

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ

автоматизації і інформаційних технологій

(факультет)

інформаційних технологій

(кафедра)

ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «БАКАЛАВР»

на тему: «Гейміфікація у системі навчання: аналіз, моделювання та програмна
реалізація»

БУГА ІВАН ВАЛЕНТИНОВИЧ

(прізвище, ім'я та по батькові студента повністю)

Київ 2024 р.

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ

автоматизації і інформаційних технологій

(факультет)

інформаційних технологій

(кафедра)

ЗАТВЕРДЖУЮ

Завідувач кафедри ІТ
к.т.н., доцент Гончаренко Т.А.

„___” _____ 2024 року

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «БАКАЛАВР»**

на тему: "Гейміфікація у системі навчання: аналіз, моделювання та програмна
реалізація"

Виконав: студент 4-го курсу, групи КН-20-1

Освітня програма: 122 «Комп'ютерні науки
(шифр і назва напрямку підготовки, спеціальності)

Буга І.В.

(прізвище та ініціали)

Керівник Вацкель В.Ю.

(прізвище та ініціали)

Рецензент _____

(прізвище та ініціали)

Київ, 2024 .

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

Факультет: автоматизації і інформаційних технологій

Кафедра: інформаційних технологій

Освітній рівень: «бакалавр за ОПП»

Освітня програма: 122. «Комп'ютерні науки»

ЗАТВЕРДЖУЮ

Завідувач кафедри ІТ

к.т.н., доцент Гончаренко Т.А.

_____ 2024 року

**З А В Д А Н Н Я
ДО ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «БАКАЛАВР»**

Буга Іван Валентинович

1. Тема роботи: Гейміфікація у системі навчання: аналіз, моделювання та програмна реалізація.

затверджена наказом ректора КНУБА № 433\2 від «29» лютого 2024р.

2. Керівник роботи: Вацкель Володимир Юрійович, асистент кафедри інформаційних технологій проектування і прикладної математики

3. Строк подання студентом роботи до захисту: червень 2024 рік

4. Зміст пояснювальної записки за розділами:

P.1. Теоретичне дослідження розвитку гейміфікації у навчанні

P.2. Удосконалення та розробка методів і моделей гейміфікації освітнього процесу

P.3. Програмна реалізація процесу системи навчання

P.4. Ергономіка інформаційних систем

5. Інформаційні слайди:

С.1. Вступ

С.2. Приклади рішень

С.3. Існуючі рішення

С.4. Засоби розробки

С.5. Результати розробки

6. Календарний план виконання кваліфікаційної роботи

Види робіт та їх зміст	Дата виконання
Р.1. Теоретичне дослідження розвитку гейміфікації у навчанні	Травень 2024 р.
Р.2. Удосконалення та розробка методів і моделей гейміфікації освітнього процесу	Травень 2024 р.
Р.3. Програмна реалізація процесу системи навчання	Травень 2024 р.
Р.4. Ергономіка інформаційних систем	Травень 2024р.
Остаточне оформлення роботи	Червень 2024 р.
Направлення роботи на рецензування, перевірку на плагіат	Червень 2024 р.
Попередній захист роботи на кафедрі	Червень 2024 р.

7. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Перевірив	
		дата	підпис
Прийом програмного продукту	Ас. Вацкель В.Ю.		

8. Дата видачі завдання: 14 лютого 2024 року

Керівник

(підпис)

Вацкель В.Ю.

(прізвище та ініціали)

Здобувач

(підпис)

Буга І.В.

(прізвище та ініціали)

АНОТАЦІЯ

У дипломній роботі розглянуто розробку програмного забезпечення "GamificationApp" для підвищення якості навчання програмістів через інтерактивні навчальні ігри. Основною метою розробки було створення зручного та ефективного інструменту для вдосконалення навичок програмування у цікавій та мотивуючій формі.

Розробка включала створення модульної архітектури, що забезпечує легкість у підтримці та розширенні функціональності програми. Використання мови програмування C# та фреймворку Windows Forms дозволило реалізувати ефективну і надійну логіку додатку, а серверна частина, реалізована на ASP.NET Core, забезпечила надійне управління даними та високу масштабованість. Дані користувачів зберігаються у файлах, що спрощує їх обробку та зберігання.

Кожна гра у програмі включає кілька рівнів складності, що дозволяє користувачам поступово підвищувати свої навички. Впровадження системи досягнень та нагород забезпечує додаткову мотивацію для користувачів. Програма підтримує функціонал реєстрації та входу користувачів, зберігає та відображає прогрес користувачів у кожній грі.

Ключові слова: гейміфікація, навчання програмуванню, Windows Forms, C#, інтерактивні ігри, користувацький прогрес.

ABSTRACT

The diploma work addresses the development of the "GamificationApp" software aimed at enhancing the quality of programmer education through interactive learning games. The main objective was to create a convenient and efficient tool for improving programming skills in an engaging and motivating manner.

The development included creating a modular architecture that ensures ease of maintenance and functionality expansion. Using the C# programming language and Windows Forms framework allowed for implementing an effective and reliable application logic, while the server-side implemented on ASP.NET Core provided reliable data management and high scalability. User data is stored in files, simplifying their processing and storage.

Each game in the application includes multiple difficulty levels, enabling users to gradually enhance their skills. The implementation of an achievement and reward system provides additional motivation for users. The program supports user registration and login functionality, saves, and displays user progress in each game.

Keywords: gamification, programming education, Windows Forms, C#, interactive games, user progress.

ЗМІСТ

ВСТУП.....	8
1. РОЗДІЛ 1. Теоретичне дослідження розвитку гейміфікації у навчанні .	10
1.1. Характеристика поняття та приклад застосування гейміфікації в навчанні.....	10
1.2. Основні методи та моделі гейміфікації	14
1.3. Опис існуючих рішень.....	17
1.4. Аналіз існуючих програм для навчання програмування	39
1.5. Постановка задачі для створення гейміфікованого ЕОІС	43
2. РОЗДІЛ 2. Алгоритмічне та математичне пояснення.....	46
2.1. Проектування архітектури системи.....	46
2.2. Бібліотеки та фреймворки для розробки додатку	52
2.3. Моделювання ігрових механік	54
3. РОЗДІЛ 3. Розробка програмного забезпечення.....	59
3.1. Методологія розробки	59
3.2. Загальні положення	60
3.3. Вибір програмних засобів.....	62
3.4. Реалізація програмного забезпечення.....	64
3.5. Опис функціональних можливостей	66
4. РОЗДІЛ 4. Ергономіка інформаційних систем	69
4.1. Ергономіка ІТ.....	69
4.2. Дизайн додатку	71
4.3. Технічно-економічне обґрунтування розробки.....	82
4.4. Математичні методи оцінки часу розробки програмного забезпечення	89
ВИСНОВКИ.....	96
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	98
ДОДАТОК А ПРЕЗЕНТАЦІЯ	102
ДОДАТОК Б ЛІСТИНГ КОДУ	105

ВСТУП

В умовах сучасного інформаційного суспільства гейміфікація стає все більш популярною у різних галузях, включаючи освіту. Гейміфікація в навчальному процесі має на меті підвищити залученість студентів та покращити їхні навчальні результати шляхом інтеграції ігрових елементів у освітні програми. Ця випускна кваліфікаційна робота присвячена розробці та модернізації об'єкту інформатизації у вигляді гейміфікованої системи навчання, яка спрямована на підвищення якості освітнього процесу та мотивації студентів.

Метою даної роботи є створення гейміфікованої освітньої інформаційної системи, яка забезпечить інтерактивне та захоплююче навчання для студентів, сприятиме розвитку їхніх навичок та знань, а також підвищить їхню зацікавленість у навчанні. Необхідність розробки гейміфікованої системи обумовлена наступними факторами:

- 1) гейміфікація дозволяє зробити навчання більш цікавим та захоплюючим, що сприяє збільшенню мотивації студентів до навчання;
- 2) використання ігрових елементів сприяє кращому засвоєнню матеріалу, оскільки студенти активно залучаються до навчального процесу;
- 3) гейміфіковані системи забезпечують миттєвий зворотний зв'язок, що дозволяє студентам оперативно коригувати свої дії та досягати кращих результатів;

Основні проектні рішення, що були прийняті при розробці гейміфікованої системи, включають:

- 1) система включає такі елементи, як бейджі, бали, рівні, лідерборди та квести, які стимулюють студентів до активної участі у навчальному процесі;
- 2) для розробки системи використовуються сучасні технології програмування, такі як C# та Windows Forms, що забезпечує високу продуктивність та зручність користування;

3) система має модульну архітектуру, яка дозволяє легко розширювати та модернізувати її функціональність відповідно до потреб користувачів.

Результати даної роботи можуть бути застосовані в різних галузях освіти, включаючи:

1) гейміфікована система може бути використана для покращення навчального процесу в загальноосвітніх та вищих навчальних закладах;

2) система може бути застосована для навчання та підвищення кваліфікації працівників у різних професійних галузях;

3) гейміфіковані елементи можуть бути інтегровані в онлайн-платформи для забезпечення більшої залученості студентів.

Актуальність теми даної роботи обумовлена зростаючою потребою у вдосконаленні освітніх методик та підвищенні ефективності навчання. Традиційні методи навчання не завжди здатні забезпечити достатню мотивацію студентів, що призводить до зниження їхньої успішності. Гейміфікація, навпаки, пропонує інноваційний підхід до навчання, який враховує психологічні особливості студентів та сприяє їхньому активному залученню в освітній процес.

Критичний аналіз відомих рішень у сфері гейміфікації свідчить про високу ефективність цього підходу у різних галузях, включаючи корпоративне навчання, охорону здоров'я, маркетинг та освіту. Проте, наявні рішення часто орієнтовані на конкретні завдання та не завжди враховують потреби освітніх закладів України. Це підкреслює доцільність та необхідність розробки власних гейміфікованих систем, які будуть адаптовані до умов української освітньої системи та сприятимуть її розвитку.

Отже, дана робота є важливим кроком у напрямку інтеграції гейміфікації в освітній процес, що має потенціал значно покращити якість навчання та підвищити мотивацію студентів до здобуття нових знань.

1. РОЗДІЛ 1. Теоретичне дослідження розвитку гейміфікації у навчанні

1.1. Характеристика поняття та приклад застосування гейміфікації в навчанні

За останні кілька років гейміфікація увійшла до переліку трендів освітніх процесів для різних категорій студентів. Гейміфікацію вивчають фахівці з навчання в університетах і компаніях та досліджують окремі навчальні заклади. Тому ми вважаємо, що варто детальніше розглянути цю технологію та визначити її характерні риси і чим вона відрізняється від суміжних категорій.

Гейміфікація - це використання ігрових механік у неігрових ситуаціях для заохочення певної поведінки [1].

Основним принципом гейміфікації з точки зору програмного забезпечення є забезпечення постійного та вимірюваного зворотного зв'язку з користувачем, що дозволяє користувачеві динамічно коригувати свою поведінку. основні аспекти гейміфікації:

- 1) динаміка - використання сценаріїв, що вимагають уваги;
- 2) використання сценаріїв, що вимагають уваги та реакції користувача в режимі реального часу;
- 3) динаміка - використання елементів сценарію, характерних для ігрового процесу, таких як віртуальні нагороди, статуси, віртуальні товари тощо;
- 4) естетика - створення загального ігрового досвіду, який сприяє емоційному залученню користувача;
- 5) соціальна взаємодія - широкий спектр методів, які забезпечують взаємодію користувачів, характерну для ігор [2].

Гейміфікацію часто ототожнюють зі спорідненими термінами "ігри" та "ігрове навчання", але, на відміну від "гейміфікації", вона має певні критерії (табл. 1.1).

Таблиця 1.1 - Критерії гри та гейміфікації

Гра	Навчання, засноване на грі (навчання через реальну гру)	Гейміфікація
Для розваги, може мати або не мати суворі правила	Гра визначає навчальні цілі (об'єкти)	Може бути набором завдань з ясними цілями і формами досягнень
Виграш чи програш – частина гри	Програшу може і не бути, так як мета, це мотивація учня дійти до кінця навчальної програми (досягти навчальної мети)	Програшу може і не бути, так як мета, це мотивація учня зробити певний дії та досягти навчальних цілей
Гра первинна – досягнення вторинні	Іноді сама гра є некритичним досягненням	Можуть бути некритичні досягнення
Ігри зазвичай важко і дорого створювати	Дорого і важко створювати	Дешевше і легше робити
Сюжет і його частини (сцени) – частина гри	Зміст подібний сценам гри	Ігрові елементи, наприклад, додаються в LMS чи іншу систему навчання, де міститься контент

Коли в навчальній літературі гейміфікацію описують як низку ігрових елементів, що використовуються для навчання та спрямовані на мотивацію студентів, у тому числі в корпоративному секторі, варто навести градацію найбільш ефективних для кінцевого користувача елементів.

Виходячи з цього, можна виділити три рівні гейміфікації:

1) використання в навчальних курсах системи балів, бейджів (що свідчать про досягнення) та оцінок слухачів (таблиць лідерів);

2) додавання сюжету або ігрової атмосфери. Поетапне подання навчальної інформації, певне ускладнення змісту від уроку до уроку, коли

перехід до нової теми подається як великий стрибок вперед. Також включає такі функції, як внутрішньосистемна взаємодія між користувачами, можливість миттєвого зворотного зв'язку та інтерактивні навчальні відео, де сюжет змінюється відповідно до поведінки учня;

3) розробка автентичних навчальних ігор, які поєднують пізнання та розваги.

У науковій літературі та практиками також висловлюються думки щодо використання гейміфікації та критеріїв її ефективності. Наприклад, зазначається, що активне використання бейджів та таблиць лідерів на заняттях для дорослих майже завжди призводить до посилення конкуренції.

Безумовно, дух суперництва спонукатиме учнів виконувати завдання швидше та краще, але якщо один з учасників показує результати, далекі від лідера в списку, за певних умов він може опустити руки і вирішити, що немає сенсу вчитися [3].

Крім того, справжня навчальна програма на основі гейміфікації повинна містити план (алгоритм) досягнення реальних навчальних цілей і зміни поведінки учнів.

Інші стверджують, що вищезгадані механізми, такі як PBL (бали, бейджі та рівні), працюють у короткостроковій перспективі.

Іншими словами, потрібні більш складні ігри та більш глибоке опрацювання легенд і мотивації працівників = геймерів для виконання конкретних завдань. Наприклад, у корпоративному навчанні гейміфікація сприймається як додаткова мотивація співробітників до роботи, коли насправді, згідно з дослідженнями, основний мотивуючий фактор - взяти під контроль своє життя.

А щоб його підвищити, їм потрібно мати можливість прогнозувати майбутнє компанії та себе. Для працівників, орієнтованих на стабільність, необхідно зробити майбутнє компанії та їхнє власне передбачуваним. Це вимагає відкритої комунікації та швидкого зворотного зв'язку від топ-

менеджменту до всіх рівнів, а для працівників, орієнтованих на розвиток, це означає можливість впливати на свою кар'єру та заробітну плату і долати організаційні бар'єри.

Для цього необхідно підвищити автономію працівників, а відкрита комунікація все ще необхідна для усунення будь-яких бар'єрів на шляху розвитку працівників.

З цього можна зробити висновок, що гейміфікація не повинна бути пріоритетом, а скоріше опцією. Так, гейміфікація може бути дуже ефективною у візуальному визнанні досягнень студентів (учнів), організації змагань всередині курсів та розробці навчального контенту.

Однак у реальному освітньому процесі необхідні чіткі та зрозумілі цілі навчання, реалістичність отриманих результатів та їхня застосовність у житті та професійній діяльності, а також організаційні реформи в освітніх організаціях та відомствах (підтримка викладачів у розробці ефективних навчальних програм, дизайн якісних навчальних аудиторій, розвиток інформаційно-комунікаційної інфраструктури в освітніх закладах тощо).

Ми вважаємо, що гейміфікація не повинна відводити учнів від запланованих результатів. Використовуючи гейміфікацію, розробники часто захоплюються механікою гри і забувають про цілі навчального курсу.

Щоб уникнути цього, важливо завжди пам'ятати, чого очікують від користувачів, які проходять курс.

Додаючи ігровий елемент, потрібно одразу подивитися, чи допоможе він учневі краще зрозуміти і надалі використовувати отриману інформацію. Потрібно визнати, що основна мета використання гейміфікації - змінити поведінку окремих людей та аудиторії в цілому, тобто гейміфікація не для розваги, а для підвищення ефективності навчального процесу.

1.2. Основні методи та моделі гейміфікації

Гейміфікація - це ефективний інструмент у розробці навчального досвіду, який використовує привабливість ігор і застосовує її в неігрових контекстах, таких як освіта та навчання. Коли гейміфікація працює добре, вона може значно підвищити рівень залученості та запам'ятовування.

1) встановіть чіткі навчальні цілі: успіх гейміфікації полягає в досягненні навчальних цілей. Чітко визначені та вимірювані навчальні цілі є основою дизайну гейміфікованого контенту. Це гарантує, що ігрові елементи узгоджуються з бажаними результатами, а вправи мають конкретну мету і сприяють навчанню;

2) використовуйте заохочення: важливим елементом гейміфікації є мотивація, яку можна значно посилити за допомогою ігрових винагород. Винагороди можуть варіюватися від балів і значків до таблиць лідерів, індикаторів прогресу і розблокування нового контенту. Однак важливо, щоб ці винагороди відповідали значущим результатам і заохочували майстерність, а не просто участь;

3) включайте інтерактивність: активне навчання завжди ефективніше за пасивне. Сюди входять вікторини, вправи з перетягуванням та вирішенням проблем. Якщо зробити учнів активними учасниками, це підвищить їхню зацікавленість у навчальному процесі, залученість і пам'ять;

4) заохочення конкуренції та співпраці: ще однією психологічною привабливістю ігор є їхній соціальний аспект. Включення елементів змагання (наприклад, таблиць лідерів і рейтингів) та співпраці (наприклад, командних завдань і групових квестів) може значно підвищити зацікавленість і залученість. Деякі люди процвітають у змаганні, тоді як інші можуть віддавати перевагу підходу, заснованому на співпраці;

5) розповідь: включення елемента розповіді може надати контекст навчальному процесу і зробити його більш цікавим і таким, що запам'ятовується. Це не обов'язково має бути складним і може бути простим,

наприклад, пов'язати завдання тематично або створити сюжетну лінію прогресу;

6) збалансування складності: важливо дотримуватися балансу між складністю та досяжністю. Завдання мають бути достатньо складними, щоб зацікавити учнів, але не повинні бути занадто складними і демотивуючими. Найкраще починати з легких завдань і поступово збільшувати їхню складність, щоб вони ускладнювалися в міру того, як учень опановує попереднє завдання;

7) забезпечуйте зворотний зв'язок: негайний і регулярний зворотний зв'язок допомагає учням зрозуміти, де вони перебувають у процесі навчання. Це мотивує учнів до вдосконалення та дозволяє виправляти помилки в режимі реального часу;

8) свобода робити помилки: відеоігри часто дозволяють учням повторно пройти рівень, якщо вони не впоралися з ним з першого разу. Впровадження подібного підходу в навчальний дизайн може допомогти учням розвинути експериментування та стійкість - якості, які є важливими для ефективного навчання.

Таким чином, ефективне використання гейміфікації в дизайні навчального досвіду ґрунтується на поєднанні цих стратегій. Йдеться про розуміння учня, знання того, що його мотивує і зацікавлює, а потім створення дизайну, який перетворює цю зацікавленість на реальні результати навчання. І, нарешті, потрібно пам'ятати про дух гри - про задоволення.

Гейміфікація навчання - це чудовий спосіб підвищити зацікавленість, але для того, щоб бути ефективною, вона має бути обережною і продуманою. Успішна гейміфікація вимагає глибокого розуміння як принципів навчання, так і ігрового дизайну [4].

Опишемо кілька практичних стратегій, які допоможуть впровадити гейміфікацію у навчальному середовищі:

1) почніть з навчальних цілей: спочатку визначте навчальні цілі та розробіть елементи гри відповідно до цих цілей. Основним моментом є

узгодження ігрових елементів з навчальними цілями, а не навпаки. Елементи гри повинні допомогти досягти цих цілей, покращуючи розуміння та запам'ятовування;

2) вибір правильної ігрової механіки: вибір ігрової механіки має вирішальне значення. Чи таблиці лідерів, значки, бали та індикатори прогресу найкраще підходять для учнів? Змагальні чи кооперативні ігри є більш ефективними? Відповіді на ці питання допоможуть вам обрати правильні механізми, які будуть резонувати з вашою цільовою аудиторією;

3) включіть механізми зворотного зв'язку: зворотний зв'язок є невід'ємною частиною будь-якого навчального процесу. Гейміфіковане навчання забезпечує миттєвий зворотний зв'язок через бали, оцінки, анімовані графіки тощо. Коли навчальна діяльність гейміфікована, досягнення та невдачі стають можливостями для навчання;

4) баланс рівнів складності: важливим аспектом розробки ігор є баланс між складністю та навичками. Якщо гра занадто легка, вона стає нудною; якщо занадто складна, вона починає дратувати. Ігрові елементи та навчальні матеріали повинні бути складними, але досяжними, щоб підтримувати мотивацію учнів [5];

5) персоналізуйте досвід: учні з більшою ймовірністю будуть зацікавлені в навчанні, якщо навчальні матеріали адаптовані до їхніх інтересів, уподобань та здібностей. Щоб персоналізувати навчальний досвід, спробуйте включити адаптивні навчальні траєкторії, аватари та елементи сторітелінгу;

6) додайте соціальні елементи: соціальні компоненти, такі як таблиці лідерів, командні завдання та дискусійні форуми, можуть підвищити залученість. Ці соціальні елементи допомагають будувати спільноту, заохочують взаємодію та глибше навчання;

7) тестуйте і вдосконалюйте: нарешті, перед запуском гейміфікованого навчання протестуйте його з невеликою групою студентів. Збираються відгуки, вносяться уточнення та коригування за потреби. Мета - створити баланс між

навчанням і розвагами, коли навчальні цілі досягаються, а учасники отримують задоволення від процесу.

