням переходами між сценарними вузлами, обробкою моральних і поведінкових параметрів та формуванням варіантів розвитку подій.

У роботі використано наступні **методи дослідження**: методи системного аналізу, UML-моделювання (діаграми варіантів використання, діяльності, класів, послідовності), алгоритмічного проєктування, використання базових принципів програмування та структур даних, порівняльного аналізу програмних аналогів, а також методи функціонального та модульного тестування.

Наукова новизна роботи полягає у розробці моделі адаптивної наративної системи, у якій сценарна траєкторія формується алгоритмічно на основі вибору користувача та накопичених параметрів стану, що забезпечує відкладені сюжетні наслідки в межах заданої структури сценарію.

Практичне значення роботи полягає у створенні завершеного програмного продукту з реалізованою логікою адаптивного сюжету, який може бути використаний як демонстраційна модель наративної системи, база для подальшого розвитку або приклад реалізації алгоритмічного моделювання поведінки в інтерактивному середовищі.

Структура та обсяг роботи. Кваліфікаційна робота складається зі вступу, трьох розділів, висновків, переліку джерел посилання та додатків. У першому розділі розглянуто предметну область і сформульовано постановку задачі, у другому - здійснено проєктування архітектури та моделей системи, у третьому - описано реалізацію та результати тестування програмного продукту. Загальний обсяг роботи становить 56 сторінок.

РОЗДІЛ 1
АНАЛІТИЧНЕ ОБҐРУНТУВАННЯ РОЗРОБКИ ПРОГРАМНОГО ПРОДУКТУ

1.1 Характеристика предметної області інтерактивних наративних систем

Світова індустрія відеоігор демонструє стабільну тенденцію до зростання та є одним із найбільш динамічних секторів цифрових розваг. Як показано на рисунку 1.1, обсяг глобального ринку відеоігор у 2024 році становить приблизно 274,63 млрд доларів США. За прогнозами аналітичних досліджень, протягом наступного десятиліття очікується значне зростання цього показника - до понад 721 млрд доларів США у 2034 році. Така динаміка розвитку обумовлена збільшенням кількості користувачів цифрових платформ, розвитком інтернет-інфраструктури, а також поширенням різноманітних форматів інтерактивного цифрового контенту.

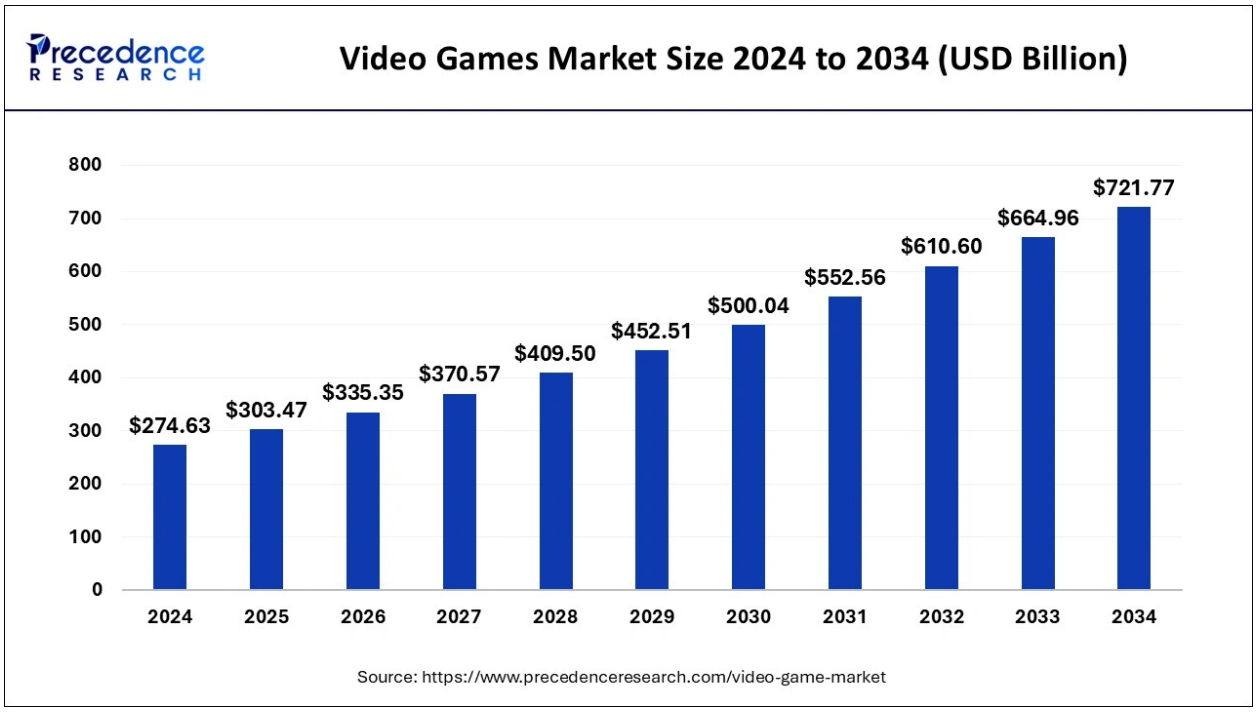

Рисунок 1.1 - Діаграма зростання ринку ігор, млрд.д.США

Джерело: [2].

Розвиток ігрового ринку сприяє появі нових типів програмних продуктів, зокрема інді-проєктів та наративно орієнтованих ігор. У цьому сегменті особливу роль відіграють інтерактивні системи, у яких основна увага приділяється сюжетній взаємодії користувача з програмним середовищем. До таких систем належать ігри у жанрі візуальної новели, де сюжет розвивається залежно від вибору користувача, а програмна логіка обробляє наслідки прийнятих рішень[3]. Зростання популярності подібних програмних продуктів обумовлює актуальність дослідження підходів до розробки інтерактивних наративних систем, що поєднують сюжетну структуру, змінні стану та алгоритмічну обробку поведінкових параметрів користувача.

Інтерактивні наративні системи є одним із напрямів розвитку цифрових розважальних і освітніх програмних продуктів. Їх основною особливістю є поєднання художнього наративу та алгоритмічної логіки, що дозволяє користувачеві впливати на перебіг подій у програмному середовищі. На відміну від традиційних лінійних історій, у таких системах подальший розвиток сюжету визначається алгоритмічно на основі вибору користувача[4].

У сучасній індустрії програмного забезпечення інтерактивні наративні системи використовуються переважно у відеоіграх жанру visual novel, interactive fiction, narrative-driven games. У цих системах користувач взаємодіє з програмним середовищем через вибір діалогів, прийняття рішень або виконання певних дій, які змінюють подальший розвиток історії.

Ключовими елементами таких систем є:

- сценарна структура (послідовність сцен і діалогів);
- система вибору (menu choices);
- змінні стану (репутація, довіра, моральний профіль);
- механізми розгалуження сюжету;
- збереження стану користувацьких рішень.

Однією з важливих особливостей інтерактивних наративних систем є можливість формування відкладених наслідків рішень. Це означає, що вибір

користувача може впливати на події, які відбуваються значно пізніше в процесі взаємодії із системою.

У контексті програмної реалізації такі системи поєднують механізми управління станами, алгоритми логічної обробки подій, моделі поведінки персонажів, системи збереження даних.

З технічної точки зору інтерактивна наративна система є програмним продуктом, який включає модулі сценарної логіки, інтерфейсу користувача, керування даними та обробки подій.

У рамках цієї роботи предметна область пов’язана з розробкою програмного продукту «Магічна аптека», що представляє собою інтерактивну наративну систему у жанрі візуальної новели. У системі користувач виступає в ролі алхіміка-аптекаря, який взаємодіє з клієнтами, аналізує їхні історії, обирає варіанти діалогів і приймає рішення щодо створення магічних рецептів.

Особливістю предметної області є поєднання: психологічної взаємодії з персонажами, алгоритмічної оцінки поведінки, системи моральних виборів, моделювання довгострокових наслідків рішень.

1.2 Порівняльний аналіз існуючих програмних рішень

Сучасні інтерактивні ігрові системи, орієнтовані на сюжет та взаємодію з користувачем, представлені широким спектром програмних продуктів, що поєднують наративні елементи, механіки прийняття рішень та різні ігрові системи. До таких продуктів належать інтерактивні історії, рольові ігри, симулятори та візуальні новели, у яких користувач взаємодіє з програмним середовищем через діалоги, вибір дій або виконання певних завдань. Особливістю подібних систем є наявність сценарної логіки, що дозволяє змінювати перебіг подій залежно від рішень користувача, а також використання змінних стану, які впливають на розвиток сюжету, поведінку персонажів та результати взаємодії.

Для аналізу було обрано програмні продукти, які мають схожі концептуальні або механічні елементи з розроблюваною системою «Магічна

аптека». Основними критеріями відбору конкурентів стали: наявність наративної складової, використання системи моральних або поведінкових рішень, інтерактивна взаємодія з персонажами, а також наявність ігрових механік, пов'язаних із аналізом ситуацій або створенням ігрових об'єктів. У результаті - було обрано три програмні рішення: Papers, Please, Potion Craft та Disco Elysium, які представляють різні підходи до реалізації наративних ігрових систем та дозволяють визначити основні функціональні та концептуальні особливості сучасних інтерактивних ігор.

Для визначення особливостей та конкурентних переваг розроблюваного програмного продукту було проведено аналіз існуючих програмних рішень, що належать до жанру інтерактивних наративних ігор.

Papers, Please (рис. 1.2) - це інді-гра, у якій користувач виконує роль прикордонного інспектора та приймає рішення щодо допуску людей через кордон. Основна механіка гри побудована на аналізі інформації та моральних виборах.

Рисунок 1.2 - гра Papers, Please

Джерело: [5].

Переваги:

- сильна моральна складова;
- наявність наслідків рішень;
- атмосфера психологічного тиску.

Недоліки:

- обмежена система діалогів;
- невелика кількість сценарних гілок;
- відсутність глибокої системи взаємодії з персонажами.

Potion Craft (рис. 1.3) - це симулятор алхіміка, у якому гравець створює різні зілля для клієнтів. Основна увага приділяється механіці створення рецептів та управлінню інгредієнтами.

