МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
УМАНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет економіки, підприємництва та інформаційних технологій

Кафедра інформаційних технологій

Допущено до захисту
"_____" _____2026 р.
Зав. кафедри, к.т.н., доцент
_____ **Ліщук Р.І.**

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

РОЗРОБКА ОСВІТНЬОГО КОМПОНЕНТУ ДЛЯ ІГРОВОЇ RPG-ПЛАТФОРМИ З ВИКОРИСТАННЯМ ШТУЧНОГО ІНТЕЛЕКТУ

Виконав – студент 4 курсу, 41-кн
групи освітньої програми
"Комп'ютерні науки"
спеціальності 122 "Комп'ютерні науки"
Павленко Вадим Володимирович

Науковий керівник – д.т.н., професор
Кулаков Павло Ігорович

Умань – 2026

ЗМІСТ

ВСТУП 7
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ 9
1.1 Аналіз предметної області та актуальність розробки 9
1.2 Огляд та аналіз існуючих аналогів 15
1.3. Обґрунтування мети та постановка задачі 23
РОЗДІЛ 2. МОДЕЛЮВАННЯ ТА МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ ОСВІТНЬОГО КОМПОНЕНТУ 26
2.1 Моделювання освітнього компоненту 26
2.2 Математична модель адаптивного навчання 29
2.3 Розробка концепції гейміфікації та ігрових механік 34
2.4 Обґрунтування архітектурних рішень та вибір засобів програмної реалізації 37
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ОСВІТНЬОГО КОМПОНЕНТУ 41
3.1 Реалізація адаптивної навчальної складової 41
3.2 Реалізація ШІ-асистента та серверної частини 45
3.3 Реалізація ігрових механік та збереження даних 49
3.4 Тестування готового освітнього компоненту 58
ВИСНОВКИ 61
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 63
ДОДАТКИ ..... **Помилка! Закладку не визначено.**
Додаток А ..... **Помилка! Закладку не визначено.**

ВСТУП

Сучасний етап розвитку інформаційних технологій характеризується активним впровадженням інтелектуальних систем у сферу освіти, що зумовлено потребою підвищення ефективності навчального процесу, індивідуалізації навчання та підвищення мотивації здобувачів освіти. Традиційні підходи до навчання часто не враховують індивідуальні особливості студентів, зокрема рівень їх підготовки, швидкість засвоєння матеріалу та особистісні інтереси, що знижує загальну ефективність освітнього процесу.

Одним із перспективних напрямів вирішення зазначеної проблеми є використання адаптивних навчальних систем, які здатні динамічно змінювати складність навчального матеріалу відповідно до поточного рівня знань користувача. Особливу роль у таких системах відіграють методи штучного інтелекту, що дозволяють моделювати процес навчання, оцінювати рівень засвоєння знань та формувати персоналізовані рекомендації.

Паралельно з розвитком адаптивних технологій значної популярності набуває гейміфікація освітнього процесу, яка передбачає інтеграцію ігрових механік у навчальне середовище. Використання елементів рольових ігор, системи досягнень, прогресії персонажа та інтерактивного сюжету дозволяє підвищити залученість користувачів, сформувати стійку мотивацію до навчання та зробити процес засвоєння знань більш привабливим.

У цьому контексті особливий інтерес становить поєднання адаптивного навчання, гейміфікації та штучного інтелекту в єдиній програмній системі. Такий підхід дозволяє не лише адаптувати навчальний контент до рівня користувача, але й забезпечити інтерактивну взаємодію з інтелектуальним асистентом, який виконує роль наставника у віртуальному ігровому середовищі.

Актуальність теми кваліфікаційної роботи зумовлена необхідністю розробки освітнього компоненту, який поєднує адаптивні алгоритми, ігрові механіки та сучасні технології штучного інтелекту з метою підвищення ефективності навчання та зацікавленості користувачів.

Метою кваліфікаційної роботи є розробка з адаптивним штучним інтелектом, гейміфікованим середовищем та індивідуалізованим рівнем складності навчальних завдань.

Об’єктом дослідження є процеси функціонування та організації інтелектуальних навчальних систем у сфері освіти, спрямованих на персоналізацію навчання.

Предметом дослідження є методи, алгоритми та програмні засоби реалізації освітнього компоненту з використанням адаптивного навчання, гейміфікації та технологій штучного інтелекту.

Методами дослідження є аналіз наукової літератури та існуючих рішень у сфері інтелектуальних навчальних систем, методи інтелектуального аналізу даних для оцінювання рівня знань користувача та адаптації складності завдань, а також методи програмної інженерії для проєктування та реалізації клієнт-серверної архітектури компонента.

Практичне значення отриманих результатів полягає у можливості використання розробленого освітнього компоненту як основи для створення повноцінних інтелектуальних навчальних платформ, а також у впровадженні елементів адаптивного навчання та гейміфікації в освітній процес.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз предметної області та актуальність розробки

У сучасних умовах розвитку цифрової освіти все більшої ваги набувають інноваційні методи організації навчального процесу, що спрямовані на підвищення мотивації, залученості та результативності здобувачів знань. Традиційні форми навчання, які спираються на лекційно-семінарську модель та стандартні тести, дедалі частіше демонструють обмежену ефективність, особливо серед покоління, яке звикло до інтерактивних цифрових середовищ [1; 2]. Ця ситуація створює потребу в пошуку нових дидактичних підходів, здатних поєднати глибоке освоєння матеріалу з високою мотиваційною залученістю користувачів. Порівняльний аналіз підходів наведено у таблиці 1.1.

Таблиця 1.1 Порівняльний аналіз традиційного і гейміфікованого підходу

Критерій	Традиційний підхід	Гейміфікований підхід
Залученість учнів	Високий рівень пасивного сприйняття матеріалу	Активна участь завдяки ігровим механізмам
Мотивація	Переважно зовнішня (оцінки, контроль)	Переважно внутрішня (особисті досягнення, цікавість)
Стиль навчання	Лінійне подання матеріалу	Динамічний, адаптивний сценарій навчання
Форма взаємодії	Одностороння комунікація	Взаємодія з системою й іншими учнями
Показники результативності	Залежність від тестування та репродуктивних завдань	Розвиток критичного мислення та креативності через виконання завдань і досягнень результатів

Гейміфікація як концепція означає використання ігрових механік та структурних елементів гри у неігрових контекстах, що сприяє підвищенню зацікавленості та участі користувачів у навчальному процесі. У педагогічній науці гейміфікація розглядається не лише як інструмент мотивації, але і як механізм, що змінює спосіб взаємодії здобувача знань із навчальним матеріалом та стимулює активне мислення. Сучасні дослідження підкреслюють, що

гейміфікація здатна підвищувати пізнавальну активність, зменшувати когнітивне перевантаження та сприяти закріпленню знань завдяки інтерактивним завданням та безпосередньому зворотному зв’язку [3, с. 45-46].

Зокрема, у огляді сучасних робіт щодо застосування штучного інтелекту в гейміфікованих навчальних середовищах зазначається, що технології машинного навчання, адаптивні алгоритми та аналітика даних дозволяють трансформувати гейміфікацію з простого набору мотиваційних механік у комплексну динамічну систему, яка моделює поведінку учня, адаптує контент та підтримує когнітивні процеси в режимі реального часу. Дослідження підтверджують, що ШІ-алгоритми підвищують індивідуалізацію та ефективність [4, с. 2-5], а синергія ШІ і гейміфікації створює адаптивні траєкторії навчання з аналітикою в реальному часі [5, с. 40].

Для обґрунтування актуальності гейміфікації в освіті важливо звернутись до емпіричних даних, що демонструють її вплив на навчальні результати. Так, у рамках експериментальних досліджень було показано, що адаптивно-гейміфіковані оцінювання у змішаному навчанні сприяють підвищенню продуктивності учнів, збільшують рівень залучення та покращують загальні результати у порівнянні зі стандартними формами оцінювання [6, с. 5-6; 9]. Практичне підтвердження вищезазначених теоретичних положень простежується у результатах дослідження. На рис. 1.1 відображено позитивну тенденцію зміни рівня розвитку соціальної компетентності учнів: після впровадження елементів рольової гри спостерігається суттєве зростання показників «високого» та «достатнього» рівнів при одночасному скороченні частки учнів із «низьким» рівнем [9].

Крім того, систематичні огляди показують, що інтеграція механізмів адаптивної гейміфікації в цифрові освітні платформи сприяє поліпшенню критичного мислення, виконавчих функцій та здатності до саморегуляції у навчальному процесі, що є ключовими складовими ефективного навчання на довгострокову перспективу [6, с. 9-11; 9].

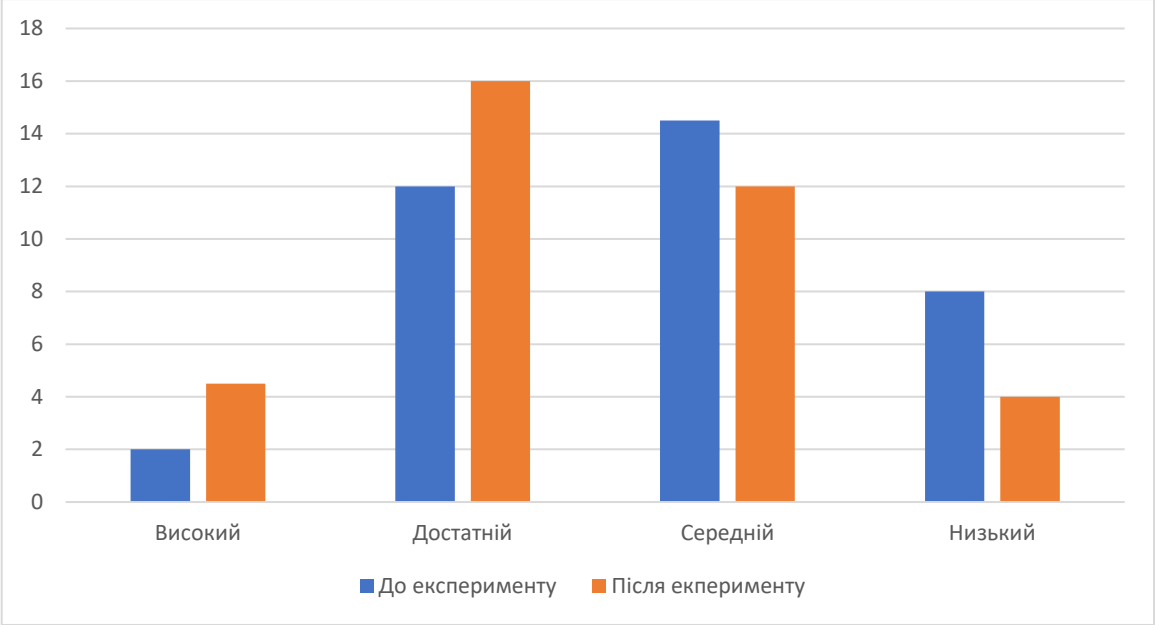


Рис. 1.1 Рівні соціальної компетентності учнів у ході дослідження

На рівні шкільної педагогіки численні дослідження підтверджують, що застосування ігрових методик підвищує рівень мотивації учнів, сприяє кращому засвоєнню матеріалу та активізує навчальну діяльність. Дослідження у сфері лінгвістичної освіти демонструють позитивний вплив гейміфікації на формування мовних компетентностей, зокрема при вивченні іноземних мов [7, с. 222-223]. Схожі результати отримані й у дослідженнях з гейміфікації як методології для студентів STEM-спеціальностей, де інтерактивні ігрові завдання сприяли зростанню академічних досягнень [8, с. 13-15].

Ці дані демонструють, що гейміфікація не лише підвищує мотиваційний компонент, але й має прямий позитивний ефект на результати навчання, особливо коли вона поєднана з інтерактивними та адаптивними елементами.

Особливий інтерес викликає застосування гейміфікації у форматі рольових ігор, що виходить за межі традиційної «бали-значки-лідерборд». RPG-ігри мають розвинену сюжетну основу, інтерактивні сценарії, систему персонажів та прогресу, що природно стимулює когнітивну активність та формування навичок прийняття рішень в умовах змінних параметрів задач. З точки зору психології навчання, ігрові середовища такого типу сприяють виникненню стану «flow» (стану потоку) – психологічного стану оптимальної концентрації та мотивації,

коли завдання відповідає рівню навичок учня, що підсилює процес засвоєння та критичного мислення [9, с. 121-124]. Загальна модель інтеграції RPG-елементів у навчальний процес зображено на рис. 1.2.

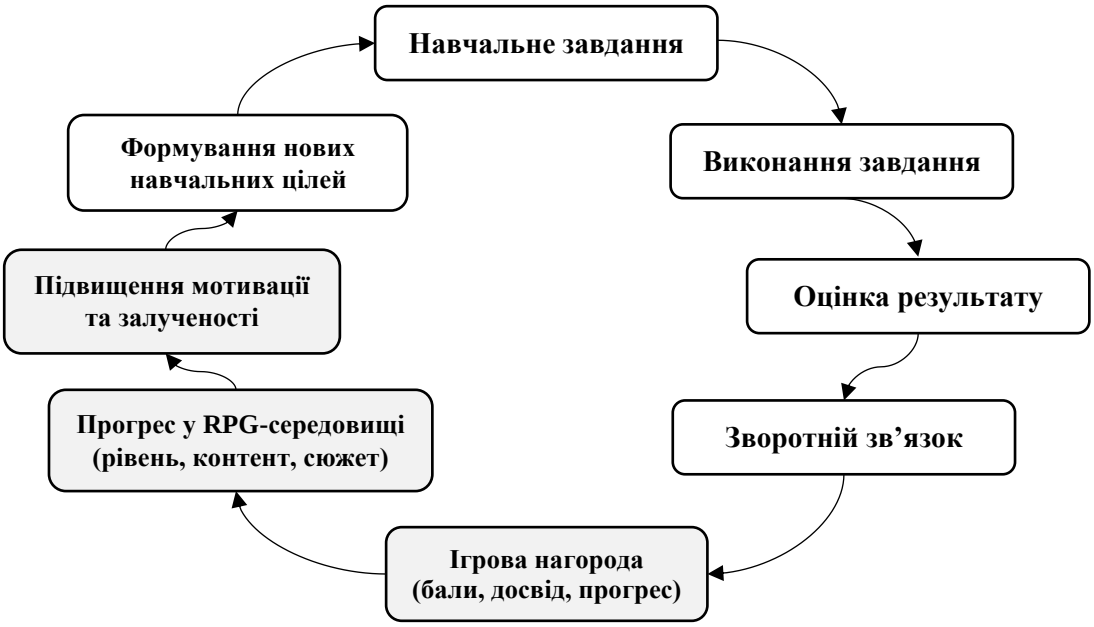


Рис. 1.2 Модель інтеграції RPG-елементів у навчальний процес

У дослідженнях з рольових ігор у навчанні зазначається, що участь у таких інтерактивних сценаріях сприяє активізації внутрішньої мотивації, поглибленню розуміння складних концепцій та розвитку міжособистісних навичок, таких як комунікація, співпраця та стратегічне мислення [9, с. 7-9]. Крім того, RPG-середовища дозволяють моделювати реальні або уявні професійні ситуації, що створює умови для практичного застосування теоретичних знань.