Успішна гейміфікація - це не просто зробити навчання цікавим. Вона підвищує якість навчання, дозволяючи учням взаємодіяти з контентом у змістовний спосіб. Впровадження гейміфікації в навчальне середовище може здатися складним завданням, але переваги підвищеної залученості, покращених результатів навчання та вищого рівня утримання учнів у навчальному процесі змушують замислитися над цим питанням.

1.3.Опис існуючих рішень

Гейміфікація як інструмент підвищення ефективності навчання набуває все більшої популярності завдяки своїй здатності залучати та мотивувати студентів через інтеграцію ігрових елементів у навчальний процес. Вибір способів гейміфікації залежить від багатьох факторів, таких як мета навчання, вік учнів, їхні інтереси та технічні можливості закладу освіти. Розглянемо основні способи гейміфікації, які можуть бути застосовані в освітньому процесі:

Бейджі та нагороди є одним із найпоширеніших способів гейміфікації, що використовуються для підвищення мотивації та залученості студентів у навчальний процес. Вони надаються за досягнення певних цілей або виконання завдань, стимулюючи студентів до подальших зусиль. Бейджі можуть бути віртуальними (електронними) або фізичними, залежно від контексту навчання. Їхня ефективність полягає в тому, що вони надають студентам відчуття досягнення та визнання.

Віртуальні бейджі зазвичай використовуються в онлайн-курсах та дистанційному навчанні. Вони можуть бути представлені у вигляді цифрових значків, сертифікатів або інших графічних елементів, що з'являються в особистих профілях студентів. Такі нагороди можна інтегрувати в системи

управління навчанням (LMS), що дозволяє автоматично відслідковувати прогрес студентів та надавати нагороди за досягнення певних етапів.

Приклади використання. На платформі Coursera студенти отримують віртуальні сертифікати та бейджі за успішне завершення курсів. Ці нагороди можна додати до профілю LinkedIn або використовувати як підтвердження набуття нових навичок.

Duolingo (рис. 2.1). мовний додаток Duolingo використовує систему бейджів для мотивації користувачів до регулярного навчання. За досягнення певних цілей, таких як завершення уроку або досягнення нової теми, користувачі отримують бейджі, що відображають їхній прогрес.

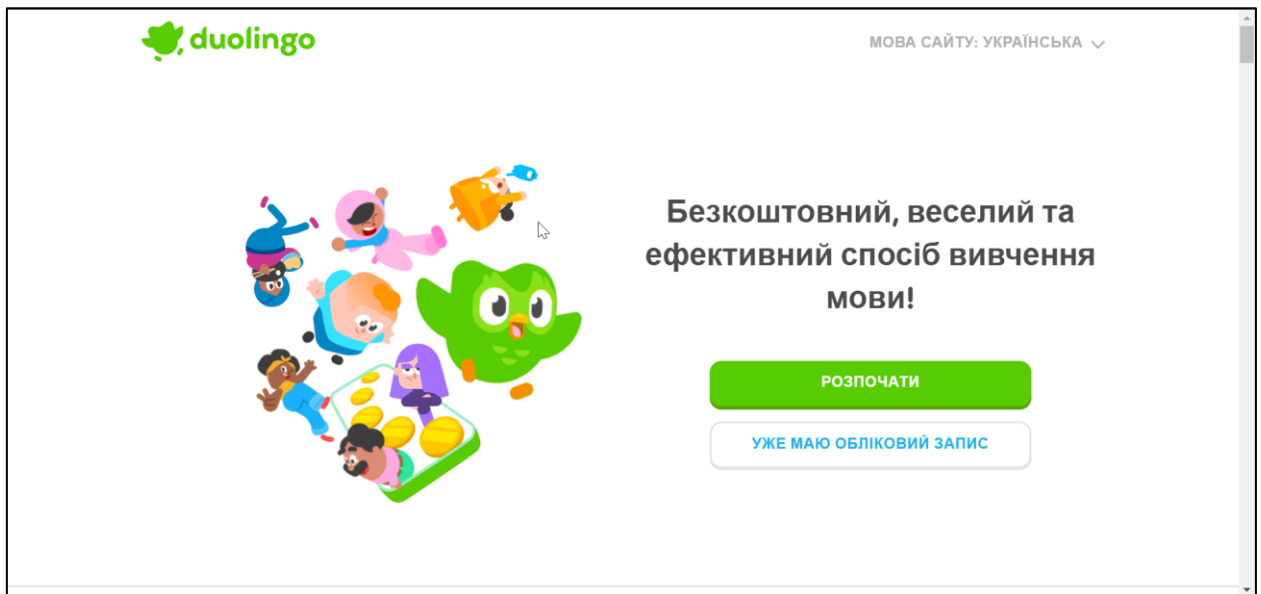


Рисунок 1.1 - Duolingo [6]

Фізичні бейджі та нагороди можуть бути використані в традиційних навчальних закладах для створення додаткового стимулу для студентів. Це можуть бути значки, медалі, дипломи або інші предмети, які студенти можуть отримати за досягнення високих результатів у навчанні, участь у конкурсах або інших заходах.

Приклади використання:

1) шкільні конкурси та олімпіади: у багатьох школах проводяться конкурси та олімпіади з різних предметів, за участь у яких студенти можуть отримати медалі, дипломи та інші нагороди. Це не тільки підвищує їхню мотивацію, але й дозволяє виявити талановитих студентів;

2) позакласні заходи: участь у спортивних змаганнях, художніх конкурсах або волонтерських проектах також може супроводжуватися нагородженням фізичними бейджами або медалями, що стимулює студентів до активної участі в житті школи чи університету.

Використання бейджів та нагород у навчальному процесі має декілька переваг:

1) підвищення мотивації: нагороди стимулюють студентів досягати нових цілей, підвищуючи їхню мотивацію до навчання;

2) визнання досягнень: студенти отримують визнання за свої зусилля, що сприяє підвищенню їхньої самооцінки та впевненості в собі;

3) візуалізація прогресу: бейджі та нагороди дозволяють студентам бачити свій прогрес, що сприяє підтриманню їхньої зацікавленості в навчанні;

4) соціальна складова: можливість поділитися своїми досягненнями з іншими, наприклад, через соціальні мережі, створює додатковий стимул для студентів.

Для успішного впровадження бейджів та нагород у навчальний процес необхідно врахувати декілька основних аспектів:

1) важливо визначити чіткі критерії для отримання нагород, щоб студенти знали, за які досягнення вони можуть отримати бейджі;

2) використання різних видів нагород для різних типів досягнень дозволяє підтримувати інтерес студентів та уникати монотонності;

3) бейджі та нагороди повинні бути інтегровані з навчальною програмою, щоб їх отримання було безпосередньо пов'язане з навчальними досягненнями;

4) надання зворотного зв'язку студентам щодо їхнього прогресу та досягнень є важливою складовою мотиваційного процесу.

Отже, використання бейджів та нагород у навчальному процесі є ефективним інструментом гейміфікації, який дозволяє підвищити мотивацію та залученість студентів, забезпечуючи при цьому визнання їхніх досягнень та візуалізацію прогресу.

Система балів дозволяє оцінювати активність та успішність студентів за допомогою нарахування балів за виконання завдань, участь у дискусіях, правильні відповіді на питання тощо. Цей спосіб гейміфікації стимулює змагання між студентами, підвищуючи їхню мотивацію до навчання. Крім того, система балів дозволяє легко відстежувати прогрес кожного студента.

Система балів ґрунтується на принципі нарахування балів за різні навчальні та позанавчальні активності. Кількість балів, які отримує студент, залежить від складності та важливості виконаних завдань або участі в заходах. Бали можуть бути накопичувальними, що дозволяє студентам досягати певних рівнів або статусів на основі їхньої активності та успіхів.

Основні елементи системи балів включають:

1) постановка завдань: завдання можуть включати домашні роботи, лабораторні роботи, участь у проєктах, активність на заняттях та в онлайн-дискусіях;

2) нарахування балів: студенти отримують бали за виконання завдань, правильні відповіді на питання, участь у групових обговореннях та інших активностях;

3) відстеження прогресу: система дозволяє відстежувати прогрес студентів, відображаючи їхні досягнення та успішність у вигляді кількості набраних балів;

4) нагородження: студенти можуть отримувати нагороди або переходити на нові рівні на основі кількості набраних балів.

Система балів сприяє підвищенню мотивації студентів шляхом створення конкурентного середовища та визнання їхніх досягнень. Основні аспекти впливу системи балів на мотивацію включають:

- 1) змагальний дух: студенти прагнуть набрати більше балів, ніж їхні однокурсники, що стимулює їх до активної участі в навчальному процесі;
- 2) визнання та винагороди: бали є формою визнання зусиль та досягнень студентів, що підвищує їхню самооцінку та мотивацію;
- 3) відстеження прогресу: можливість бачити свій прогрес та досягнення допомагає студентам ставити нові цілі та досягати їх.

Приклади використання системи балів у різних навчальних закладах та платформах:

- 1) Edmodo - це платформа для управління навчанням, яка дозволяє вчителям створювати завдання та оцінювати студентів за допомогою балів. Студенти можуть бачити свої досягнення та порівнювати їх із досягненнями однокурсників, що стимулює їх до активної участі в навчанні;

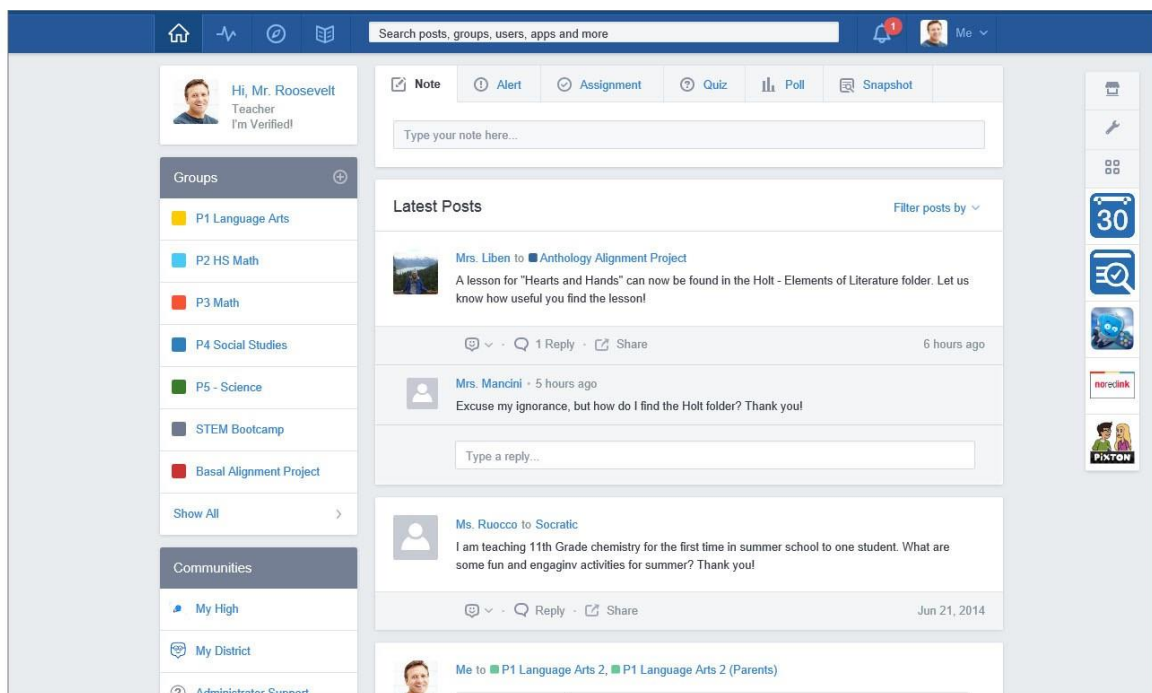


Рисунок 1.2 - Edmodo [7]

2) Classcraft використовує систему балів для створення гейміфікованого навчального середовища. Студенти отримують бали за виконання завдань, участь у групових обговореннях та допомогу однокурсникам. Накопичені бали можна використовувати для розблокування нових рівнів та отримання нагород;

3) Khan Academy використовує систему балів для мотивації студентів до вивчення нових тем та виконання завдань. Студенти отримують бали за проходження уроків та тестів, що дозволяє їм відстежувати свій прогрес та досягнення.

Використання системи балів у навчальному процесі має кілька переваг:

1) система балів стимулює студентів до активної участі в навчанні, оскільки вони прагнуть набрати більше балів та отримати визнання;

2) студенти можуть бачити свої досягнення у вигляді балів, що дозволяє їм відстежувати свій прогрес та ставити нові цілі;

3) система балів створює змагальне середовище, яке сприяє підвищенню мотивації студентів до навчання;

4) система балів може бути адаптована до різних видів діяльності та завдань, що дозволяє враховувати індивідуальні особливості студентів.

Для успішного впровадження системи балів у навчальний процес необхідно врахувати декілька основних аспектів:

1) важливо визначити чіткі критерії для нарахування балів, щоб студенти розуміли, за які завдання та активності вони можуть отримати бали;

2) система балів повинна бути прозорою та зрозумілою для студентів, щоб вони могли легко відстежувати свої досягнення та прогрес;

3) для підтримання інтересу студентів важливо використовувати різноманітні завдання та активності, за які нараховуються бали;

4) надання зворотного зв'язку студентам щодо їхніх досягнень та прогресу є важливою складовою мотиваційного процесу.

Отже, система балів є ефективним інструментом гейміфікації, який дозволяє оцінювати активність та успішність студентів, підвищувати їхню мотивацію та створювати змагальне середовище в навчальному процесі.

Розподіл навчального матеріалу на рівні з поступовим ускладненням завдань є ще одним ефективним способом гейміфікації. Перехід на новий рівень здійснюється після успішного завершення завдань попереднього рівня. Така структура сприяє поступовому засвоєнню знань та підтримує інтерес студентів, оскільки кожен новий рівень представляє собою новий виклик.

Система рівнів базується на принципі поступового ускладнення завдань, що дозволяє студентам покроково засвоювати нові знання та навички. Основні елементи системи рівнів включають:

1) початковий рівень: студенти починають з найпростіших завдань, які допомагають закріпити базові знання та навички;

2) проміжні рівні: кожен наступний рівень містить завдання більшої складності, які вимагають від студентів застосування раніше здобутих знань та навичок;

3) кінцевий рівень: найвищий рівень містить найскладніші завдання, які дозволяють оцінити повне засвоєння навчального матеріалу.

Система рівнів та прогресу сприяє підвищенню мотивації студентів шляхом надання їм відчуття досягнення та підтримки інтересу до навчання. Основні аспекти впливу системи рівнів на мотивацію включають:

1) відчуття досягнення - студенти отримують задоволення від успішного завершення рівня та переходу на новий, що стимулює їх до подальших зусиль;

2) поступове ускладнення - завдання поступово ускладнюються, що дозволяє студентам крок за кроком засвоювати нові знання та навички, не відчуваючи перевантаження;

3) нові виклики - кожен новий рівень представляє собою новий виклик, що підтримує інтерес студентів до навчання.

Наведемо приклади використання системи рівнів у різних навчальних платформах та програмах.

Khan Academy використовує систему рівнів для структурованого навчання. Студенти починають з базових тем і поступово переходять до більш складних. Прогрес відстежується, і студенти можуть бачити, які теми вони освоїли, а які ще потребують уваги.

На платформі Codecademy (рис. 2.3) студенти вивчають програмування, проходячи курси, розбиті на рівні. Кожен рівень містить завдання різної складності, які допомагають закріпити отримані знання та навички. Після завершення кожного рівня студенти отримують віртуальні нагороди та бачать свій прогрес.

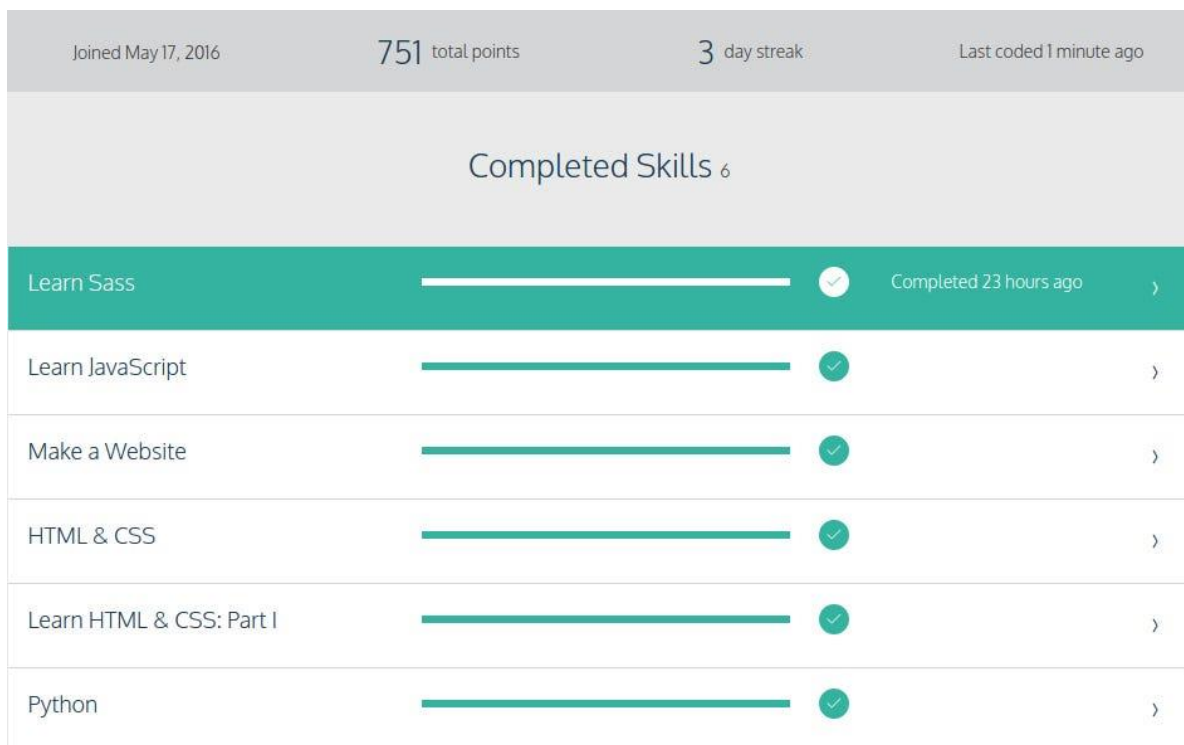


Рисунок 1.3 - Позначення прогресу в Codecademy [8]

Duolingo використовує систему рівнів для вивчення мов. Користувачі починають з найпростіших уроків і поступово переходять до більш складних, набираючи бали та досягаючи нових рівнів. Кожен рівень вимагає засвоєння певної кількості слів та граматичних правил.

Використання системи рівнів та прогресу в навчальному процесі має кілька переваг:

- 1) система рівнів дозволяє студентам крок за кроком засвоювати нові знання та навички, що сприяє глибшому розумінню матеріалу;
- 2) кожен новий рівень представляє собою новий виклик, що підтримує інтерес студентів до навчання та запобігає втраті мотивації;
- 3) студенти отримують задоволення від успішного завершення рівня та переходу на новий, що підвищує їхню самооцінку та впевненість у собі;
- 4) система рівнів дозволяє враховувати індивідуальні особливості та темпи навчання кожного студента, забезпечуючи оптимальні умови для засвоєння матеріалу.

Для успішного впровадження системи рівнів та прогресу у навчальний процес необхідно врахувати декілька основних аспектів:

- 1) чітка структура рівнів - важливо визначити чітку структуру рівнів та критерії для переходу на новий рівень, щоб студенти знали, які завдання їм потрібно виконати для досягнення наступного рівня;
- 2) різноманітність завдань - завдання на кожному рівні повинні бути різноманітними та поступово ускладнюватися, щоб підтримувати інтерес студентів та забезпечувати їхній поступовий розвиток;
- 3) відстеження прогресу - система повинна надавати можливість відстежувати прогрес студентів та надавати їм зворотний зв'язок щодо їхніх досягнень та успіхів;
- 4) мотиваційні елементи - використання нагород, бейджів та інших мотиваційних елементів на кожному рівні сприяє підвищенню мотивації студентів та їхньої залученості у навчальний процес.

Отже, система рівнів та прогресу є ефективним інструментом гейміфікації, який дозволяє структурувати навчальний матеріал, підтримувати інтерес студентів та забезпечувати їхнє поступове засвоєння знань та навичок.

Лідерборди використовуються для відображення рейтингу студентів на основі їхніх досягнень у навчанні. Це створює дух змагання та мотивує студентів покращувати свої результати, щоб піднятися на вищу позицію в рейтингу. Важливо, щоб таблиці лідерів оновлювалися в режимі реального часу та були доступні для перегляду всім учасникам навчального процесу.

Лідерборди базуються на принципі змагання, де студенти змагаються за кращі результати в різних завданнях та активностях. Основні елементи лідербордів включають:

1) збір даних - система збирає дані про досягнення студентів, такі як бали за виконання завдань, участь у проєктах, активність у класі тощо;

2) ранжування - студенти ранжуються на основі їхніх досягнень, що дозволяє створювати рейтинг найкращих студентів;

3) оновлення в реальному часі - лідерборди повинні оновлюватися в режимі реального часу, щоб відображати актуальні результати та підтримувати змагальний дух серед студентів;

4) доступність - таблиці лідерів повинні бути доступні для перегляду всім учасникам навчального процесу, включаючи студентів, викладачів та адміністрацію.

Лідерборди (рис. 3.4) мають значний вплив на мотивацію студентів, оскільки вони створюють змагальне середовище та стимулюють студентів до покращення своїх результатів.