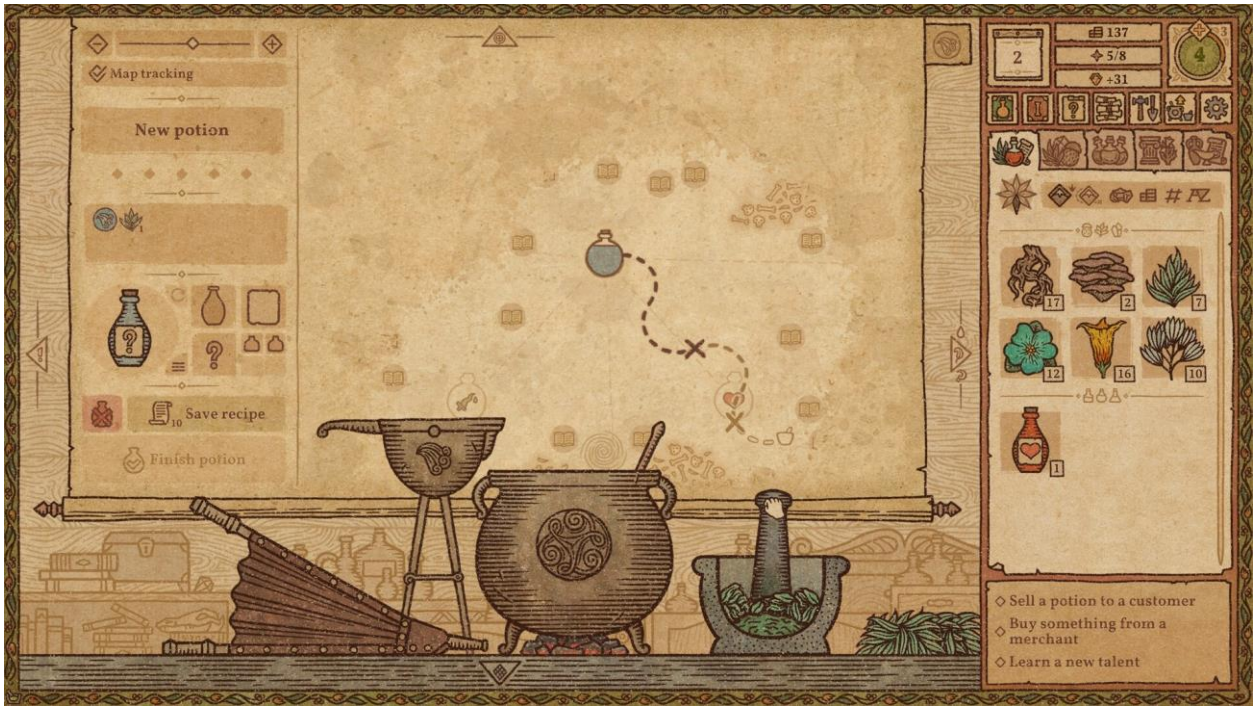

Рисунок 1.3 - гра Potion Craft

Джерело: [6].

Переваги:

- детальна система алхімії;
- атмосферний дизайн;
- інтерактивний процес створення зілля.

Недоліки:

- обмежена сюжетна складова;
- мінімальна система моральних рішень;
- відсутність складної поведінкової моделі персонажів.

Disco Elysium (рис. 1.4) - рольова гра з глибокою наративною структурою та системою психологічних параметрів персонажа.

Рисунок 1.4 - гра Disco Elysium

Джерело: [7].

Переваги:

- складна система діалогів;
- психологічна модель персонажів;
- багаторівнева сюжетна структура.

Недоліки:

- висока складність системи;
- значні ресурси для розробки;
- відсутність вузької тематичної спеціалізації.

Для більш наочного порівняння функціональних особливостей розглянутих програмних продуктів було сформовано узагальнену таблицю 1.1, у якій наведено основні характеристики аналізованих ігор.

На основі аналізу можна визначити ключові характеристики сучасних наративних систем:

Таблиця 1.1 - Порівняльний аналіз конкурентів

Характеристика	**Papers, Please**	**Potion Craft**	**Disco Elysium**
Жанр	Наративна інді-гра, симулятор інспектора	Симулятор алхіміка	Наративна рольова гра
Основна механіка	Аналіз документів та прийняття рішень	Створення зілля та управління інгредієнтами	Діалогова взаємодія та розвиток персонажа
Тип взаємодії з користувачем	Аналіз інформації та вибір дій	Створення рецептів та обслуговування клієнтів	Діалоги, дослідження світу, прийняття рішень
Наявність моральних рішень	✓	частково	✓
Система персонажів	частково	обмежена	✓
Алгоритмічна логіка	✓	✓	✓
Сюжетна глибина	середня	низька	висока
Варіативність розвитку подій	обмежена	мінімальна	висока
Складність системи	середня	середня	висока

Класифікація жанрів також підтверджується системою тегів сучасних ігрових платформ [8]

На основі проведеного порівняльного аналізу можна зробити висновок, що сучасні наративні ігрові системи поєднують різні підходи до реалізації інтерактивності: аналіз інформації та моральні рішення, механіки створення ігрових об’єктів, а також складні системи діалогів і розвитку персонажів. Розроблювана система «Магічна аптека» поєднує окремі елементи цих підходів,

зокрема систему моральних рішень, взаємодію з персонажами та механіку створення магічних рецептів.

Розроблювана система **«Магічна аптека»** поєднує елементи цих підходів, зокрема:

- систему моральних рішень;
- психологічну взаємодію з персонажами;
- механіку створення магічних рецептів;
- систему відкладених наслідків.

Це дозволяє створити інтерактивну систему, яка поєднує наративну складову та алгоритмічну обробку поведінкових параметрів.

Цільовою аудиторією програмного продукту є користувачі, які цікавляться сюжетно-орієнтованими інді-іграми та візуальними новелами, готові до текстової взаємодії з персонажами та прийняття рішень, що впливають на фінальний результат. Потенційний користувач не потребує спеціальних технічних навичок, однак має бути зацікавлений у послідовному проходженні сценарію, аналізі ситуацій та виборі варіантів дій. Таким чином, продукт орієнтований передусім на аудиторію, для якої важливими є наративна складова, моральний вибір і варіативність завершення гри.

1.3 Формування вимог до системи «Магічна аптека»

На основі аналізу предметної області та існуючих програмних рішень було сформовано основні вимоги до програмного продукту «Магічна аптека».

Система повинна забезпечувати такі функціональні вимоги:

1. реалізацію діалогової взаємодії між персонажами;
2. надання користувачеві можливості вибору одного з доступних варіантів дії у межах сценарного вузла;
3. збереження та оновлення внутрішніх параметрів системи, зокрема показників карми, репутації та допоміжного параметра довіри клієнта;
4. формування доступних варіантів рецептів відповідно до характеристик клієнта;

5. обробку наслідків прийнятих користувачем рішень;
6. виведення результатів взаємодії з клієнтами;
7. алгоритмічне визначення переходів між сценарними вузлами та варіантів завершення гри залежно від поточного стану системи;
8. ведення та відображення журналу прийнятих рішень у межах проходження гри.

До нефункціональних вимог належать: зручність користувацького інтерфейсу, стабільність роботи програмного продукту, можливість масштабування сценарної структури, підтримка різних платформ, ефективне управління ресурсами.

Системні вимоги до продукту наступні - програмний продукт повинен бути реалізований із використанням платформи Ren'Py, що забезпечує: підтримку сценарної логіки, інтеграцію з мовою програмування Python, механізми управління сценами, систему збереження станів [9].

Система повинна включати такі логічні компоненти системи: сценарна логіка, обробка вибору користувача, робота з рецептами, механізм збереження стану та інтерфейс користувача

У процесі функціонування програмного продукту «Магічна аптека» система обробляє вхідні дані, які формуються на основі дій користувача та сценарних подій гри.

До вхідних даних належать вибір користувачем варіантів діалогових реплік, вибір інгредієнтів для створення магічних рецептів.

На основі таких вхідних даних формуються поточні значення змінних стану системи, зокрема рівня довіри персонажів, морального профілю та репутації. Крім того, враховуються параметри поточної сюжетної сцени, які визначають доступні варіанти взаємодії. В ході обробки вхідних даних система формує відповідні результати, що визначають подальший розвиток ігрового процесу.

До вихідних даних належать відображення наступних діалогів або сцен, зміна значень внутрішніх параметрів персонажів, формування результатів

взаємодії з клієнтами, а також оновлення показників репутації та морального профілю користувача. У результаті обробки цих даних система визначає подальший варіант розвитку подій відповідно до обраних рішень та відображає відповідні події у програмному середовищі.

Висновки до розділу 1

У першому розділі було проведено аналіз предметної області інтерактивних наративних систем, що дозволило визначити основні особливості програмних продуктів цього типу. Розглянуто сучасні підходи до створення наративно орієнтованих ігор, у яких сюжет формується залежно від дій користувача, а програмна логіка забезпечує обробку наслідків прийнятих рішень.

У межах дослідження було також проаналізовано сучасний стан ринку ігрових програмних продуктів та визначено тенденції його розвитку. Встановлено, що індустрія відеоігор демонструє стабільне зростання, що сприяє появі нових типів інтерактивних систем, зокрема інді-проєктів та наративно орієнтованих ігор. Особливу увагу в таких програмних продуктах приділяють сюжетній взаємодії з користувачем, використанню систем морального вибору та моделюванню поведінки персонажів. що підтверджується аналізом відгуків користувачів сучасних ігрових платформ [10].

Для визначення функціональних особливостей майбутнього програмного продукту було проведено порівняльний аналіз існуючих аналогів, серед яких розглянуто ігри Papers, Please, Potion Craft та Disco Elysium. У результаті аналізу встановлено їх основні переваги та обмеження, а також визначено ключові підходи до реалізації наративної логіки, взаємодії з персонажами та алгоритмічної обробки рішень користувача.

На основі проведеного аналізу предметної області та існуючих програмних рішень було сформовано основні вимоги до програмного продукту «Магічна аптека». Визначено функціональні вимоги, пов’язані з реалізацією системи діалогів, обробкою вибору користувача, механікою створення магічних рецептів

та збереженням стану ігрових параметрів. Також сформовано нефункціональні та системні вимоги, які визначають особливості програмної архітектури, структури даних та взаємодії компонентів системи.

Отримані результати аналітичного дослідження можуть стати основою для подальшого етапу роботи - моделювання та проєктування архітектури програмного продукту.

РОЗДІЛ 2
МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ АРХІТЕКТУРИ СИСТЕМИ

У цьому розділі буде розглянуто моделі поведінки користувача, архітектурну структуру системи та алгоритми обробки моральних і поведінкових параметрів, що забезпечують адаптивність сюжетної логіки програмного продукту.

2.1 Моделювання поведінки користувача та сценарних процесів

Моделювання поведінки користувача у розробленому програмному продукті здійснюється з урахуванням специфіки інтерактивних наративних систем, у яких основною формою взаємодії є прийняття рішень у межах заданого сценарію. У контексті гри «Магічна аптека» користувач виступає у ролі аптекаря, що взаємодіє з клієнтами, аналізує їхні проблеми та обирає один із запропонованих варіантів дії. Така взаємодія має не лише функціональний, але й змістовний характер, оскільки кожне рішення інтерпретується як моральний вибір.

На відміну від традиційних комп’ютерних ігор, де поведінка користувача часто обмежується виконанням механічних дій, у візуальній новелі вона пов’язана з інтерпретацією тексту, аналізом ситуації та прийняттям рішень у межах заданого наративу. Це вимагає використання моделей, які враховують не лише послідовність дій, але й їхній вплив на загальний стан системи та розвиток сюжетної лінії.

Основною структурною одиницею сценарію є сценарний вузол, який можна розглядати як окремий стан системи. У межах реалізованого програмного продукту кожен такий вузол відповідає консультації з клієнтом. Він включає відображення діалогу, додаткові репліки, що залежать від внутрішніх характеристик персонажа, а також варіанти вибору, доступні користувачу. Після

здійснення вибору система переходить до наступного стану, що відповідає новому етапу сценарію.