У порівнянні з класичними елементами гейміфікації, де мотивація здебільшого побудована на зовнішніх «нагородах», RPG-підхід органічно інтегрує навчальні завдання у сюжетну структуру, що підтримує внутрішню мотивацію та довгострокову зацікавленість.

Сучасний розвиток навчальних технологій дедалі більше акцентує увагу не лише на загальному дизайні гейміфікації, але й на можливостях її адаптації під конкретного учня. Зокрема, використання штучного інтелекту в гейміфікованих навчальних системах дозволяє реалізувати динамічну персоналізацію

навчального маршруту, адаптуючи складність завдань, темп подання матеріалу, а також тип зворотнього зв’язку у залежності від поведінкових даних та успішності користувача [10].

Адаптивні алгоритми здатні коригувати навчальні траєкторії так, щоб підтримувати баланс між викликом та рівнем навичок учня, що безпосередньо пов’язано зі станом потоку та оптимальним навчальним темпом. Це дозволяє уникати ситуацій, коли завдання стають надто простими або надто складними, що може призводити до втрати мотивації або фрустрації [10; 11]. Загальний принцип адаптивної персоналізації навчання зображено на рис. 1.3.

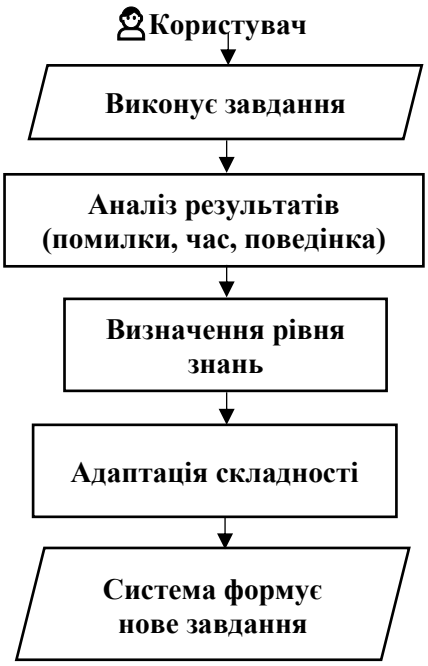


Рис. 1.3 Принцип адаптивної персоналізації навчання

Сьогодні такі адаптивні підходи активно досліджуються в рамках штучного інтелекту та освітніх технологій. Вони включають механізми поведінкового моделювання, аналізу помилок, прогнозування складності наступних завдань на основі вбудованих моделей учнівських профілів. Це створює основу для побудови інтелектуальних навчальних середовищ, що здатні враховувати унікальні освітні потреби кожного користувача [11].

Подальший розвиток адаптивних навчальних систем безпосередньо пов’язаний із впровадженням методів штучного інтелекту, які виступають

ключовим інструментом забезпечення глибокої персоналізації та інтерактивності освітнього процесу. Якщо адаптивність визначає здатність системи підлаштовуватись під користувача, то саме штучний інтелект забезпечує механізми, за допомогою яких така адаптація стає можливою та динамічною [12, с. 252].

У сучасних інтелектуальних навчальних системах ШІ використовується для побудови моделей знань користувача, що відображають рівень засвоєння матеріалу, характер помилок, темп навчання та інші поведінкові параметри. Такі моделі формуються на основі аналізу великого обсягу даних, включаючи результати виконання завдань, час реакції, частоту повторень і навіть послідовність дій користувача [12, с. 252-253]. Це дозволяє не лише фіксувати поточний рівень знань, але й прогнозувати подальший розвиток навчальної траєкторії та адаптувати її у реальному часі.

Особливого значення набувають алгоритми, що враховують динаміку засвоєння та забування матеріалу. Використання таких підходів дозволяє своєчасно визначати моменти, коли знання потребують повторення, та формувати відповідні навчальні завдання [12, с. 253]. Крім того, сучасні системи прагнуть враховувати ефект випадкового вгадування, що підвищує точність оцінювання реального рівня підготовки користувача. Штучний інтелект забезпечує більш глибоке та об’єктивне моделювання навчального процесу у порівнянні з традиційними підходами.

Окремим напрямом використання ШІ є інтеграція інтелектуальних асистентів у навчальні системи. На відміну від статичних підказок або шаблонних пояснень, ШІ-асистент здатний взаємодіяти з користувачем у діалоговому режимі, адаптувати стиль пояснення, відповідати на запитання та надавати рекомендації з урахуванням індивідуального рівня підготовки. Це створює ефект персонального наставника, що суттєво підвищує залученість користувача та сприяє глибшому розумінню навчального матеріалу. Зокрема, у межах проєкту на базі Національного технічного університету «Харківський політехнічний інститут» інтеграція ШІ-асистента для розв’язування задач з

алгоритмізації дозволила скоротити середній час виконання контрольних завдань на 27% [13, с. 11].

Інтеграція штучного інтелекту з гейміфікацією дозволяє перейти від статичних ігрових механік до динамічних адаптивних ігрових середовищ. Зокрема, ШІ може використовуватись для регулювання складності ігрових завдань, керування темпом прогресу, підбору індивідуальних нагород та навіть адаптації сюжетних елементів відповідно до поведінки користувача [14].

Разом із тим, аналіз сучасних рішень показує, що використання штучного інтелекту часто обмежується окремими аспектами навчального процесу. Це зумовлено обмеженістю технічної інфраструктури, недостатнім рівнем цифрової компетентності користувачів, відсутністю усталених методичних підходів, а також фінансовими обмеженнями впровадження таких технологій. [12, с. 254]

Використання штучного інтелекту є ключовим фактором розвитку сучасних адаптивних навчальних систем та гейміфікованих освітніх середовищ. Поєднання алгоритмів аналізу даних, моделей знань користувача та інтелектуальних асистентів створює передумови для формування нових підходів до організації навчального процесу.

1.2 Огляд та аналіз існуючих аналогів

У межах дослідження доцільно проаналізувати існуючі програмні рішення та цифрові освітні платформи, що реалізують елементи адаптивності, штучного інтелекту та гейміфікації у навчальному процесі. Такий аналіз дозволяє визначити ключові технологічні підходи та виявити переваги й обмеження сучасних систем, що є основою для формування вимог до розроблюваного рішення.

Розглянемо три сучасні системи-аналоги: Knewton, Squirrel AI та Duolingo. Вибір зазначених платформ обумовлений їх використанням адаптивного навчання, елементів штучного інтелекту та гейміфікації, що дає змогу визначити основні тенденції розвитку освітніх технологій.

Knewton є однією з перших комерційних платформ у сфері адаптивного

навчання, що реалізує алгоритмічну персоналізацію освітнього контенту. Система не надає власних навчальних курсів, а функціонує як адаптивний модуль, який інтегрується у сторонні освітні продукти та забезпечує підбір матеріалу відповідно до індивідуальної навчальної траєкторії користувача. Основу її роботи становить аналіз поведінкових даних, моделювання рівня знань користувача та прогнозування оптимальної послідовності навчання на основі алгоритмів машинного навчання [15, с. 8]. Основні характеристики системи наведено в таблиці 1.2.

Таблиця 1.2 Основні характеристики Knewton

Параметр	Опис
Тип системи	Адаптивна навчальна платформа
Основна функція	Персоналізація навчального контенту на основі алгоритмів машинного навчання
Підхід до адаптивності	Аналіз поведінкових даних учня, прогнозування оптимального шляху навчання
Інтеграція	Вбудовується у LMS або освітні курси сторонніх видавців
Предметна спеціалізація	Традиційні академічні дисципліни (математика, науки, мови)
ШІ-асистент	Відсутній як діалоговий наставник, рекомендаційний рівень через адаптацію контенту
Гейміфікація	Немає сюжетної чи RPG-мотивації
Доступ	Платний, комерційний B2B продукт

Архітектурно система використовує набір адаптивних алгоритмів, що забезпечують оцінювання поточного рівня знань користувача, виявлення прогалин та оптимізацію послідовності подачі навчального матеріалу. На основі аналізу поведінкових даних формується індивідуальна навчальна траєкторія та здійснюється підбір наступних завдань з урахуванням особливостей користувача.

Основною перевагою платформи Knewton є високий рівень адаптивності навчання, що досягається за рахунок аналізу результатів користувача та прогнозування його подальшого прогресу. Система формує персоналізовані траєкторії навчання та надає розширені аналітичні звіти, які містять інформацію

про прогрес, виявлені прогалини та рекомендації щодо подальших дій. Інтеграція з системами керування навчанням (LMS) забезпечує гнучкість використання та можливість масштабування платформи [15, с. 8, 11].

Разом із тим, система має низку обмежень. Зокрема, гейміфікація практично відсутня: не підтримуються ігрові механіки, сюжетна мотивація та рольова взаємодія. Крім того, платформа не передбачає повноцінної взаємодії з користувачем через ШІ-асистента, обмежуючись рекомендаційною функцією. Також Knewton потребує попередньо підготовленого контенту, оскільки не генерує навчальні сценарії або інтерактивні наративи, що звужує можливості її використання у творчих і гейміфікованих середовищах [15, с. 8, 11].

Переваги та недоліки узагальнено у таблиці 1.3.

Таблиця 1.3 Підсумкова таблиця переваг та недоліків Knewton

Категорія	Переваги	Недоліки
Адаптивність	Формує персоналізовані траєкторії навчання, оптимізує послідовність тем	Потребує підготовленого контенту від автора, не створює навчальні сценарії автоматично
Аналітика	Розширені звіти для інструкторів: прогрес, прогалини, рекомендації	Не забезпечує активну взаємодію з учнем через ШІ-асистента
Інтеграція	Можливість вбудовування у LMS	Не підтримує ігрові механіки, сюжетні елементи, рольову взаємодію
Мотивація	Підтримка учня через адаптацію матеріалу	Відсутність інтегрованої гейміфікації чи RPG-прогресу
Використання	Підходить для великих освітніх платформ та академічних курсів	Не адаптована до творчих або ігрових сценаріїв

Платформа Knewton забезпечує високий рівень персоналізації навчального процесу завдяки використанню адаптивних алгоритмів. Водночас її орієнтація виключно на оптимізацію навчальної траєкторії без застосування механізмів мотивації та інтерактивної взаємодії обмежує рівень залучення користувача,

особливо у порівнянні з гейміфікованими середовищами, де навчання інтегрується у сюжетну структуру.

Squirrel AI – це освітня платформа, що активно застосовує ШІ для адаптації навчального процесу відповідно до поведінкових даних користувача. Система використовує knowledge-maps – структуровані схеми знань, що дозволяють оцінювати не лише поточний рівень знань учня, а й будувати графи залежностей між темами для коригування шляху навчання [16; 17, с. 308-310].

Основні характеристики системи наведено в таблиці 1.4.

Таблиця 1.4 Основні характеристики Squirrel AI

Параметр	Опис
Тип системи	ШІ-платформа адаптивного навчання
Основна функція	Персоналізація навчального процесу на основі knowledge-maps та поведінкових даних учня
Підхід до адаптивності	Динамічне коригування завдань у реальному часі, прогнозування оптимального шляху опанування знань
Інтеграція	Переважно освітні заклади, використання у масштабних шкільних платформах
Предметна спеціалізація	Традиційні академічні дисципліни (математика, науки, мови)
ШІ-асистент	Немає повноцінного діалогового наставника
Гейміфікація	Мінімальна, немає сюжетної або RPG-мотивації
Доступ	Комерційна система; ліцензійна підписка для шкіл та освітніх центрів

Платформа Squirrel AI широко застосовується в освітніх закладах і демонструє можливість масштабної персоналізації навчального процесу [17, с. 308]. Система формує профіль користувача на основі аналізу відповідей, часу реакції та успішності, що дозволяє генерувати адаптивні рекомендації щодо подальшого навчання.

Основною перевагою платформи є високий рівень адаптивності, який забезпечується динамічним коригуванням завдань у реальному часі. Використання knowledge-maps дозволяє виявляти прогалини у знаннях та визначати логічні зв’язки між темами, що сприяє формуванню індивідуальної навчальної траєкторії. Аналіз поведінкових і результативних показників

користувача підвищує точність підбору навчального матеріалу та ефективність засвоєння знань [17, с. 308-310].

Разом із тим, система має низку обмежень. Зокрема, гейміфікація реалізована мінімально: відсутні сюжетні та рольові механіки, що знижує рівень мотивації користувача. Платформа орієнтована переважно на академічні дисципліни та менш придатна для міждисциплінарних або творчих сценаріїв. Крім того, Squirrel AI не передбачає повноцінного ШІ-асистента у вигляді діалогового наставника, обмежуючись рекомендаційною підтримкою без глибокої інтерактивної взаємодії [17, с. 308-310].

Переваги та обмеження платформи Squirrel ШІ узагальнено у таблиці 1.5.

Таблиця 1.5 Підсумкова таблиця переваг та недоліків Squirrel AI

Категорія	Переваги	Недоліки
Адаптивність	Персоналізовані траєкторії на основі knowledge-maps, динамічне коригування завдань	Потребує підготовленого контенту від автора, не створює навчальні сценарії автоматично
Аналітика	Аналіз поведінки, часу реакції та успішності, точне виявлення прогалин	Відсутня активна взаємодія через ШІ-асистента
Інтеграція	Масштабованість, інтеграція з LMS	Не підтримує сюжет, RPG та ігрові механіки
Мотивація	Підвищення ефективності через адаптивні завдання	Відсутня гейміфікація та емоційна мотивація
Використання	Широко застосовується в школах, ефективна для академічних дисциплін	Не придатна для творчих і сюжетно-орієнтованих сценаріїв

Платформа Squirrel AI є одним із найбільш розвинених прикладів систем адаптивного навчання, що використовують штучний інтелект для побудови персоналізованих навчальних траєкторій. Водночас відсутність ШІ-асистента та інтегрованих мотиваційних механізмів, зокрема сюжетних і рольових елементів, обмежує її здатність забезпечувати тривале залучення користувачів порівняно з гейміфікованими середовищами.