RANKING	TEAM	MATCHES	KILL POINTS	PLAYER POINTS	TOTAL POINTS	WIN
01	Player Name	05	20	17	09	03
02	Player Name	05	20	17	09	03
03	Player Name	05	20	17	09	03
04	Player Name	05	20	17	09	03
05	Player Name	05	20	17	09	03

Рисунок 1.4 - Приклад лідерборду [9]

Основні аспекти впливу лідербордів на мотивацію включають:

1) змагальний дух - студенти прагнуть досягти кращих результатів, щоб піднятися на вищу позицію в рейтингу, що стимулює їх до активної участі в навчальному процесі;

2) визнання досягнень - високі позиції в таблицях лідерів служать формою визнання зусиль та досягнень студентів, що підвищує їхню самооцінку та мотивацію;

3) підтримка інтересу - постійна можливість покращувати свої результати та бачити свої досягнення в порівнянні з іншими студентами підтримує інтерес до навчання.

Наведемо приклади використання лідербордів у різних навчальних платформах та програмах. Khan Academy використовує лідерборди для відображення прогресу та досягнень студентів. Студенти можуть бачити свої результати в порівнянні з іншими учнями, що стимулює їх до покращення своїх показників;

Classcraft використовує лідерборди для створення змагального навчального середовища. Студенти отримують бали за виконання завдань та участь у проєктах, а їхні результати відображаються в таблицях лідерів. Це стимулює їх до активної участі та покращення своїх досягнень;

Duolingo використовує лідерборди для мотивації користувачів до регулярного вивчення мов. Користувачі можуть бачити свої результати в порівнянні з іншими учнями, що стимулює їх до покращення своїх показників та досягнення нових цілей.

Використання лідербордів у навчальному процесі має кілька переваг:

1) лідерборди стимулюють студентів до активної участі в навчанні, оскільки вони прагнуть досягти кращих результатів та піднятися на вищу позицію в рейтингу;

2) таблиці лідерів служать формою визнання зусиль та досягнень студентів, що підвищує їхню самооцінку та мотивацію;

3) лідерборди створюють змагальне середовище, яке сприяє підвищенню мотивації студентів до навчання;

4) студенти можуть бачити свої досягнення в порівнянні з іншими учнями, що дозволяє їм відстежувати свій прогрес та ставити нові цілі.

Для успішного впровадження лідербордів у навчальний процес необхідно врахувати декілька основних аспектів:

1) чіткі критерії оцінювання - важливо визначити чіткі критерії для ранжування студентів, щоб вони розуміли, за які досягнення отримують бали та як формується їхній рейтинг;

2) оновлення в реальному часі - лідерборди повинні оновлюватися в режимі реального часу, щоб відображати актуальні результати та підтримувати змагальний дух серед студентів;

3) доступність - таблиці лідерів повинні бути доступні для перегляду всім учасникам навчального процесу, включаючи студентів, викладачів та адміністрацію;

4) позитивне підкріплення - важливо забезпечити позитивне підкріплення для студентів, які досягають високих результатів, щоб підтримувати їхню мотивацію та заохочувати до подальших успіхів.

Отже, лідерборди є ефективним інструментом гейміфікації, який дозволяє створити змагальне середовище, підвищити мотивацію студентів та забезпечити визнання їхніх досягнень.

Квести та місії є інтерактивними завданнями, які об'єднують навчальний матеріал у захоплюючі історії або пригоди. Виконуючи квести, студенти отримують не лише знання, але й задоволення від процесу навчання. Квести можуть включати різноманітні завдання, такі як розгадування загадок, вирішення логічних задач, проходження тестів тощо.

Квести та місії базуються на принципі інтерактивності та залучення студентів до навчального процесу через цікаві та захоплюючі завдання. Основні елементи квестів та місій включають:

1) сюжетна лінія - кожен квест або місія має захоплюючу сюжетну лінію, яка пов'язує навчальний матеріал з інтерактивними завданнями;

2) різноманітність завдань - квести включають різноманітні завдання, такі як розгадування загадок, вирішення логічних задач, проходження тестів, виконання практичних завдань тощо;

3) етапи та рівні - квести можуть бути розбиті на кілька етапів або рівнів, кожен з яких представляє собою новий виклик та сприяє поступовому засвоєнню знань;

4) нагороди та мотиваційні елементи - за успішне виконання завдань студенти отримують нагороди, такі як бейджі, бали, підвищення рівня тощо.

Квести та місії мають значний вплив на мотивацію студентів, оскільки вони роблять процес навчання цікавим та захоплюючим. Основні аспекти впливу квестів та місій на мотивацію включають:

1) захоплюючий процес навчання - сюжетні лінії та цікаві завдання роблять навчання цікавим та захоплюючим, що сприяє підвищенню мотивації студентів;

2) інтерактивність - квести вимагають активної участі студентів, що сприяє кращому засвоєнню знань та розвитку навичок;

3) відчуття досягнення - успішне виконання завдань та отримання нагород сприяє підвищенню самооцінки студентів та підтримує їхню мотивацію до подальших зусиль.

Приклади використання квестів та місій у різних навчальних платформах та програмах. Classcraft використовує квести та місії для створення захоплюючого навчального процесу. Студенти проходять через серію завдань та викликів, об'єднаних сюжетною лінією, що сприяє інтерактивному засвоєнню навчального матеріалу.

Minecraft: Education Edition (рис. 2.5): у цій платформі вчителі можуть створювати навчальні квести та місії, використовуючи можливості гри Minecraft. Студенти виконують завдання у вигляді гри, що робить процес навчання цікавим та захоплюючим.



Рисунок 1.5 - Minecraft: Education Edition [10]

Kahoot! дозволяє створювати інтерактивні вікторини та квести, які можна використовувати для перевірки знань студентів та підвищення їхньої мотивації до навчання.

Використання квестів та місій у навчальному процесі має кілька переваг:

- 1) підвищення мотивації - квести та місії роблять процес навчання цікавим та захоплюючим, що сприяє підвищенню мотивації студентів;
- 2) інтерактивне навчання - квести вимагають активної участі студентів, що сприяє кращому засвоєнню знань та розвитку навичок;
- 3) поступове засвоєння знань - квести розбиті на кілька етапів або рівнів, що дозволяє студентам поступово засвоювати нові знання та навички;
- 4) відчуття досягнення: успішне виконання завдань та отримання нагород підвищує самооцінку студентів та підтримує їхню мотивацію до подальших зусиль.

Для успішного впровадження квестів та місій у навчальний процес необхідно врахувати декілька основних аспектів:

- 1) цікава сюжетна лінія - важливо створити захоплюючу сюжетну лінію, яка буде інтегрована з навчальним матеріалом та завданнями;
- 2) різноманітність завдань - квести повинні включати різноманітні завдання, які сприяють розвитку різних навичок та засвоєнню різноманітного навчального матеріалу;
- 3) етапи та рівні - квести повинні бути розбиті на кілька етапів або рівнів, щоб студенти могли поступово засвоювати нові знання та навички;
- 4) нагороди та мотиваційні елементи - використання нагород та інших мотиваційних елементів сприяє підвищенню мотивації студентів та підтримує їхню зацікавленість у навчанні.

Отже, квести та місії є ефективним інструментом гейміфікації, який дозволяє інтегрувати навчальний матеріал у захоплюючі історії та завдання, підвищуючи мотивацію студентів та сприяючи кращому засвоєнню знань та навичок.

Ігрові симуляції дозволяють студентам застосовувати теоретичні знання на практиці в ігровому контексті. Це можуть бути бізнес-симуляції, симуляції управління проектами, симуляції соціальних ситуацій тощо. Ігрові симуляції сприяють розвитку го мислення, вміння приймати рішення та працювати в команді.

Ігрові симуляції базуються на створенні реалістичних сценаріїв, які дозволяють студентам зануритися у різні ситуації та застосовувати свої знання і навички для вирішення проблем. Основні елементи ігрових симуляцій включають:

- 1) симуляції створюються на основі реальних ситуацій, які студенти можуть зустріти у професійному житті;
- 2) студенти беруть активну участь у симуляціях, приймаючи рішення та виконуючи завдання у відповідь на змінні умови;
- 3) після завершення симуляції студенти отримують зворотний зв'язок щодо своїх дій та рішень, що дозволяє їм аналізувати свої помилки та покращувати свої навички;
- 4) ігрові симуляції можуть бути адаптовані для різних навчальних цілей та рівнів складності.

Ігрові симуляції мають значний вплив на навчання, оскільки вони дозволяють студентам застосовувати теоретичні знання на практиці та розвивати важливі професійні навички. Основні аспекти впливу ігрових симуляцій на навчання включають:

- 1) симуляції дозволяють студентам застосовувати теоретичні знання у реалістичних ситуаціях, що сприяє кращому засвоєнню матеріалу;
- 2) участь у симуляціях вимагає від студентів аналізу ситуацій, прийняття рішень та розв'язання проблем, що сприяє розвитку го мислення;
- 3) симуляції надають студентам можливість приймати рішення у різних умовах, що сприяє розвитку навичок прийняття рішень;

4) багато симуляцій вимагають від студентів роботи у командах, що сприяє розвитку комунікативних та командних навичок.

Приклади використання ігрових симуляцій у різних навчальних платформах та програмах. SimCityEDU: SimCityEDU - це освітня версія популярної гри SimCity, яка використовується для навчання студентів основам управління містом, планування та вирішення соціальних та економічних проблем. Студенти використовують свої знання для створення та управління віртуальним містом, що допомагає їм зрозуміти складність та взаємозв'язок різних аспектів міського життя.

Capsim пропонує бізнес-симуляції для навчання управлінню бізнесом та стратегічного планування. Студенти приймають роль менеджерів компаній та приймають рішення щодо виробництва, маркетингу, фінансів та інших аспектів бізнесу. Це дозволяє їм застосовувати теоретичні знання на практиці та розвивати управлінські навички.

Labster (рис. 2.6) пропонує віртуальні лабораторні симуляції для навчання студентів природничих наук. Студенти можуть проводити експерименти у віртуальних лабораторіях, використовуючи реалістичні інструменти та методи. Це дозволяє їм засвоювати складні концепції та розвивати навички лабораторної роботи.



Рисунок 1.6 - Labster [11]

The Beer Game - це симуляція ланцюга постачання, розроблена для навчання студентів основам управління ланцюгами постачання та логістики. Учасники приймають ролі виробників, дистриб'юторів та роздрібних продавців, приймаючи рішення щодо замовлень та запасів. Це допомагає студентам зрозуміти складність ланцюгів постачання та важливість координації між різними ланками.

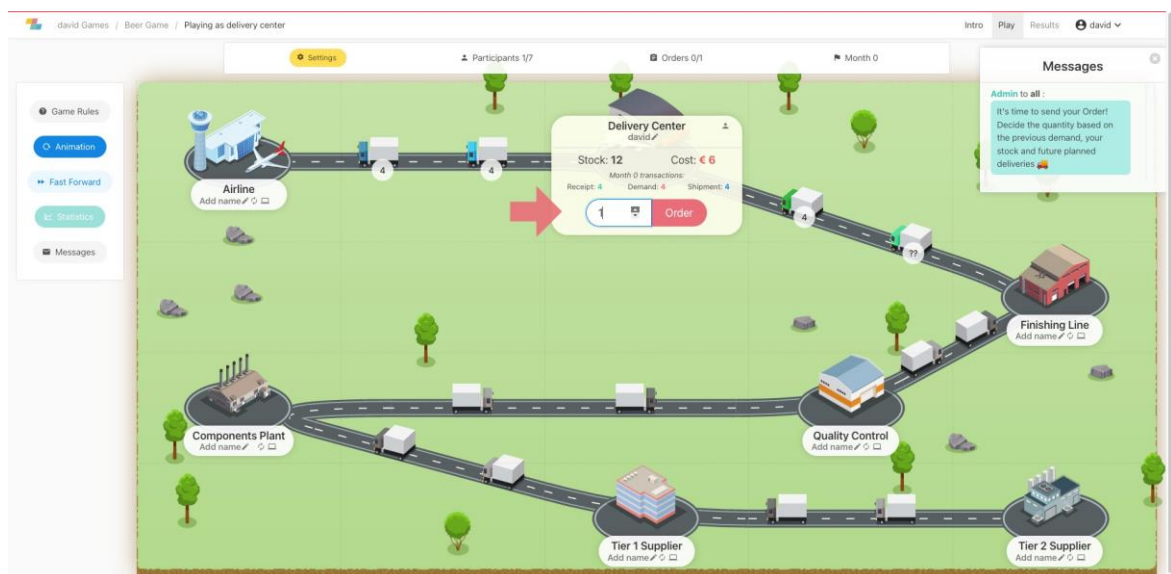


Рисунок 1.7 - The Beer Game [12]

Використання ігрових симуляцій у навчальному процесі має кілька переваг:

- 1) ігрові симуляції дозволяють студентам застосовувати теоретичні знання на практиці, що сприяє кращому засвоєнню матеріалу;
- 2) симуляції вимагають від студентів аналізу ситуацій, прийняття рішень та розв'язання проблем, що сприяє розвитку го мислення;
- 3) симуляції надають студентам можливість приймати рішення у різних умовах, що сприяє розвитку навичок прийняття рішень;
- 4) багато симуляцій вимагають від студентів роботи у командах, що сприяє розвитку комунікативних та командних навичок;
- 5) симуляції створюють безпечне середовище, де студенти можуть експериментувати з різними стратегіями та приймати рішення без ризику реальних наслідків.

Для успішного впровадження ігрових симуляцій у навчальний процес необхідно врахувати декілька основних аспектів:

- 1) реалістичність сценаріїв - важливо створити реалістичні сценарії, які будуть інтегровані з навчальним матеріалом та завданнями;
- 2) інтерактивність - симуляції повинні бути інтерактивними, щоб студенти могли активно брати участь та приймати рішення у відповідь на змінні умови;
- 3) зворотний зв'язок - після завершення симуляції важливо надавати студентам зворотний зв'язок щодо їхніх дій та рішень, щоб вони могли аналізувати свої помилки та покращувати свої навички;
- 4) гнучкість - симуляції повинні бути гнучкими та адаптованими для різних навчальних цілей та рівнів складності;
- 5) підтримка та навчання викладачів - важливо забезпечити підтримку та навчання викладачів для ефективного використання ігрових симуляцій у навчальному процесі.

Отже, ігрові симуляції є ефективним інструментом гейміфікації, який дозволяє студентам застосовувати теоретичні знання на практиці, розвивати критичне мислення, навички прийняття рішень та командної роботи.

Використання автономних ігор та мобільних додатків для навчання дозволяє студентам займатися вивченням матеріалу у зручній для них час та у зручному місці. Такі додатки можуть включати різноманітні завдання, вікторини, тренувальні вправи та інші ігрові елементи. Мобільні додатки також можуть забезпечувати інтерактивний зворотний зв'язок, що сприяє кращому засвоєнню знань.

Автономні ігри та мобільні додатки базуються на принципах інтерактивності, доступності та гнучкості навчання. Основні елементи автономних ігор та мобільних додатків включають:

1) додатки містять інтерактивні завдання, вікторини, тренувальні вправи та інші ігрові елементи, що дозволяють студентам активно брати участь у навчальному процесі;

2) мобільні додатки доступні на різних платформах (iOS, Android), що дозволяє студентам використовувати їх у будь-який час та у зручному для них місці;

3) студенти можуть займатися вивченням матеріалу у зручному для них темпі, повертатися до складних тем та повторювати матеріал стільки разів, скільки потрібно;

4) мобільні додатки забезпечують інтерактивний зворотний зв'язок, що дозволяє студентам аналізувати свої помилки та покращувати свої знання.

Автономні ігри та мобільні додатки мають значний вплив на навчання, оскільки вони забезпечують гнучкість, доступність та інтерактивність навчального процесу. Основні аспекти впливу автономних ігор та мобільних додатків на навчання включають:

1) підвищення мотивації - інтерактивні завдання та ігрові елементи роблять навчання цікавим та захоплюючим, що сприяє підвищенню мотивації студентів;

2) гнучкість навчання - студенти можуть займатися у зручній для них час та у зручному місці, що дозволяє їм ефективніше планувати свій час та займатися навчанням у своєму темпі;

3) покращення засвоєння знань - інтерактивний зворотний зв'язок дозволяє студентам аналізувати свої помилки та покращувати свої знання;

4) індивідуалізація навчання - мобільні додатки можуть адаптуватися до індивідуальних потреб та рівня знань кожного студента, що забезпечує оптимальні умови для засвоєння матеріалу.

Приклади використання автономних ігор та мобільних додатків у навчальному процесі. Duolingo - популярний мобільний додаток для вивчення мов, який використовує ігрові елементи, такі як вікторини, завдання на запам'ятовування слів та граматики. Додаток надає інтерактивний зворотний зв'язок, що дозволяє студентам відразу бачити свої помилки та покращувати свої знання;

Khan Academy пропонує мобільний додаток, який дозволяє студентам вивчати різні предмети, включаючи математику, науку, історію та інші. Додаток містить інтерактивні відеоуроки, завдання та вікторини, що сприяє кращому засвоєнню знань;

Quizlet - мобільний додаток, що дозволяє студентам створювати та використовувати флеш-карти для запам'ятовування інформації. Додаток також містить інтерактивні вікторини та ігри, що робить процес навчання цікавим та захоплюючим;

Brilliant пропонує мобільний додаток для вивчення математики та природничих наук через інтерактивні завдання та курси. Додаток використовує ігрові елементи та надає зворотний зв'язок, що сприяє глибокому розумінню матеріалу.

Використання автономних ігор та мобільних додатків у навчальному процесі має кілька переваг:

1) студенти можуть займатися у зручний для них час та у зручному місці, що дозволяє їм ефективніше планувати свій час та займатися навчанням у своєму темпі;

2) мобільні додатки містять інтерактивні завдання та ігрові елементи, що робить навчання цікавим та захоплюючим;

3) додатки забезпечують інтерактивний зворотний зв'язок, що дозволяє студентам аналізувати свої помилки та покращувати свої знання;

4) мобільні додатки доступні на різних платформах, що дозволяє студентам використовувати їх у будь-який час та у зручному для них місці;

5) індивідуалізація навчання: додатки можуть адаптуватися до індивідуальних потреб та рівня знань кожного студента, що забезпечує оптимальні умови для засвоєння матеріалу.

Для успішного впровадження автономних ігор та мобільних додатків у навчальний процес необхідно врахувати декілька основних аспектів:

1) важливо забезпечити інтерактивність завдань та ігрових елементів, щоб студенти могли активно брати участь у навчальному процесі;

2) мобільні додатки повинні забезпечувати інтерактивний зворотний зв'язок, щоб студенти могли аналізувати свої помилки та покращувати свої знання;

3) додатки повинні бути доступні на різних платформах та мати зручний інтерфейс, щоб студенти могли легко використовувати їх у будь-який час та у зручному для них місці;

4) додатки повинні адаптуватися до індивідуальних потреб та рівня знань кожного студента, що забезпечить оптимальні умови для засвоєння матеріалу;

5) важливо забезпечити підтримку та навчання викладачів для ефективного використання автономних ігор та мобільних додатків у навчальному процесі.

Отже, автономні ігри та мобільні додатки є ефективним інструментом гейміфікації, який дозволяє студентам займатися вивченням матеріалу у зручній для них час та у зручному місці, забезпечуючи інтерактивний зворотний зв'язок та індивідуалізацію навчання.

Вибір способів гейміфікації залежить від конкретних цілей навчання та особливостей студентів. Використання бейджів, системи балів, рівнів, лідербордів, квестів, ігрових симуляцій та мобільних додатків дозволяє значно підвищити мотивацію та залученість студентів до навчального процесу. Важливо також враховувати технічні можливості та ресурси навчального закладу при впровадженні гейміфікації.

1.4. Аналіз існуючих програм для навчання програмування

Програмування є однією з найпопулярніших та затребуваних навичок у сучасному світі. Існує багато програм та платформ, які допомагають студентам вивчати програмування. Далі розглянемо деякі з них, аналізуючи їхні переваги та недоліки.

Codecademy - це інтерактивна платформа для навчання програмуванню, яка пропонує курси з різних мов програмування, включаючи Python, JavaScript, HTML/CSS, Ruby та інші. Платформа використовує інтерактивні завдання, які студенти можуть виконувати прямо в браузері [13].

Coursera - це онлайн-платформа, яка пропонує курси, спеціалізації та ступені від провідних університетів та компаній. Платформа пропонує курси з програмування на різних мовах, таких як Python, Java, C++, та інші [14].

freeCodeCamp - це некомерційна платформа для навчання програмуванню, яка пропонує безкоштовні курси з різних мов програмування та технологій, таких як HTML/CSS, JavaScript, Python та інші. Платформа

також надає можливість працювати над реальними проектами для неприбуткових організацій [15].

Udacity - це онлайн-платформа, яка пропонує курси та нанодегрі (nanodegrees) з різних напрямів, включаючи програмування, аналіз даних, штучний інтелект та інші. Платформа співпрацює з провідними компаніями для створення навчальних програм [16].

edX - це онлайн-платформа для навчання, створена Гарвардським університетом та Массачусетським технологічним інститутом. Платформа пропонує курси з програмування та інших предметів від провідних університетів та компаній [17].

SoloLearn - це мобільний додаток та платформа для навчання програмуванню, яка пропонує курси з різних мов програмування, таких як Python, JavaScript, C++, Java та інші. Платформа використовує інтерактивні завдання та вікторини для навчання [18].

HackerRank - це платформа для практики програмування, яка пропонує задачі та змагання з різних мов програмування. Платформа використовується також для підготовки до технічних співбесід [19].