Сукупність сценарних вузлів утворює логічну структуру, яку можна інтерпретувати як орієнтований граф. Вершини цього графа відповідають окремим етапам взаємодії (наприклад, консультаціям із клієнтами), а ребра - переходам між ними, що залежать від вибору користувача. У даній реалізації граф має переважно лінійну структуру, оскільки порядок клієнтів є фіксованим, однак кожен вузол містить альтернативні варіанти розвитку подій. Таким чином, забезпечується поєднання лінійного сценарію та локальної варіативності.

Поведінка користувача у системі формується як послідовність прийнятих рішень, що накопичуються протягом проходження гри. Важливою особливістю є те, що система не розглядає кожне рішення ізольовано, а враховує попередній досвід взаємодії. Це реалізується через використання внутрішніх параметрів стану, таких як показники карми та репутації, які змінюються після кожного вибору.

З формальної точки зору процес взаємодії користувача із системою можна представити як перехід між станами, що описується наступним співвідношенням: $S(t+1) = f(S(t), A(t))$, де S(t) - поточний стан системи, що включає значення внутрішніх параметрів, A(t) - дія користувача (вибір рецепту), а S(t+1) - новий стан системи після обробки цієї дії. У програмній реалізації ця модель представлена у вигляді послідовного оновлення змінних karma_meter та reputation_score після кожного вибору користувача, що безпосередньо відображено у програмному коді [11].

У такій моделі генерація сюжетної лінії розглядається як послідовне формування індивідуального маршруту проходження через множину сценарних станів. Конкретна траєкторія визначається функцією переходу, яка враховує попередній стан системи та вибір користувача.

У кожному сценарному вузлі система формує обмежений набір варіантів рішень. У поточній реалізації для кожного клієнта пропонується рівно два варіанти рецептів, що відповідають його психотипу. Такий підхід дозволяє

зберегти баланс між варіативністю та зрозумілістю інтерфейсу. Надмірна кількість варіантів могла б ускладнити процес прийняття рішень, тоді як обмеження до двох забезпечує чіткість і швидкість взаємодії.

Важливим елементом моделі є параметр прихованого наміру клієнта (hidden intent), який використовується для формування додаткових діалогових реплік. У залежності від значення цього параметра система може змінювати характер відповіді аптекаря, що створює ефект глибшої взаємодії з персонажем. Зазначений параметр впливає на діалогову частину та частково змінює внутрішній стан персонажа (рівень довіри), але не впливає на визначення фінальної кінцівки гри, а виконує роль наративного елементу, що підвищує рівень занурення користувача.

Ще одним важливим компонентом моделі є журнал подій, який фіксує всі значущі дії користувача. Кожен запис у журналі містить інформацію про клієнта, обраний рецепт, його ефект, а також зміни внутрішніх параметрів системи. Такий підхід дозволяє зберігати історію проходження та забезпечує узгодженість сценарію. Крім того, журнал може використовуватися для відображення результатів гри після завершення основного сценарію.

Сценарний процес у системі має чітко визначену структуру. Він починається з вступної сцени, яка вводить користувача у контекст гри, після чого відбувається послідовна обробка клієнтів. Кожна консультація супроводжується вибором користувача та зміною параметрів стану. Після завершення всіх консультацій система переходить до етапу підбиття підсумків, де на основі накопичених значень визначається фінальний результат.

Для формалізації взаємодії користувача із системою доцільно використати діаграму варіантів використання (рис. 2.1), яка відображає основні сценарії роботи. У межах цієї діаграми гравець виступає основним актором, який ініціює всі дії, тоді як система забезпечує обробку запитів, збереження даних та формування результатів. Такий підхід дозволяє чітко визначити функціональні межі системи та її ключові можливості.

Крім того, доцільним є використання діаграми діяльності (рис. 2.2), яка відображає послідовність дій під час проходження гри. Вона дозволяє візуалізувати процес переходу між станами, включаючи обробку клієнтів, вибір варіантів та оновлення внутрішніх параметрів. Перевірка наявності мотиву реалізується через аналіз параметра hidden_intent клієнта.

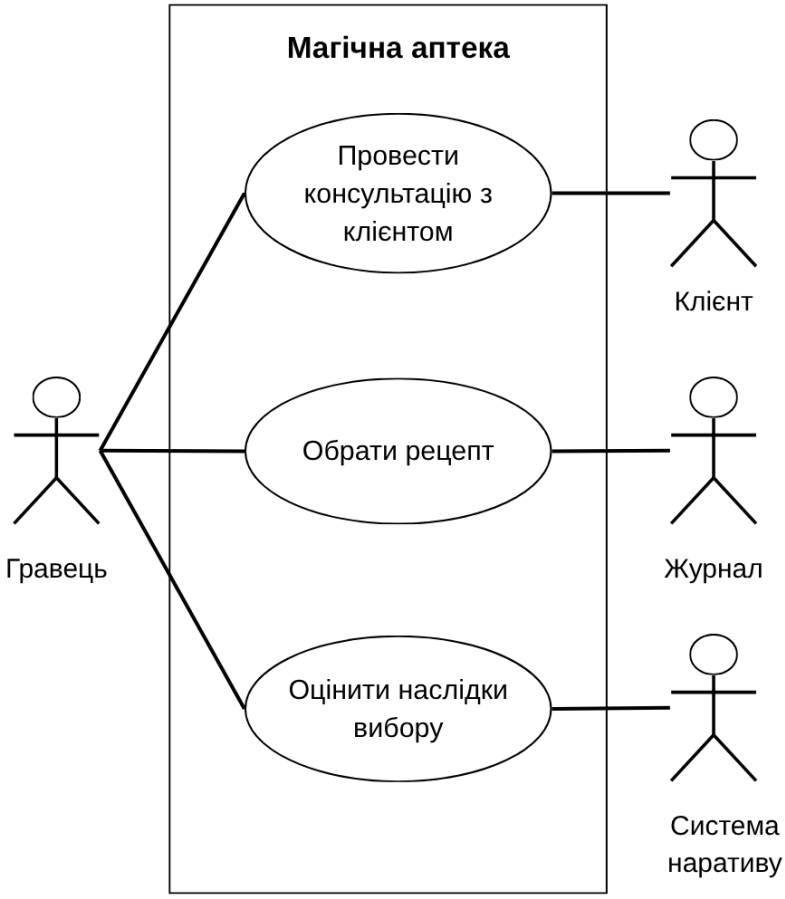

Рисунок 2.1 - Діаграма варіантів використання (Use Case)

Джерело: розроблено автором

Діаграма діяльності відображає основний цикл роботи гри, який реалізовано через послідовну обробку клієнтів у циклі while . Після ініціалізації змінних для кожного клієнта виконується консультація: виводиться історія, перевіряється параметр hidden_intent, після чого гравець обирає один із запропонованих рецептів. Обраний варіант впливає на значення карми та репутації, а також записується до журналу. Процес повторюється, доки є клієнти,

що відповідає перевірці «Ще клієнт?». Після завершення циклу виконується аналіз накопичених параметрів і визначається кінцівка гри, що узгоджується з фінальними гілками діаграми.

Рисунок 2.2 - Діаграма діяльності (Activity Diagram)

Джерело: розроблено автором.

Таким чином, модель поведінки користувача у розробленій системі базується на поєднанні сценарної структури, накопичення параметрів та обмеженого вибору на кожному етапі. Це дозволяє забезпечити як керованість процесу, так і варіативність фінального результату, що відповідає сучасним підходам до проєктування інтерактивних наративних систем.

2.2 Архітектурна структура та компонентна модель

Архітектура програмного продукту «Магічна аптека» визначає спосіб організації програмних компонентів, їх взаємодію та принципи побудови системи. У рамках даного проєкту архітектурний опис побудовано на основі логічного виділення основних складових системи, кожна з яких відповідає за окремий аспект функціонування гри.

Слід зазначити, що наведені компоненти є логічною моделлю системи та не реалізовані у вигляді окремих програмних модулів. Фактична реалізація виконана у середовищі Ren’Py у вигляді сценарних блоків (label) та допоміжних функцій на Python.

Обраний підхід дозволяє забезпечити зрозумілість структури програмного продукту, спростити процес розробки та створити основу для подальшого розширення функціональності. При цьому архітектура адаптована до специфіки інтерактивної візуальної новели, де основним елементом є сценарна логіка та обробка вибору користувача.

У розробленій системі можна виділити декілька ключових компонентів, які взаємодіють між собою та формують єдину архітектурну модель.

Центральним елементом є сценарна логіка гри, яка реалізується через послідовність сценарних блоків. У програмному коді ця логіка представлена у вигляді окремих сценарних секцій (label), що визначають порядок виконання дій. Зокрема, основний сценарій включає запуск гри, послідовну обробку клієнтів, перегляд журналу рішень та визначення кінцівки. Така структура забезпечує чітке розділення етапів виконання та дозволяє керувати переходами між ними.

Другим важливим компонентом є обробка вибору користувача реалізована за допомогою механізму меню (menu) середовища Ren’Py, а формування доступних варіантів здійснюється через функцію recommend_recipes. Вона відповідає за формування варіантів дій та обробку вибраного користувачем рішення. У поточній реалізації цей модуль представлений функцією формування доступних рецептів для кожного клієнта, а також механізмом вибору одного з двох варіантів. Після здійснення вибору система виконує відповідні зміни внутрішнього стану.

Окрему роль у системі відіграє модуль предметної логіки, який описує основні сутності гри. У межах реалізації до таких сутностей належать клієнт та рецепт. Клієнт містить інформацію про персонажа, його історію, психотип та прихований намір, тоді як рецепт описує можливий варіант рішення, включаючи інгредієнти, ефект та вплив на параметри системи. Використання окремих класів для представлення цих сутностей дозволяє структурувати дані та спростити їх подальшу обробку.

Ще одним ключовим компонентом є збереження стану гри, яке реалізовано через глобальні змінні karma_meter та reputation_score, що оновлюються після кожного вибору користувача.. Зміна цих параметрів відбувається після кожного вибору користувача, що дозволяє накопичувати результати рішень протягом гри.

Важливою складовою архітектури є журнал подій, який реалізований як структура для збереження історії взаємодії. Журнал подій реалізований у вигляді списку (persistent.journal), у якому зберігаються словники з інформацією про прийняті рішення. Кожен запис містить інформацію про клієнта, обраний рецепт, наслідки цього вибору, а також зміну параметрів карми та репутації. Журнал використовується для відображення результатів після завершення основного сценарію, що забезпечує логічну завершеність ігрового процесу [12].

У поточній реалізації механізм persistent використовується насамперед для збереження журналу рішень і службових даних, необхідних для відображення результатів проходження. Повноцінне довготривале збереження прогресу між

окремими ігровими сесіями у межах роботи не реалізовувалося і розглядається як напрям подальшого розвитку.

Інтерфейс користувача є окремим рівнем системи, що відповідає за відображення тексту, сцен та варіантів вибору. У даній реалізації він побудований на базі можливостей середовища Ren’Py, що дозволяє поєднувати сценарний опис із програмною логікою. Інтерфейс забезпечує візуалізацію процесу взаємодії, але не містить складної логіки, оскільки основна обробка виконується на рівні сценарію та внутрішніх функцій.