Duolingo – це одна з найпопулярніших платформ для вивчення мов, яка

поєднує елементи адаптивного навчання та гейміфікації. Система використовує алгоритми персоналізації, щоб визначати індивідуальний рівень знань користувача та формувати завдання відповідно до його прогресу [18, с. 1].

Основні характеристики платформи наведено в таблиці 1.6.

Таблиця 1.6 Основні характеристики Duolingo

Параметр	Опис
Тип системи	Гейміфікована мовна платформа
Основна функція	Персоналізація навчання на основі історії відповідей, помилок і успіхів із адаптивним підбором вправ
Підхід до адаптивності	Динамічне генерування вправ у реальному часі з повторенням матеріалу та формуванням індивідуальної траєкторії
Інтеграція	Масове використання через мобільні додатки та веб-платформу для широкого кола користувачів
Предметна спеціалізація	Вивчення іноземних мов (понад 40), фокус на лексиці, граматиці та мовних навичках
ШІ-асистент	Частковий: підказки та пояснення без повноцінного діалогового наставника
Гейміфікація	Рівні, очки, бейджі, рейтинги; відсутність сюжетної та RPG-логіки
Доступ	Freemium: безкоштовна версія + платна підписка з додатковими можливостями

Основними перевагами платформи Duolingo є широке масове охоплення та доступність, що забезпечується підтримкою понад 40 мов та наявністю мільйонів користувачів по всьому світу. Такий масштаб дозволяє проводити ефективне навчання у різних вікових групах та культурних середовищах. Висока мотивація користувачів досягається за рахунок інтенсивного застосування гейміфікаційних механік: система використовує рівні, бейджі, очки досвіду та рейтингові таблиці, що стимулює регулярне навчання та формує ефект «занурення», який підсилює засвоєння матеріалу. Адаптивність завдань забезпечується алгоритмами підбору вправ відповідно до історії відповідей користувача, врахуванням помилок та успіхів, а також повторенням матеріалу для закріплення знань, що дозволяє формувати персоналізовану траєкторію навчання. Крім того, Duolingo пропонує

короткі інтерактивні модулі з оперативним зворотнім зв’язком, що підтримує високий рівень залучення та знижує когнітивне навантаження на користувача [18, с. 2; 19].

Разом із цим, платформа має обмеження. ШІ-асистент обмежується перевіркою правильності виконання вправ і наданням підказок, але не аналізує поведінку користувача, наприклад рівень мотивації, настрій чи стратегічні помилки. Інтеграція сюжетних механік або RPG-логіки відсутня, тож навчання не формує повноцінного ігрового світу. Платформа орієнтована переважно на мовну спеціалізацію і менш придатна для інших навчальних дисциплін. Крім того, розширені функції, як відсутність реклами чи необмежене навчання, доступні лише у платній підписці Duolingo Plus [19]. Переваги та обмеження платформи Duolingo узагальнено у таблиці 1.7.

Таблиця 1.7 Підсумкова таблиця переваг та недоліків Duolingo

Категорія	Переваги	Недоліки
Адаптивність	Формує індивідуальні траєкторії на основі відповідей і успішності, динамічно підбирає та повторює матеріал	Орієнтована на мовний контент, не враховує поведінку поза вправами та процеси вгадування
Аналітика	Оцінка відповідей і швидкий зворотній зв’язок через інтерактивні модулі	Відсутня повноцінна діалогова взаємодія та глибокий аналіз поведінки і мотивації
Інтеграція	Мобільний і веб-доступ, масове охоплення	Не підтримує сюжетні механіки, RPG-прогрес і складні ігрові елементи
Мотивація	Гейміфікація (XP, рівні, бейджі, рейтинги), ефект залучення	Відсутня сюжетна інтеграція та глибока ігрова взаємодія
Використання	Широке застосування для вивчення мов, можлива безкоштовна підписка	Обмеження мовною спеціалізацією, непридатність для інших дисциплін

Duolingo демонструє достатній рівень гейміфікації та адаптивності для масового вивчення мов, проте його функціональні можливості обмежені мовним контекстом і не включають повноцінного ШІ-асистента, інтегрованого сюжету або

RPG-механік. Це визначає перспективи розвитку навчальних систем, у яких штучний інтелект не лише адаптує завдання, а й виступає інтерактивним наставником, навчальний процес інтегрується у сюжетну логіку, а складність завдань коригується з урахуванням індивідуальних характеристик користувача та результатів його навчання. Таким чином, Duolingo є прикладом часткової реалізації адаптивності та гейміфікації, проте не забезпечує повноцінної інтеграції ШІ-взаємодії, сюжетного середовища та персоналізованого управління складністю.

Проведений порівняльний аналіз сучасних освітніх платформ, зокрема Knewton, Squirrel AI та Duolingo, показав, що існуючі рішення орієнтовані на реалізацію окремих аспектів навчального процесу, проте не забезпечують їх комплексної інтеграції. Узагальнену характеристику зазначених систем наведено у таблиці 1.8.

Таблиця 1.8 Порівняльна характеристика існуючих систем-аналогів

Критерій	Knewton	Squirrel AI	Duolingo
Тип системи	Адаптивна навчальна платформа	ШІ-платформа адаптивного навчання	Гейміфікована мовна платформа
Рівень адаптивності	Високий (алгоритмічна персоналізація)	Високий (knowledge-maps, поведінковий аналіз)	Середній (адаптація вправ)
ШІ-взаємодія	Відсутня або мінімальна	Обмежена (рекомендації)	Часткова (підказки, перевірка)
Гейміфікація	Відсутня	Мінімальна	Висока (XP, рівні, рейтинги)
Сюжет / RPG-елементи	Відсутні	Відсутні	Відсутні
Персоналізація складності	Так	Так	Частково
Міждисциплінарність	Обмежена	Обмежена	Відсутня (фокус на мовах)
Інтерактивність	Низька	Середня	Висока
Масштабованість	Висока (LMS)	Висока	Дуже висока (масовий продукт)
Основний недолік	Відсутність гейміфікації та ШІ-взаємодії, платна	Відсутність гейміфікації та ШІ-асистента, платна	Відсутність глибокого ШІ та сюжетної інтеграції, базові аспекти гейміфікації

Аналіз наведених даних свідчить, що платформи Knewton та Squirrel AI забезпечують високий рівень адаптивності навчання за рахунок використання алгоритмічної персоналізації та аналізу поведінки користувача. Водночас вони не передбачають використання повноцінного ШІ-асистента як інтерактивного елемента взаємодії та практично не застосовують гейміфікаційні або сюжетно-рольові механіки. Крім того, ці системи є комерційними, що обмежує їх доступність.

У свою чергу, платформа Duolingo демонструє високий рівень залученості користувачів завдяки розвиненій гейміфікації, однак її адаптивність має обмежений характер і зосереджена переважно на підборі вправ. Функціональні можливості системи обмежені вузькою предметною областю, а інтелектуальна взаємодія з користувачем реалізована частково. Оцінювання знань базується переважно на роботі з окремими мовними одиницями, що не дозволяє формувати цілісну модель рівня підготовки користувача.

Таким чином, сучасні освітні платформи реалізують або адаптивність навчального процесу, або механізми мотивації через гейміфікацію, однак не поєднують ці підходи з повноцінною інтелектуальною взаємодією у межах єдиного середовища. Це обумовлює доцільність розробки освітнього компоненту, в якому адаптивний ШІ-асистент, динамічна зміна складності завдань та гейміфіковане середовище функціонують як взаємопов’язані компоненти.

1.3. Обґрунтування мети та постановка задачі

На основі проведеного аналізу теоретичних підходів, методів адаптивного навчання та існуючих програмних рішень встановлено, що ключовою проблемою сучасних освітніх систем є відсутність комплексної інтеграції адаптивності, гейміфікації та інтелектуальної підтримки користувача.

Більшість існуючих рішень забезпечують або персоналізацію навчального процесу, або підвищення мотивації за рахунок ігрових механік, проте не реалізують їх узгоджену взаємодію. Недостатня глибина адаптації складності

навчальних завдань, обмеженість гейміфікаційних підходів та відсутність ШІ-асистента як активного учасника навчального процесу знижують ефективність таких систем.

У зв’язку з цим метою кваліфікаційної роботи є розробка освітнього компоненту з адаптивним ШІ-асистентом та гейміфікацією навчального процесу з урахуванням індивідуального рівня підготовки користувача в межах інтерактивного RPG-середовища.

Для досягнення поставленої мети необхідно вирішити такі основні завдання:

– проаналізувати існуючі підходи до побудови освітніх компонент та адаптивного навчання;

– дослідити методи персоналізації навчального процесу та оцінки рівня знань користувача;

– розробити модель адаптивного визначення складності завдань з урахуванням індивідуальних характеристик користувача;

– спроєктувати архітектуру компонента з інтегрованим ШІ-асистентом;

– реалізувати механізми гейміфікації навчального процесу у вигляді RPG-середовища;

– забезпечити взаємодію користувача з освітнім компонентом через інтелектуального асистента;

– оцінити ефективність запропонованого підходу.

Особливістю обраної стратегії є поєднання адаптивного навчання з ігровими механіками. У запропонованому рішенні навчальний процес інтегрується у сюжетну структуру RPG-середовища, де прогрес користувача безпосередньо залежить від успішності виконання навчальних завдань.

Крім того, важливим елементом дослідження є використання адаптивних алгоритмів оцінки знань, які враховують не лише правильність відповідей, але й динаміку навчання, повторюваність помилок, швидкість виконання завдань та інші поведінкові характеристики користувача. Такий підхід дозволяє більш

точно визначати рівень підготовки користувача та оптимізувати складність навчального матеріалу [20, с. 2-5].

Отже, поєднання адаптивного навчання, гейміфікації та ШІ-взаємодії створює передумови для підвищення ефективності навчального процесу та рівня залученості користувачів, що визначає перспективність даного рішення.

РОЗДІЛ 2. МОДЕЛЮВАННЯ ТА МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ ОСВІТНЬОГО КОМПОНЕНТУ

2.1 Моделювання освітнього компоненту

На основі проведеного аналізу предметної області та існуючих аналогів сформулюємо задачу розробки освітнього компоненту з адаптивною складністю, ШІ-асистентом та гейміфікованим RPG-середовищем.

Розроблюване рішення представляє собою інтерактивне програмне середовище, у якому навчальний процес інтегровано у ігровий контекст. Основна ідея полягає у тому, що користувач (студент або учень) взаємодіє з навчальним матеріалом через виконання завдань, які є частиною ігрового процесу. Прогрес у грі безпосередньо залежить від успішності виконання навчальних завдань, що забезпечує тісний зв’язок між навчанням і мотивацією.

Компонент функціонуватиме як адаптивний, тобто здатний змінювати рівень складності завдань, тип навчального контенту та характер взаємодії залежно від індивідуальних характеристик користувача. Для цього використовуються алгоритми аналізу поведінкових даних та простежування знань, які дозволяють оцінювати рівень знань, визначати прогалини та формувати персоналізовану траєкторію навчання.

Ключовою складовою є ШІ-асистент, який забезпечує діалогову взаємодію з користувачем. Асистент виконує функції наставника: пояснює матеріал, відповідає на запитання, надає підказки та адаптує стиль комунікації відповідно до рівня підготовки користувача. Крім того, ШІ-асистент може враховувати поведінкові аспекти взаємодії, що дозволяє підвищити рівень персоналізації та залученості.

У межах компонента виділяються такі основні ролі користувачів:

– Студент (користувач) – основний учасник, який проходить навчання у форматі RPG-середовища, виконує завдання, отримує винагороди та взаємодіє з ШІ-асистентом;
– ШІ-асистент – виконує роль наставника, надає пояснення та підказки, адаптує навчання;

- Адміністратор – здійснює наповнення бази знань та коригує навчальні сценарії.

Основні сценарії взаємодії користувача включають:

- Реєстрація та ініціалізація профілю – користувач вводить базові дані, формується початковий рівень знань.
- Проходження навчальних завдань – користувач виконує завдання, інтегровані у ігровий процес (відкриття об’єктів, виконання квестів, створення предметів, взаємодія з ігровим середовищем тощо).
- Адаптація складності – змінює рівень складності завдань залежно від результатів користувача за допомогою алгоритмів.
- Взаємодія з ШІ-асистентом – користувач отримує пояснення, підказки та відповіді на запитання.
- Отримання винагород – за виконання завдань користувач отримує досвід (XP), рівні, прогрес по сюжету та новий контент.
- Аналіз прогресу – відображає статистику навчання користувача, ігровий процес (рівень, стан персонажа), результати зберігаються у профілі користувача для забезпечення безперервності навчання.

Вхідні дані включають:

- профіль користувача (рівень знань, історія відповідей, поведінкові характеристики);
- база навчальних завдань (категорії, складність, тематика);
- параметри гейміфікації (рівні, сюжет, нагороди, прогрес);
- запити до ШІ-асистента.

Вихідні дані:

- адаптований рівень складності завдань;
- оцінка знань користувача;
- ігрові показники (XP, рівень, стан);
- відповіді ШІ-асистента.

Для відображення структури середовища, сценаріїв взаємодії та потоків вхідних/вихідних даних побудовано UML-діаграму [21, с. 4-15], на рис. 2.1.