Таблиця 1.1 - Аналіз аналогічних рішень

Рішення	Переваги	Недоліки
Codecademy	студенти можуть виконувати завдання безпосередньо на платформі, що забезпечує практичне засвоєння матеріалу	деякі курси не охоплюють складніші теми та концепції програмування, що може бути недостатнім для більш просунутих студентів
	платформа пропонує курси з багатьох мов програмування та технологій, що дозволяє студентам вибрати найбільш цікаві для них напрями	деякі курси та можливості доступні лише за підпискою, що може бути обмеженням для деяких користувачів
	студенти отримують миттєвий зворотний зв'язок щодо своїх рішень, що допомагає швидко виправляти помилки	

Продовження таблиці 1.1

Coursera	курси створюються провідними університетами та компаніями, що забезпечує високу якість навчального матеріалу;	багато курсів та програм є платними, що може бути обмеженням для деяких студентів;
	студенти можуть отримувати сертифікати про завершення курсів, що може бути корисним для кар'єрного зростання	онлайн-курси вимагають високої самоорганізації та дисципліни від студентів, що може бути складним для деяких людей
	платформа пропонує курси, спеціалізації та ступені, що дозволяє студентам вибрати оптимальний формат навчання	
freeCodeCamp	всі курси та матеріали на платформі є безкоштовними, що робить її доступною для всіх бажаючих	платформа не надає офіційних сертифікатів про завершення курсів, що може бути недоліком для тих, хто хоче підтвердити свої навички
	студенти мають можливість працювати над реальними проектами, що допомагає їм отримати практичний досвід	хоча платформа пропонує багато курсів, їхня кількість може бути меншою в порівнянні з іншими платформами
	платформа має активну спільноту, яка допомагає студентам у вирішенні проблем та надає підтримку	
Udacity	курси створюються у співпраці з провідними компаніями, що забезпечує актуальність та високу якість матеріалу;	курси та нанодегрі є платними, що може бути обмеженням для деяких студентів;
	курси включають багато практичних завдань та проектів, що допомагає студентам застосовувати отримані знання на практиці	курси вимагають значного часу та зусиль для успішного завершення, що може бути складним для зайнятих людей
	платформа надає кар'єрну підтримку, включаючи	

	допомогу у створенні резюме та підготовці до співбесід	
--	--	--

Продовження таблиці 1.1

edX	курси створюються провідними університетами та компаніями, що забезпечує високу якість навчального матеріалу;	багато курсів та програм є платними, що може бути обмеженням для деяких студентів;
	платформа пропонує курси з різних мов програмування та технологій, що дозволяє студентам вибрати найбільш цікаві для них напрями	онлайн-курси вимагають високої самоорганізації та дисципліни від студентів, що може бути складним для деяких людей
	студенти можуть отримувати сертифікати про завершення курсів, що може бути корисним для кар'єрного зростання	
SoloLearn	додаток доступний на мобільних пристроях, що дозволяє студентам вчитися у будь-який час та у зручному місці	деякі курси не охоплюють складніші теми та концепції програмування, що може бути недостатнім для більш просунутих студентів
	студенти можуть виконувати завдання та вікторини, що сприяє кращому засвоєнню матеріалу	платформа не надає офіційних сертифікатів про завершення курсів, що може бути недоліком для тих, хто хоче підтвердити свої навички
	платформа має активну спільноту, де студенти можуть обговорювати питання та допомагати один одному	
HackerRank	платформа пропонує багато практичних завдань та змагань, що допомагає студентам покращити свої навички програмування	платформа більше орієнтована на практичні завдання, ніж на теоретичне навчання, що може бути недоліком для новачків;
	HackerRank пропонує задачі, що часто зустрічаються на технічних співбесідах, що дозволяє студентам підготуватися до них	на відміну від інших платформ, HackerRank не пропонує структурованих курсів, що може ускладнити процес навчання для деяких студентів
	платформа має активну спільноту, де студенти можуть	

	обговорювати рішення задач та обмінюватися досвідом	
--	---	--

Загалом, існуючі програми для навчання програмування пропонують різні підходи та формати навчання, що дозволяє студентам вибрати найбільш зручний та ефективний для них спосіб засвоєння матеріалу. Вибір платформи залежить від багатьох факторів, таких як рівень знань студента, його цілі та переваги у навчанні. Кожна з розглянутих платформ має свої переваги та недоліки, які варто враховувати при виборі оптимального варіанту навчання.

1.5. Постановка задачі для створення гейміфікованого ЕОІС

Електронне навчання - це ігрова дисципліна, яка занурює користувача в навчальний процес. Сама форма навчання, що використовує елементи середовища вищої освіти, вже містить елементи гри.

Її цільовою аудиторією є спільнота студентів, організована за групами, потоками та спеціалізаціями.

Крім того, вже існують правила оцінювання успішності студентів, реєстрації проміжних результатів (еквівалентом є модульні та екзаменаційні відомості) та отримання основного результату - сертифікату про завершення навчання.

Впровадження акредитованих програм передбачає отримання сертифікату як проміжного результату.

У процесі навчання викладачі оцінюють результати навчання студентів за допомогою балів.

Викладачі також можуть отримувати зворотній зв'язок від студентів у вигляді смайликів за результатами оцінювання студентами електронних навчальних ресурсів з дисципліни, за результатами анонімних опитувань студентів та за відгуками на форумах з дисципліни. Додаткові ігрові елементи можуть бути введені як на рівні окремих дисциплін, так і на рівні ЕНК.

У загальній моделі гейміфікації модель октету уточнюється відповідно до контексту освітнього процесу та посилює використання результатів діяльності студентів в іграх разом з викладачем при оцінюванні результатів навчання.

Ця модель використовує автоматизовану систему обліку діяльності студентів в системі JetIQ у формі змішаного та дистанційного навчання.

Механіка такої моделі полягає у наданні різних нагород (статусів, значків та медалей) за різні форми активності та досягнення у навчанні. Нагороди відображаються в журналі активності студента та використовуються викладачем у сценаріях оцінювання навчання.

Проаналізувавши проблеми впровадження модулів гейміфікації на прикладах відомих світових додатків та навчальних платформ, ми вдосконалимо модулі гейміфікації та моніторингу активності студентів у системі JetIQ з метою визначити основні виклики:

- 1) сформулювати загальну модель гейміфікованого середовища електронного навчання, використовуючи теорії гейміфікації, гейміфікації навчання та мотивації;
- 2) удосконалити моделі заробляння балів, значків та медалей для зміни статусу за результатами діяльності студентів в EOIC;
- 3) розробити модель гейміфікації на основі моделі Octalysis;
- 4) розробити модель використання результатів гейміфікації для оцінювання студентів;
- 5) розробити та вдосконалити модулі гейміфікації для використання в EOIC;
- 6) розробити сценарії використання результатів гейміфікації для оцінювання знань та навичок студентів.

Загалом, вивчаючи різноманітні приклади гейміфікації та моделі гейміфікації, можна зробити висновок, що позитивний процес гейміфікації

може мотивувати навчання та створювати інтерактивне інформаційне освітнє середовище.

Вивчення різних прикладів гейміфікації та моделей гейміфікації дозволяє зробити висновок, що активні процеси гейміфікації можуть мотивувати навчання і створювати інтерактивне інформаційне освітнє середовище.

Інтерактивна природа ігор сприяє розвитку взаємодії між гравцями, викладачами, різними програмними агентами та ігровими персонажами. Міждисциплінарність знань, тренд в освіті та бізнесі, також активно розвивається в процесі гейміфікації.

Він об'єднує різні дисципліни та розвиває навички поєднання та збалансування знань з різних дисциплін та сфер, таких як наука, технології та соціальні науки. Наприклад, всі навчальні ігри мають елемент оцінювання гравців.

Ділові ігри, засновані на професійних дисциплінах, дуже часто визначають ролі гравців і характер аудиторії, з якою студенти будуть взаємодіяти в майбутньому.

Наприклад, гра "Замовник і розробник" вимагає від студентів визначення вимог до програмного продукту, знання предметної області та систематизації отриманих результатів.

Важливою метою впровадження гейміфікації є залучення учасників гри до цілеспрямованої діяльності, яка дозволяє досягти конкретних результатів навчання на основі гри, залучити користувачів, мотивувати їх до навчання та створити бажання розвиватися в запропонованому напрямку, та/або зосередитися на розвитку спеціальних навичок, таких як сфокусуватися на розвитку спеціальних навичок. Наприклад, стати більш уважним, динамічним, розвинути критичне мислення, навчитися користуватися технологіями тощо.

2. РОЗДІЛ 2. Алгоритмічне та математичне пояснення

2.1. Проєктування архітектури системи

Використання сучасних інструментів та технологій у розробці додатка для підвищення якості роботи програмістів дозволяє забезпечити високу продуктивність, надійність та зручність використання. Основні засоби розробки, включаючи мови програмування, рамки та бібліотеки, системи керування базами даних, інструменти для розробки та управління проектом, інструменти для тестування та контролю якості, а також інструменти для управління проектами та співпраці, забезпечують всебічний підхід до створення ефективного та якісного додатка.

Для нашого проекту ми обрали мову програмування C#, Windows Forms для створення користувацького інтерфейсу, а також середовище розробки Visual Studio. Такий вибір зумовлений кількома основними факторами, які забезпечують високу продуктивність, зручність розробки та ефективність у реалізації поставлених задач.

C# є сучасною об'єктно-орієнтованою мовою програмування, яка широко використовується у розробці різноманітних додатків, включаючи десктопні, веб- та мобільні додатки. Вона надає потужні засоби для створення надійного та масштабованого програмного забезпечення, що відповідає високим стандартам якості. Завдяки своїй синтаксичній схожості з іншими популярними мовами програмування, такими як Java та C++, C# є легкою для вивчення та освоєння, що забезпечує швидкий старт у розробці проекту [20].

Windows Forms обрано як технологію для створення користувацького інтерфейсу нашого додатка. Цей фреймворк є частиною .NET Framework і дозволяє швидко та ефективно створювати інтерактивні та зручні для користувача інтерфейси. Windows Forms надає великий набір готових елементів управління, таких як кнопки, текстові поля, таблиці, діаграми та інші компоненти, що дозволяє значно скоротити час розробки. Крім того,

Windows Forms добре інтегрується з іншими технологіями .NET, що забезпечує безшовну взаємодію між різними частинами додатка [21].

Visual Studio обрано як основне середовище розробки завдяки його потужним можливостям та інструментам, які значно спрощують процес розробки програмного забезпечення (рис. 2.1).

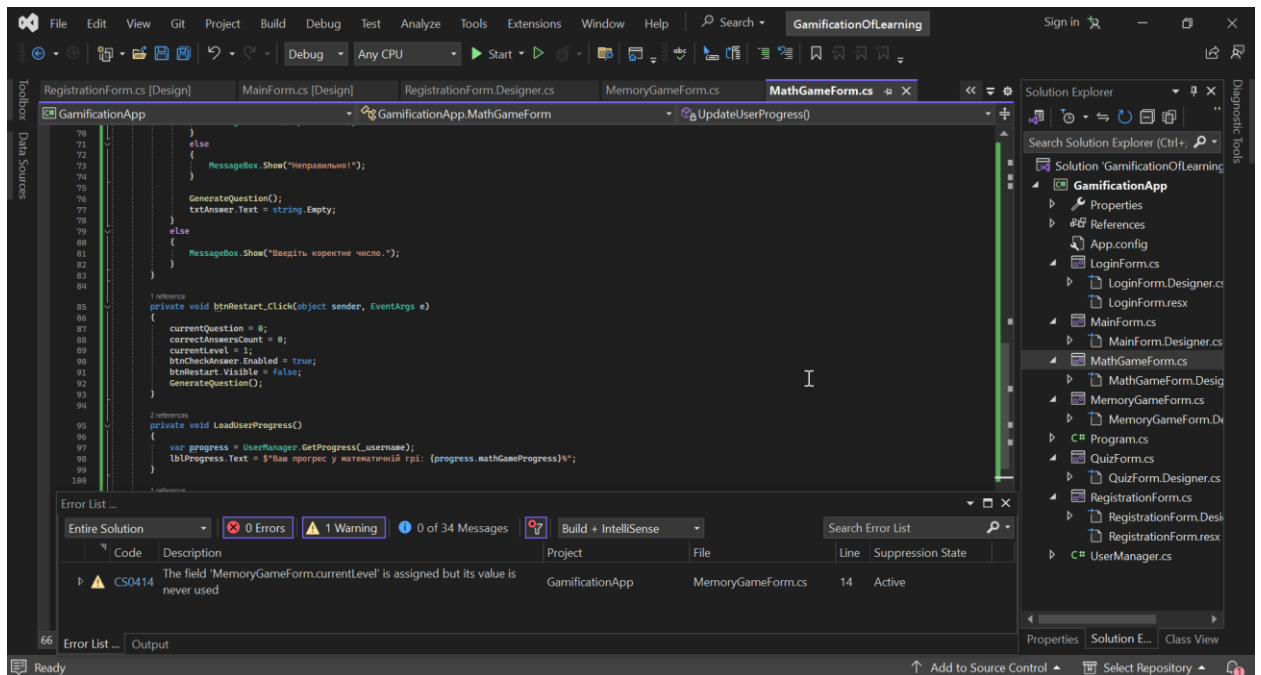


Рисунок 2.1 - Інтерфейс розробки в Visual Studio

Visual Studio надає зручний та інтуїтивно зрозумілий інтерфейс, що дозволяє розробникам швидко орієнтуватися у проекті та ефективно працювати з кодом. Однією з основних переваг Visual Studio є його інтеграція з системами контролю версій, такими як Git, що забезпечує зручне управління кодом та спільну роботу над проектом. Крім того, Visual Studio включає потужні засоби для відлагодження, тестування та аналізу коду, що дозволяє забезпечити високу якість програмного забезпечення на всіх етапах розробки [22].

Таким чином, вибір C#, Windows Forms та Visual Studio для нашого проекту обґрунтований їхніми потужними можливостями, зручністю

використання та ефективністю у реалізації задач. Це дозволяє забезпечити високу продуктивність розробки, надійність та масштабованість додатка, а також зручність для кінцевих користувачів.

Опис архітектури системи

Архітектура нашої системи для підвищення якості роботи програмістів розроблена з урахуванням принципів модульності, масштабованості та підтримки. Основні компоненти системи розглянемо у вигляді діаграми класів (рис. 1.2).

Опис діаграми класів:

1) User - клас представляє користувача системи. Містить інформацію про користувача, таку як ідентифікатор, ім'я користувача, пароль та електронну пошту. Містить метод Authenticate, який перевіряє правильність імені користувача та пароля;

2) Course - клас представляє курс, який містить модулі. Містить інформацію про курс, таку як ідентифікатор, назва та опис. Містить метод AddModule, який додає модуль до курсу;

3) Module - клас представляє модуль, який містить завдання. Містить інформацію про модуль, таку як ідентифікатор, назва та зміст. Містить метод AddTask, який додає завдання до модуля;

4) Task - клас представляє завдання, яке оцінюється за допомогою методу Evaluate. Містить інформацію про завдання, таку як ідентифікатор, опис та кількість балів;

5) Progress - клас представляє прогрес користувача в системі. Містить інформацію про користувача, курс, модуль, завдання та отримані бали. Містить метод UpdateProgress, який оновлює прогрес користувача.

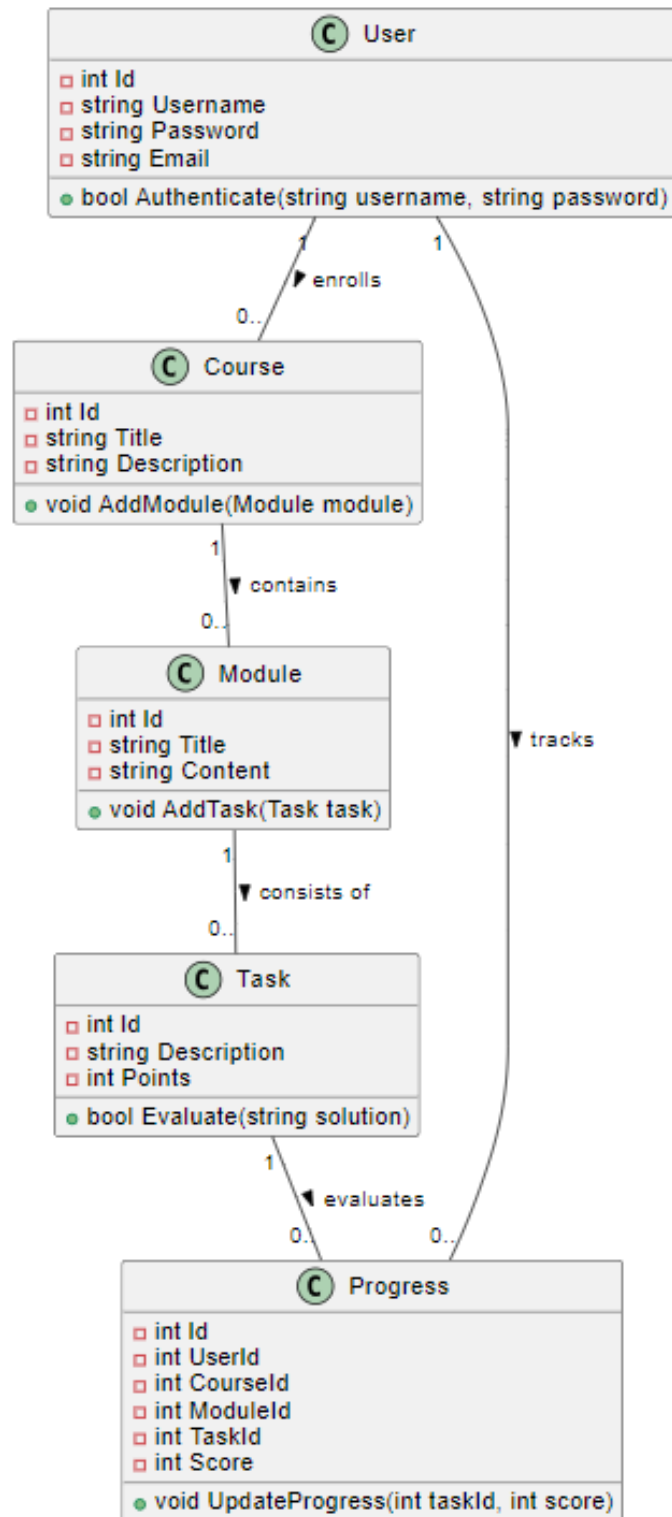


Рисунок 2.2 - Діаграма класів

Також розробимо діаграму послідовності (рис. 2.3)

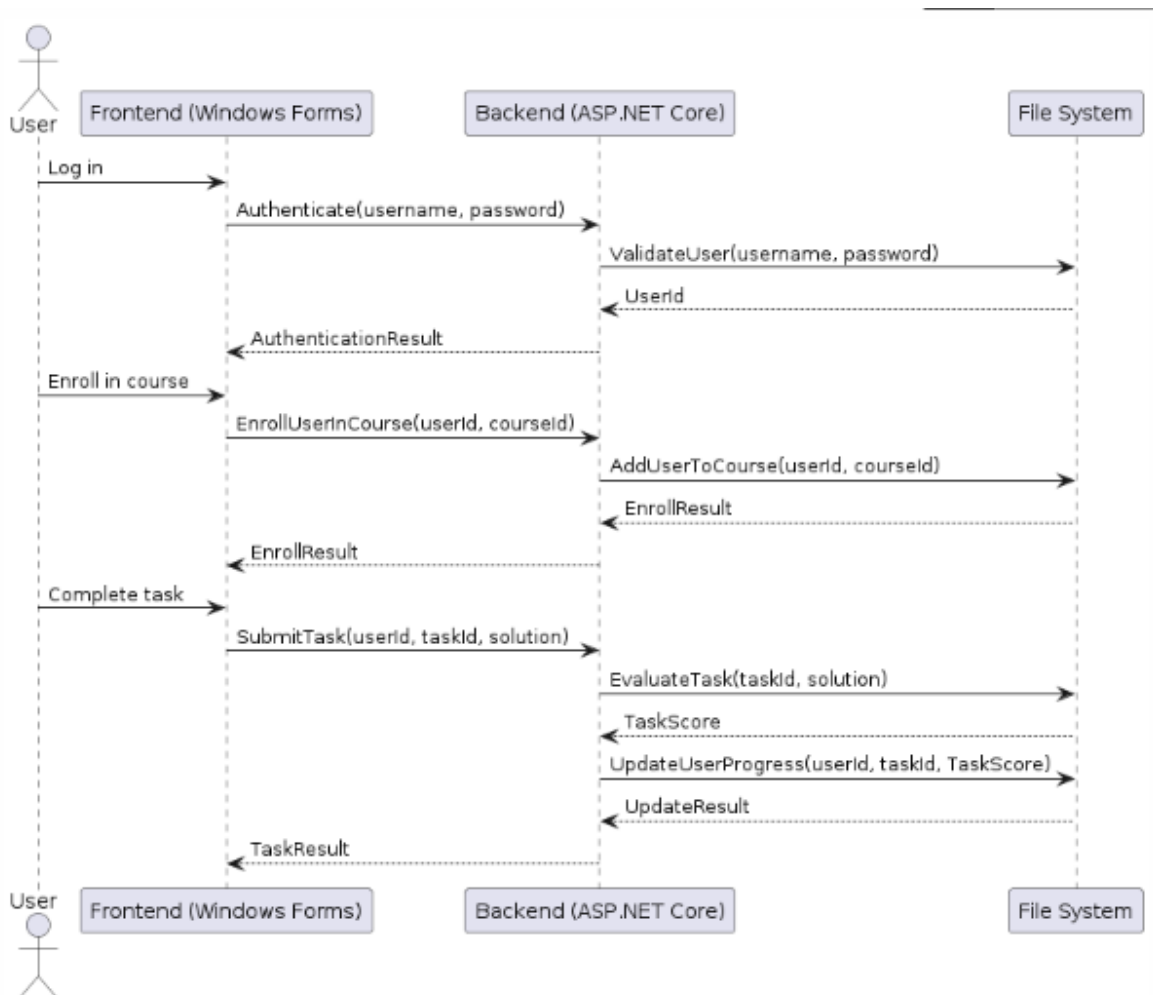


Рисунок 2.3 - Діаграма послідовності

А також діаграму активності (рис. 2.4) Діаграма активності ілюструє процес використання додатка "GamificationApp" від моменту його запуску користувачем до завершення гри та оновлення прогресу користувача. Процес починається з того, що користувач запускає додаток, після чого відображається форма входу. Користувач вводить свої логін та пароль. Якщо автентифікація успішна, відображається головне меню.