З погляду логічного подання (рис. 2.3) систему доцільно розглядати як модель з трьома умовно виділеними складовими: представлення, логіки та даних. Складова представлення охоплює інтерфейс користувача, який забезпечує взаємодію з гравцем. Складова логіки відповідає за обробку сценарію, вибору та переходів між станами. Складова даних включає структури, що зберігають інформацію про клієнтів, рецепти та стан гри.

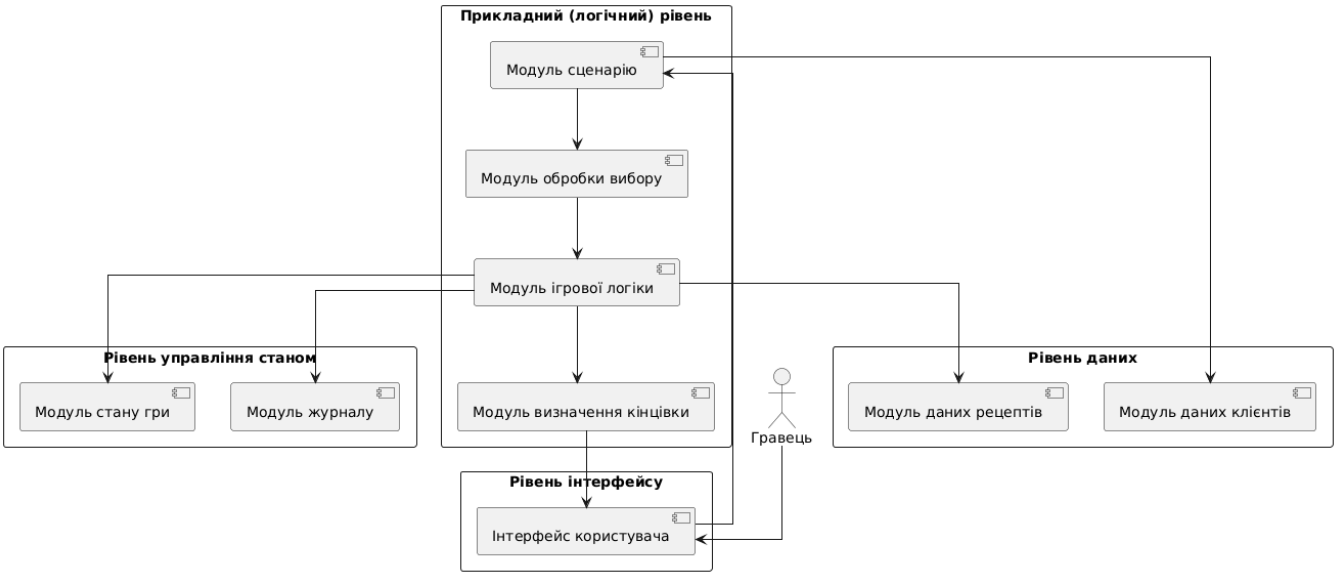

Рисунок 2.3 - Діаграма компонентів (Component diagram)

Джерело: розроблено автором

Архітектура системи також передбачає формалізацію внутрішньої структури даних, які використовуються у програмному продукті. Для цього доцільно застосувати діаграму класів (рис. 2.4), яка відображає основні сутності системи та їх взаємозв’язки.

Архітектура програмного продукту є спрощеною та обумовлена специфікою платформи Ren’Py

Ключовими структурами даних є класи «Client» та «Recipe», а також допоміжні функції для обробки логіки. Клас «Client» описує клієнта та містить такі характеристики, як ім’я, історія, психотип та прихований намір, що використовується для варіації діалогів. Клас «Recipe» визначає можливі варіанти рішень та включає назву рецепту, перелік інгредієнтів, ефект, а також вплив на моральні параметри системи.

Крім того, у системі використовується набір змінних стану, зокрема показники карми та репутації, які змінюються залежно від вибору користувача та використовуються для визначення фінального результату гри.

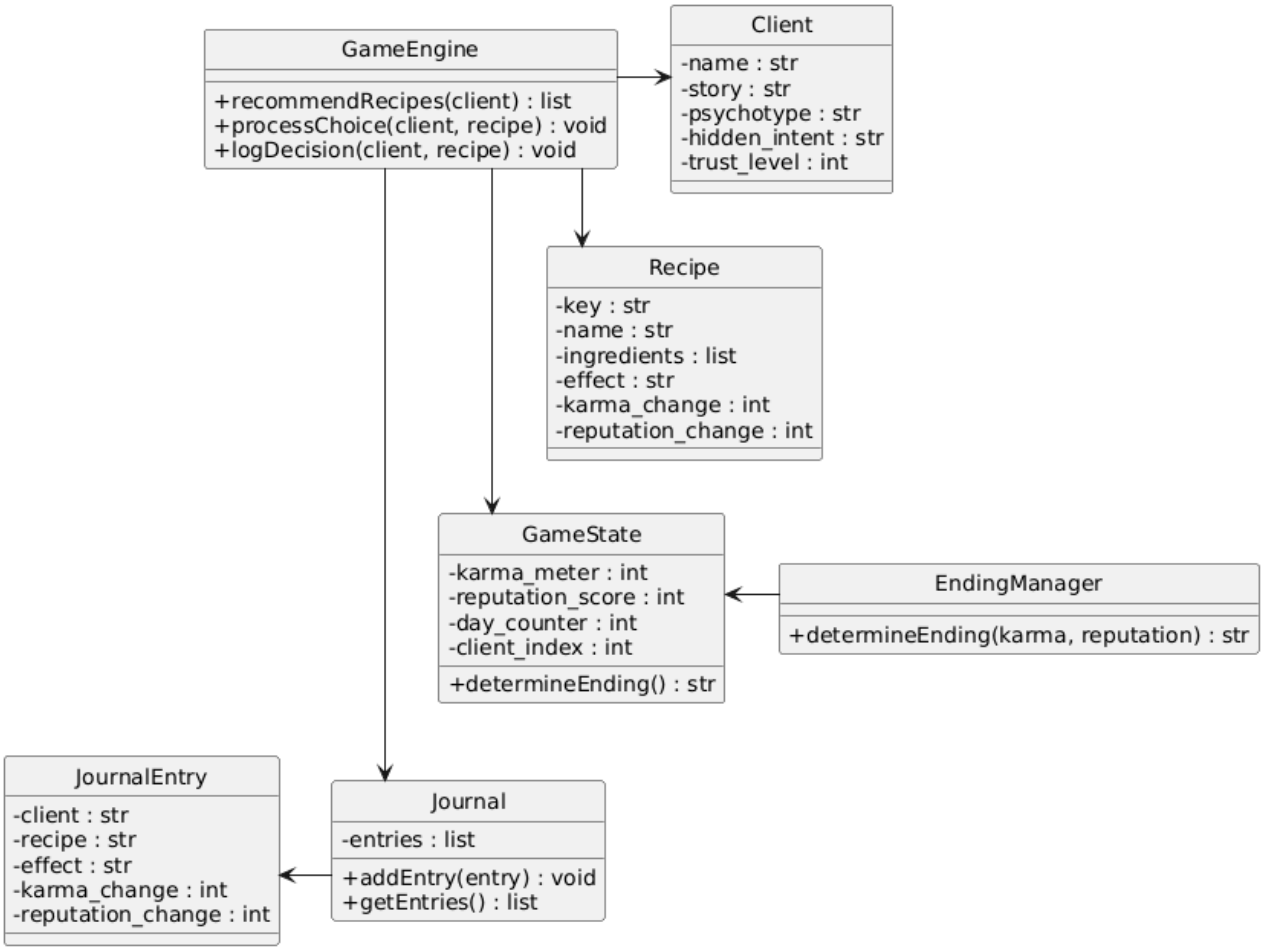

Рисунок 2.4 - Діаграма класів (Class diagram)

Джерело: розроблено автором

Взаємодія між рівнями відбувається послідовно. Користувач здійснює вибір через інтерфейс, після чого цей вибір передається до логічного рівня. Логічний рівень обробляє його, змінює стан системи та повертає результат для відображення. Таким чином формується замкнений цикл взаємодії.

Особливістю даної архітектури є використання подієво-орієнтованого підходу. Кожна дія користувача розглядається як подія, яка ініціює виконання відповідного сценарного блоку. У результаті цього відбувається оновлення внутрішніх параметрів системи та перехід до наступного стану.

Крім того, важливим аспектом є відокремлення логіки від контенту. Текстові дані, що відображаються користувачу, зберігаються окремо від алгоритмічної частини, що дозволяє змінювати сценарій без внесення змін у структуру програми. Це значно спрощує підтримку та розвиток системи.

Запропонована архітектура також передбачає можливість масштабування. Система може бути розширена шляхом додавання нових клієнтів, рецептів або параметрів без необхідності зміни базової логіки роботи. Це досягається завдяки використанню структурованих даних та модульного підходу до організації коду.

Таким чином, архітектура програмного продукту забезпечує цілісність, гнучкість та можливість подальшого розвитку системи. Вона відповідає вимогам інтерактивних наративних систем і дозволяє ефективно реалізувати механізм впливу рішень користувача на визначення подальшого розвитку подій.

2.3 Алгоритми обробки моральних та поведінкових параметрів

Однією з ключових особливостей розробленого програмного продукту є використання моральних та поведінкових параметрів, які визначають реакцію системи на дії користувача. У контексті гри «Магічна аптека» такі параметри виступають основним механізмом формування адаптивної сюжетної логіки, оскільки саме вони впливають на фінальний результат проходження.

На відміну від більш складних моделей, у яких використовується велика кількість параметрів, у даній реалізації було обрано спрощений, але ефективний

підхід. Основними змінними стану системи є показники карми (karma_meter) та репутації (reputation_score), які накопичуються протягом усього ігрового процесу. Саме ці два параметри використовуються для визначення фінальної кінцівки гри.

Зміна значень параметрів відбувається безпосередньо у процесі взаємодії користувача з клієнтами. Після кожної консультації користувач обирає один із двох запропонованих рецептів, кожен з яких має визначений вплив на моральні характеристики. Таким чином, кожен вибір супроводжується зміною значень внутрішніх змінних системи.

Формально процес оновлення параметрів можна описати наступним співвідношенням:

$$P = P + A$$

де P - поточне значення параметра, а A - зміна, що залежить від обраного користувачем варіанту. Значення A може бути як додатним, так і від’ємним, що дозволяє відображати як позитивні, так і негативні наслідки рішень.

У програмному коді це реалізовано шляхом додавання значень karma_change та reputation_change обраного рецепта до відповідних змінних стану.

У межах кожного сценарного вузла (консультації) система виконує послідовність дій: отримання інформації про клієнта, формування доступних варіантів рішень, обробка вибору користувача та оновлення внутрішніх параметрів. Таким чином формується динамічна система, у якій стан змінюється поступово, залежно від сукупності прийнятих рішень.