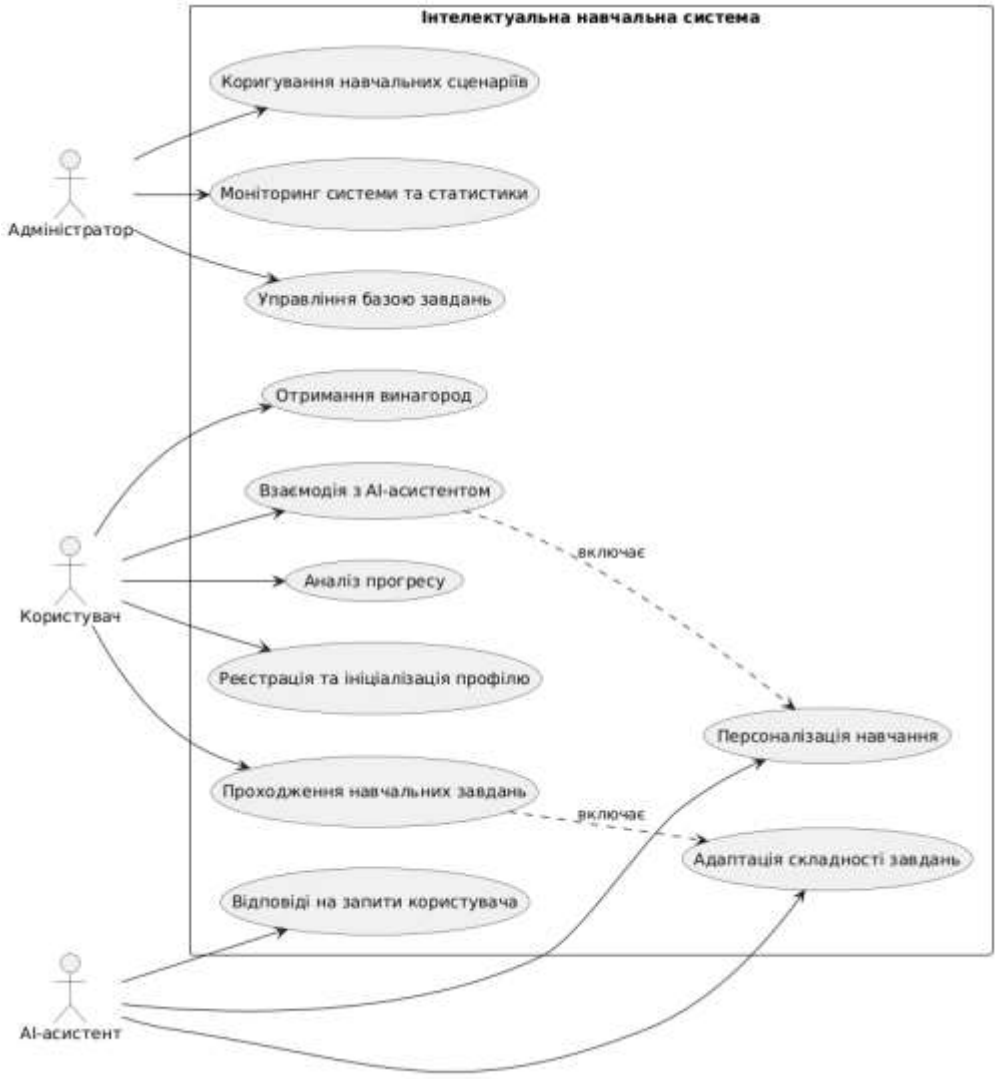


Рис. 2.1 UML-діаграма компоненту

Для забезпечення високої якості навчального процесу та комфортної взаємодії користувачів визначимо ключові показники:

- точність адаптації рівня складності (не менше 80% відповідності рівню користувача);
- час реакції середовища (швидкість відповіді ШІ-асистента та адаптації контенту)
- рівень залученості користувача (визначається через частоту взаємодій та тривалість сесій);
- ефективність навчання (зростання показників успішності користувача);
- надійність та цілісність даних.

Паралельно з показниками ефективності необхідно враховувати обмеження до середовища, які визначать межі його функціонування та накладуть ліміти на алгоритми та обчислювальні ресурси. Основні обмеження:

- обчислювальні ресурси для обробки даних та роботи ШІ-алгоритмів;
- обмеження часу відповіді для забезпечення інтерактивності;
- залежність якості адаптації від повноти та якості навчального контенту;
- обмеження складності алгоритмів для забезпечення масштабованості середовища;
- обмеження формату контенту лише текстовими або тестовими запитаннями.

Такий підхід дозволяє чітко визначити критерії якості освітнього компонента та межі його застосування, що є необхідним для подальшої розробки математичної моделі адаптивного навчання та алгоритмів ШІ-асистента.

2.2 Математична модель адаптивного навчання

Однією з основних цілей розроблюваного освітнього компонента є створення механізму, який дозволяє адаптувати навчальний процес до індивідуальних особливостей користувача. Адаптивність компоненту забезпечується через кількісне оцінювання рівня знань, прогнозування успішності навчання та динамічне регулювання складності навчальних завдань. Для досягнення цих цілей необхідно розробити математичну модель, яка формалізує процес навчання та дозволяє описати його закономірності в термінах ймовірності, когнітивних процесів та поведінкових характеристик учня.

У запропонованому середовищі рівень знань користувача з певного предмету формується як ймовірність того, що конкретна навчальна навичка була засвоєна. Позначимо цю ймовірність через K_t, де t – момент часу, що відповідає конкретній навчальній взаємодії:

$$K_t = P(L_t) \tag{1}$$

Тут L_t відображає стан «знання засвоєно». Таким чином, значення K_t находиться в межах від 0 до 1, де 0 відповідає повній відсутності знань, а 1 – повному засвоєнню матеріалу. Таке представлення дозволяє описувати процес навчання як динамічний ймовірнісний процес, що змінюється в часі.

Для моделювання оновлення знань після кожної взаємодії користувача середовище використовує байєсівський підхід до простежування знань. Ця модель дозволяє враховувати ймовірність помилок користувача під час виконання завдання, а також можливість випадкового вгадування правильної відповіді [22].

Нехай S – ймовірність помилки, G – ймовірність випадкового вгадування, а T – ймовірність того, що користувач вивчив новий матеріал у процесі виконання завдання. Тоді ймовірність того, що навичка засвоєна після правильної відповіді, визначається наступним чином [22]:

$$P(L_t|\text{вірно}) = \frac{P(L_{(t-1)})(1-S)}{P(L_{(t-1)})(1-S)+\left(1-P(L_{(t-1)})\right)G} \quad (2)$$

Аналогічно, у разі неправильної відповіді, ймовірність оновлюється за формулою [22]:

$$P(L_t|\text{невірно}) = \frac{P(L_{(t-1)})S}{P(L_{(t-1)})S+\left(1-P(L_{(t-1)})\right)(1-G)} \quad (3)$$

Після цього враховується ймовірність навчання T, що дозволяє моделі відображати поступове покращення знань користувача навіть після правильної відповіді, яка могла відбутися випадково:

$$K_{(t+1)} = P(L_t) + \left(1 - P(L_t)\right)T \quad (4)$$

Таким чином, ВКТ дозволяє побудувати гнучкий і точний механізм відстеження прогресу учня, який адаптується до його реальних знань та випадкових факторів.

Одним із критичних факторів у процесі навчання є також забування матеріалу. Згідно з експериментами Еббінгауза, рівень засвоєних знань з часом експоненційно зменшується. Моделюється за допомогою функції:

$$R(t) = e^{-\frac{t}{\lambda}} \quad (5)$$

де *R(t)* – коефіцієнт збереження знань, *t* – час, що минув з моменту останнього повторення матеріалу, а λ – параметр стабільності пам’яті, який відображає індивідуальні особливості учня [23, с. 57]. Графічну інтерпретацію процесу забування згідно з наведеною моделлю представлено на рис. 2.2.

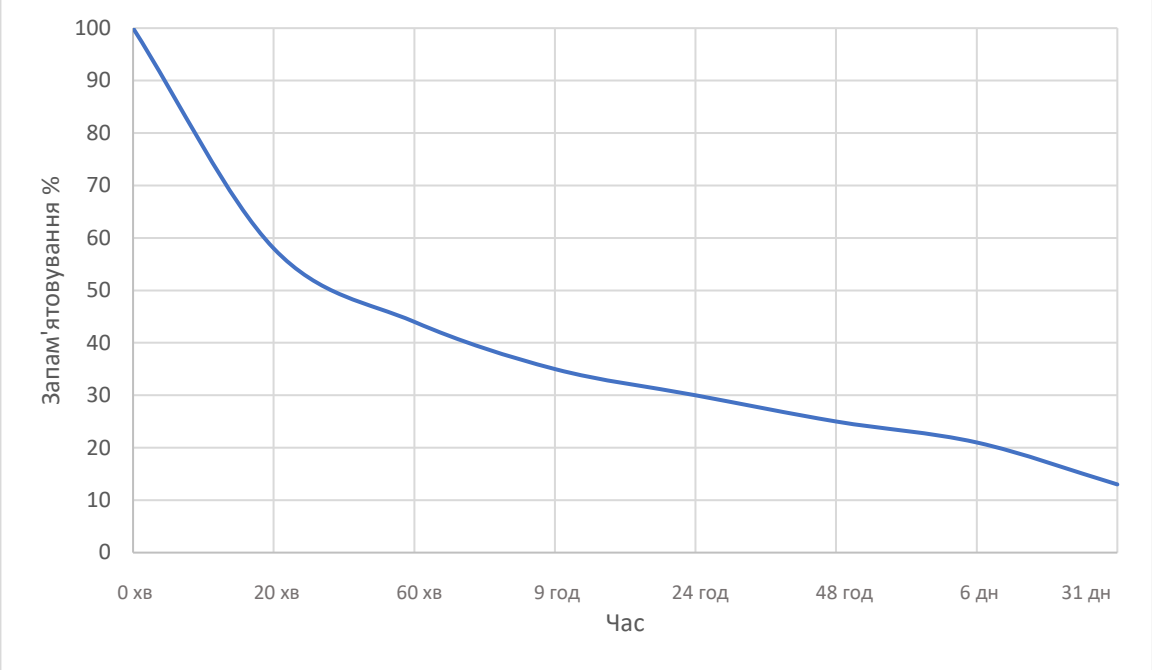


Рис. 2.2 Крива забування Еббінгауза

Застосування цього коефіцієнта дозволяє отримати ефективний рівень знань, який враховує як факт засвоєння, так і його актуальність у поточний момент:

$$K_{(\text{ефект})} = K_t * R(t) \tag{6}$$

Отже, компонента не лише оцінює знання, а й враховує їх втрату з часом, що дозволяє формувати більш точні рекомендації щодо повторення матеріалу.

Поведінкові характеристики користувача також є важливим елементом адаптивної моделі. Зокрема, точність відповідей визначається як відношення кількості правильних відповідей до загальної кількості виконаних завдань:

$$A = \frac{N_{(\text{правильне})}}{N_{(\text{загальне})}} \tag{7}$$

Крім того, серії правильних чи неправильних відповідей дозволяють оцінити стабільність поведінки користувача:

$$S = S_{(\text{вірно})} - S_{(\text{невірно})} \tag{8}$$

Ці метрики дозволяють моделювати психологічні аспекти навчання, зокрема мотивацію, увагу та концентрацію.

Для підвищення залученості користувача у навчальний процес впроваджується модель емоційного стану ШІ-асистента [25]. Емоційний стан описується вектором:

$$E_t = (trust_t, mood_t, mystery_t) \quad (9)$$

де $trust_t \in [0,1]$ – довіра ШІ-асистента до користувача на момент часу t, $mood_t \in [0,1]$ – поточний емоційний настрій ШІ-асистента, що впливає на тон комунікації, $mystery_t \in [0,1]$ – рівень поточної сили асистента, який показує скільки ще відповідей він може дати.

Оновлення стану вектора емоцій відбувається за дискретним правилом:

$$E_{(t+1)} = E_t + \Delta E \quad (10)$$

де $\Delta E = (\Delta trust, \Delta mood, \Delta mystery)$ визначається на основі взаємодії користувача. Наприклад, якщо користувач успішно виконує завдання, збільшується довіра $(\Delta trust > 0)$ та настрій ШІ-асистента $(\Delta mood > 0)$, а рівень загадковості змінюється відповідно до того, як часто гравець користується асистентом.

Формально, адаптивне навчання можна розбити на три взаємопов'язані підзадачі:

1. Задача оцінювання – визначення рівня знань користувача K_t, що описує ймовірність засвоєння конкретної навчальної навички.
2. Задача прогнозування – оцінка ймовірності правильної відповіді користувача на конкретне завдання з урахуванням поточного стану знань та попередніх результатів.
3. Задача оптимізації – підбір оптимальної складності завдань D_t, яка максимізує ефективність навчання.

У загальному вигляді, задачі можна представити як оптимізаційну задачу:

$$maxD_t F(K_t, D_t) \quad (11)$$

де $F(K_t, D_t)$ – функція ефективності навчання, яка інтегрує ймовірність засвоєння знань K_t та обрану складність завдання D_t, $K_t \in [0,1]$ – поточний рівень знань користувача з конкретного предмету або навички, $D_t \in [0,1]$ – обрана складність

завдання, нормована до одиничного інтервалу, $maxD_t$ означає, що середовище прагне обрати таку складність, яка максимізує навчальний ефект при збереженні мотивації та залученості учня [26].

Адаптивна складність завдань визначається як комбінована функція базової складності, точності користувача, серії відповідей та ефективного рівня знань:

$$d = d_{(\text{базове})} + \alpha(A - 0{,}5) + \beta S + \gamma\left(K_{(\text{ефект})} - 0{,}5\right) \quad (12)$$

Ця формула дозволяє враховувати різні аспекти навчальної діяльності. Вагові коефіцієнти α, β, γ визначають значущість відповідних параметрів, а їхнє налаштування дозволяє адаптувати середовище під конкретну групу учнів. У запропонованому рішенні перевага надається ефективному рівню знань користувача, оскільки саме він є найточнішим індикатором готовності до більш складних завдань [27].

Для підтримки оптимального навчального навантаження середовище також застосовує концепцію flow-зони (стану потоку), що дозволяє уникати як надмірно легких, так і надмірно складних завдань. Обмеження flow-зони формалізується через інтервал допустимих значень складності:

$$d \in \left[d_{(\text{базове})} - \Delta, d_{(\text{базове})} + \Delta\right] \quad (13)$$

Функція обрізки забезпечує, що остаточне значення складності завдання завжди залишатиметься в межах flow-зони:

$$d = clip\left(d, d_{(\text{базове})} - \Delta, d_{(\text{базове})} + \Delta\right) \quad (14)$$

Цей підхід дозволяє підтримувати баланс між мотивацією користувача та ефективністю навчання [27].

Таким чином, математична модель адаптивного навчання інтегрує ймовірнісні методи (ВКТ), когнітивні моделі (ефект забування), поведінкову аналітику та концепцію оптимального навантаження. Вона дозволяє моделювати індивідуальний процес навчання кожного користувача та забезпечує персоналізацію складності завдань, що у свою чергу сприяє більш ефективному засвоєнню матеріалу та підвищенню залученості користувача в ігровому навчальному середовищі.

2.3 Розробка концепції гейміфікації та ігрових механік

У нашого освітнього компоненту гейміфікація відіграє ключову роль у підвищенні мотивації користувачів, забезпеченні їх залученості та формуванні стійкого інтересу до процесу навчання. На відміну від традиційних підходів, де навчання подається у вигляді послідовності завдань, у розроблюваному рішенні передбачається інтеграція навчального процесу в ігрове середовище, що базується на принципах рольових ігор [28].