У головному меню користувач вибирає гру, яку хоче грати, і гра запускається. Під час гри користувач виконує завдання, а результати завдання

відправляються на сервер для перевірки. Сервер перевіряє результати і зберігає їх у файлі. Після цього гра показує результати і оновлює прогрес користувача.

Якщо автентифікація не успішна, користувачу відображається повідомлення про помилку, і він має можливість ввести логін та пароль повторно. Цей цикл триває до тих пір, поки користувач не введе правильні дані для входу.

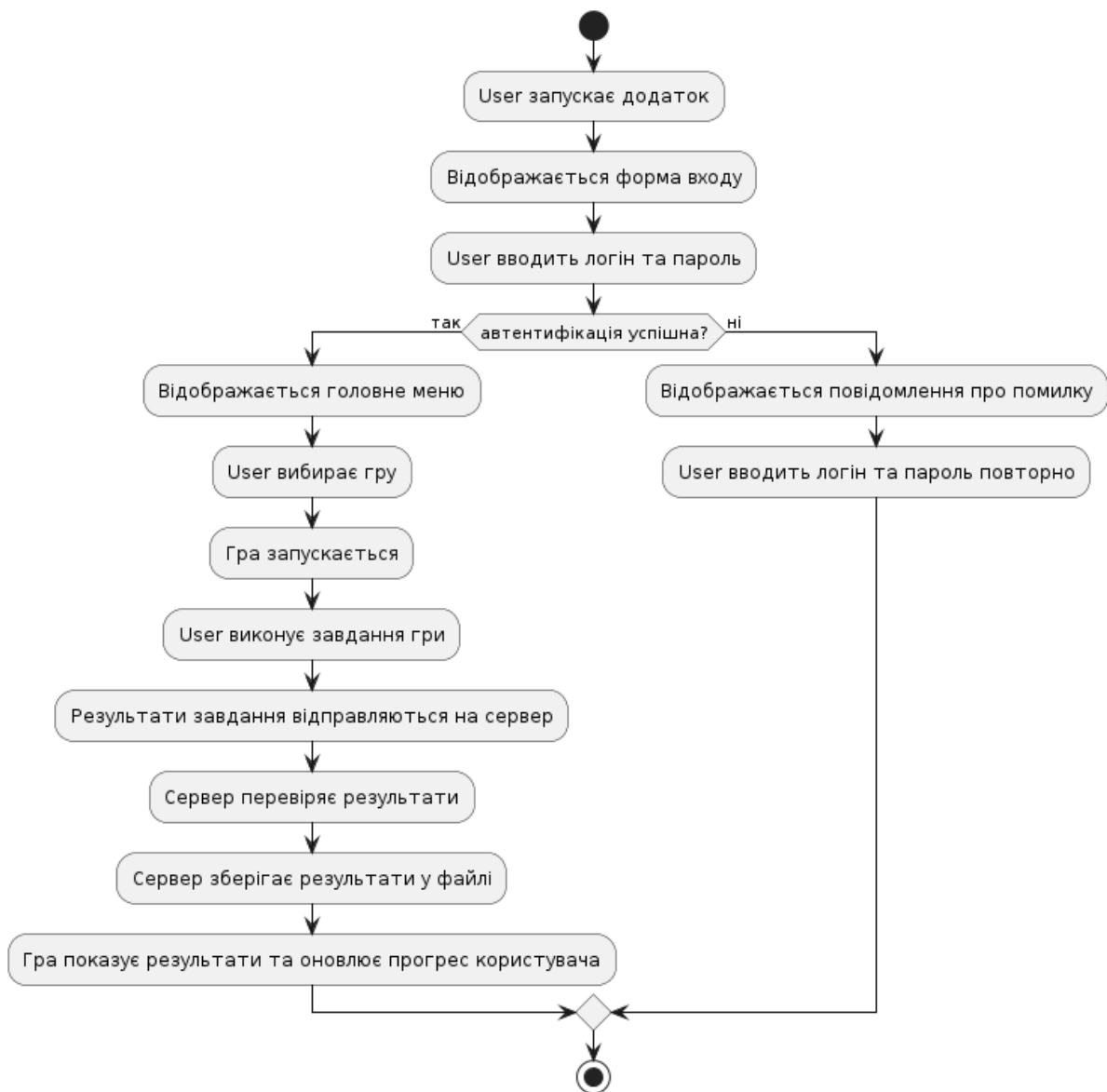


Рисунок 2.4 - Діаграма активності

Архітектура системи забезпечує модульність та масштабованість, дозволяючи легко додавати нові функції та розширювати систему у майбутньому. Використання сучасних інструментів та технологій забезпечує високу продуктивність та надійність системи.

2.2.Бібліотеки та фреймворки для розробки додатку

Для розробки гейміфікованої системи навчання було використано декілька ключових технологій, бібліотек та фреймворків, що забезпечують ефективність, надійність та зручність у розробці та подальшій експлуатації програмного забезпечення.

Однією з основних технологій, використаних у розробці, є Windows Forms, що є частиною .NET Framework. Windows Forms дозволяє швидко та ефективно створювати інтерактивні та зручні для користувача інтерфейси. Цей фреймворк надає великий набір готових елементів управління, таких як кнопки, текстові поля, таблиці, діаграми та інші компоненти, що дозволяє значно скоротити час розробки. Завдяки великій кількості вбудованих компонентів, розробник може легко створювати складні інтерфейси з мінімальними зусиллями. Крім того, Windows Forms добре інтегрується з іншими технологіями .NET, що забезпечує безшовну взаємодію між різними частинами додатку.

Для реалізації серверної частини програми було використано ASP.NET Core, який забезпечує надійне управління даними та високу масштабованість. ASP.NET Core є кросплатформовим фреймворком з відкритим кодом для побудови сучасних хмарних додатків. Використання цього фреймворку дозволяє забезпечити високий рівень продуктивності та безпеки, що є критично важливим для систем, які обробляють значні обсяги даних та

потребують швидкого відгуку. ASP.NET Core підтримує модульну архітектуру, що полегшує розширення функціональності системи та її підтримку. Завдяки цьому фреймворку, можна легко впроваджувати нові функції та оптимізувати наявні, що робить систему гнучкою та адаптивною до змін.

Основною мовою програмування, використаною для розробки, є C#. Це сучасна об'єктно-орієнтована мова програмування, яка широко використовується для створення різноманітних додатків, включаючи десктопні, веб- та мобільні додатки. C# надає потужні засоби для створення надійного та масштабованого програмного забезпечення, що відповідає високим стандартам якості. Завдяки своїй синтаксичній схожості з іншими популярними мовами програмування, такими як Java та C++, C# є легкою для вивчення та освоєння, що забезпечує швидкий старт у розробці проекту. Використання C# також забезпечує високу продуктивність та безпеку коду, що є важливим аспектом для розробки надійного програмного забезпечення.

Для роботи з базою даних було обрано Entity Framework – ORM (Object-Relational Mapping) фреймворк, який значно спрощує взаємодію з базою даних. Entity Framework дозволяє працювати з даними у вигляді об'єктів, що значно полегшує процес розробки та забезпечує високу продуктивність роботи з даними. Цей фреймворк підтримує автоматичне створення бази даних, що дозволяє уникнути помилок під час її налаштування та забезпечує зручне управління даними. Завдяки Entity Framework, можна легко здійснювати складні запити до бази даних, забезпечуючи при цьому високу продуктивність та надійність.

Основним середовищем розробки було обрано Visual Studio завдяки його потужним можливостям та інструментам, які значно спрощують процес

розробки програмного забезпечення. Visual Studio надає зручний та інтуїтивно зрозумілий інтерфейс, що дозволяє розробникам швидко орієнтуватися у проекті та ефективно працювати з кодом. Однією з основних переваг Visual Studio є його інтеграція з системами контролю версій, такими як Git, що забезпечує зручне управління кодом та спільну роботу над проектом. Крім того, Visual Studio включає потужні засоби для відлагодження, тестування та аналізу коду, що дозволяє забезпечити високу якість програмного забезпечення на всіх етапах розробки. Вбудовані інструменти для роботи з базами даних, веб-розробки та розробки мобільних додатків роблять Visual Studio універсальним рішенням для розробників.

Використання цих бібліотек та фреймворків забезпечує ефективну реалізацію всіх необхідних функцій додатку, забезпечуючи при цьому гнучкість та масштабованість системи. Завдяки сучасним технологіям розробки, система може легко адаптуватися до нових вимог та впроваджувати нові функціональні можливості, забезпечуючи високу продуктивність та зручність для кінцевих користувачів. Інтеграція різних технологій дозволяє створювати комплексні рішення, що відповідають високим стандартам якості та задовольняють потреби користувачів.

Таким чином, вибір інструментів та технологій для розробки гейміфікованої системи навчання забезпечує її надійність, масштабованість та високу продуктивність. Це дозволяє створювати сучасні та зручні для користувачів додатки, які сприяють ефективному навчанню та підвищенню мотивації студентів.

2.3. Моделювання ігрових механік

Моделювання та оптимізація алгоритмів для нарахування балів

Математична формула для нарахування балів:

Для кожного завдання, виконаного студентом, можна нараховувати бали на основі складності завдання, часу виконання та точності відповіді. Формула для нарахування балів може виглядати так:

$$B = C * \left(1 - \frac{T}{T_{max}}\right) * A$$

де:

B — кількість нарахованих балів,

C — коефіцієнт складності завдання,

T — фактичний час виконання завдання,

T_{max} — максимальний дозволений час для виконання завдання,

A — точність відповіді (від 0 до 1).

Алгоритми для розподілу рівнів:

Для розподілу рівнів можна використовувати накопичені бали. Наприклад, студенти можуть переходити на новий рівень після досягнення певної кількості балів. Можна застосувати наступний алгоритм:

1. Визначити рівні та порогові значення балів для кожного рівня.
2. При кожному оновленні балів студента перевіряти, чи досягнуто нового рівня.
3. Якщо досягнуто, оновлювати рівень студента та надавати відповідні бонуси або доступ до нових функцій.

Алгоритм для оцінки прогресу:

Прогрес можна оцінювати за допомогою формули, яка враховує загальну кількість виконаних завдань, їхню складність та успішність виконання:

$$P = \frac{\sum_{i=1}^n B_i}{\sum_{i=1}^n C_i}$$

де:

P — прогрес студента,

n — кількість виконаних завдань,

B_i — бали за i -те завдання,

C_i — складність i -го завдання.

Використання теорії ігор для аналізу ефективності гейміфікаційних елементів

Стратегічні ігри та рівновага Неша:

Можна моделювати навчальний процес як стратегічну гру, де кожен студент є гравцем, а його стратегії — це вибір завдань та способи їх виконання. Метою є максимізація отриманих балів при мінімізації затраченого часу.

Гравці: студенти.

Стратегії: вибір завдань, рівень участі, витрачений час.

Виграш: нараховані бали та досягнення.

Застосування рівноваги Неша дозволяє визначити, які стратегії будуть оптимальними для студентів у різних ситуаціях.

Ігрові дерева для моделювання рішень:

Ігрові дерева можна використовувати для моделювання прийняття рішень студентами. Кожен вузол дерева представляє вибір завдання або дію, а гілки — можливі результати та нараховані бали.

Використання теорії ймовірностей та Марковських процесів

Ймовірність успіху завдання:

Можна використовувати теорію ймовірностей для моделювання ймовірності успіху виконання завдання залежно від складності та рівня підготовки студента.

$$P(\text{успіх}) = \frac{S}{C}$$

де:

$P(\text{успіх})$ — ймовірність успішного виконання завдання,

S — рівень підготовки студента,

C — складність завдання.

Марковські процеси для моделювання прогресу: Марковські процеси можна застосовувати для моделювання прогресу студента, де станами є рівні або етапи навчання, а переходи між станами залежать від ймовірності успішного виконання завдань.

Матриця перехідних ймовірностей:

$$\begin{bmatrix} P_{11} & P_{12} & \dots & P_{1n} \\ P_{21} & P_{22} & \dots & P_{2n} \\ \dots & \dots & \dots & \dots \\ P_{n1} & P_{n2} & \dots & P_{nn} \end{bmatrix}$$

де:

P_{ij} — ймовірність переходу зі стану i в стан j .

Оцінка ефективності гейміфікаційних елементів

Аналіз результатів:

Для оцінки ефективності гейміфікації можна використовувати методи статистичного аналізу. Наприклад, можна порівняти середній бал, час виконання завдань та інші показники до і після впровадження гейміфікації.

Регресійний аналіз:

Використання регресійного аналізу дозволяє визначити вплив різних гейміфікаційних елементів на успішність студентів. Модель регресії може

включати такі змінні, як кількість виконаних завдань, середній бал, кількість отриманих нагород тощо.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n + \varepsilon$$

де:

Y — успішність студента,

X_i — змінні, що описують різні гейміфікаційні елементи,

β_i — коефіцієнти регресії,

ε — випадкова похибка.

3. РОЗДІЛ 3. Розробка програмного забезпечення

3.1. Методологія розробки

Вибір методології

Для розробки гейміфікованої системи навчання була обрана методологія Agile. Цей гнучкий підхід до управління проектами дозволяє швидко адаптуватися до змін і забезпечує високу якість кінцевого продукту. Agile методологія включає ітеративний процес розробки, що дає змогу регулярно отримувати робочий продукт на кожному етапі проекту.

Agile (принцип методології, обґрунтування вибору)

Методологія Agile базується на принципах гнучкого управління проектами, які включають співпрацю, орієнтацію на кінцевого користувача, ітеративний підхід до розробки та постійне вдосконалення. Проект розбивається на невеликі цикли, що називаються спринтами, кожен з яких завершується створенням робочого продукту. Це дозволяє постійно отримувати зворотний зв'язок і вносити необхідні корективи. Такий підхід забезпечує гнучкість у розробці, дозволяючи швидко реагувати на змінювані вимоги та впроваджувати нові функції за потребою.

Кожен спринт завершується ретроспективою, де аналізується виконана робота та визначаються шляхи для покращення процесів і продукту. Цей процес сприяє поступовому вдосконаленню якості проекту, що є важливим аспектом для забезпечення високого рівня користувацького задоволення. Постійне вдосконалення є невід'ємною частиною Agile методології, що дозволяє виявляти та виправляти помилки на ранніх стадіях розробки, тим самим забезпечуючи якісний кінцевий продукт.

Гнучкість методології Agile дозволяє швидко реагувати на змінювані вимоги та впроваджувати нові функції за потребою. Це особливо важливо для

розробки гейміфікованих систем, де вимоги можуть змінюватися в залежності від зворотного зв'язку користувачів. Гейміфіковані системи потребують постійного оновлення та вдосконалення, щоб залишатися актуальними та цікавими для користувачів. Agile підхід дозволяє адаптувати проект до нових викликів та можливостей, забезпечуючи при цьому інтерактивність та залученість користувачів.

Обрання Agile методології для розробки гейміфікованої системи навчання було обґрунтоване необхідністю гнучкого підходу до управління проектом. Це дозволяє швидко реагувати на змінювані вимоги та забезпечує високий рівень якості кінцевого продукту. Agile підхід також сприяє залученню всіх зацікавлених сторін у процес розробки, що підвищує загальну ефективність проекту та задоволеність користувачів.

Використання Agile методології забезпечує ефективне управління часом та ресурсами, дозволяючи зосередитися на найважливіших завданнях, поступово вдосконалюючи проект та впроваджуючи нові функціональні можливості на основі отриманого зворотного зв'язку від тестування. Agile дозволяє створювати високоякісний продукт, який відповідає потребам кінцевих користувачів та легко адаптується до нових вимог. Завдяки цьому підходу можна досягти високої якості програмного забезпечення, що сприятиме ефективному навчанню та задоволенню користувачів.

3.2. Загальні положення

Розробка програмного забезпечення гейміфікованої системи навчання включає декілька ключових аспектів, що визначають загальну стратегію та підхід до створення ефективного та надійного продукту. У цьому розділі розглядаються основні етапи розробки, використані методології та технології, а також загальні принципи, що лежать в основі проекту.

Розробка програмного забезпечення починається з детального аналізу вимог та визначення основних функціональних можливостей системи. На цьому етапі важливо чітко розуміти, які завдання має вирішувати система, які функції повинні бути реалізовані, і як ці функції будуть використовуватися кінцевими користувачами. На основі цього аналізу формується технічне завдання, яке включає в себе всі необхідні вимоги до системи та визначає обсяг роботи.

Далі починається етап реалізації програмного забезпечення. На цьому етапі використовуються обрані мови програмування, фреймворки та бібліотеки для написання коду системи. Реалізація включає розробку користувацького інтерфейсу, серверної частини, бази даних та інших компонентів системи. Важливо забезпечити високу якість коду, дотримання стандартів програмування та використання найкращих практик розробки.

Одним із ключових аспектів розробки є тестування програмного забезпечення. Тестування включає перевірку функціональності системи, виявлення та виправлення помилок, а також забезпечення відповідності системи вимогам технічного завдання. Тестування може включати як ручне, так і автоматизоване тестування, а також використання різних типів тестів, таких як юніт-тести, інтеграційні тести, системні тести та інші.

Завершальним етапом розробки є впровадження та підтримка системи. Це включає розгортання системи на робочих серверах, забезпечення її безперебійної роботи, а також надання підтримки користувачам та виправлення виявлених проблем. Підтримка також включає внесення змін та оновлень до системи, що можуть бути необхідними для вдосконалення функціональності або виправлення помилок.

3.3. Вибір програмних засобів

Вибір програмних засобів для розробки гейміфікованої системи навчання був ретельним процесом, що включав аналіз численних інструментів та технологій з урахуванням специфічних вимог проекту. Кожен із обраних засобів мав забезпечити високу продуктивність, надійність, безпеку та зручність використання.

3.3.1. Обґрунтування вибору C#

Мова програмування C# була обрана завдяки її потужності та гнучкості, що дозволяє створювати складні додатки з високими вимогами до продуктивності. C# підтримує об'єктно-орієнтовану парадигму, що сприяє модульності та масштабованості коду. Крім того, інтеграція з .NET Framework надає доступ до великої кількості бібліотек та інструментів, що полегшує розробку та зменшує час на впровадження нових функцій.

3.3.2. Використання Windows Forms для створення інтерфейсу

Windows Forms був обраний для розробки користувацького інтерфейсу завдяки своїй простоті та ефективності. Цей фреймворк дозволяє швидко створювати зручні та інтуїтивно зрозумілі інтерфейси, що є важливим для забезпечення гарного користувацького досвіду. Windows Forms також забезпечує тісну інтеграцію з іншими компонентами .NET, що дозволяє легко взаємодіяти з серверною частиною та базою даних.

3.3.3. Переваги ASP.NET Core для серверної частини

ASP.NET Core був обраний для розробки серверної частини додатку через його високу продуктивність та кросплатформеність. Цей фреймворк дозволяє створювати надійні та масштабовані веб-додатки, що можуть

працювати на різних операційних системах. Модульна архітектура ASP.NET Core спрощує розширення функціональності та підтримку додатку, що є критично важливим для довгострокового успіху проекту.

3.3.4. Використання Entity Framework для роботи з даними

Entity Framework був обраний для управління базою даних завдяки своїй здатності значно спрощувати взаємодію з даними. ORM (Object-Relational Mapping) підхід дозволяє працювати з даними як з об'єктами, що полегшує розробку та зменшує кількість помилок. Entity Framework також забезпечує високий рівень продуктивності та надійності при виконанні складних запитів до бази даних.

3.3.5. Інтеграція з Visual Studio

Visual Studio був обраний як основне середовище розробки завдяки своїм потужним можливостям та інтеграції з іншими інструментами. Visual Studio надає зручні інструменти для написання, відлагодження та тестування коду, що дозволяє значно скоротити час розробки та підвищити якість програмного забезпечення. Крім того, інтеграція з Git забезпечує зручне управління версіями коду та спільну роботу над проектом.

3.3.6. Використання Git для управління версіями

Git був обраний для управління версіями коду завдяки своїй надійності та широкому розповсюдженню. Ця система контролю версій дозволяє ефективно відстежувати зміни у коді, забезпечуючи можливість повернення до попередніх версій у разі необхідності. Крім того, Git забезпечує зручне спільне редагування коду, що полегшує командну роботу.

Таким чином, вибір програмних засобів був обґрунтований специфічними вимогами проекту та необхідністю забезпечення високої продуктивності, надійності та зручності використання. Використання сучасних інструментів та технологій дозволяє створювати ефективні та надійні рішення, що відповідають високим стандартам якості та задовольняють потреби користувачів.

3.4. Реалізація програмного забезпечення

Розробка програмного забезпечення для гейміфікованої системи навчання була розділена на кілька ключових етапів, які включають проектування архітектури, реалізацію користувацького інтерфейсу, серверної частини та бази даних, а також тестування і впровадження системи. Кожен з цих етапів має свої особливості та вимоги, що забезпечують ефективність та надійність кінцевого продукту.

3.4.1. Проектування архітектури

Архітектура системи була розроблена з урахуванням принципів модульності, масштабованості та підтримки. Основні компоненти системи включають користувацький інтерфейс, серверну частину, базу даних та модулі взаємодії. Користувацький інтерфейс реалізований за допомогою Windows Forms, що забезпечує зручність використання та високу швидкість розробки. Серверна частина реалізована на основі ASP.NET Core, що забезпечує високу продуктивність та безпеку. База даних управляється за допомогою Entity Framework, що полегшує роботу з даними та забезпечує їх цілісність.