Особливістю даної реалізації є використання накопичувальної моделі, у якій кожне рішення впливає на загальний результат. Система не аналізує окремі дії ізольовано, а враховує їх у сукупності, що дозволяє отримати більш узгоджений і логічний фінальний результат.

Крім безпосереднього оновлення параметрів, у системі також реалізовано механізм збереження історії рішень. Для цього використовується журнал подій, який фіксує інформацію про кожну консультацію, включаючи клієнта, обраний

рецепт та його ефект. Це дозволяє не лише відображати результати після завершення гри, але й забезпечує прозорість алгоритму та можливість аналізу прийнятих рішень.

Після завершення всіх сценарних подій система переходить до етапу визначення кінцівки. Для цього використовується окремий алгоритм, який аналізує накопичені значення карми та репутації. На основі цих значень система обирає один із можливих варіантів завершення гри.

Логіка визначення кінцівки базується на використанні порогових значень. Залежно від співвідношення параметрів система класифікує результат як позитивний, нейтральний або негативний. У межах реалізації передбачено декілька варіантів завершення, що дозволяє відобразити різні наслідки поведінки користувача.

Алгоритм визначення кінцівки можна описати як послідовність етапів. Спочатку відбувається збір значень параметрів після завершення всіх консультацій. Далі система виконує їх порівняння з визначеними умовами. На основі цього порівняння визначається тип кінцівки, після чого здійснюється перехід до відповідного сценарного блоку.

Таким чином, алгоритми обробки моральних та поведінкових параметрів у розробленій системі забезпечують накопичення результатів рішень, їх аналіз та формування адаптивного фінального результату. Це дозволяє створити інтерактивне середовище, у якому поведінка користувача безпосередньо впливає на розвиток подій та завершення гри.

Висновки до розділу 2

У другому розділі було розглянуто процес моделювання поведінки користувача, побудову сценарних процесів та проєктування архітектури програмного продукту «Магічна аптека». Основну увагу було приділено формалізації взаємодії користувача із системою, структурі сценарної логіки та принципам організації програмних компонентів.

Архітектура системи є спрощеною та адаптованою до особливостей середовища Ren’Py, у якому логіка, дані та інтерфейс тісно інтегровані в межах єдиного сценарного підходу.

Було визначено, що основою функціонування гри є сценарна модель, яка може бути представлена у вигляді орієнтованого графа станів. Кожен стан відповідає окремому етапу взаємодії з клієнтом, а переходи між станами здійснюються на основі вибору користувача. Такий підхід дозволяє поєднати послідовну структуру сценарію з варіативністю розвитку подій.

У процесі проєктування було сформовано архітектуру системи, що базується на модульному підході. Виділення окремих компонентів, зокрема сценарної логіки, обробки вибору, стану гри та журналу подій, дозволило забезпечити зрозумілість структури програмного продукту, а також створити передумови для його подальшого розширення. Запропонована архітектура забезпечує гнучкість, цілісність та можливість масштабування системи без суттєвих змін базової логіки.

Особливу увагу було приділено алгоритмам обробки моральних та поведінкових параметрів. У межах реалізації використовується накопичувальна модель, у якій значення карми та репутації змінюються залежно від рішень користувача. Це дозволяє враховувати сукупність дій протягом усього ігрового процесу та формувати узгоджений фінальний результат.

Розроблений алгоритм визначення кінцівки базується на аналізі накопичених параметрів і використанні порогових значень, що дозволяє реалізувати декілька варіантів завершення гри. Такий підхід забезпечує адаптивність системи та створює умови для формування різних сценаріїв залежно від поведінки користувача.

Отримані результати моделювання та проєктування можуть бути використані при подальшій реалізації програмного продукту, а також при вдосконаленні механізмів інтерактивної наративної взаємодії. Зокрема, запропонована структура дозволяє розширювати систему шляхом додавання

нових сценарних елементів, параметрів та варіантів розвитку сюжету без порушення її загальної логіки.

РОЗДІЛ 3
РЕАЛІЗАЦІЯ ТА ПЕРЕВІРКА ПРАЦЕЗДАТНОСТІ ПРОДУКТУ

3.1 Технологічна реалізація та структура коду

У даному підрозділі розглядається практична реалізація програмного продукту «Магічна аптека», а також структура програмного коду, що забезпечує функціонування ігрової логіки, взаємодію з користувачем та обробку результатів його вибору. Реалізація виконана з використанням рушія Ren'Py, який поєднує сценарний підхід до розробки з можливістю використання мови Python для реалізації логіки.

На відміну від класичних програмних систем, де архітектура будується на основі окремих модулів і класів, у даному проєкті використано змішаний підхід. Основна частина взаємодії реалізована через сценарні конструкції Ren'Py, тоді як логіка обробки даних та стану гри реалізована за допомогою Python-коду у блоці init python.

Початковий етап реалізації включає оголошення персонажів, змінних стану та ресурсів гри (рис. 3.1).

```
define ap = Character("Аптекар", color="#92A5C1")
define cl = Character("Клієнт", color="#C19292")
define narr = Character(None)

default karma_meter = 0
default reputation_score = 0
default day_counter = 0

image apteka = Transform("apteka.png", size=(1920, 1080))
image success = Transform("success.png", size=(1920, 1080))
image neutral = Transform("neutral.png", size=(1920, 1080))
image failure = Transform("failure.png", size=(1920, 1080))
```

Рисунок 3.1 - Фрагмент коду ініціалізації персонажів та змінних

Джерело: розроблено автором

У даному фрагменті визначено персонажів гри, а також глобальні змінні, що відповідають за стан гри. Зокрема, змінні karma_meter та reputation_score використовуються для накопичення результатів вибору гравця, а змінна day_counter відображає прогрес проходження.

Далі у блоці init python реалізовано основні структури даних, які використовуються у грі (рис. 3.2).

```
class Recipe(object):
    def __init__(self, key, name, ingredients, effect, karma_change, reputation_change):
        self.key = key
        self.name = name
        self.ingredients = ingredients
        self.effect = effect
        self.karma_change = karma_change
        self.reputation_change = reputation_change

class Client(object):
    def __init__(self, name, story, psychotype, hidden_intent="neutral"):
        self.name = name
        self.story = story
        self.psychotype = psychotype
        self.hidden_intent = hidden_intent
        self.trust_level = 0
```

Рисунок 3.2 - Фрагмент коду опису класів Recipe та Client

Джерело: розроблено автором

Клас Recipe представляє рецепт, який може бути запропонований клієнту. Він містить ключ, назву, список інгредієнтів, ефект, а також зміни показників карми та репутації. Клас Client описує клієнта гри, включаючи його історію, психотип та прихований намір, який впливає на поведінку під час консультації.

Основні дані гри задаються у вигляді словника клієнтів (рис. 3.3) та списку рецептів (рис. 3.4).

```
clients = [
    Client("Анна", "Мене мучать нічні кошмари. Я не можу спати.", "troubled", "neutral"),
    Client("Богдан", "Я завжди боюся вийти з дому. Хочу стати сміливішим.", "coward", "aggressive"),
    Client("Катерина", "Хочу, щоб мене покохали. Будь-якою ціною.", "romantic", "manipulative"),
    Client("Дмитро", "Я повинен дізнатися правду про свою родину.", "seeker", "truth")
]

if not hasattr(persistent, "journal"):
    persistent.journal = []
```

Рисунок 3.3 - Фрагмент коду ініціалізації клієнтів

Джерело: розроблено автором

```
recipes = {
    "calming_tea": Recipe(
        key="calming_tea",
        name="Чай для заспокоєння",
        ingredients=["лаванда", "вербена", "мед"],
        effect="зменшує тривогу та надає спокою",
        karma_change=1,
        reputation_change=1
    ),
    "love_potion": Recipe(
        key="love_potion",
        name="Зілля кохання",
        ingredients=["троянда", "серце русалки"],
        effect="розпалює романтичні почуття",
        karma_change=-2,
        reputation_change=-1
    ),
    "truth_serum": Recipe(
        key="truth_serum",
        name="Сироватка правди",
        ingredients=["полин", "срібний корінь"],
        effect="змушує говорити правду",
        karma_change=-1,
        reputation_change=0
    ),
    "courage_elixir": Recipe(
        key="courage_elixir",
        name="Еліксир сміливості",
        ingredients=["барвінок", "вовче серце"],
        effect="надає впевненості й сміливості",
        karma_change=1,
        reputation_change=1
    ),
    "prophecy_potion": Recipe(
        key="prophecy_potion",
        name="Напій пророцтва",
        ingredients=["ковила", "очі тритона", "соняшник"],
        effect="відкриває завісу майбутнього",
        karma_change=0,
        reputation_change=2
    )
}
```

Рисунок 3.4 - Фрагмент коду ініціалізації рецептів

Джерело: розроблено автором

Такий підхід дозволяє легко розширювати гру, додаючи нові рецепти або клієнтів без зміни основної логіки. Кожен клієнт має власні характеристики, які впливають на доступні варіанти вибору.

Функція recommend_recipes(client) відповідає за формування варіантів вибору для гравця (рис. 3.5).

```
def recommend_recipes(client):
    if client.psychotype == "troubled":
        return ["calming_tea", "truth_serum"]
    elif client.psychotype == "coward":
        return ["courage_elixir", "truth_serum"]
    elif client.psychotype == "romantic":
        return ["love_potion", "calming_tea"]
    elif client.psychotype == "seeker":
        return ["truth_serum", "prophecy_potion"]
    return ["calming_tea", "truth_serum"]
```

Рисунок 3.5 - Фрагмент коду функції підбору рецептів

Джерело: розроблено автором

У цій функції використовується умовна логіка, яка визначає доступні рецепти залежно від психотипу клієнта. Це забезпечує варіативність ігрового процесу та відповідає концепції нелінійного сценарію.

Визначення кінцівки гри реалізовано окремою функцією (рис. 3.6).

```
def determine_ending(karma, reputation):
    if karma >= 3 and reputation >= 3:
        return "saintly"
    elif karma >= 1:
        return "good"
    elif karma <= -3:
        return "evil"
    elif karma <= -1:
        return "bad"
    else:
        return "neutral"
```

Рисунок 3.6 - Фрагмент коду функції визначення кінцівки

Джерело: розроблено автором

Функція determine_ending() аналізує значення карми та репутації та повертає тип кінцівки. Це дозволяє централізувати логіку визначення результату гри.

Для збереження історії рішень використовується журнал, який реалізовано через механізм persistent Ren'Py (рис. 3.7).

```
def log_decision(client_name, recipe_key):
    rec = recipes[recipe_key]
    entry = {
        "client": client_name,
        "recipe": rec.name,
        "effect": rec.effect,
        "karma_change": rec.karma_change,
        "reputation_change": rec.reputation_change,
    }
    persistent.journal.append(entry)
```

Рисунок 3.7 - Фрагмент коду функції логування рішень

Джерело: розроблено автором

Функція log_decision() додає запис у журнал, який містить інформацію про клієнта, обраний рецепт та його ефект. Це дозволяє відобразити історію гри після завершення проходження.