Основна ідея полягає у тому, що користувач взаємодіятиме з навчальним контентом не як з абстрактними задачами, а як з елементами ігрового світу, де виконання завдань матиме безпосередній вплив на розвиток персонажа, просування сюжетом та відкриття нових можливостей. Такий підхід дозволить трансформувати навчання у цілеспрямований і захопливий процес, у якому знання виступатимуть інструментом досягнення ігрових цілей [28].

Концептуально компонент розглядатиметься як однокористувацьке інтерактивне середовище, у якому поєднуватимуться механіки розвитку персонажа, дослідження ігрового світу, виконання квестів та взаємодії з інтелектуальним асистентом. Центральним елементом виступатиме персонаж користувача, який характеризуватиметься набором параметрів, що відображають як ігрові, так і навчальні аспекти прогресу. До таких параметрів належатимуть рівень персонажа, набір характеристик (сила, витривалість, інтелект тощо), а також показники знань у відповідних навчальних дисциплінах.

Складова розвитку персонажа буде реалізована через механізм накопичення досвіду та підвищення рівня. Виконання навчальних завдань, інтегрованих у квести, супроводжуватиметься отриманням винагород, що стимулюватиме користувача до активної участі у навчальному процесі. Таким чином, забезпечуватиметься пряма залежність між освітньою активністю та ігровим прогресом, що є одним із ключових принципів ефективної гейміфікації.

Важливою складовою стане механізм характеристик персонажа, який забезпечуватиме додатковий рівень персоналізації. Характеристики можуть впливати на доступність певних дій, ефективність взаємодії з ігровими об’єктами

або результати виконання окремих завдань. Це дозволить формувати індивідуальні стилі проходження та підсилюватиме відчуття унікальності користувацького досвіду.

Для підвищення глибини взаємодії передбачається впровадження інвентарної механіки, яка забезпечуватиме можливість зберігання, використання, екіпірування та комбінування різноманітних ігрових предметів. Предмети матимуть різну функціональність: від підсилення характеристик до надання допоміжних ефектів під час виконання навчальних завдань. Додатково планується реалізація механіки створення ігрових об’єктів, що дозволятиме створювати нові предмети шляхом комбінування наявних ресурсів, стимулюючи дослідження та розвиток стратегічного мислення, а також поєднання цього у навчальну частину.

Суттєвим елементом гейміфікації стане квестова механіка, яка організовуватиме навчальний процес у вигляді послідовності завдань із чітко визначеними цілями та винагородами. Квести можуть мати різну структуру – від лінійних до багаторівневих, що передбачатимуть виконання декількох взаємопов’язаних дій. Навчальні завдання інтегруватимуться у квести як їх невід’ємна частина, забезпечуючи природне включення освітнього контенту у ігровий процес.

Ігровий світ передбачатиме можливість взаємодії з неігровими персонажами та об’єктами через діалогову механіку. Особливу роль відіграватиме ШІ-асистент, який буде реалізований у вигляді ігрового персонажа-компаньйона (привида), що супроводжуватиме користувача протягом усього процесу. Асистент забезпечуватиме навчальну підтримку, надаватиме пояснення та підказки, а також адаптуватиме свою поведінку залежно від дій користувача та контексту взаємодії.

Для підвищення динамічності ігрового процесу передбачається реалізація бойової механіки та наявність противників. Успішність у таких сценаріях частково залежатиме від рівня знань користувача, що забезпечуватиме інтеграцію когнітивних та ігрових навичок. Додатково планується впровадження

базових механік керування персонажем, включаючи рух, уникнення атак (перекати) та інші дії, що сприятимуть підвищенню залученості.

Механіка торгівлі дозволятиме здійснювати обмін ресурсами, придбання предметів та управління інвентарем, вводячи елементи економічної взаємодії. Це сприятиме розвитку навичок планування та прийняття рішень. Кастомізація персонажа та налаштування інтерфейсу забезпечуватимуть адаптацію середовища під індивідуальні вподобання користувача.

Окрему увагу передбачається приділити реалізації механізму збереження прогресу. Такий підхід дозволить фіксувати стан гри, характеристики персонажа, результати навчання та інші параметри, забезпечуючи можливість відновлення сесії без втрати досягнень. Це є важливим з точки зору забезпечення безперервності навчального процесу.

Детальна схема взаємодії обраних ігрових механік та модулів наведена на рисунку 2.3



Рис. 2.3 Схема взаємодії ігрових механік та модулів

Додаткові допоміжні механіки, такі як відстеження квестів, компас, підказки та інші елементи інтерфейсу, будуть спрямовані на покращення навігації у ігровому середовищі та зниження когнітивного навантаження на користувача. Це дозволить зосередити увагу на навчальному контенті без втрати орієнтації у грі.

Таким чином, запропонована концепція гейміфікації базується на комплексному поєднанні ігрових механік, що взаємодіятимуть між собою та підсилюватимуть одна одну. Інтеграція навчальних завдань у структуру гри забезпечуватиме природну мотивацію до навчання, тоді як адаптивні елементи дозволятимуть враховувати індивідуальні особливості користувача. Це створюватиме передумови для ефективного засвоєння знань у середовищі, що поєднує освітні та ігрові підходи.

2.4 Обґрунтування архітектурних рішень та вибір засобів програмної реалізації

Розробка освітнього компоненту вимагає обґрунтованого підходу до вибору технологічного стеку та методів реалізації. Обрані рішення повинні забезпечувати не лише функціональність середовища, але й його масштабованість, інтерактивність, адаптивність до користувача та ефективність обробки даних у реальному часі.

У якості основного середовища розробки клієнтської частини було обрано платформу-рушій Unity із використанням мови програмування C# [29]. Такий вибір обумовлений необхідністю реалізації інтерактивного гейміфікованого середовища, у якому навчальний процес інтегрується у ігрову механіку. Unity надає розвинені засоби для створення як двовимірних, так і тривимірних сцен, що дозволяє моделювати повноцінне ігрове середовище з персонажами, квестами, елементом взаємодії та візуалізацією прогресу користувача. Крім того, важливою перевагою є компонентна архітектура рушія, яка дозволяє модульно організувати логіку середовища, зокрема відокремити складові навчання, інвентарю, взаємодії з NPC та ШІ-асистента. Додатковим аргументом на користь Unity є доступ до розгалуженої екосистеми готових рішень у Unity Asset Store [29]. Це дозволяє значно прискорити розробку завдяки використанню високоякісних графічних ресурсів, готових моделей персонажів та інструментарію для побудови інтерфейсів, що дає змогу зосередити основні зусилля на проектуванні унікальної освітньої логіки та геймплейних механік.

Використання мови програмування C# є логічним продовженням вибору Unity, оскільки вона забезпечує тісну інтеграцію з рушієм та надає широкі можливості для реалізації об’єктно-орієнтованої логіки. Окрім цього, C# дозволяє ефективно працювати з асинхронними процесами, що є критично важливим для взаємодії з ШІ-сервером, зокрема під час відправлення запитів та отримання відповідей без блокування основного ігрового процесу [30].

Для реалізації інтелектуальної складової компоненту було обрано локальну мовну модель, розгорнуту за допомогою середовища Ollama, з використанням моделі qwen2.5:3b [31]. Такий підхід має ряд суттєвих переваг у порівнянні з використанням хмарних API. Насамперед, це забезпечує незалежність від зовнішніх сервісів та стабільність роботи навіть за відсутності інтернет-з’єднання. Крім того, локальне виконання дозволяє контролювати обробку даних користувача, що є важливим з точки зору конфіденційності. Суттєвим фактором є також те, що на відміну від платних хмарних API, локальна модель є повністю безкоштовною у використанні. Це дозволяє зняти будь-які обмеження на кількість запитів до ШІ-асистента без додаткових витрат на підписки чи оплату за кожен токен [32].

Обрана модель має порівняно невелику кількість параметрів, що робить її придатною для локального використання без значних обчислювальних ресурсів, але водночас забезпечує достатній рівень якості генерації тексту. Це дозволяє реалізувати як навчальну взаємодію, що передбачає пояснення матеріалу та надання підказок, так і рольову складову, яка підсилює гейміфікацію середовища.

Серверна частина, що відповідає за взаємодію з мовною моделлю, реалізована з використанням мови програмування Python. Вибір цієї мови обумовлений її широким застосуванням у сфері штучного інтелекту та обробки даних, а також наявністю великої кількості бібліотек для реалізації мережевих сервісів [33]. У межах середовища Python використовується для побудови REST API, яке забезпечує обмін даними між клієнтською частиною та ШІ-модулем [34].

Для збереження стану та прогресу користувача було обрано формат JSON. Це рішення обумовлене необхідністю зберігати структуровані дані, що

включають як прості параметри (наприклад, кількість правильних відповідей), так і складні ієрархічні структури, зокрема статистику по окремих навчальних дисциплінах. Формат JSON забезпечує простоту серіалізації та десеріалізації об’єктів, що дозволяє швидко зберігати та відновлювати стан середовища. Важливою перевагою є також його універсальність, оскільки JSON однаково добре підтримується як у рушії Unity, так і у серверній частині, реалізованій на Python. Це спрощує обмін даними між елементами компоненту та забезпечує узгодженість структури даних [35].

Архітектурно взаємодія організована за принципом клієнт-сервер, де Unity виступає клієнтом, який формує запити на основі дій користувача, а Python-сервер виконує їх обробку та передає сформований запит до мовної моделі. Отримана відповідь проходить додаткову обробку, зокрема виділення структурованих даних, після чого повертається до клієнта. Такий підхід дозволяє розділити логіку гри та логіку штучного інтелекту, що значно спрощує підтримку та масштабування. Загальна структура елементів середовища та їх взаємозв’язки представлені на рис. 2.4.

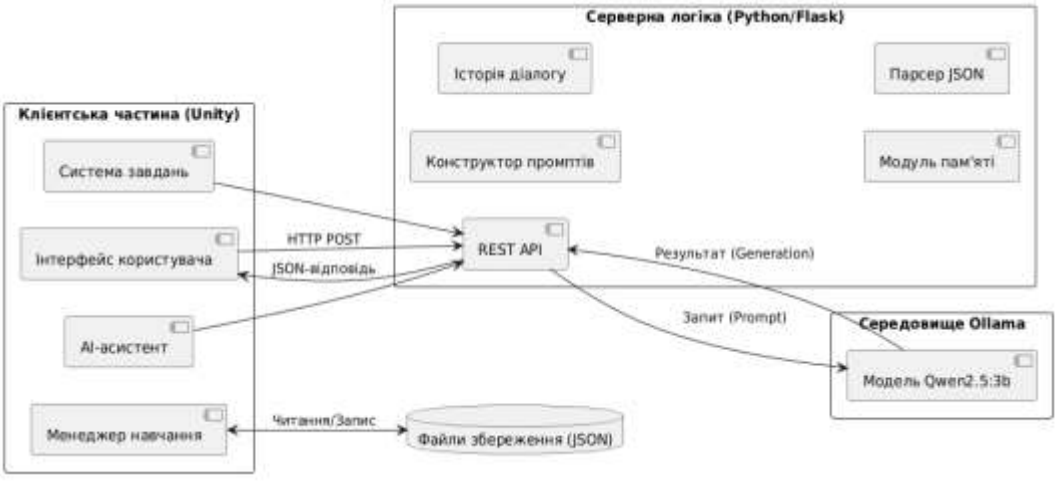


Рис. 2.4 Структурна схема взаємодії елементів програмного комплексу

Описана взаємодія між клієнтською частиною на Unity, серверною логікою на Python та моделлю Ollama qwen2.5:3b формує цілісний цикл обробки даних. Схематично послідовність дій, від моменту введення запиту користувачем до отримання відповіді від локальної мовної моделі, представлена на рис. 2.5.



Рис 2.5 Потік обробки запиту від користувача

Таким чином, проведено комплексне моделювання освітнього компоненту. Визначено архітектурні вимоги, розроблено математичну модель адаптивного підбору складності та обґрунтовано вибір технологічного стека (Unity, Python, Ollama). Запропонована структура взаємодії компонентів та гейміфіковані механіки створюють основу для програмної реалізації рішення, що забезпечить високий рівень залучення користувачів та персоналізацію навчального процесу.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ОСВІТНЬОГО КОМПОНЕНТУ

3.1 Реалізація адаптивної навчальної складової

У рамках реалізації адаптивна навчальна складова була інтегрована безпосередньо у клієнтську частину з використанням мови програмування C#, що дозволяє виконувати обчислення в режимі реального часу без додаткового навантаження на сервер. Вона виконує ключову роль у забезпеченні взаємодії користувача з середовищем, обробці навчальних завдань та інтеграції з ШІ-асистентом.

Центральним компонентом є клас PlayerLearningManager, який відповідає за управління навчальною статистикою користувача та оновлення рівня знань. У ньому зберігаються як загальні показники (кількість правильних і неправильних відповідей, серії), так і деталізована статистика за окремими предметами. Клас забезпечує фіксацію результатів, облік підказок і пропусків, а також формування узагальненого стану навчання. Частина програмного коду наведена у Додатку А.

Обробка результатів виконується методом RegisterResult, який оновлює локальні та глобальні показники користувача і рівень знань. Окремо для моделювання використовується алгоритм на основі BKT, реалізований у методі UpdateKnowledge. Складова використовує накопичувальний підхід до оцінювання, а застосування Mathf.Clamp01 забезпечує обмеження значень у допустимих межах, що гарантує коректну роботу алгоритму підбору завдань.

```
private float UpdateKnowledge(KnowledgeState ks, bool correct)
{
   float p = ks.pKnow; float S = ks.slip; float G = ks.guess;
   float T = ks.learn;
   float pGivenObs;
   if (correct)
   {
      pGivenObs = (p * (1 - S)) / (p * (1 - S) + (1 - p) * G);
   }
   else
   {
      pGivenObs = (p * S) / (p * S + (1 - p) * (1 - G));
   }
   float pNext = pGivenObs + (1 - pGivenObs) * T;
   return Mathf.Clamp01(pNext);
}
```

Клас PlayerLearningState виконує роль контейнера навчальних параметрів, зберігаючи як загальні характеристики користувача, так і статистику за окремими предметами. Його часткову програмну реалізацію також наведено у Додатку А.