3.4.2. Реалізація користувацького інтерфейсу

Користувацький інтерфейс був реалізований за допомогою Windows Forms, що дозволяє швидко створювати зручні та інтуїтивно зрозумілі інтерфейси. Основні елементи інтерфейсу включають форми для реєстрації та входу користувачів, головне меню, інтерфейс для вибору та запуску ігор, а також форми для відображення результатів та прогресу користувачів. Кожна форма була розроблена з урахуванням принципів ергономіки та зручності використання, що забезпечує приємний користувацький досвід.

3.4.3. Реалізація серверної частини

Серверна частина додатку була реалізована за допомогою ASP.NET Core, що забезпечує надійне управління даними та високу масштабованість системи. Сервер обробляє запити від клієнтської частини, забезпечуючи автентифікацію користувачів, обробку результатів ігор, збереження та оновлення прогресу користувачів. Завдяки використанню ASP.NET Core, серверна частина може легко розширюватися та адаптуватися до нових вимог, що забезпечує довгострокову підтримку та розвиток системи.

3.4.4. Реалізація бази даних

База даних була реалізована за допомогою Entity Framework, що дозволяє працювати з даними у вигляді об'єктів. Це значно спрощує процес розробки та забезпечує високу продуктивність роботи з даними. Основні таблиці бази даних включають таблиці користувачів, ігор, результатів і прогресу. Entity Framework забезпечує автоматичне створення та оновлення схеми бази даних, що полегшує її адміністрування та підтримку.

3.4.5. Тестування та впровадження системи

Після завершення основних етапів розробки було проведено ретельне тестування системи. Тестування включало перевірку функціональності користувацького інтерфейсу, серверної частини та бази даних. Було проведено як ручне, так і автоматизоване тестування, що дозволило виявити та виправити всі знайдені помилки. Після успішного завершення тестування система була впроваджена на робочі сервери та надана для використання кінцевим користувачам.

Таким чином, реалізація програмного забезпечення гейміфікованої системи навчання включала кілька ключових етапів, кожен з яких був спрямований на забезпечення високої якості, надійності та ефективності кінцевого продукту. Використання сучасних технологій та інструментів дозволило створити систему, що відповідає високим стандартам якості та задовольняє потреби користувачів.

3.5. Опис функціональних можливостей

Гейміфікована система навчання розроблена з метою підвищення мотивації та залученості користувачів через інтерактивні елементи. Основні функціональні можливості та інтерфейс були розроблені з урахуванням потреб користувачів та вимог до сучасних навчальних платформ.

Інтерактивні ігри

Система включає декілька інтерактивних ігор, кожна з яких розроблена з метою розвитку різних навичок та знань у користувачів.

1. Вікторина

Вікторина представляє собою серію запитань з варіантами відповідей. Користувач обирає правильний, на його думку, варіант. Після вибору відповіді

система негайно надає зворотний зв'язок про правильність відповіді. Запитання можуть бути з різних галузей знань, що дозволяє адаптувати вікторину до конкретних навчальних цілей.

2. Гра на пам'ять

Гра на пам'ять складається з відображення випадкових чисел від 1 до 4 у певному порядку. З кожним рівнем кількість чисел збільшується, що ускладнює завдання. Користувач має запам'ятати порядок чисел і повторити його. Ця гра допомагає покращити когнітивні здібності користувача, зокрема пам'ять та концентрацію.

3. Математична гра

Математична гра пропонує користувачам вирішувати серію математичних задач. Задачі можуть бути різного рівня складності, включаючи базові арифметичні операції, алгебраїчні рівняння та логічні задачі. Користувач повинен вирішити кожну задачу за обмежений час. Після завершення кожної задачі система негайно надає зворотний зв'язок про правильність рішення.

Відстеження прогресу

Система дозволяє користувачам відстежувати свій прогрес у виконанні завдань та досягненнях. Кожна гра зберігає результати користувача, що дозволяє бачити їхній розвиток та досягнення у реальному часі. Це сприяє підвищенню мотивації та ефективності навчання, оскільки користувачі можуть бачити свої успіхи та працювати над покращенням результатів.

Автоматична оцінка та зворотний зв'язок

Програма включає механізм автоматичної оцінки результатів завдань та надання зворотного зв'язку. Це дозволяє користувачам негайно дізнаватися про свої помилки та виправляти їх, що значно підвищує ефективність навчання.

Автоматична оцінка забезпечує об'єктивність та швидкість процесу, що є важливим аспектом ефективного навчання. Зворотний зв'язок включає інформацію про правильність відповіді.

Реєстрація та вхід користувача

Система передбачає процес реєстрації та входу користувача для забезпечення персоналізованого досвіду та збереження індивідуального прогресу. Під час реєстрації користувач вказує своє ім'я та пароль. Після успішної реєстрації користувач може входити в систему, використовуючи свої облікові дані. Це дозволяє системі зберігати та відстежувати прогрес користувача, а також забезпечувати доступ до персональних налаштувань та досягнень.

Таблиця лідерів

Для стимулювання конкуренції та підвищення мотивації користувачів система включає таблицю лідерів. Таблиця відображає найкращі результати користувачів у різних іграх, показуючи кількість набраних очок та місце в загальному рейтингу. Це спонукає користувачів до покращення своїх результатів та досягнення нових висот у навчанні. Таблиця лідерів регулярно оновлюється, що забезпечує актуальність даних та прозорість системи.

4. РОЗДІЛ 4. Ергономіка інформаційних систем

4.1. Ергономіка ІТ

Вимоги до зручності користування програмним забезпеченням

Ергономіка програмного забезпечення є ключовим аспектом, що впливає на ефективність та зручність використання системи кінцевими користувачами. Вимоги до програмного забезпечення з точки зору користувача включають кілька важливих аспектів. Перш за все, інтерфейс користувача повинен бути інтуїтивно зрозумілим. Це означає, що користувачі повинні мати можливість легко орієнтуватися в системі та виконувати необхідні дії без додаткових зусиль. Логічна структура інтерфейсу та зрозумілі елементи управління є основою для досягнення цього завдання.

Другою важливою вимогою є візуальна привабливість та зручність для зору. Дизайн інтерфейсу повинен бути приємним для ока, включаючи використання гармонійної кольорової гами, чітких шрифтів та достатнього контрасту між текстом і фоном. Це допомагає зменшити навантаження на зір користувачів та забезпечує комфортне використання програми протягом тривалого часу. Важливо уникати надмірного використання яскравих та насичених кольорів, які можуть викликати втоми очей, та зосередитися на спокійних і гармонійних відтінках.

Третім аспектом є адаптивність програмного забезпечення. Система повинна коректно працювати на різних пристроях з різними розмірами екранів, що забезпечує зручне використання програми як на стаціонарних комп'ютерах, так і на мобільних пристроях. Це дозволяє користувачам використовувати програму в будь-який час і в будь-якому місці, що значно підвищує її доступність та зручність.

Ефективність виконання завдань також є критично важливим аспектом. Інтерфейс повинен бути розроблений таким чином, щоб користувачі могли

швидко та ефективно виконувати свої завдання. Це включає мінімізацію кількості кроків для досягнення цілі, логічне групування функцій та забезпечення швидкого доступу до основних інструментів. Кожна дія повинна бути простою та зрозумілою, що знижує ризик помилок та підвищує продуктивність користувачів.

Зворотний зв'язок є невід'ємною частиною ергономіки програмного забезпечення. Користувачі повинні отримувати негайний зворотний зв'язок про свої дії, що допомагає їм зрозуміти результати та, у разі помилок, швидко їх виправити. Це може бути візуальний, звуковий або текстовий зворотний зв'язок, який інформує користувача про успішне виконання дії або про помилку, що сталася. Важливо забезпечити, щоб зворотний зв'язок був зрозумілим та своєчасним.

Проектування компонентів інтерфейсу

Проектування компонентів інтерфейсу користувача є критично важливим етапом у розробці програмного забезпечення. Цей процес включає кілька ключових аспектів, які забезпечують створення зручного, ефективного та привабливого інтерфейсу для користувачів.

Кнопки та елементи управління є одними з найважливіших компонентів інтерфейсу. Вони повинні бути достатньо великими, щоб їх було легко натискати, особливо на сенсорних екранах. Чітке позначення кнопок, яке пояснює їх функцію, допомагає користувачам швидко зрозуміти, що робить кожна кнопка. Розташування кнопок повинно бути логічним та зручним для користувача, що забезпечує легкість у виконанні необхідних дій.

Меню та навігація повинні бути організовані таким чином, щоб користувачі могли легко знайти необхідні функції. Логічна структура меню з чіткою ієрархією дозволяє швидко орієнтуватися в інтерфейсі. Навігація

повинна бути простою та інтуїтивно зрозумілою, що зменшує час, необхідний для виконання завдань, та підвищує загальну продуктивність користувачів.

Форми для введення даних повинні бути зрозумілими та простими у використанні. Поля введення мають бути достатньо великими для зручного введення інформації, а помилки введення повинні бути миттєво відображені з чіткими інструкціями для виправлення. Це забезпечує зручність введення даних та знижує ризик помилок, що можуть виникнути під час роботи з програмою.

Використання піктограм та графічних елементів допомагає швидко орієнтуватися в інтерфейсі. Піктограми повинні бути зрозумілими та інтуїтивно зрозумілими, що підвищує ефективність взаємодії з додатком. Важливо уникати надмірного використання графічних елементів, щоб не перевантажувати інтерфейс та не ускладнювати користувацький досвід.

Зворотний зв'язок та підтвердження дій є важливими компонентами проектування інтерфейсу. Користувачі повинні отримувати миттєвий зворотний зв'язок про свої дії. Це може бути у вигляді підказок, повідомлень про успішне виконання дій або помилки, а також візуальних ефектів, що підтверджують виконання дії. Такі підтвердження допомагають користувачам зрозуміти, що їхні дії були успішними або потребують коригування.

Проектування компонентів інтерфейсу з урахуванням цих аспектів забезпечує створення зручного, ефективного та привабливого для користувачів програмного забезпечення. Це дозволяє підвищити задоволеність користувачів, зменшити кількість помилок та покращити загальний користувацький досвід. Інтерфейс, розроблений з урахуванням ергономічних принципів, сприяє більш ефективному використанню програми та підвищенню продуктивності користувачів.

4.2. Дизайн додатку

Дизайн додатку відіграє ключову роль у забезпеченні зручності та привабливості програмного забезпечення. Основна мета дизайну - створення інтерфейсу, який буде зручним у використанні та привабливим для кінцевого користувача. У цьому підрозділі розглянемо дизайн кожного вікна додатку.

Вхід

Вікно входу до системи (рис. 4.1) забезпечує користувачам можливість входу до свого облікового запису. Основні елементи цього вікна включають поля для введення логіну та пароля, а також кнопки для входу та реєстрації. Дизайн інтерфейсу використовує яскраві кольори для кнопок, що робить їх легко помітними та доступними для користувача. Поля введення мають достатній розмір, що забезпечує зручність введення даних.

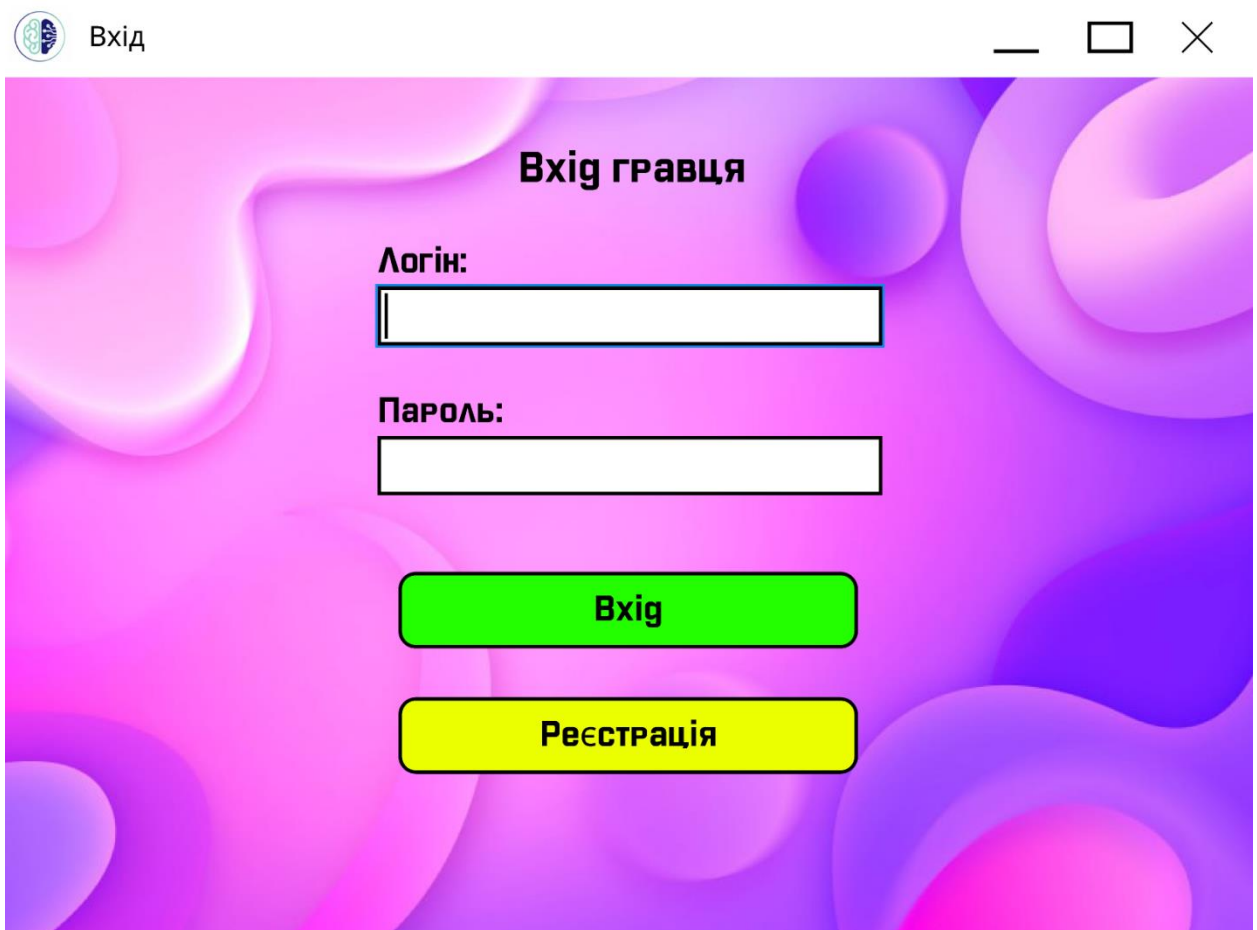
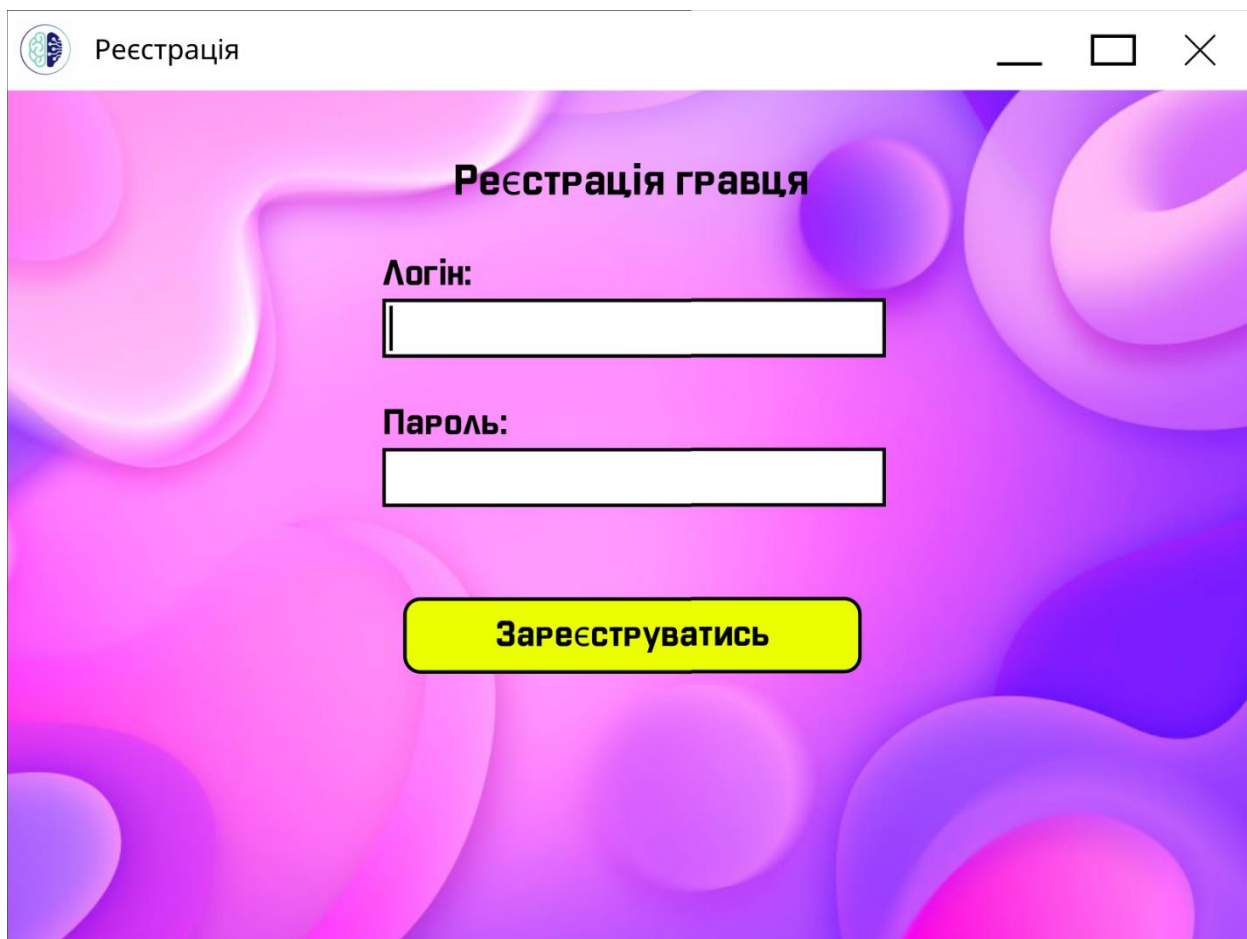


Рисунок 4.1 "Вхід"

Реєстрація

Вікно реєстрації (рис. 4.2) дозволяє новим користувачам створити обліковий запис. Вікно містить поля для введення логіну та пароля, а також кнопку для завершення реєстрації. Дизайн цього вікна аналогічний дизайну вікна входу, що забезпечує зручність та єдність інтерфейсу.



Реєстрація

Реєстрація гравця

Логін:

Пароль:

Зареєструватись

Рисунок 4.2 “Реєстрація”

Список ігор

Вікно списку ігор (рис. 4.3) відображає доступні навчальні ігри. Кожна гра має свій окремий блок з коротким описом та кнопкою для початку гри. Дизайн вікна дозволяє користувачам швидко орієнтуватися та обирати бажану гру. Яскраві кольори та зрозумілі піктограми допомагають користувачам легко знайти потрібну інформацію.

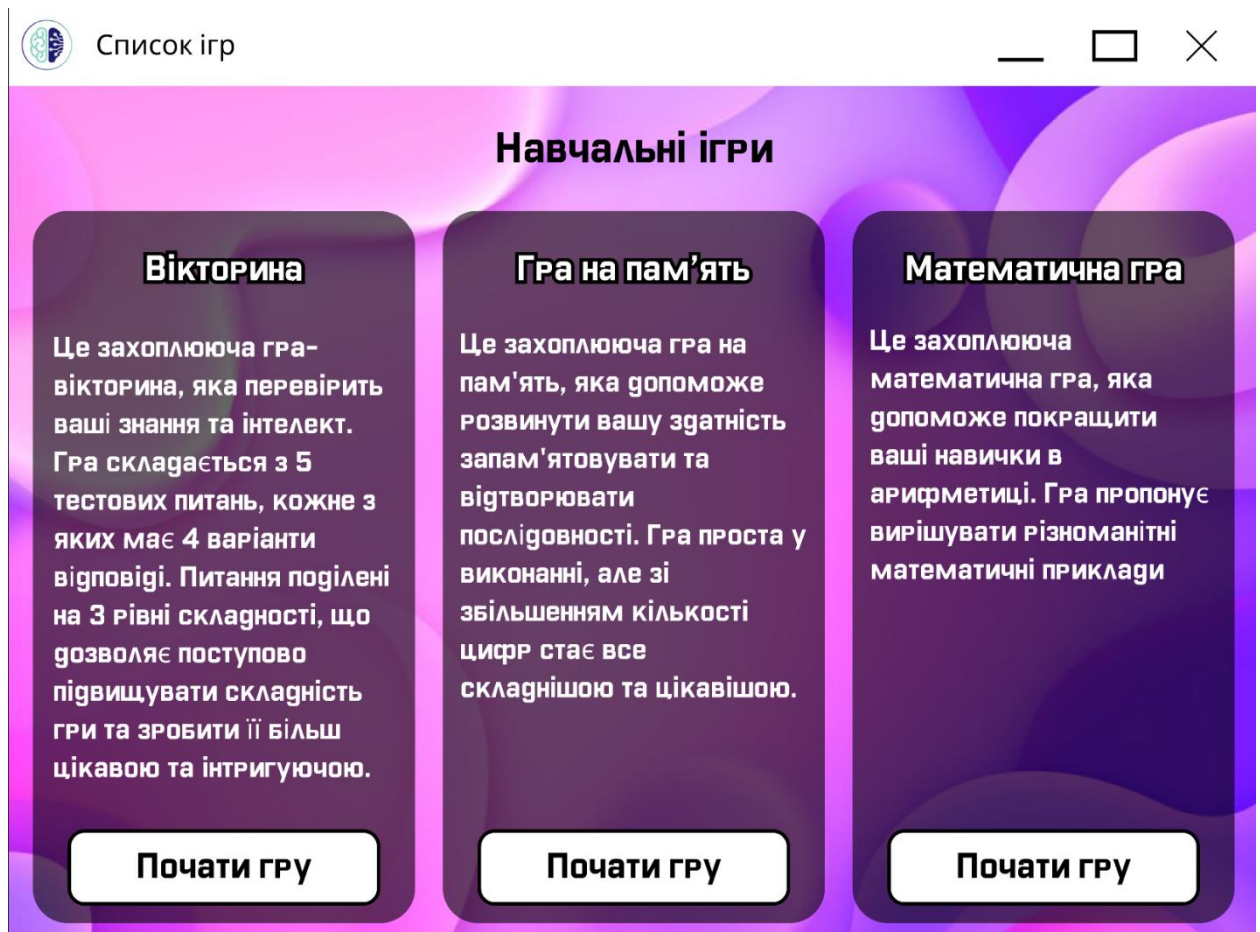


Рисунок 4.3 "Навчальні ігри"

Вікторина

Вікно вікторини (рис. 4.4) містить запитання з варіантами відповідей. Користувач обирає правильний, на його думку, варіант, після чого отримує зворотний зв'язок. Вікно також містить таблицю лідерів, яка відображає результати інших користувачів, що стимулює конкуренцію та підвищує мотивацію. Дизайн вікна забезпечує зручне розташування всіх елементів, що дозволяє користувачам легко взаємодіяти з програмою.