Основний сценарій гри реалізовано у label start(рис. 3.8).

```
label start:
    $ day_counter = 1
    $ client_index = 0
    $ karma_meter = 0
    $ reputation_score = 0
    $ persistent.journal = []

    scene apteka
    ap "Вітаю у «Магічній аптеці»!"
    $ intro_text = (
        "У цій грі ви будете слухати клієнтів, "
        "пропонувати їм зілля та спостерігати "
        "за наслідками. Ваші дії впливають на "
        "карму та репутацію."
    )
    narr "[intro_text]"
    while client_index < len(clients):
        $ current_client = clients[client_index]
        call consult(current_client) from _call_consult
        $ day_counter += 1
        $ client_index += 1

    call show_journal from _call_show_journal
    call ending from _call_ending
    return
```

Рисунок 3.8 - Фрагмент коду початку гри

Джерело: розроблено автором

У цьому фрагменті відбувається ініціалізація змінних, запуск вступної сцени та організація циклу обробки клієнтів. Використання циклу while дозволяє послідовно обробити кожного клієнта.

Ключовим елементом ігрової логіки є процедура consult. (рис. 3.9, 3.10).

```
label consult(client):
    scene apteka

    $ greeting = random.choice(greetings)
    ap "[greeting]"
    cl "[client.story]"

    if client.hidden_intent == "aggressive":
        ap "Репліка агресивного мотиву"
        $ client.trust_level -= 1
    elif client.hidden_intent == "manipulative":
        ap "Репліка маніпулятивного мотиву"
        $ client.trust_level -= 1
    elif client.hidden_intent == "truth":
        ap "Репліка мотиву правди"

    $ options = recommend_recipes(client)
```

Рисунок 3.9 - Фрагмент коду консультації клієнта

Джерело: розроблено автором

```
$ karma_meter += recipes[chosen_recipe].karma_change
$ reputation_score += recipes[chosen_recipe].reputation_change

$ log_decision(client.name, chosen_recipe)

ap "Я рекомендую вам [recipes[chosen_recipe].name]."
cl "[recipes[chosen_recipe].effect]. Дякую, спробую!"
return
```

Рисунок 3.10 - Фрагмент коду консультації клієнта

Джерело: розроблено автором

У межах цієї процедури реалізовано повний цикл взаємодії з клієнтом. Спочатку відображається привітання та історія клієнта, після чого залежно від

прихованого наміру змінюється рівень довіри. Далі формується список доступних рецептів і відображається меню вибору.

Механізм вибору реалізовано через конструкцію menu, яка є стандартним інструментом Ren'Py(рис. 3.11).

```
menu:
    "Виберіть рецепт для [client.name]"

    "[recipes[options[0]].name] (інгредієнти: [', '.join(recipes[options[0]].ingredients)])":
        $ chosen_recipe = options[0]

    "[recipes[options[1]].name] (інгредієнти: [', '.join(recipes[options[1]].ingredients)])":
        $ chosen_recipe = options[1]
```

Рисунок 3.11 - Фрагмент коду меню вибору рецепта

Джерело: розроблено автором

Після вибору рецепта відбувається оновлення значень карми та репутації, а також запис рішення у журнал. Це забезпечує накопичувальний ефект рішень і впливає на фінальний результат.

Після обробки всіх клієнтів виконується відображення журналу(рис. 3.12).

```
label show_journal:
    scene apteka
    ap "Давайте переглянемо, які рішення ви приймали."

    if len(persistent.journal) == 0:
        narr "Журнал поки порожній."
        return

    $ journal_index = 0
    while journal_index < len(persistent.journal):
        $ entry = persistent.journal[journal_index]
        $ journal_text = (
            "Клієнт: %s, рецепт: %s, "
            "ефект: %s, зміна карми: %s, "
            "зміна репутації: %s"
        ) % (
            entry["client"],
            entry["recipe"],
            entry["effect"],
            entry["karma_change"],
            entry["reputation_change"]
        )
        "[journal_text]"
        $ journal_index += 1

    return
```

Рисунок 3.12 - Фрагмент коду відображення журналу

Джерело: розроблено автором

У цьому фрагменті використовується цикл для послідовного виведення всіх записів журналу. Це дозволяє гравцю проаналізувати свої рішення.

Завершальний етап гри реалізовано у label ending(рис. 3.13).

У цьому блоці викликається функція визначення кінцівки, після чого виконується перехід до відповідної сцени. Кожна кінцівка реалізована окремим label, що дозволяє легко додавати нові варіанти завершення гри.

```
label ending:
    $ final = determine_ending(karma_meter, reputation_score)

    if final == "saintly":
        jump ending_saintly
    elif final == "good":
        jump ending_good
    elif final == "neutral":
        jump ending_neutral
    elif final == "bad":
        jump ending_bad
    else:
        jump ending_evil
```

Рисунок 3.13 - Фрагмент коду визначення кінцівки

Джерело: розроблено автором

Таким чином, реалізація програмного продукту «Магічна аптека» базується на поєднанні сценарного підходу Ren'Py та програмної логіки на Python. Такий підхід дозволяє забезпечити гнучкість системи, простоту розширення та відповідність розробленій моделі поведінки гри. Основні компоненти системи реалізовані у вигляді функцій і структур даних, що забезпечує зрозумілість коду та його відповідність поставленим завданням.

3.2 Використання програмного продукту

Використання програмного продукту «Магічна аптека» передбачає взаємодію користувача з інтерактивною наративною системою, у якій основною формою діяльності є прийняття рішень у процесі діалогів із персонажами.

Відповідно до поставленого завдання кваліфікаційної роботи, користувач виступає у ролі аптекаря, який обслуговує клієнтів, аналізує їхні історії та приймає рішення щодо вибору відповідних рецептів . Такий підхід дозволяє реалізувати не лише ігрову, але й аналітичну взаємодію із системою.

На початковому етапі використання програмного продукту користувач запускає гру, після чого відбувається відображення вступної сцени(рис. 3.14). У цій сцені подається коротке пояснення механіки гри, що дозволяє користувачу зрозуміти основні принципи взаємодії із системою. Вступ виконує важливу роль адаптації користувача до інтерфейсу та логіки гри, оскільки забезпечує контекст і формує очікування щодо подальшого процесу.

Рисунок 3.14 - Початковий екран гри з описом механіки

Джерело: розроблено автором

Після завершення вступної частини система переходить до основного циклу роботи, який полягає у послідовній обробці клієнтів. Кожен клієнт представляє окремий сценарний вузол, у межах якого користувач взаємодіє із системою через діалог. У процесі використання програми користувач читає

історію клієнта, яка містить опис його ситуації(рис. 3.15), після чого система формує додаткові репліки залежно від внутрішніх характеристик персонажа.

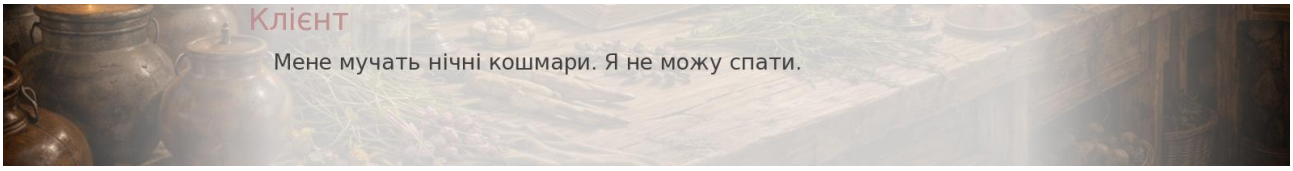

Рисунок 3.15 - Діалог з клієнтом із відображенням історії

Джерело: розроблено автором

Важливим елементом використання програмного продукту є інтерпретація текстової інформації. На відміну від традиційних ігор, де користувач виконує механічні дії, у даній системі ключовим є аналіз змісту діалогу. Користувач має оцінити ситуацію, визначити можливі наслідки та прийняти рішення, яке найбільш відповідає його стратегії поведінки. Це формує інтелектуальну складову взаємодії з програмним продуктом.

Після ознайомлення з інформацією про клієнта система надає користувачу можливість вибору одного з двох варіантів рецепту (рис. 3.16). Вибір реалізується через інтерфейс меню, що є стандартним механізмом середовища Ren’Py. Кожен варіант містить назву рецепту, перелік інгредієнтів, що дозволяє користувачу оцінити можливі наслідки свого рішення

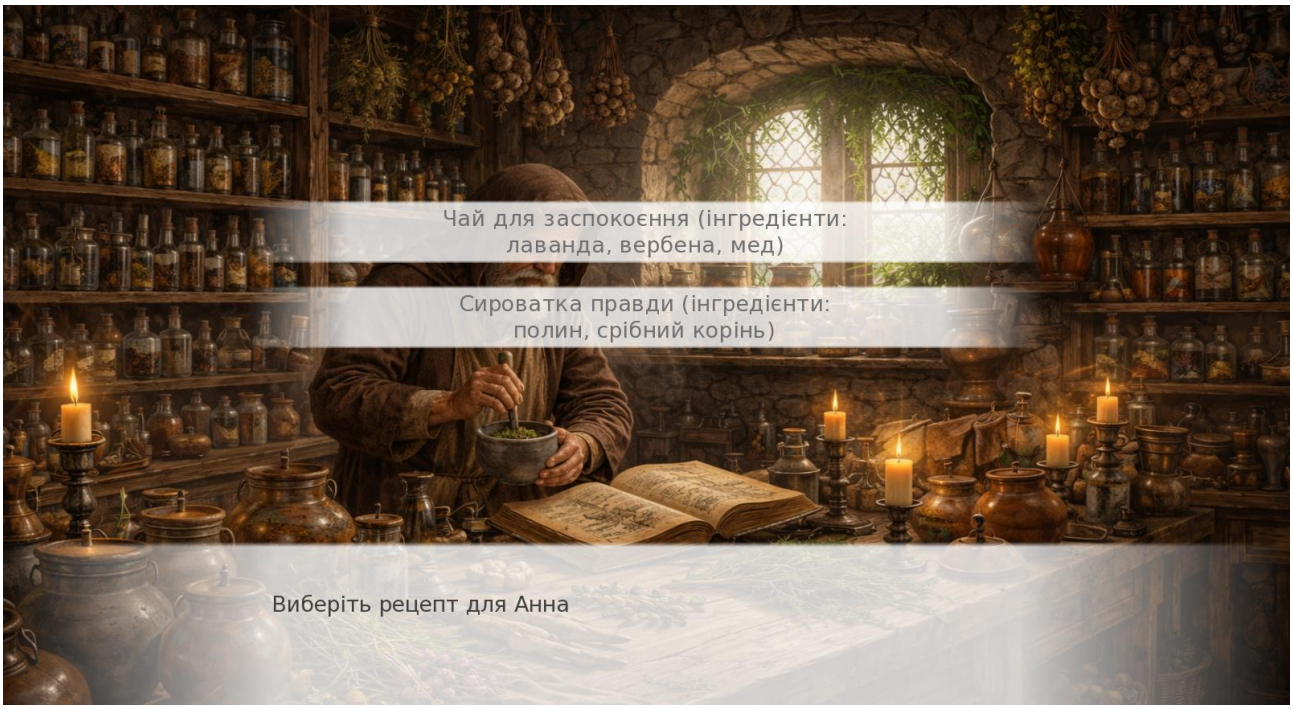

Рисунок 3.16 - Меню вибору рецепту з переліком інгредієнтів

Джерело: розроблено автором

У процесі використання системи важливою є не лише сама дія вибору, але й розуміння її наслідків. Кожен обраний рецепт впливає на внутрішні параметри системи, зокрема рівень карми та репутації. Ці параметри змінюються після кожної взаємодії, що створює накопичувальний ефект. Таким чином, користувач формує власну стратегію проходження гри, яка може бути орієнтована на позитивні, нейтральні або негативні результати.