На основі накопичених даних складова формує адаптивну модель навчання. Ключовим є метод GetAdaptiveDifficulty, який визначає складність наступного завдання з урахуванням BKT, точності відповідей та серії успіхів користувача. Використання вагових коефіцієнтів забезпечує плавну зміну складності без різких стрибків, підтримуючи стабільний навчальний процес. Додатково враховується час із моменту останнього повторення, що моделює ефект забування та наближає складову до реальних когнітивних процесів.

```
public int GetAdaptiveDifficulty(SubjectType subject, int baseDifficulty)
{
   if (!State.subjectStats.TryGetValue(subject, out var stats)) return baseDifficulty;
   float retention = GetRetention(stats);
   float effectiveKnowledge = stats.knowledge.pKnow * retention;
   int total = stats.correct + stats.wrong;
   float accuracy = total > 0 ? (float)stats.correct / total : 0.5f;
   int streak = stats.streakSubjectCorrect - stats.streakSubjectWrong;
   float alpha = 2.0f;  // вплив accuracy
   float beta = 0.5f;   // вплив streak
   float gamma = 3.0f;  // вплив BKT (найважливіше)
   float difficultyFloat = baseDifficulty
      + alpha * (accuracy - 0.5f)
      + beta * streak
      + gamma * (effectiveKnowledge - 0.5f);
   int difficulty = Mathf.RoundToInt(difficultyFloat);
   int minDifficulty = Mathf.Max(1, baseDifficulty - 2);
   int maxDifficulty = baseDifficulty + 2;
   return Mathf.Clamp(difficulty, minDifficulty, maxDifficulty);
}
```

Всі навчальні завдання формалізовано у вигляді окремих об’єктів класу TaskRequirement, реалізованого на основі ScriptableObject [36]. Такий підхід дозволяє зберігати навчальний контент незалежно від програмної логіки, забезпечуючи зручність редагування та масштабування. Кожне завдання містить ключові параметри, зокрема предмет, тип, текст умови, варіанти та правильні відповіді, рівень складності, підказки та обмеження часу виконання. Детальна структура класу наведена у Додатку А.

Централізоване управління завданнями здійснюється через компонент TaskDatabase, який відповідає за збереження та вибірку завдань. Вибір конкретного завдання відбувається з урахуванням адаптивного рівня складності, що визначається на основі поточного стану знань користувача. Додатково враховується історія попередніх завдань, що дозволяє уникати їх частого повторення та підвищує ефективність навчального процесу [36]. Частковий код у винесено у Додаток А.

Важливим елементом адаптивної навчальної складової є складова відображення навчальних завдань, яка реалізована у вигляді окремого контролера інтерфейсу TaskUIController. Цей компонент відповідає за відображення запитань, обробку введених відповідей, а також взаємодію з користувачем у процесі виконання завдання.

Даний клас забезпечує динамічне оновлення інтерфейсу залежно від типу завдання, предмету та рівня складності. Особливістю реалізації є використання різних компонентів рендерингу для різних типів завдань, що дозволяє адаптувати відображення до специфіки навчального матеріалу.

Обробка відповіді користувача здійснюється методом OnSubmit, який виконує збір даних із форм та їх валідацію. Перевірка правильності делегується класу TaskValidator (код наведено у Додатку А), що забезпечує розмежування ігрової логіки та оцінювання знань. Додатково використання звукових ефектів і затримок підвищує інтерактивність та якість зворотного зв’язку з користувачем.

У клієнтській частині також реалізовано допоміжні механізми, зокрема підказки та пропуск завдань, які виступають елементами збору даних для адаптивної моделі. Використання підказки фіксується як ознака недостатнього засвоєння матеріалу та обробляється методом RegisterHint, що впливає на коригування рівня знань. Пропуск завдання прирівнюється до неправильної відповіді, що дозволяє зберігати цілісність даних та коректно змінювати складність подальших завдань.

Інтеграція з ШІ-асистентом реалізована у вигляді окремого компонента – класу AssistantAI (код наведено у Додатку А), який забезпечує діалогову

взаємодію з користувачем та формування запитів до серверної частини. Передбачено два режими роботи асистента: ігровий та навчальний.

У межах даного розділу основну увагу приділено навчальному режиму, який реалізує логіку контекстної адаптації. Він оперує динамічними даними про дисципліну, криву забування та стан моделі ВКТ, передаючи ШІ не лише тип завдання, а й параметри knowledge та level. Це дозволяє визначати характер помилки користувача та адаптувати відповідь асистента.

Крім того, реалізовано користувацький інтерфейс, що забезпечує відображення ключових елементів взаємодії. Як показано на рис. 3.1, інтерфейс включає панель завдань, статистику навчання та діалогове вікно ШІ.



Рис. 3.1 Інтерфейс навчальної складової

Значну увагу приділено реалізації зворотного зв’язку з користувачем. Складова забезпечує миттєве інформування про правильність відповіді, відображення прогресу та оновлення показників у реальному часі з використанням анімацій, звукових ефектів і візуальних індикаторів. Навчальні завдання інтегровані в ігровий процес (квести, відкриття об’єктів, взаємодія із середовищем), що підвищує залученість користувача.

Для узагальнення архітектури та відображення зв’язків між елементами розроблено діаграму класів, наведену на рисунку 3.2.

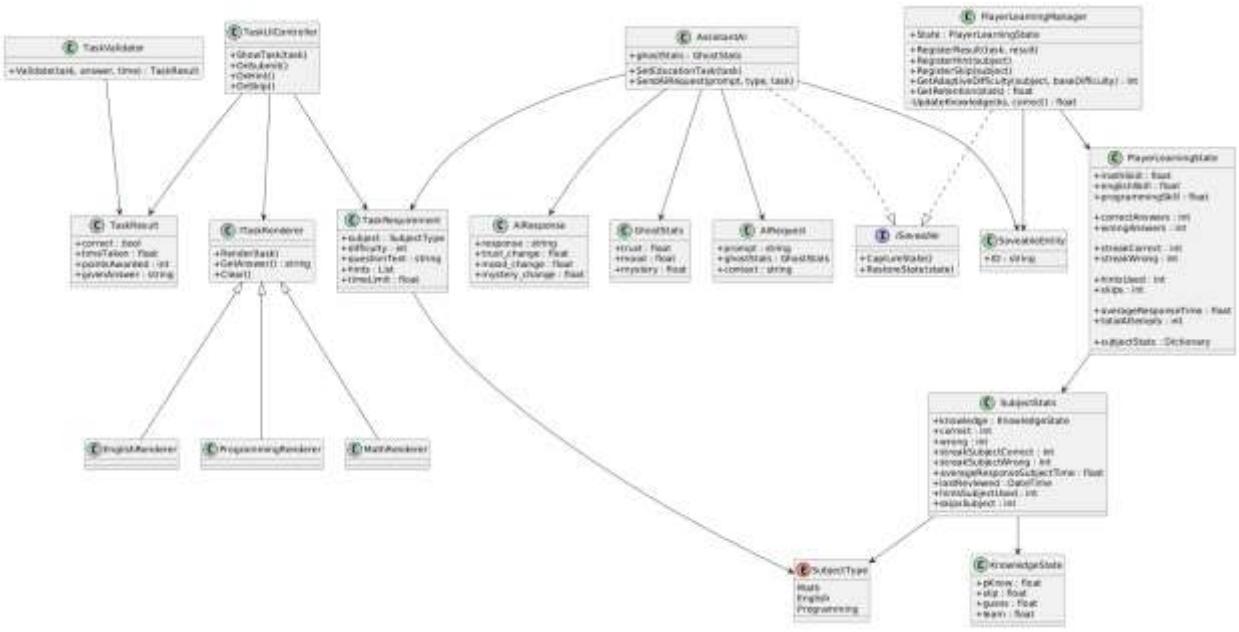


Рис. 3.2 – UML-діаграма класів клієнтської частини

Наведені на схемі зв’язки демонструють модульність складової: логіка управління навчанням відокремлена від візуальної частини, а сервіс взаємодії з ШІ реалізований як незалежний компонент, що працює через інтерфейси. Детальна програмна реалізація ключових елементів наведена у Додатку А.

Таким чином, навчальна складова забезпечує повний цикл взаємодії користувача з навчальним середовищем – від подання завдань до обробки результатів і інтеграції з ШІ. Використання модульного підходу та адаптивної моделі забезпечує її гнучкість і масштабованість.

3.2 Реалізація ШІ-асистента та серверної частини

Штучний інтелект у розробленому середовищі виконує роль інтерактивного асистента, який забезпечує підтримку користувача під час навчання, надає пояснення, формує підказки та інтегрується у ігровий процес як окремий персонаж. Реалізація даного компонента базується на клієнт-серверній архітектурі, що дозволяє розділити логіку інтерфейсу, обробку запитів та генерацію відповідей.

У межах реалізації було використано локальну мовну модель qwen2.5:3b, що запускається через середовище Ollama, а серверна частина побудована на

мові програмування Python із використанням веб-фреймворку Flask. Клієнтська частина, реалізована у середовищі Unity, взаємодіє із сервером через HTTP-запити, передаючи текст запиту, контекст та параметри стану асистента [37].

На стороні клієнта ШІ-асистент реалізований у вигляді компонента AssistantAI (частковий програмний код компонента наведено у Додатку А), який відповідає за обробку введення користувача, формування запитів до серверу та відображення отриманих відповідей. Даний компонент інтегрований у користувацький інтерфейс гри та підтримує як загальну взаємодію (RPG-режим), так і навчальний режим.

Ключовою функцією клієнтської частини є формування структурованого запиту до ШІ-асистента. На відміну від звичайних чат-ботів, запит містить не лише текст користувача, а й контекст: ігрові показники, стан квестів та навчальну аналітику, що забезпечує персоналізацію відповідей. Метод BuildRPGPrompt виконує селективний збір даних, а у разі виявлення низької успішності NeedsLearningFromStats доповнює запит інструкціями для адаптації пояснень до рівня знань користувача.

```
private string BuildRPGPrompt(string prompt)
{
   StringBuilder sb = new StringBuilder();
   sb.AppendLine("[Question]");
   sb.AppendLine(prompt);
   bool needsStats = NeedsStats(lower);
   bool needsQuests = NeedsQuests(lower);
   bool needsLearning = NeedsLearning(lower);
   bool weakPlayer = NeedsLearningFromStats();
   if (needsStats)
   {
      sb.AppendLine("\n[Player Stats]");
      sb.AppendLine(GetStatsSummary());
   }
   if (needsQuests)
   {
      sb.AppendLine("\n[Quests]");
      sb.AppendLine(GetQuestSummary());
   }
   if (needsLearning || weakPlayer)
   {
      sb.AppendLine("\n[Player Learning]");
      sb.AppendLine(GetLearningSummary());
   }
   return sb.ToString();
}
```

Запит включає такі складові: текст питання користувача, параметри персонажа, інформацію про активні квести, а також дані про рівень знань користувача. Завдяки цьому модель отримує повну картину поточного стану гри та навчального процесу.

Візуалізацію процесу взаємодії користувача з інтелектуальним асистентом безпосередньо у ігровому середовищі представлено на рисунку 3.6.



Рисунок 3.3 – Інтерфейс взаємодії користувача з ШІ-асистентом

Обмін даними між клієнтом і сервером здійснюється у форматі JSON через HTTP POST-запити [38]. Відправка та асинхронне отримання відповіді реалізовані за допомогою класу UnityWebRequest [39]. Сервер повертає JSON-об’єкт із текстом відповіді та параметрами емоційного стану асистента (trust, mood, mystery). Отримані дані десеріалізуються в об’єкт AIResponse і використовуються для оновлення інтерфейсу та взаємодії з користувачем.

Серверна частина реалізована мовою Python із використанням мікрофреймворку Flask. Основною функцією сервера є приймання запитів від клієнта, формування контексту для мовної моделі та обробка результатів генерації. Сервер містить два основних API-методи: обробку запитів /ask та скидання стану сесії /reset. Метод /ask виконує повний цикл обробки, що включає синхронізацію стану асистента з Unity та взаємодію з локальною мовною моделлю через Ollama API [40].

```
chat_history = []
MAX_HISTORY = 4
def trim_history():
  global chat_history
  if len(chat_history) > MAX_HISTORY:
    chat_history = chat_history[-MAX_HISTORY:]
def update_memory():
  global memory_summary, chat_history
  summary_prompt = [
    { "role": "system",
      "content": (
        "Summarize the interaction briefly.\n"
        "Focus on:\n"
        "- player attitude\n"
        "- topics discussed\n"
        "- relationship evolution\n"
        "Keep it under 2 sentences.")}]
  summary_prompt.extend(chat_history)
  payload = {"model": MODEL,
    "messages": summary_prompt,
    "stream": False,
    "options": {
      "temperature": 0.2,
      "num_predict": 60 }}
  try:
    r = requests.post(OLLAMA_URL, json=payload)
    r.raise_for_status()
    result = r.json()
    summary = result.get("message", {}).get("content", "")
    if summary:
      memory_summary = summary
  except Exception as e:
    print("[MEMORY ERROR]", e)
```

Після отримання відповіді сервер виконує розбір JSON-блоку, що дозволяє автоматично оновлювати параметри персонажа на основі контексту розмови. Окремо реалізовано механізм «пам'яті» (update_memory), який виконує підсумовування діалогу при перевищенні ліміту повідомлень, що дозволяє зберігати контекст взаємодії при обмеженому вікні контексту моделі. Логіка сервера та функції стиснення історії наведені у Додатку А.

Для генерації відповідей використовується локальна мовна модель, яка запускається за допомогою інструментарію Ollama. Взаємодія з моделлю здійснюється через HTTP API, що дозволяє інтегрувати її у серверну логіку без необхідності використання зовнішніх хмарних сервісів, забезпечуючи конфіденційність даних та автономність роботи.

Особливістю реалізації є використання складного системного запиту, що визначає рольову модель асистента. У даному випадку він представлений як ігровий персонаж (дух) із власним стилем спілкування та правилами поведінки.

Окрім текстової відповіді, модель генерує числові дельти параметрів trust, mood і mystery, що дозволяє узгоджувати діалог із внутрішнім станом персонажа.

Для підтримки зв’язності діалогу реалізовано механізм короткострокової пам’яті. Сервер зберігає обмежену кількість останніх повідомлень (MAX_HISTORY), які додаються до кожного нового запиту.

Крім того, реалізовано механізм узагальнення історії взаємодії, який формує стислий опис попередніх діалогів. Це зменшує навантаження на модель та дозволяє зберігати контекст навіть після очищення історії повідомлень. Метод update_memory автоматично запускається при досягненні критичної кількості повідомлень. Отримане узагальнення зберігається у змінній memory_summary і додається до системного запиту, забезпечуючи довготривалу пам’ять асистента без перевантаження контексту.

У процесі реалізації особливу увагу було приділено обробці помилок, які можуть виникати під час взаємодії з сервером або мовною моделлю. У випадку збою мережі або перевантаження моделі середовище повертає інформативне повідомлення користувачу та не допускає аварійного завершення роботи ігрового клієнта.