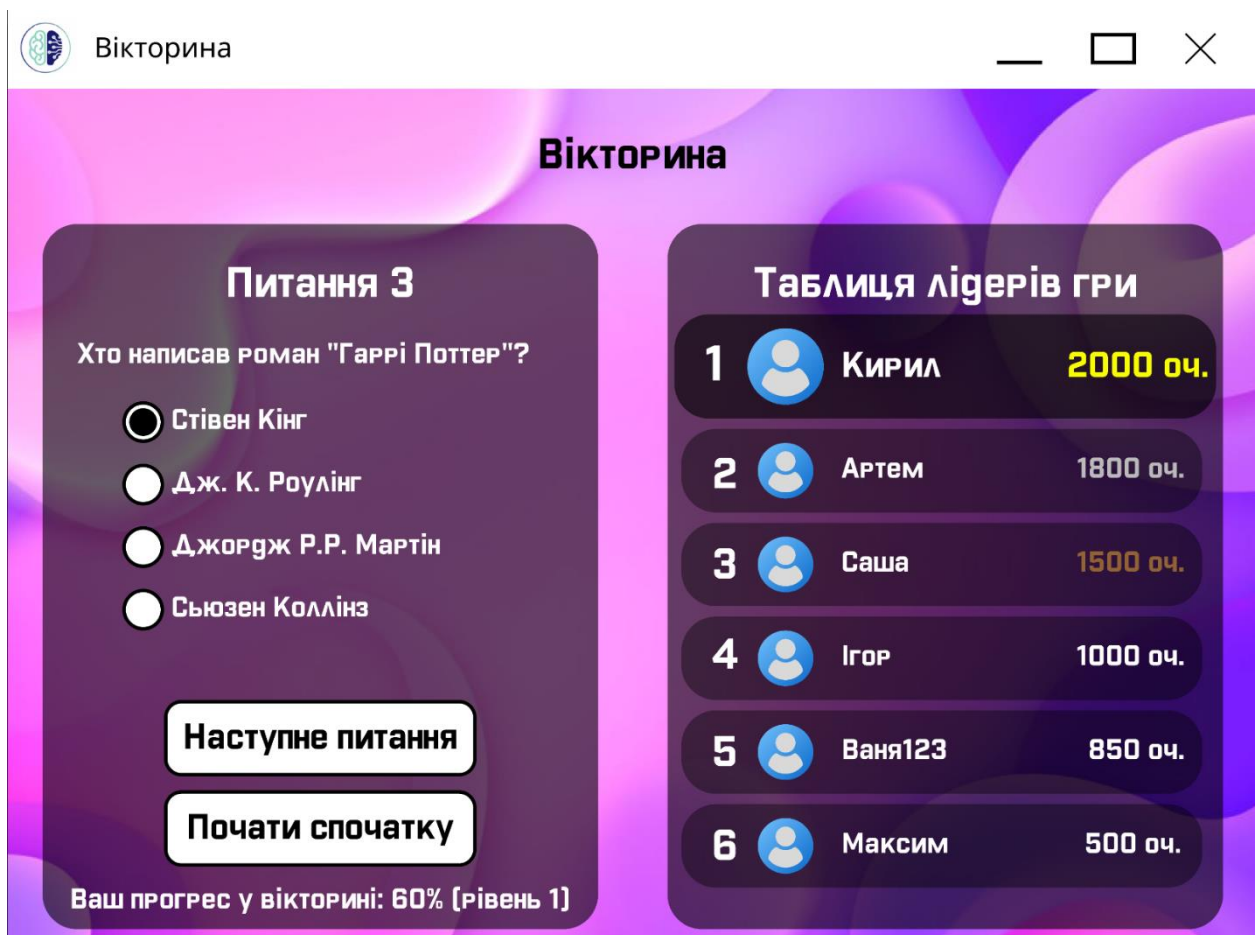


Рисунок 4.4 “Вікторина”

Гра на пам'ять

Вікно гри на пам'ять (рис. 4.5) містить завдання, де користувач повинен запам'ятати та повторити послідовність чисел. З кожним рівнем кількість чисел збільшується, що робить гру більш складною та цікавою. Вікно також містить таблицю лідерів, що стимулює користувачів до покращення своїх результатів. Дизайн вікна включає яскраві кольори та чіткі інструкції, що забезпечує зручність та зрозумілість для користувачів.

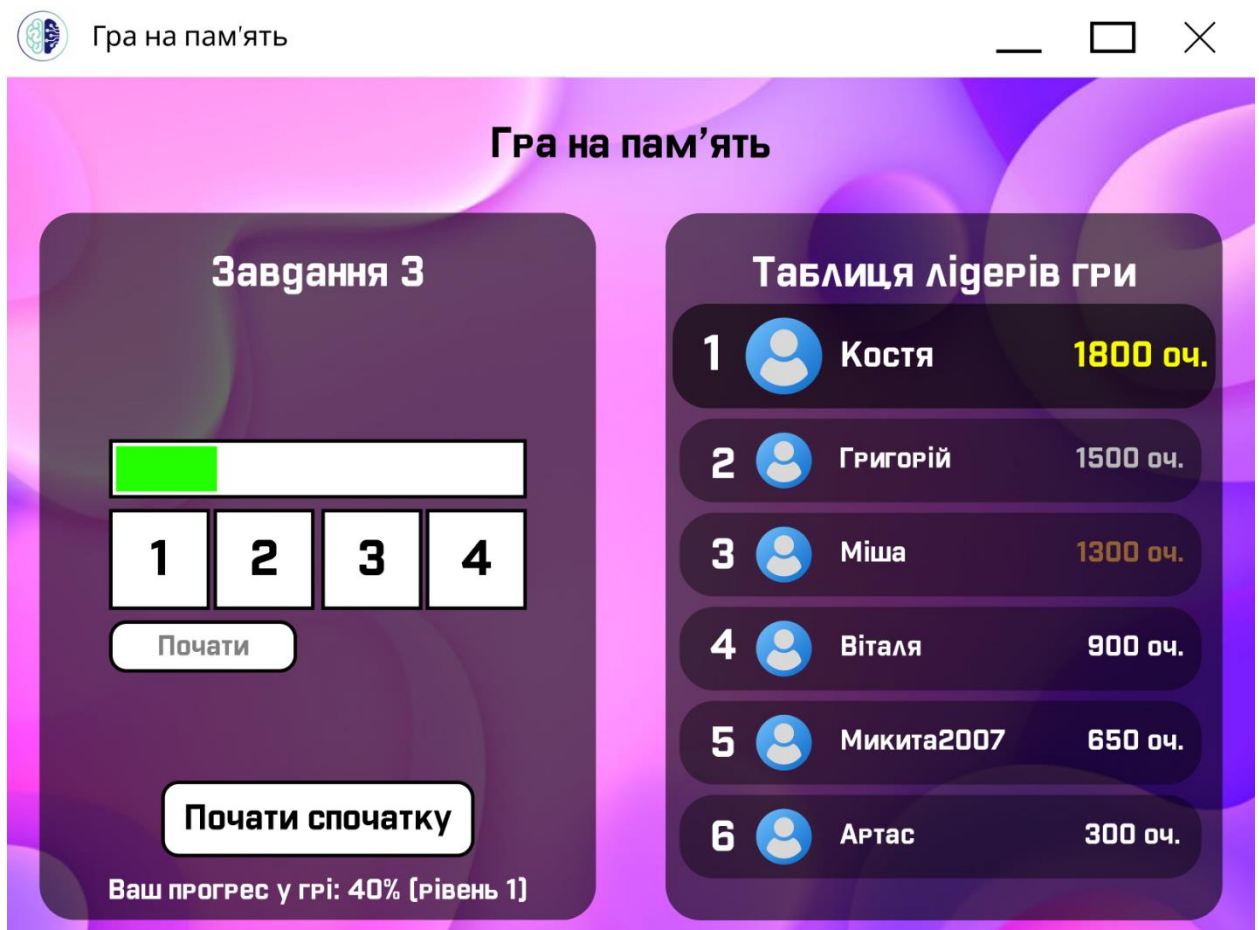


Рисунок 4.5 “Гра на пам’ять”

Математична гра

Вікно математичної гри (рис. 4.6) пропонує користувачам вирішувати різноманітні математичні задачі. Користувач вводить відповідь у поле введення та отримує зворотний зв'язок про правильність рішення. Вікно також містить таблицю лідерів, що відображає результати інших користувачів. Дизайн вікна забезпечує зручне розташування елементів та зрозумілий інтерфейс для користувачів.

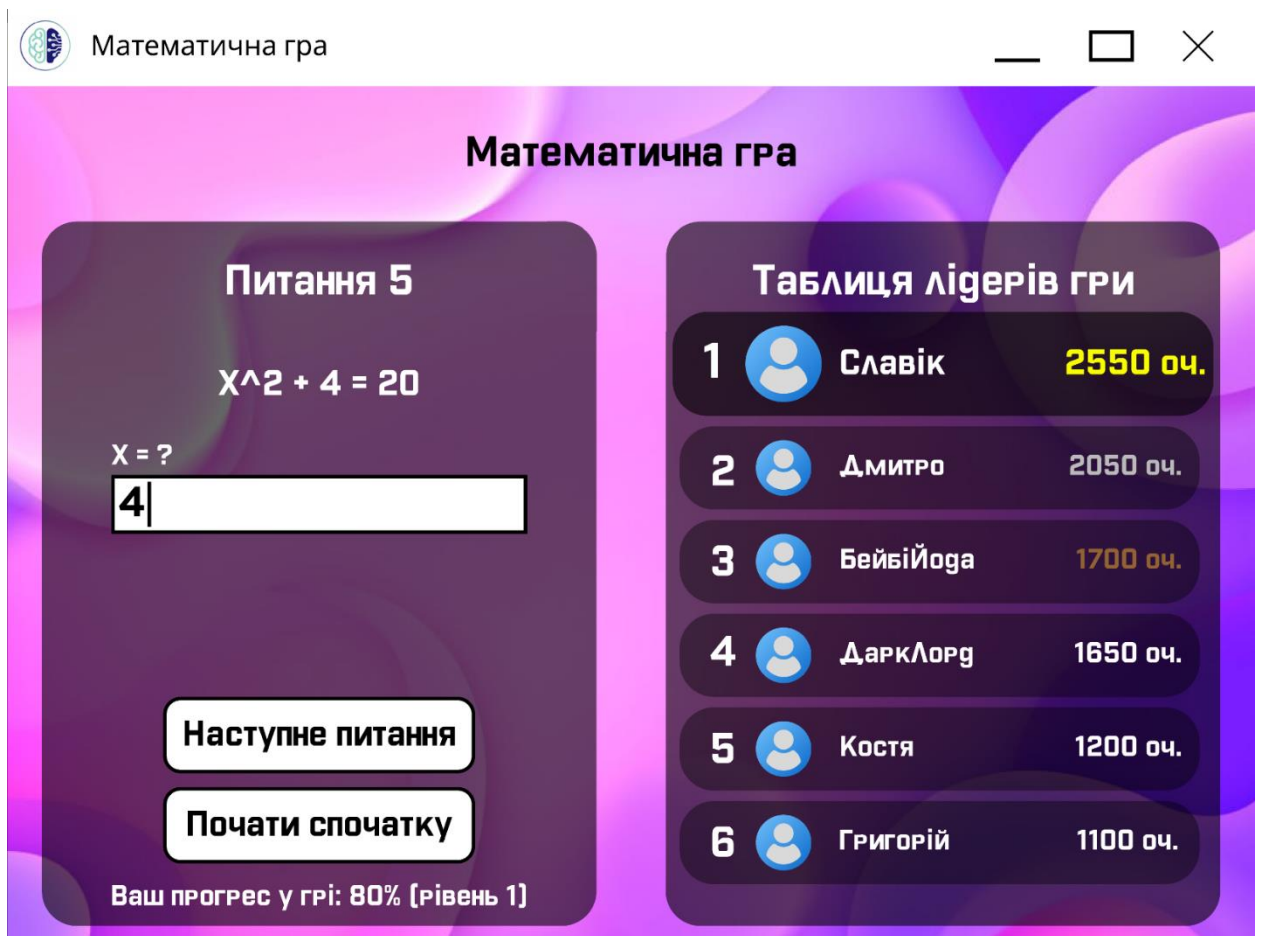


Рисунок 4.6 “Математична гра”

Таблиця лідерів

Вікно таблиці лідерів (рис. 4.7) відображає найкращі результати користувачів у різних іграх. Таблиця лідерів стимулює конкуренцію та підвищує мотивацію користувачів до покращення своїх результатів. Дизайн вікна забезпечує зручний та зрозумілий формат відображення інформації, що дозволяє користувачам легко знайти свою позицію в рейтингу.

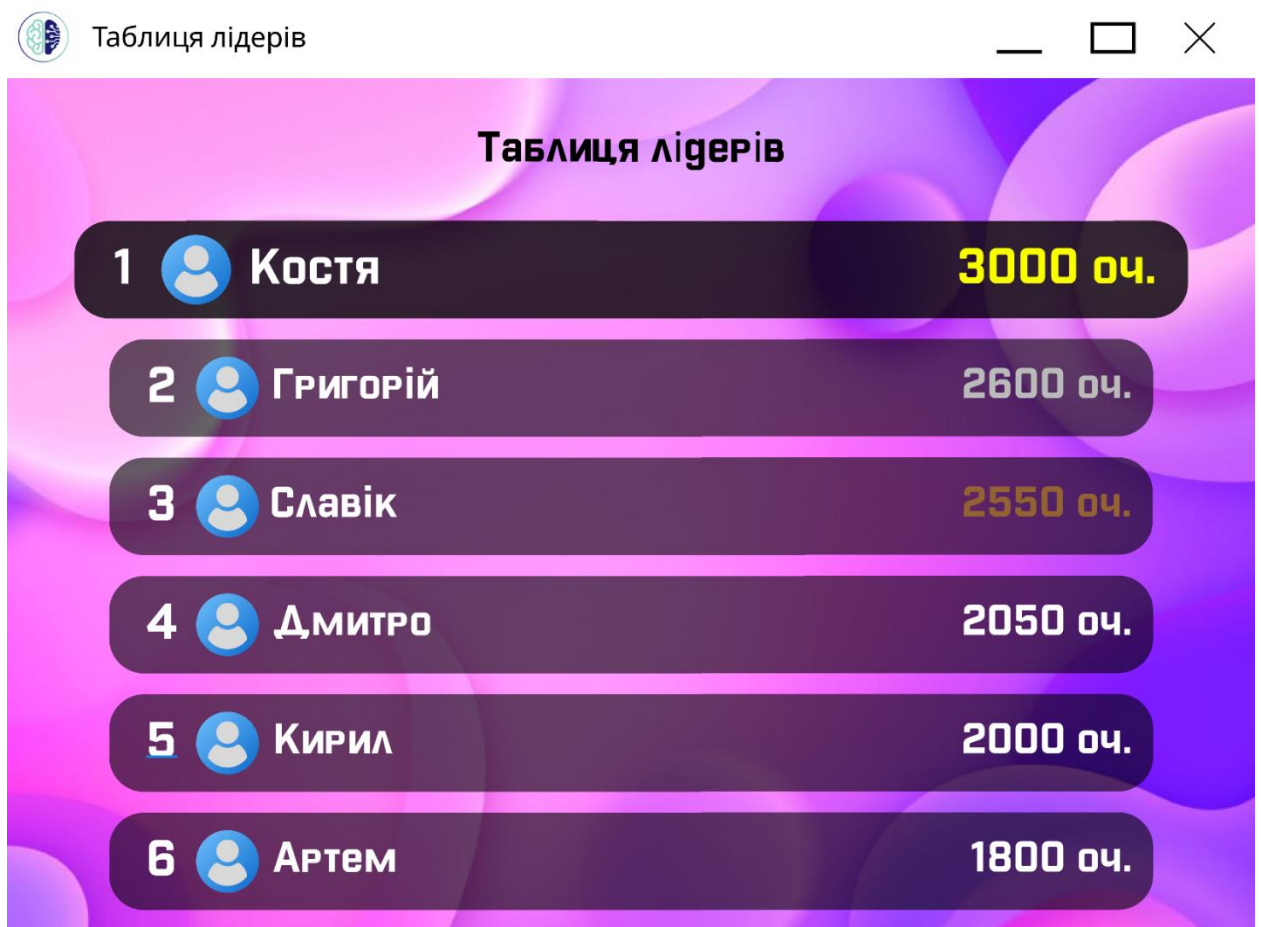


Рисунок 4.7 “Таблиця лідерів”

Досягнення

Вікно досягнень (рис. 4.8) відображає досягнення користувача у різних іграх. Кожне досягнення має свою назву, опис та кількість набраних очок. Дизайн вікна включає зрозумілі піктограми та приємну кольорову гаму, що забезпечує зручне та привабливе відображення інформації.

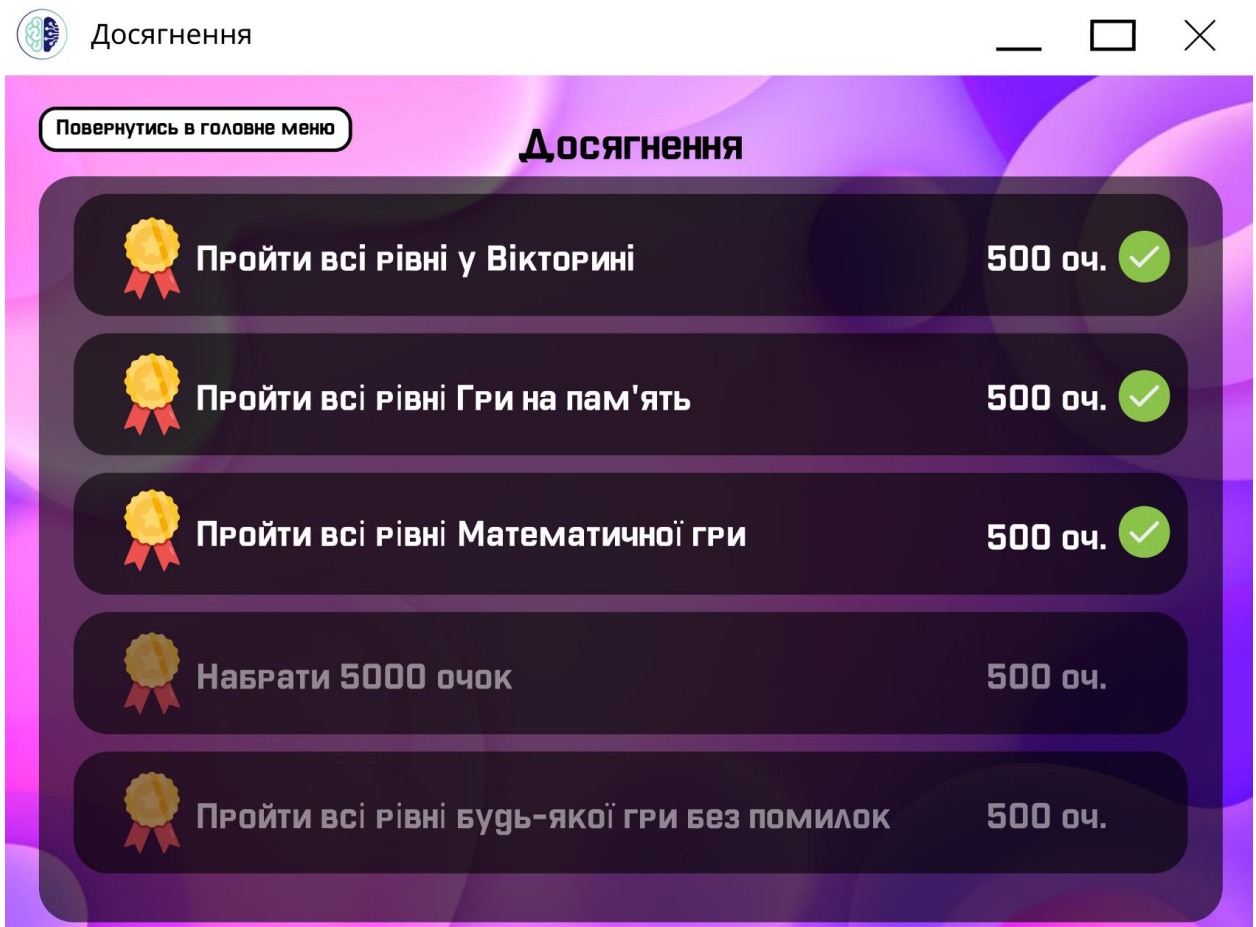


Рисунок 4.8 “Досягнення”

Профіль

Вікно профілю (рис. 4.9) містить інформацію про користувача, включаючи його ім'я, кількість набраних очок, рівень, кількість виконаних завдань та інші статистичні дані. Дизайн вікна забезпечує зручне відображення всієї необхідної інформації про користувача, що дозволяє йому швидко орієнтуватися в своєму прогресі.