Після здійснення вибору система відображає реакцію клієнта та результат застосування рецепту(рис. 3.17). Це може бути як позитивний, так і негативний ефект, що залежить від відповідності обраного рішення ситуації клієнта. Такий підхід дозволяє створити ефект зворотного зв’язку, який є важливим для формування досвіду користувача.

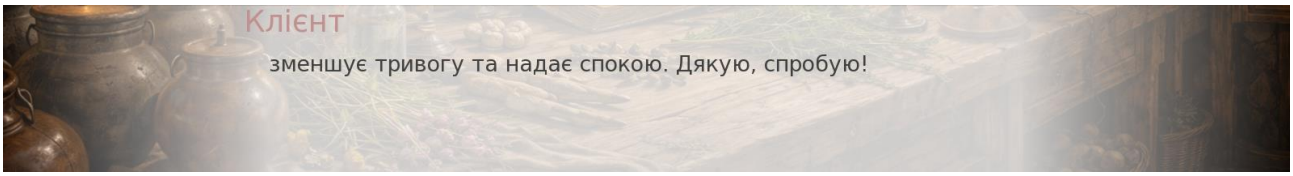

Рисунок 3.17 - Момент після вибору рецепту з реакцією клієнта

Джерело: розроблено автором

У процесі подальшого використання програмного продукту користувач проходить декілька таких сценарних вузлів, кожен з яких має унікальний зміст. Незважаючи на лінійну структуру проходження, кожен вибір впливає на загальний результат, що забезпечує варіативність фіналу гри.

Особливу роль у використанні програмного продукту відіграє журнал рішень (рис. 3.18). Після кожної взаємодії з клієнтом система автоматично додає запис до журналу, який містить інформацію про прийняте рішення. Журнал дозволяє користувачу відстежувати свою поведінку та аналізувати прийняті рішення після завершення гри.

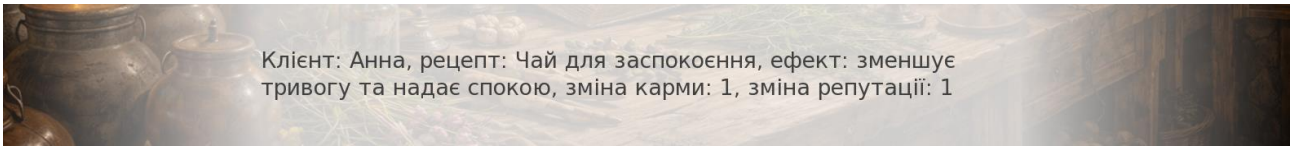

Рисунок 3.18 - Відображення журналу рішень після проходження

Джерело: розроблено автором

З точки зору користувацького досвіду журнал виконує функцію аналітичного інструменту. Він дозволяє не лише переглянути історію взаємодії, але й оцінити логіку власних рішень. Це особливо важливо у контексті навчального або демонстраційного використання програмного продукту.

Після обробки всіх клієнтів система переходить до завершального етапу використання, який полягає у визначенні кінцівки гри. На цьому етапі відбувається аналіз накопичених значень карми та репутації, після чого система визначає тип фінального результату.

Користувач отримує один із можливих варіантів завершення, який відображає загальну характеристику його поведінки протягом гри. Це може бути позитивна, нейтральна або негативна кінцівка, що відповідає обраній стратегії (рис. 3.19).

Рисунок 3.19 - Екран визначення з прикладом позитивної кінцівки

Джерело: розроблено автором

Використання програмного продукту не потребує спеціальних технічних знань. Користувачеві достатньо базових навичок роботи з комп’ютером, що робить систему доступною для широкого кола користувачів. Це відповідає нефункціональним вимогам до зручності використання.

У межах практичного використання програмного продукту можливі різні сценарії застосування. Зокрема, система може використовуватися як розважальний продукт, демонстраційна модель інтерактивної наративної системи або навчальний інструмент для вивчення принципів алгоритмічного прийняття рішень. У кожному з цих випадків користувач взаємодіє із системою через однакові механізми, але отримує різний досвід залежно від мети використання.

З технічної точки зору використання програмного продукту передбачає виконання сценарних блоків у середовищі Ren’Py, що забезпечує стабільну роботу системи та коректну обробку подій. Користувач не взаємодіє безпосередньо з програмним кодом, що дозволяє зосередитися на ігровому процесі.

Таким чином, процес використання програмного продукту «Магічна аптека» є послідовною взаємодією користувача із сценарною системою, у межах якої відбувається аналіз інформації, прийняття рішень та оцінка їхніх наслідків. Реалізовані механізми забезпечують зрозумілість, варіативність та логічну завершеність ігрового процесу, що відповідає вимогам до інтерактивних наративних систем.

3.3 Тестування програмного продукту

Після завершення етапу розробки програмного продукту «Магічна аптека» було проведено комплексну перевірку його працездатності з метою підтвердження коректності реалізації функціональних можливостей, стабільності виконання сценаріїв та відповідності програмного продукту поставленим вимогам. Особливістю тестування даного програмного продукту є його наративна природа, що передбачає необхідність перевірки не лише технічних аспектів, але й логічної узгодженості сценаріїв [13].

У процесі тестування було перевірено коректність запуску програмного продукту та ініціалізацію основних змінних. Було встановлено, що при кожному новому запуску гри відбувається правильне встановлення початкових значень змінних karma_meter, reputation_score та day_counter, що забезпечує незалежність кожного проходження. Також було підтверджено, що журнал рішень очищується перед початком нового сеансу, що виключає накопичення даних між різними проходженнями.

Наступним етапом тестування стала перевірка основного ігрового циклу. Було досліджено коректність послідовної обробки клієнтів у межах циклу, що реалізує взаємодію користувача з системою. У результаті встановлено, що кожен

клієнт обробляється рівно один раз, а після завершення взаємодії система переходить до наступного етапу без пропусків або повторів. Також було перевірено граничну ситуацію, за якої список клієнтів може бути порожнім, і встановлено, що у поточній реалізації передбачається наявність клієнтів, що забезпечує стабільність виконання сценарію.

У процесі тестування було перевірено коректність відображення діалогів. Було встановлено, що текст історії клієнта, привітання та відповіді системи відображаються без помилок, а підстановка змінних у текст виконується правильно. Також було підтверджено, що механізм випадкового вибору привітання забезпечує варіативність початкових реплік, не порушуючи загальної логіки взаємодії.

Окрему увагу було приділено перевірці механізму обробки прихованих параметрів клієнтів. Було досліджено поведінку системи при різних значеннях параметра hidden_intent та встановлено, що відповідні умовні переходи виконуються коректно. Зокрема, у випадках, коли клієнт має агресивний або маніпулятивний намір, система змінює внутрішній параметр довіри, що відповідає закладеній логіці.

Модульне тестування у межах даного програмного продукту реалізовано шляхом перевірки окремих функціональних елементів системи із використанням контрольних сценаріїв. На відміну від повноцінного автоматизованого unit-тестування, яке передбачає використання спеціалізованих бібліотек, у даній роботі застосовано підхід ручного тестування логічних функцій у відокремленому вигляді.

Зокрема, було перевірено функцію determine_ending, яка відповідає за визначення фінального результату гри. Для цього було сформовано набір тестових значень параметрів карми та репутації, що дозволило перевірити коректність переходів між різними типами кінцівок. У результаті встановлено, що функція повертає правильний результат для всіх граничних і проміжних значень.

Функція recommend_recipes була протестована шляхом підстановки різних типів клієнтів із заданими психотипами. Це дозволило підтвердити коректність формування списку доступних варіантів вибору та відповідність логіки підбору рецептів характеристикам клієнта.

Окремо було перевірено функцію log_decision, яка відповідає за запис інформації до журналу. У процесі тестування було встановлено, що кожен виклик функції створює коректний запис із необхідними полями та зберігає його у структурі даних без втрати інформації.

Таким чином, модульне тестування дозволило підтвердити правильність роботи ключових логічних компонентів системи, що забезпечують обробку рішень користувача, формування варіантів вибору та визначення фінального результату гри.

У ході тестування було перевірено роботу механізму формування варіантів вибору. Було встановлено, що функція підбору рецептів повертає коректний набір значень для кожного типу клієнта. Також було перевірено ситуацію можливого повернення порожнього списку, у результаті чого встановлено, що у поточній реалізації передбачено значення за замовчуванням, що виключає відсутність варіантів вибору.

Перевірка механізму вибору користувача підтвердила, що система коректно обробляє вибраний варіант та переходить до наступного етапу сценарію. Було також досліджено можливість повторного виконання вибору, у результаті чого встановлено, що кожна дія застосовується лише один раз, що виключає дублювання ефектів.

У процесі тестування було перевірено правильність зміни внутрішніх параметрів системи. Було встановлено, що значення карми та репутації змінюються відповідно до властивостей обраного рецепту. Проведені тестові сценарії з різними комбінаціями виборів підтвердили коректність накопичення значень та їх вплив на подальший перебіг гри.

Особливу увагу було приділено перевірці механізму журналювання. Було встановлено, що після кожної взаємодії з клієнтом система додає запис до

журналу, який містить інформацію про прийняте рішення. Було також перевірено коректність відображення журналу після завершення гри, що підтвердило правильність порядку записів та повноту збережених даних.

У межах тестування було перевірено поведінку системи при повторному запуску гри. Було встановлено, що журнал очищується на початку нового проходження, що виключає дублювання записів та забезпечує коректність відображення інформації. Така поведінка відповідає поточній логіці реалізації.

Окремим етапом було тестування механізму визначення кінцівки. Було проведено серію тестових проходжень із різними комбінаціями значень карми та репутації. У результаті встановлено, що система коректно визначає фінальний результат відповідно до накопичених значень. Було підтверджено можливість отримання різних типів кінцівок залежно від обраної стратегії проходження.

Було також перевірено граничні значення параметрів, зокрема ситуації з максимально можливими позитивними та негативними значеннями. У результаті встановлено, що алгоритм визначення кінцівки працює коректно та не допускає некоректної класифікації результату.

У процесі тестування було досліджено коректність переходів між сценами. Було встановлено, що зміна сцен відбувається без помилок, а всі графічні ресурси завантажуються коректно. Також було перевірено відсутність накладання зображень та порушення послідовності відображення.

Додатково було перевірено стабільність роботи програмного продукту при багаторазовому запуску та різних сценаріях проходження. У результаті встановлено, що система працює стабільно, без аварійних завершень або збоїв у логіці.