Це забезпечує стабільну роботу навіть за умов нестабільного з’єднання. Додатково реалізовано булеву змінну isWaitingResponse, яка блокує відправку нових запитів до отримання відповіді, запобігаючи утворенню черги та перевантаженню сервера.

Реалізований ШІ-асистент є комплексною частиною, що поєднує клієнтську логіку, серверну обробку та інтеграцію з мовною моделлю. Використання клієнт-серверної архітектури забезпечує гнучкість і масштабованість, а контекстна та емоційна моделі підвищують якість взаємодії користувача.

3.3 Реалізація ігрових механік та збереження даних

Реалізація ігрових механік спрямована на забезпечення інтерактивності, залученості користувача та інтеграції навчального процесу в гейміфіковане

середовище. Ігрова складова реалізована в Unity з використанням C# і включає набір взаємопов’язаних компонентів, що відповідають за різні аспекти гри.

Основою гейміфікації є механіка персонажа, яка охоплює базові характеристики, рівні та механізми розвитку. Ці параметри впливають на взаємодію з середовищем, зокрема на складність завдань і якість підказок ШІ-асистента. Реалізація складової базується на класі PlayerStats, частковий код наведено у Додатку А.

Для зручності користувача всі поточні характеристики та прогрес розвитку відображаються у спеціалізованому інтерфейсі (рис. 3.4).



Рисунок 3.4 – Інтерфейс відображення характеристик та стану персонажа

Механіка інвентарю реалізує механізми збереження, використання та екіпірування предметів, забезпечуючи взаємодію гравця з економікою ігрового світу. Кожен предмет має унікальний набір властивостей (вага, категорія, вплив на характеристики персонажа). Управління ресурсами здійснюється через клас Inventory, який виконує роль центрального менеджера об'єктів. Частковий вихідний код представлено у додатку А.

Візуальна частина представлена багатовіконним інтерфейсом, який розділяє інвентар та комірки швидкого доступу для екіпірування (рис. 3.5).



Рисунок 3.5 – Інтерфейс управління інвентарем та екіпіруванням

Квестова механіка забезпечує постановку завдань та контроль їх виконання, виступаючи основним інструментом спрямування ігрового та навчального процесу. Кожен квест містить опис, набір кроків (цілей) та стан виконання. Реалізація базується на менеджері QuestManager, який відстежує події в ігровому світі (збір предметів, відвідування локацій, діалоги) та оновлює статус активних завдань. Частковий код представлено у додатку А.

Для моніторингу поточних завдань у грі реалізовано журнал квестів, який дозволяє користувачеві бачити активні цілі, опис завдань та прогрес виконання кожного кроку (рис. 3.6).

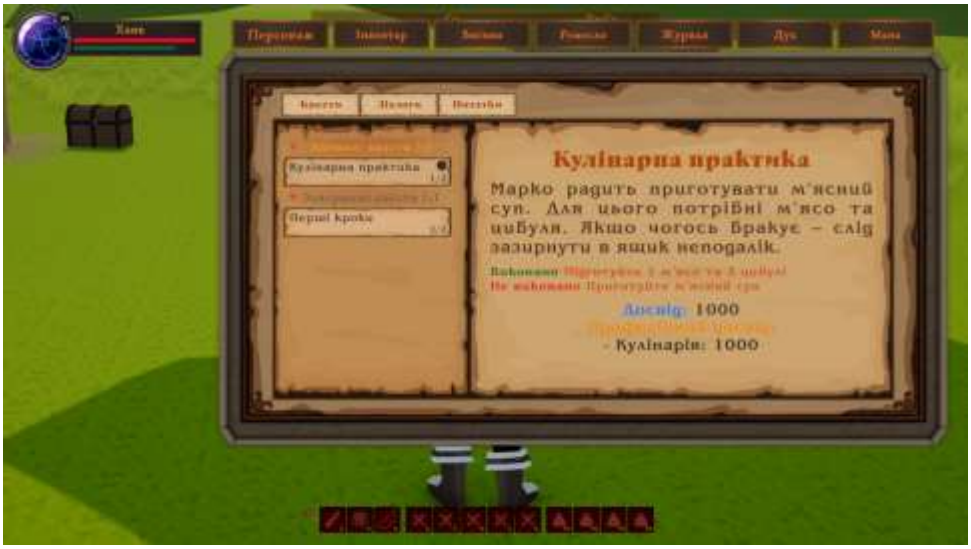


Рисунок 3.6 – Інтерфейс журналу квестів та відстеження завдань

Бойова механіка реалізує взаємодію персонажа з противниками та включає механіки атаки, отримання шкоди, витрат ресурсів, а також рухи ухилення (перекати). Обробка бойових дій здійснюється у реальному часі з урахуванням характеристик персонажа. Основна логіка управління боєм, що включає обробку комбо-атак та стану переривання, зосереджена у класі PlayerCombat.

Візуалізація бойової сцени, на якій продемонстровано взаємодію гравця з ворогом та роботу анімаційних тригерів, наведена на рис. 3.7.



Рис. 3.7 – Демонстрація бойової механіки в ігровому середовищі

Механіки руху персонажа, зокрема переміщення, стрибки та перекати, реалізовані із використанням системи введення Unity Input System [41] та фізичного рушія. Вони забезпечують плавну взаємодію користувача з ігровим світом. Логіка обробки переміщення у просторі, розрахунку напрямку відносно камери та виконання маневрів ухилення описана у класі PlayerController. Частковий вихідний код представлено у додатку А.

На рис. 3.8 продемонстровано процес переміщення персонажа в ігровій сцені з активною візуалізацією анімаційних станів.



Рис. 3.8 – Демонстрація переміщення та динаміки персонажа

Діалогова механіка забезпечує взаємодію користувача з неігровими персонажами, зокрема ШІ-асистентом. Вона реалізована через інтерфейс введення тексту (або вибору готового варіанту) та відображення відповідей. Керування логікою розмов, перевірка передумов (наявність активних або завершених квестів) та відображення варіантів відповідей гравця покладено на менеджер DialogueManager. Частковий вихідний код представлено у додатку А.

Інтерфейс діалогового вікна, що включає історію повідомлень, іконку персонажа та динамічні кнопки вибору, зображено на рис. 3.9.



Рис. 3.9 – Інтерфейс діалогової механіки та вибір варіантів відповіді

Механіка торгівлі та створення ігрових об’єктів забезпечує економічну складову гри. Користувач може обмінювати ресурси, купувати або створювати предмети. Крафт реалізується шляхом поєднання ресурсів за заданими рецептами. Логіка інтерфейсу та перевірки створення предметів реалізована у менеджері CraftingUIManager, який взаємодіє з інвентарем та характеристиками гравця.

Візуальне представлення вікна створення, що містить список доступних рецептів, необхідні інгредієнти, вимоги до виробничої локації та стан обраної професії, наведено на рис. 3.10.



Рис. 3.10 – Інтерфейс створення та перелік інгредієнтів

Механіка торгівлі інтегрована в загальну економічну модель гри та дозволяє гравцеві взаємодіяти з NPC для оновлення спорядження або збуту знайдених ресурсів. Реалізація базується на менеджері TradeManager, який керує станом транзакції, перевіряє баланс золота обох сторін та забезпечує коректне переміщення об’єктів між інвентарями.

Візуальний інтерфейс вікна торгівлі, що відображає інвентарі обох сторін, слоти обміну, кошти, опис предметів та підсумкову вартість угоди, представлено на рис. 3.11.



Рис. 3.11 – Інтерфейс торгівлі з відображенням балансу та предметів

Окрім наведених ключових механік, програмна реалізація включає низку додаткових елементів, таких як налаштування, сховища предметів, механізми взаємодії з оточенням, візуальні ефекти, звукове супроводжування, кастомізація, створення профілю персонажа і т.д.

Також, у середовищі реалізований механізм збереження через серіалізацію даних у форматі JSON. Такий підхід обрано завдяки його простоті, універсальності та зручній інтеграції з C# і Python. Збереження виконується на стороні клієнта (Unity), що зменшує залежність від зовнішніх сервісів і забезпечує автономність роботи.

Архітектурно механізм збереження побудований на основі інтерфейсу, який реалізується класами, що потребують збереження свого стану. Зокрема, використовується інтерфейс ISaveable, який визначає методи для отримання та відновлення стану об’єкта.

Використання цього інтерфейсу дозволяє механізму збереження автоматично проходити по всіх об'єктах у сцені, що його реалізують, збирати їх дані у єдиний файл та відновлювати ігровий прогрес при наступному запуску додатка. Загальну схему процесу збереження/завантаження представлено на рисунку 3.12.

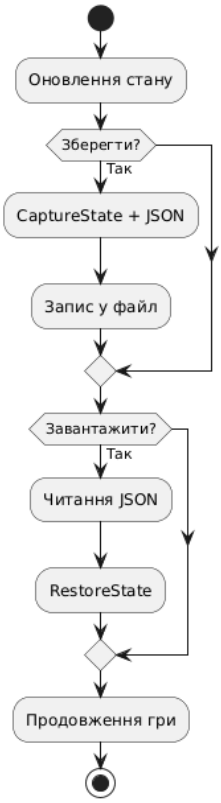


Рис. 3.12 – Схема процесу збереження та завантаження даних

Практична реалізація інтерфейсу передбачає визначення логіки збору та відновлення даних у конкретних класах. Зокрема, метод CaptureState формує об’єкт, який містить усі дані, необхідні для подальшого відновлення стану середовища.

Відновлення стану здійснюється за допомогою методу RestoreState, який десеріалізує отримані дані та відновлює внутрішній стан об’єкта.

Керування процесом серіалізації та десеріалізації даних здійснюється централізовано через клас SaveManager. Він реалізує патерн Singleton, забезпечуючи єдину точку доступу до функцій збереження з будь-якої частини програми, та відповідає за роботу з файловою системою, створення скріншотів для слотів збереження та менеджмент об'єктів, що реалізують інтерфейс ISaveable. Частковий вихідний код представлено у додатку А.

Інтерфейс забезпечує інтуїтивний доступ до управління слотами збережень, що дозволяє швидко оперувати сесіями.

```
public interface ISaveable
{
   string GetSaveID();
   object CaptureState();
   void RestoreState(object state);
}
private IEnumerator SaveGameRoutine(int slot)
{
   var saveables = FindObjectsByType<MonoBehaviour>(FindObjectsSortMode.None)
      .OfType<ISaveable>();
   SaveGameWrapper wrapper = new() {saveVersion = 1,
      saveTime = DateTime.Now.ToString("yyyy-MM-dd HH:mm:ss")};
   foreach (var saveable in saveables) {
      object state = saveable.CaptureState();
      string json = JsonUtility.ToJson(state);
      wrapper.objects.Add(new SaveObjectEntry {
         id = saveable.GetSaveID(),
         json = json,
         type = state.GetType().AssemblyQualifiedName }); }
   string path = GetSaveFilePath(slot);
   File.WriteAllText(path, JsonUtility.ToJson(wrapper, true));
}
public void LoadGame(int slot)
{
   string json = File.ReadAllText(GetSaveFilePath(slot));
   var wrapper = JsonUtility.FromJson<SaveGameWrapper>(json);
   var saveables = FindObjectsByType<MonoBehaviour>(FindObjectsSortMode.None)
      .OfType<ISaveable>();
   foreach (var entry in wrapper.objects) {
      var saveable = saveables.FirstOrDefault(s => s.GetSaveID() == entry.id);
      if (saveable == null) continue;
      Type type = Type.GetType(entry.type);
      object data = JsonUtility.FromJson(entry.json, type);
      saveable.RestoreState(data);}
}
```

Візуальна реалізація елементів збереження/завантаження даних представлена на рис. 3.13.



Рис. 3.13 Інтерфейс механізму збереження прогресу

У підсумку, реалізовані ігрові механіки формують цілісне інтерактивне середовище, у якому навчальний процес інтегрується у гейміфікований контекст. Використання різних ігрових складових забезпечує підвищення мотивації користувача та ефективність засвоєння навчального матеріалу. Реалізований механізм збереження даних забезпечує надійне збереження стану гри та навчального прогресу користувача, підтримує складні структури даних та дозволяє ефективно відновлювати їх при повторному запуску середовища.

3.4 Тестування готового освітнього компоненту

У процесі тестування перевірено роботу клієнтської частини, реалізованої в середовищі Unity. Встановлено, що інтерфейс коректно відображає всі елементи, а взаємодія відбувається без затримок і помилок. Ігрові механіки (рух персонажа, бойова механіка, інвентар, квести та діалоги) функціонують стабільно та відповідають заданій логіці.

Окрему увагу було приділено тестуванню серверної частини, реалізованої з використанням мови програмування Python. Встановлено, що сервер коректно обробляє HTTP-запити, передає дані до ШІ-моделі та повертає результати у відповідному форматі. Взаємодія між клієнтом і сервером є стабільною та без втрати даних.

Тестування ШІ-асистента показало, що він здатен генерувати змістовні та релевантні відповіді на запити користувача. Отримані відповіді є логічно узгодженими, зрозумілими та відповідають контексту навчального процесу. ШІ-асистент успішно виконує функції підказок, пояснення матеріалу та супроводу користувача в ігровому середовищі.

У ході тестування адаптивної навчальної складової було підтверджено коректність роботи механізмів аналізу дій користувача та зміни рівня складності завдань. Складова адекватно реагує на результати виконання завдань, змінюючи складність відповідно до рівня підготовки користувача, що підтверджує ефективність реалізованих алгоритмів адаптації.

Також було перевірено роботу механізму збереження даних. Результати тестування показали, що процеси серіалізації та десеріалізації даних у формат JSON виконуються коректно. Дані користувача, включаючи прогрес, характеристики персонажа та стан навчальної складової, успішно зберігаються та відновлюються при повторному запуску середовища.

Разом із тим, у процесі тестування було виявлено окремі недоліки, що впливають на зручність використання освітнього компонента. Зокрема, спостерігається відносно тривалий час генерації відповідей ШІ-асистента, який у середньому становить від 40 секунд до 90 секунд залежно від складності запиту (рис. 3.15). Така затримка обумовлена використанням локально розгорнутої мовної моделі та обмеженими обчислювальними ресурсами апаратного забезпечення.