Таким чином, у результаті проведеного тестування було підтверджено працездатність програмного продукту «Магічна аптека», коректність реалізації його функціональних можливостей та відповідність поставленим вимогам. Проведена перевірка показала, що всі основні компоненти системи функціонують у межах очікуваної поведінки, що дозволяє вважати програмний продукт готовим до практичного використання.

3.4 Обмеження та можливості розвитку системи

Програмний продукт «Магічна аптека», розроблений у межах даної кваліфікаційної роботи, реалізує базову модель інтерактивної наративної системи з використанням механіки вибору та накопичення результатів. Незважаючи на коректну роботу та відповідність поставленим вимогам, система має певні обмеження, що зумовлені як обраними технічними рішеннями, так і обсягом реалізації в межах навчального проєкту. Аналіз цих обмежень дозволяє визначити напрями подальшого розвитку програмного продукту та підвищення його функціональності.

Одним із основних обмежень є відносно проста структура ігрового процесу. Хоча користувач має можливість обирати варіанти рішень, загальна послідовність взаємодії з клієнтами є фіксованою. Кожен клієнт обробляється у визначеному порядку, що зменшує варіативність проходження гри та обмежує можливість формування унікальних сценаріїв при повторному використанні. Така структура є доцільною для початкової реалізації, проте у перспективі може бути розширена за рахунок динамічного формування сценарних гілок.

Іншим обмеженням є обмежена кількість варіантів вибору у кожній ситуації. У поточній реалізації користувачу пропонується лише два варіанти рецепту, що спрощує процес прийняття рішень, але водночас знижує глибину взаємодії. Розширення кількості варіантів або введення багаторівневих рішень дозволило б підвищити складність гри та зробити її більш варіативною.

Також слід зазначити, що частина внутрішніх параметрів системи реалізована лише частково. Зокрема, параметр довіри клієнта змінюється у процесі взаємодії, однак не впливає на подальший розвиток подій або доступні варіанти вибору. Це обмежує можливість використання даного параметра як повноцінного механізму моделювання поведінки персонажів.

До обмежень можна віднести і спосіб збереження історії рішень. Хоча використовується механізм persistent, журнал очищується при кожному новому запуску гри, що не дозволяє накопичувати довгострокову статистику або

реалізувати повторні проходження з урахуванням попереднього досвіду користувача.

Узагальнюючи, основні обмеження системи можна сформулювати таким чином:

- лінійна структура сценарію без динамічного розгалуження;
- обмежена кількість варіантів вибору у кожній ситуації;
- неповна інтеграція внутрішніх параметрів у логіку гри;
- відсутність довготривалого збереження прогресу;
- обмежена кількість клієнтів та сценарних ситуацій.

Водночас зазначені обмеження не є критичними і можуть бути усунені у процесі подальшого розвитку системи. Більше того, вони визначають потенційні напрями вдосконалення програмного продукту.

Одним із ключових напрямів розвитку є розширення наративної моделі гри. Це може включати додавання нових клієнтів, ускладнення їхніх історій, а також введення розгалужених сценаріїв, у яких рішення користувача впливатимуть не лише на кінцевий результат, але й на перебіг наступних подій. Такий підхід дозволить створити більш глибоку та нелінійну структуру гри.

Перспективним є також розвиток системи внутрішніх параметрів. Зокрема, параметр довіри клієнта може бути інтегрований у логіку формування варіантів вибору або впливати на реакцію персонажів. Це дозволить зробити взаємодію більш реалістичною та адаптивною.

Ще одним напрямом розвитку є вдосконалення системи збереження даних. Використання повноцінного механізму збереження прогресу дозволить реалізувати багатосесійне використання програмного продукту, а також створити систему досягнень або статистики користувача.

Доцільним є також розширення функціональних можливостей інтерфейсу користувача. Це може включати покращення візуального оформлення, додавання анімацій, звукового супроводу та інтерактивних елементів, що підвищить загальну якість користувацького досвіду.

Основні можливості розвитку системи можна узагальнити наступним чином:

- розширення кількості сценаріїв та клієнтів;
- впровадження нелінійної структури сюжету;
- інтеграція додаткових параметрів у логіку гри;
- реалізація системи збереження та прогресу;
- покращення інтерфейсу та візуального оформлення;
- додавання нових механік взаємодії.

Таким чином, програмний продукт «Магічна аптека» є базовою реалізацією інтерактивної наративної системи, яка може бути розширена та вдосконалена у різних напрямах. Виявлені обмеження не знижують цінності розробленого рішення, а навпаки визначають перспективи його подальшого розвитку. Це підтверджує можливість використання даного програмного продукту як основи для створення більш складних та функціонально насичених систем у майбутньому.

Висновки до розділу 3

У третьому розділі кваліфікаційної роботи було реалізовано програмний продукт «Магічна аптека», який представляє собою інтерактивну візуальну новелу, створену з використанням рушія Ren’Py та мови програмування Python. У процесі розробки було впроваджено основні функціональні компоненти системи, зокрема механізм діалогової взаємодії з користувачем, вибір варіантів дій, а також обробку наслідків прийнятих рішень.

Реалізація програмного продукту базується на використанні сценарної моделі, в якій користувач послідовно взаємодіє з клієнтами, обираючи відповідні варіанти рішень. Застосування об’єктно-орієнтованого підходу дозволило структуровано описати основні сутності системи, зокрема клієнтів та рецепти, а також забезпечити гнучкість у подальшому розширенні функціоналу.

У ході розробки було реалізовано систему внутрішніх параметрів, що включає показники карми та репутації, які змінюються залежно від дій користувача. На основі цих параметрів формується підсумковий результат роботи системи у вигляді різних варіантів кінцівок, що забезпечує варіативність проходження гри та підвищує її інтерактивність.

Проведене функціональне тестування підтвердило коректність роботи основних компонентів програмного продукту, зокрема обробки діалогів, вибору варіантів рішень, зміни внутрішніх параметрів та визначення кінцевого результату. Усі ключові сценарії виконуються без помилок, що свідчить про стабільність реалізованої логіки.

Аналіз використання програмного продукту показав, що система забезпечує зрозумілу та послідовну взаємодію користувача з інтерфейсом, дозволяє приймати рішення на основі представленої інформації та отримувати відповідний результат. Реалізований журнал рішень надає можливість відстежувати дії користувача, що підвищує прозорість та інформативність роботи системи.

Отже, у результаті виконання третього розділу було досягнуто поставленої мети щодо реалізації програмного продукту, підтверджено його працездатність та відповідність визначеним вимогам, що свідчить про завершеність етапу розробки та готовність системи до практичного використання.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було досліджено теоретичні основи створення інтерактивних візуальних новел із використанням платформи Ren’Py. Особливу увагу приділено особливостям жанру, ролі інтерактивності, механізмам вибору користувача та впливу рішень на перебіг ігрового процесу. Також проаналізовано функціональні можливості платформи Ren’Py, її переваги у створенні подібних програмних продуктів та доцільність використання у межах поставленого завдання.

У процесі роботи було спроєктовано модель програмного продукту, яка передбачає послідовну взаємодію користувача з клієнтами, систему вибору рішень із відповідними наслідками, а також використання внутрішніх параметрів для формування результату. Значну увагу приділено структуризації даних, зокрема опису сутностей клієнтів і рецептів, а також логіці обробки дій користувача.

У ході реалізації розроблено програмний продукт у вигляді інтерактивної візуальної новели, в якій користувач приймає рішення щодо вибору дій, що впливають на зміну показників карми та репутації. На основі цих параметрів формується підсумковий результат у вигляді різних варіантів кінцівок, що забезпечує варіативність проходження та підвищує інтерактивність системи. Також реалізовано журнал рішень, що дозволяє відстежувати дії користувача протягом гри.

Проведене тестування підтвердило працездатність основних компонентів програмного продукту, коректність реалізації сценарної логіки, а також стабільність роботи системи вибору та формування кінцевого результату. Реалізований інтерфейс забезпечує зрозумілу та послідовну взаємодію користувача з системою.

У результаті виконання кваліфікаційної роботи було досягнуто поставленої мети — розроблено інтерактивний програмний продукт на платформі Ren’Py, який відповідає визначеним вимогам та демонструє

можливості алгоритмічного формування сценарної траєкторії, переходів між подіями та фінального результату на основі рішень користувача.

Розроблений продукт може бути використаний як основа для подальшого розвитку та вдосконалення у сфері інтерактивних ігрових систем.

ПЕРЕЛІК ДЖЕРЕЛ ТА ПОСИЛАННЯ

1. Murray J. H. Hamlet on the Holodeck: The Future of Narrative in Cyberspace. Updated edition [Електронний ресурс]. - Cambridge, MA: MIT Press, 2017. URL: https://mitpress.mit.edu/9780262533485/hamlet-on-the-holodeck/ (дата звернення: 13.03.2026).
2. Presedence Research Official Site URL: https://www.precedenceresearch.com/video-game-market (дата звернення 13.03.2026)
3. Camingue J., Carstensdottir E., Melcer E. F. What is a visual novel? // Proceedings of the ACM on Human-Computer Interaction. 2021. Vol. 5, CHI PLAY, Article 285. URL: https://doi.org/10.1145/3474712 (дата звернення: 14.03.2026).
4. Bostan B., Marsh T. The “Interactive” of Interactive Storytelling: Customizing the Gaming Experience // Entertainment Computing - ICEC 2010 / ed. by H. S. Yang, R. Malaka, J. Hoshino, J. H. Han. Berlin ; Heidelberg : Springer, 2010. P. 472-475. (Lecture Notes in Computer Science ; Vol. 6243). URL: https://doi.org/10.1007/978-3-642-15399-0_63 (дата звернення: 14.03.2026).
5. Papers, Please Official Site URL: https://papersplea.se/ (дата звернення 14.03.2026)
6. Steam Store URL: https://store.steampowered.com/app/1210320/Potion_Craft_Alchemist_Simulator/?l=ukrainian (дата звернення 15.03.2026)
7. Steam Store URL: https://store.playstation.com/ru-ua/product/EP6757-PPSA03329_00-0902745586622586 (дата звернення 15.03.2026)
8. Steamworks Documentation. Steam tags [Електронний ресурс]. URL: https://partner.steamgames.com/doc/store/tags (дата звернення: 15.03.2026).
9. Ren’Py Documentation [Електронний ресурс]. URL: https://www.renpy.org/doc/html/index.html (дата звернення: 15.03.2026).
10. Steamworks Documentation. User reviews URL: https://partner.steamgames.com/doc/store/reviews (дата звернення: 16.03.2026).
11. El Ariss O., Xu D. Handbook of Finite State Based Models and Applications. - CRC Press, 2011. URL: https://api.pageplace.de/preview/DT0400.9781439846193_A23985397/preview-9781439846193_A23985397.pdf (дата звернення: 16.03.2026).
12. Ren’Py Documentation. Persistent Data [Електронний ресурс]. URL: https://www.renpy.org/doc/html/persistent.html (дата звернення: 16.03.2026).
13. ASTQB / ISTQB. 4.2 Black-Box Test Techniques [Електронний ресурс]. URL: https://astqb.org/4-2-black-box-test-techniques/ (дата звернення: 12.04.2026).