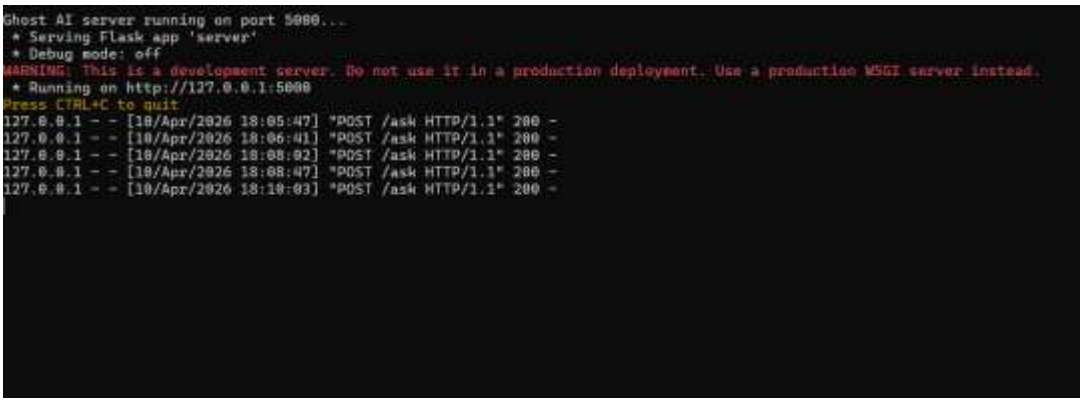


Рис. 3.15 – Фрагмент журналу сервера з часом обробки запитів

Водночас зазначений недолік частково нівелюється особливостями реалізації компоненту, зокрема наявністю гейміфікованих механік, а також інтерактивною взаємодією користувача з ігровим світом. У період очікування відповіді користувач має можливість виконувати інші ігрові дії (дослідження світу, взаємодія з об’єктами, перегляд інвентарю тощо), що знижує суб’єктивне сприйняття затримки та підтримує загальний рівень залученості.

Додатково варто відзначити лінгвістичні особливості роботи ШІ-асистента. Під час тестування було виявлено, що найвищий рівень точності та релевантності відповідей досягається при використанні англійської мови. При

взаємодії українською мовою спостерігається певне зниження якості генерації контенту. Це зумовлено специфікою навчання використовуваної моделі, де англомовний корпус даних є домінуючим, а також технічними обмеженнями архітектури моделі щодо обробки флективних мов. Отже, для отримання максимально деталізованих відповідей користувачам рекомендовано використовувати англійську мову, хоча підтримка української залишається функціонально доступною.

Також у деяких випадках можливі незначні затримки при обробці запитів або переходах між станами середовища. Окремі елементи інтерфейсу потребують подальшого вдосконалення з точки зору зручності навігації, сприйняття та візуального оформлення. Узагальнені результати тестування зображені у табл. 3.1.

Таблиця 3.1 Результати тестування функціональних модулів

№	Модуль	Що тестується	Результат
1	UI	Відображення інтерфейсу	Успішно
2	Клієнт-сервер	Передача даних	Успішно
3	ШІ	Генерація відповідей	Успішно
4	Адаптація	Зміна складності	Успішно
5	Збереження	JSON збереження/завантаження	Успішно
6	Навантаження	Час відповіді	Частково

Незважаючи на зазначені недоліки, результати проведеного тестування підтверджують загальну працездатність та стабільність розробленого освітнього компоненту. Усі основні функціональні елементи успішно інтегровані та взаємодіють між собою відповідно до визначеної архітектури.

Таким чином, створений освітній компонент може бути використаний як основа для подальшого розвитку, оптимізації та розширення функціональних можливостей.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було розроблено освітній компонент з адаптивним ШІ, гейміфікацією та індивідуальним рівнем складності. Основною метою роботи було створення програмного середовища, яке поєднує навчальний процес із інтерактивною ігровою взаємодією та забезпечує індивідуалізацію складності завдань відповідно до рівня користувача.

У процесі реалізації було виконано аналіз предметної області, розглянуто сучасні підходи до створення навчальних систем та інтерактивних застосунків, а також визначено основні вимоги до функціональності програмного забезпечення. На основі отриманих результатів було спроєктовано архітектуру компоненту та реалізовано його ключові модулі.

Розроблений компонент включає клієнтську частину, реалізовану в середовищі Unity, серверну частину для обробки запитів та взаємодії з ШІ-моделлю, а також механізм збереження даних користувача. Окрему увагу приділено реалізації адаптивної навчальної складової, який базується на аналізі результатів виконання завдань, історії помилок, часу відповіді та поведінкових показників користувача.

Інтегрований ШІ-асистент забезпечує підтримку користувача в режимі реального часу, генерує пояснення навчального матеріалу, підказки та адаптує рівень складності відповідно до контексту взаємодії. Додатково реалізовано ігрові механіки, що підвищують рівень залученості користувача та формують мотиваційний компонент навчального процесу.

У ході тестування було підтверджено коректність роботи основних функціональних модулів компонента, стабільність взаємодії між клієнтською та серверною частинами, а також працездатність механізмів збереження та відновлення даних. Водночас було виявлено окремі обмеження, зокрема значний час генерації відповідей ШІ-асистента, що зумовлено використанням локально розгорнутої мовної моделі та обмеженими обчислювальними ресурсами.

Незважаючи на зазначені недоліки, результати роботи підтверджують працездатність запропонованого підходу та ефективність поєднання навчальних і ігрових механік у єдиному освітньому компоненті. Розроблене середовище може бути використане як основа для подальшого вдосконалення, масштабування та впровадження більш складних алгоритмів адаптації та штучного інтелекту.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Матківський М. П., Тарас Т. М., Лучкевич Є. Р. Роль гейміфікації у покращенні мотивації та навчальних результатів учнів середньої школи в умовах цифрової трансформації освіти в Україні. Педагогічна Академія: наукові записки. 2025. https://doi.org/10.5281/zenodo.14717082
2. Мізюк В. А., Хижняк А. В., Хренова В. В. Використання адаптивних навчальних платформ для персоналізації дистанційного навчання. Педагогічна Академія: наукові записки. 2025. https://doi.org/10.5281/zenodo.14605125
3. Карабін С. В. Гейміфікація в освітньому процесі як засіб розвитку молодших школярів. Репозитарій ТНПУ. 2019. http://dspace.tnpu.edu.ua/bitstream/123456789/14039/1/Karabin_Gameification_educational.pdf
4. Прус О. В. Використання штучного інтелекту в адаптивних навчальних системах: перспективи розвитку освіти України. Репозитарій ВНТУ. 2024. С. 1–15. https://ir.lib.vntu.edu.ua/bitstream/handle/123456789/41952/20967.pdf
5. Городянська Л. В. Синергія мотиваційних стратегій і технологій штучного інтелекту у формуванні креативних здібностей майбутніх управлінців. Вісник Полтавського державного аграрного університету. Серія «Публічне управління та адміністрування». 2025. № 4. С. 38–41. https://doi.org/10.32782/pdau.pma.2025.4.6
6. Вербовецький Д. В. Аналіз досвіду впровадження гейміфікації в освітній процес. Інститут інформаційних технологій. 2023. С. 1–15. https://lib.iitta.gov.ua/id/eprint/734560/
7. Шикун А. В. Гейміфікація в онлайн-навчанні іноземних мов як засіб підвищення мотивації та комунікативної активності студентів. Інноваційна педагогіка. 2025. № 83(2). https://www.innovpedagogy.od.ua/archives/2025/83/part_2/45.pdf
8. Сажнєва Ю. М. Гейміфікація як інструмент для підвищення активності учнів на уроках інформатики. На Урок. 2024. https://naurok.com.ua/pedagogichniy-

dosvid-geymifikaciya-yak-instrument-dlya-pidvischennya-aktivnosti-uchniv-na-urokah-informatiki-424775.html

9. Олефіренко Н. В., Добрунов О. С. Досвід застосування рольових ігор для розвитку соціальної компетентності. OpenEdu КНУ. 2025. С. 1–12. https://openedu.kubg.edu.ua/journal/index.php/openedu/article/download/522/461/3176

10. Равіндран Д.-р. Д. Гейміфікація на основі штучного інтерактивного навчання: зміна методологій викладання. Актуальні проблеми в системі освіти: загальноосвітній заклад середньої освіти – доуніверситетська підготовка – заклад вищої освіти. 2025. № 1(5). С. 29–38. https://jrnl.kai.edu.ua/index.php/APSE/article/view/20249

11. Коцюбівська К., Тимошенко О., Хрущ С., Мельник І. Використання інструментів штучного інтелекту під час розроблення плану персоналізованого навчання. Цифрова платформа: інформаційні технології в соціокультурній сфері. 2025. № 8(1). С. 52–60. https://doi.org/10.31866/2617-796X.8.1.2025.335531

12. Грищенко І. М. Штучний інтелект і гейміфікація як інструменти підвищення якості освітніх послуг в системі публічного управління освітою. Наука та інновації. 2025. № 7745. С. 1–20. https://www.nayka.com.ua/index.php/investplan/article/download/7745/7875/16996

13. Тумбрукакі А. В., Роговська М. Г. Новітні AI-технології в освітньому середовищі: систематизація досліджень і прикладні аспекти. 2025. https://doi.org/10.5281/zenodo.17395536

14. Трофименко О., Дика А., Лобода Ю., Толокнов А., Бондаренко М. Алгоритми штучного інтелекту в розробці ігор. Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка». 2025. № 3(27). С. 109–119. https://doi.org/10.28925/2663-4023.2025.27.701

15. Круглик В., Осадчий В., Павленко Л., Симоненко С. Формування відкритого освітнього середовища з використанням технологій штучного інтелекту: аналіз та класифікація. Освітологічний дискурс. 2024. № 45(2). С. 6–15. https://od.kubg.edu.ua/index.php/journal/article/view/1134/904

16. Squirrel AI Learning. Офіційний сайт. 2026. https://squirrelAI.com

17. Шпарик О. Досвід Китаю щодо дигіталізації освіти: Squirrel AI. Інститут педагогіки НАПН України. 2020. С. 308–311. https://lib.iitta.gov.ua/721234/1/Comparative_2020_web-308-311.pdf

18. Кучерявий І. В., Романюк О. В. Гейміфікація в освіті на прикладі програмної платформи для вивчення мов Duolingo. ВНТУ. 2023. С. 1–10. https://ir.lib.vntu.edu.ua/handle/123456789/39427

19. Як працюють алгоритми Duolingo і як кожен з нас їх щодня вдосконалює. 2023. Texty.org.ua. https://texty.org.ua/articles/108916/yak-pracyuyut-alhorytmy-duolingo-i-yak-kozhen-z-nas-yih-shodnya-vdoskonalyuye/

20. Запорожцева Ю. В. Персоналізація в навчанні як сучасний освітній тренд. Житомирський обласний інститут післядипломної педагогічної освіти. 2024. С. 1–11. https://imso.zippo.net.ua/wp-content/uploads/2024/12/3.%D0%97%D0%B0%D0%BF%D0%BE%D1%80%D0%BE%D0%B6%D1%86%D0%B5%D0%B2%D0%B0.pdf

21. Комлева Н. О. UML-орієнтована інформаційна технологія для неперервних задач максимального покриття з об’єктами довільної форми. Вісник Харківського національного університету імені В. Н. Каразіна. Серія «Математичне моделювання. Інформаційні технології. Автоматизовані системи управління». 2025. № 65. С. 1–17. https://periodicals.karazin.ua/mia/article/download/28375/24782

22. Bayesian knowledge tracing. 2026. In Wikipedia. https://en.wikipedia.org/wiki/Bayesian_knowledge_tracing

23. Мокін Б. І., Мокін О. Б., Войцеховська О. О., Пасєка Б. В. Ідентифікація еквівалентної моделі процесу забування знань в кожній зі «смуг забування». Вісник Вінницького політехнічного інституту. 2024. № 3. С. 56–65. https://doi.org/10.31649/1997-9266-2024-174-3-56-65

24. Крива забування Еббінгауза. 2026. Вікіпедія. https://uk.wikipedia.org/wiki/Крива_забування_Еббінгауза

25. Крамар А. В., Кравчук Я. Я., Осадчук С. І. Адаптивне навчання чат-ботів на основі динамічного аналізу поведінкових патернів користувачів. Вчені записки. 2025. № 4. С. 1–10. https://www.tech.vernadskyjournals.in.ua/journals/2025/4_2025/part_2/26.pdf

26. Василик М. С. та ін. Адаптивне навчання як чинник оптимізації освітнього процесу у закладах вищої освіти. Гірська школа Українських Карпат. 2024. № 30. С. 5–10. https://lib-repo.pnu.edu.ua/handle/123456789/23004

27. Modeling the Efficiency of AI-Based Adaptive Learning Systems. Інститут інформаційних технологій НАПН України. 2025. https://lib.iitta.gov.ua/id/eprint/748072/

28. Гейміфікація на основі RPG. AhaSlides. 2025. https://ahaslides.com/uk/blog/gamification-learning-platforms/

29. Unity Technologies. Unity Real-time Development Platform. 2026. https://unity.com

30. Unity Technologies. Unity User Manual (C# scripting). 2026. https://docs.unity3d.com/Manual/UnityManual.html

31. Ollama. Ollama GitHub Repository. GitHub. 2026. https://github.com/ollama/ollama

32. Розгортання LLM локально: усе, що потрібно знати. Dev.ua. 2025. https://dev.ua/blogs/posts/rozhortannia-llm-lokalno

33. Використання Python для задач штучного інтелекту. Вчені записки ДУІКТ. 2024. https://con.dut.edu.ua/index.php/communication/article/view/2809

34. Що таке REST API і як він працює. DAN.IT Education. 2026. https://dan-it.com.ua/uk/blog/shho-take-rest-api-i-yak-vin-praczyuye/

35. IETF. RFC 8259: The JavaScript Object Notation (JSON) Data Interchange Format. 2017. https://datatracker.ietf.org/doc/html/rfc8259

36. ScriptableObject. Unity Technologies. 2026. https://docs.unity3d.com/Manual/class-ScriptableObject.html

37. Слюсар В. П., Гусаковський І. П. Практичні аспекти розгортання великих мовних моделей у локальній мережі. Вісник НАУ. 2024. С. 1–10. https://slyusar.kiev.ua/Slyusar_LLM__Network.pdf

38. JSON Serialization. Unity Technologies. 2015. https://docs.unity3d.com/ru/530/Manual/JSONSerialization.html

39. UnityWebRequest. Unity Technologies. 2026. https://docs.unity3d.com/6000.4/Documentation/ScriptReference/Networking.UnityWebRequest.html

40. Generate a chat message. Ollama Documentation. 2026. https://docs.ollama.com/api/chat

41. Input System. Unity Technologies. 2026. https://docs.unity3d.com/Packages/com.unity.inputsystem@1.19/manual/index.html