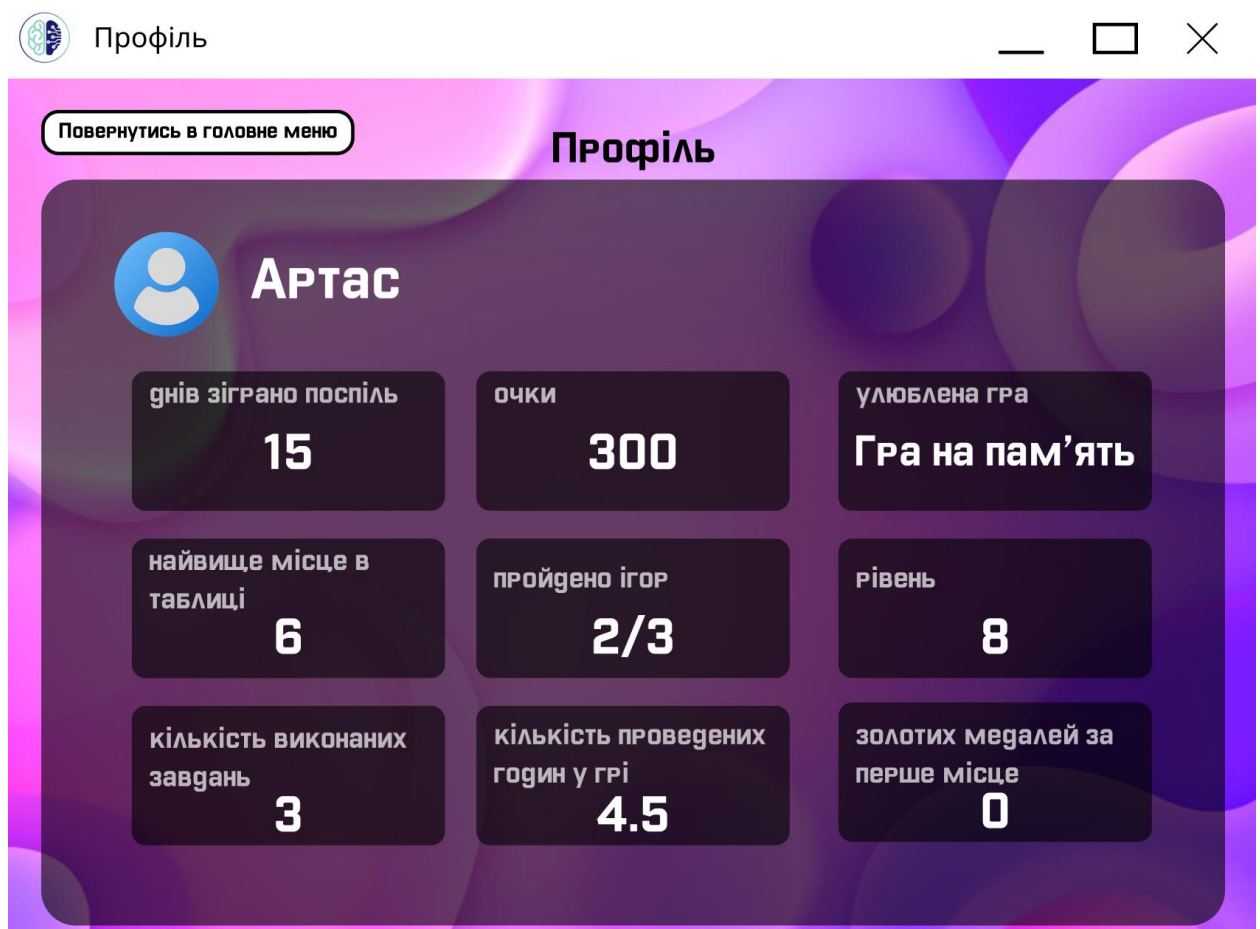


Рисунок 4.9 “Профіль”

Головне меню

Головне меню (рис. 4.10) забезпечує навігацію до основних розділів додатку: список ігор, профіль, досягнення та таблиця лідерів. Меню має простий та інтуїтивно зрозумілий дизайн, що дозволяє користувачам легко знаходити необхідні функції та переходити між різними розділами програми.

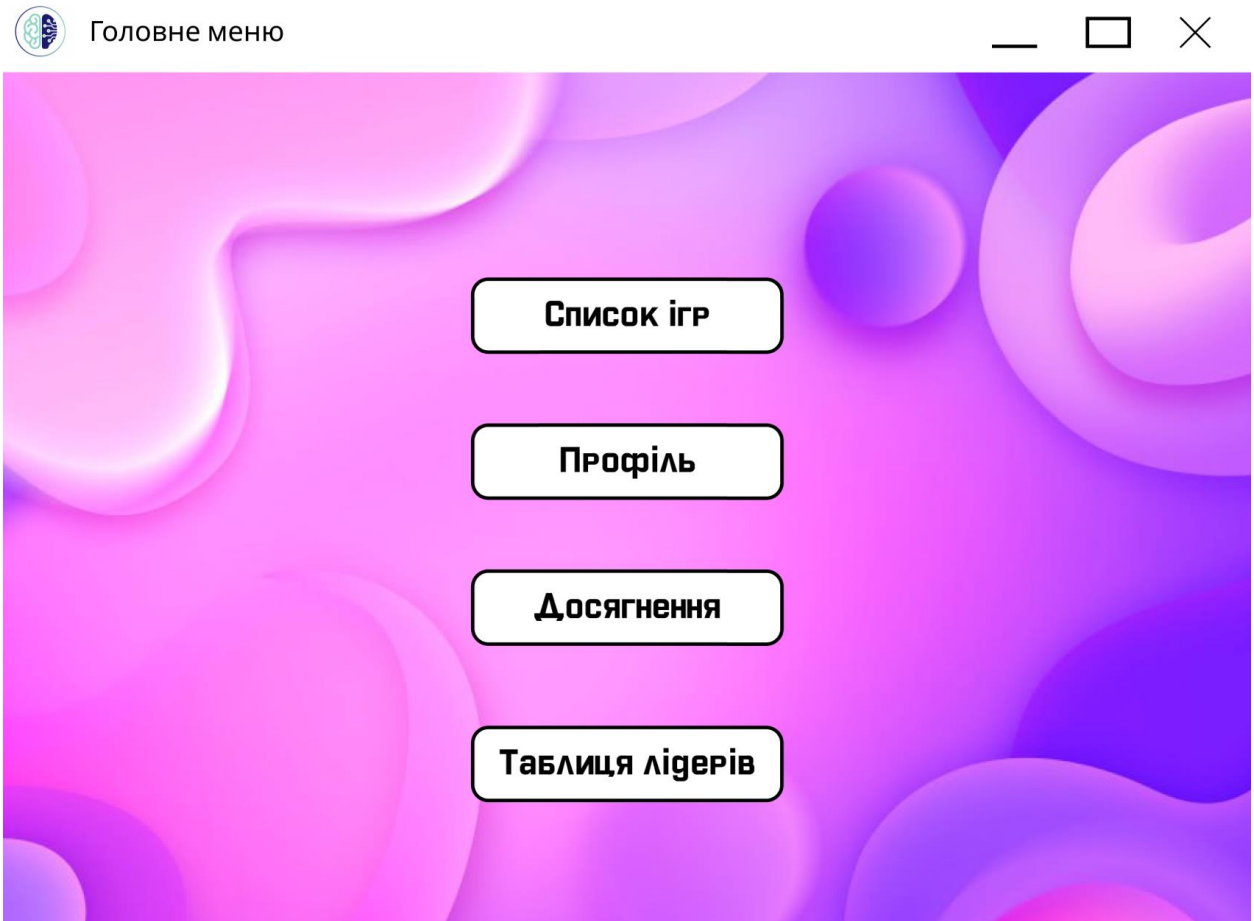


Рисунок 4.10 “Головне меню”

Дизайн кожного вікна додатку був розроблений з урахуванням принципів ергономіки та зручності використання. Це забезпечує комфортне та привабливе використання програми, підвищуючи мотивацію користувачів до навчання та досягнення нових знань та навичок.

4.3. Технічно-економічне обґрунтування розробки

Резюме проекту

Проект розробки гейміфікованої системи навчання спрямований на створення інтерактивного інструменту, що поєднує освітні функції з елементами гри для підвищення мотивації та залученості користувачів. Основна мета проекту - забезпечити ефективне та привабливе навчання, яке відповідатиме сучасним вимогам та потребам користувачів. Ця система розробляється для широкого кола користувачів, включаючи навчальні заклади, корпоративні тренінгові центри та індивідуальних користувачів, які прагнуть підвищити свої знання та навички.

Гейміфікована система навчання включає кілька інтерактивних ігор, кожна з яких розроблена з метою розвитку різних навичок та знань у користувачів. Основні ігри включають вікторину, гру на пам'ять та математичну гру, які дозволяють користувачам перевіряти та покращувати свої знання у різних галузях. Крім того, система містить функції відстеження прогресу, автоматичної оцінки результатів та надання зворотного зв'язку, що допомагає користувачам отримувати миттєві результати своїх дій та швидко виправляти помилки.

Система розробляється з урахуванням принципів ергономіки та зручності використання, що забезпечує комфортне та ефективне навчання. Інтерфейс додатку є інтуїтивно зрозумілим та привабливим, що сприяє зручності використання для користувачів різного віку та рівня технічної підготовки. Додаткові функції, такі як реєстрація та вхід користувачів, таблиця лідерів та досягнення, забезпечують додаткову мотивацію та стимулюють користувачів до подальшого навчання та досягнення нових результатів.

Опис проектованого продукту або виду послуг

Розроблена гейміфікована система навчання включає кілька ключових компонентів, що забезпечують інтерактивне та ефективне навчання користувачів. Цей продукт поєднує освітні функції з елементами гри, що сприяє підвищенню мотивації та залученості користувачів. Основні функціональні можливості та компоненти системи включають.

Система містить кілька інтерактивних ігор, розроблених для розвитку різних навичок та знань користувачів. Основні ігри включають вікторину, гру на пам'ять та математичну гру. Вікторина складається з серії запитань з варіантами відповідей, призначена для перевірки та покращення знань користувачів у різних галузях. Гра на пам'ять допомагає користувачам розвивати та покращувати здатність запам'ятовувати та відтворювати послідовності чисел. Математична гра пропонує користувачам вирішувати математичні задачі різної складності для тренування логічного мислення та математичних навичок.

Користувачі можуть відстежувати свій прогрес у виконанні завдань та досягненнях. Це включає графіки та діаграми, що візуально відображають прогрес користувачів та допомагають їм розуміти свої досягнення та області для покращення. Система автоматично оцінює результати завдань та надає користувачам негайний зворотний зв'язок. Це включає інформацію про правильність відповідей та пояснення правильних рішень, що допомагає користувачам краще розуміти матеріал та швидко виправляти помилки.

Система передбачає процес реєстрації нових користувачів та входу для вже зареєстрованих. Це забезпечує персоналізований досвід та збереження індивідуального прогресу користувачів. Для стимулювання конкуренції та підвищення мотивації користувачів система включає таблицю лідерів, яка відображає найкращі результати користувачів у різних іграх. Це спонукає користувачів до покращення своїх результатів та досягнення нових висот у навчанні.

Система відображає досягнення користувачів у різних іграх, що служить додатковим стимулом для продовження навчання та досягнення нових цілей. Дизайн інтерфейсу був розроблений з урахуванням принципів ергономіки та зручності використання. Всі елементи інтерфейсу логічно структуровані та легко доступні, що забезпечує комфортне та ефективне використання програми.

Оцінка ринку збуту

Ринок збуту для гейміфікованої системи навчання є досить широким та перспективним, охоплюючи різні сегменти освіти та професійного розвитку. Основними цільовими сегментами є навчальні заклади, корпоративні тренінгові центри та індивідуальні користувачі, які прагнуть підвищити свої знання та навички.

Навчальні заклади, такі як школи, коледжі та університети, є ключовими споживачами подібних систем, оскільки вони постійно шукають нові методи та інструменти для підвищення якості навчання та залучення студентів до навчального процесу. Гейміфіковані системи навчання можуть стати ефективним доповненням до традиційних методів навчання, підвищуючи мотивацію студентів та сприяючи кращому засвоєнню матеріалу.

Корпоративні тренінгові центри також становлять значний сегмент ринку збуту. Багато компаній інвестують у професійний розвиток своїх співробітників, використовуючи сучасні технології для навчання та підвищення кваліфікації. Гейміфіковані системи навчання можуть бути ефективно використані для проведення тренінгів, семінарів та курсів підвищення кваліфікації, підвищуючи залученість співробітників та покращуючи результати навчання.

Індивідуальні користувачі, включаючи студентів та професіоналів, які прагнуть самостійно підвищувати свої знання та навички, також є важливим

сегментом ринку. Зростаючий інтерес до самостійного навчання та розвитку відкриває нові можливості для впровадження гейміфікованих систем навчання, які можуть бути доступні через інтернет та мобільні додатки.

Аналіз конкурентного середовища показує, що на ринку вже існують кілька подібних продуктів, які пропонують гейміфіковані рішення для навчання. Проте, розроблене програмне забезпечення відрізняється високою якістю інтерфейсу, широкими можливостями для персоналізації навчального процесу та інтеграції з іншими інструментами, що забезпечує конкурентну перевагу. Основними конкурентами є такі продукти, як XYZ Learning Game та ABC Training Platform, які мають обмежені можливості у порівнянні з нашим рішенням.

Враховуючи зростаючий попит на сучасні технології в освіті та професійному розвитку, а також широкі можливості для впровадження гейміфікованих систем навчання, ринок збуту для даного програмного забезпечення є перспективним та має великий потенціал для зростання та розвитку.

Аналіз конкуренції

На ринку існує кілька подібних продуктів, які пропонують гейміфіковані рішення для навчання. Основними конкурентами нашого програмного забезпечення є такі продукти, як XYZ Learning Game та ABC Training Platform. Ці продукти мають певні обмеження у порівнянні з нашим рішенням, що дозволяє нам виділятися на ринку завдяки високій якості інтерфейсу, широким можливостям для персоналізації навчального процесу та інтеграції з іншими інструментами.

XYZ Learning Game пропонує базові функції гейміфікації, такі як вікторини та прості ігри на розвиток пам'яті. Однак, цей продукт не має можливостей для детального відстеження прогресу користувачів та надання

зворотного зв'язку в режимі реального часу. Крім того, інтерфейс XYZ Learning Game є менш інтуїтивно зрозумілим та привабливим, що може впливати на зручність використання та загальний користувацький досвід.

ABC Training Platform орієнтована переважно на корпоративні тренінги та пропонує обмежену кількість інтерактивних ігор. Основний акцент цього продукту робиться на проведення семінарів та тренінгів, що обмежує його застосування у широкому спектрі навчальних закладів та для індивідуальних користувачів. Крім того, ABC Training Platform не забезпечує високого рівня інтеграції з іншими освітніми інструментами та платформами, що може бути важливим для користувачів, які шукають комплексні рішення для навчання.

Наше програмне забезпечення має низку переваг, які забезпечують конкурентну перевагу на ринку. Висока якість інтерфейсу та зручність використання дозволяють користувачам легко орієнтуватися у системі та ефективно використовувати її функції. Широкі можливості для персоналізації навчального процесу, включаючи налаштування рівнів складності та адаптацію змісту до потреб користувачів, забезпечують індивідуальний підхід до навчання.

План виробництва

Процес виробництва гейміфікованої системи навчання включає кілька ключових етапів, кожен з яких має свої специфічні завдання та вимоги. Це забезпечує системний підхід до розробки програмного забезпечення, що гарантує високу якість кінцевого продукту та задоволення потреб користувачів.

Перший етап – проектування, включає визначення функціональних вимог, проектування архітектури системи та розробку інтерфейсу користувача. На цьому етапі здійснюється аналіз потреб користувачів, вивчаються існуючі рішення та визначаються ключові функції, які необхідно реалізувати. Розробка

детального технічного завдання та прототипів інтерфейсу допомагає створити чітке бачення кінцевого продукту.

Другий етап – розробка, охоплює написання коду, створення бази даних та інтеграцію різних компонентів системи. Основною метою цього етапу є реалізація функціональних можливостей, визначених на етапі проектування. Програмісти пишуть код, створюють алгоритми, які забезпечують роботу системи, та інтегрують різні модулі для забезпечення їхньої взаємодії.

Третій етап – тестування, включає перевірку функціональності, виявлення та виправлення помилок. Це критично важливий етап, оскільки він забезпечує якість та стабільність роботи системи. Тестувальники перевіряють систему на відповідність вимогам, тестують різні сценарії використання та ідентифікують можливі проблеми. Виправлення помилок та оптимізація коду на цьому етапі забезпечують високу якість кінцевого продукту.

Четвертий етап – впровадження, передбачає розгортання системи на робочих серверах та надання доступу користувачам. На цьому етапі здійснюється налаштування серверів, інсталяція програмного забезпечення та конфігурація системи для забезпечення її коректної роботи в робочому середовищі. Крім того, проводиться навчання користувачів та надання технічної підтримки для забезпечення безперебійного функціонування системи.

П'ятий етап – підтримка, включає забезпечення безперебійної роботи системи, виправлення виявлених проблем, внесення змін та оновлень. Це триваючий процес, який передбачає моніторинг роботи системи, реагування на запити користувачів та впровадження нових функцій у відповідь на зворотний зв'язок та змінні потреби ринку. Підтримка також включає оновлення системи для забезпечення її безпеки та відповідності сучасним технологічним вимогам.

Організаційний та юридичний план

Організаційний план проекту передбачає чітке визначення завдань та відповідальності на кожному етапі розробки, а також ефективне управління ресурсами та процесами розробки. Основні завдання включають планування, моніторинг прогресу, управління ризиками та координацію всіх етапів роботи. Важливо забезпечити чіткий зв'язок між усіма учасниками процесу та регулярне звітування про хід виконання проекту.

На етапі розробки необхідно виконувати всі завдання, пов'язані з написанням коду, створенням алгоритмів, розробкою інтерфейсу користувача та інтеграцією різних компонентів системи. Важливо реалізувати всі функціональні можливості, визначені в технічному завданні, забезпечуючи високу якість та стабільність роботи програмного забезпечення.

Тестування є критично важливим етапом, який включає перевірку функціональності системи, виявлення та виправлення помилок. Це забезпечує відповідність вимогам та високу якість кінцевого продукту. Всі виявлені проблеми повинні бути оперативно виправлені, а система оптимізована для забезпечення найкращого користувацького досвіду.

Після завершення розробки та тестування система впроваджується на робочих серверах та надається доступ користувачам. На цьому етапі здійснюється налаштування серверів, інсталяція програмного забезпечення та конфігурація системи для забезпечення її коректної роботи в робочому середовищі. Важливо також провести навчання користувачів та забезпечити технічну підтримку для безперебійної роботи системи.

Юридичний план передбачає захист прав інтелектуальної власності, дотримання вимог законодавства та укладення необхідних договорів з партнерами та користувачами. Захист прав інтелектуальної власності включає реєстрацію авторських прав на програмне забезпечення та використання

ліцензій для захисту від несанкціонованого використання або копіювання. Дотримання вимог законодавства включає забезпечення відповідності програмного забезпечення місцевим та міжнародним нормативам і стандартам, таким як правила конфіденційності даних та захист персональної інформації користувачів.

4.4. Математичні методи оцінки часу розробки програмного забезпечення

Точне планування часу на розробку програмного забезпечення є критично важливим аспектом успішного управління проектами. Незалежно від типу програмного забезпечення, точна оцінка часу дозволяє ефективно розподілити ресурси, уникнути перевитрат, дотриматися дедлайнів та забезпечити високу якість кінцевого продукту. Це особливо актуально для гейміфікованих додатків для навчання, які потребують ретельного планування через їхню складну функціональність та інтерактивність.

Гейміфіковані додатки для навчання включають елементи ігрового дизайну, такі як бали, рівні, нагороди та інші мотиваційні механізми, що стимулюють користувачів до навчання. Вони часто включають інтерактивні модулі, персоналізовані профілі користувачів, систему зворотного зв'язку та аналізу результатів. Розробка таких додатків вимагає не лише технічних знань, але й розуміння психології користувачів та педагогічних принципів.

Враховуючи високу складність і багатокomпонентність гейміфікованих додатків, важливо забезпечити точне планування кожного етапу розробки. Це включає дизайн, програмування, тестування та інтеграцію компонентів. Ретельне планування допомагає передбачити можливі труднощі та ризики, забезпечити їхнє вчасне вирішення і, в результаті, створити якісний продукт, який буде ефективним інструментом для навчання.

Огляд методів оцінки часу розробки

Існує кілька методів оцінки часу розробки програмного забезпечення, які дозволяють проектним командам ефективно планувати свої ресурси та уникати перевитрат часу і бюджету. Найпоширеніші методи включають:

Експертні оцінки: Залучення досвідчених фахівців для оцінки часу, необхідного для виконання завдань. Цей метод базується на інтуїції та досвіді експертів і часто використовується у поєднанні з іншими методами для підвищення точності.

Аналогічні оцінки: Використання даних з минулих подібних проектів для оцінки часу на поточний проект. Цей метод дозволяє швидко отримати оцінку, проте може бути неточним, якщо умови проектів суттєво відрізняються.

Метод PERT (Program Evaluation and Review Technique): Статистичний метод, який використовує три оцінки часу (оптимістичну, ймовірну та песимістичну) для розрахунку очікуваного часу виконання завдання. Метод PERT дозволяє врахувати невизначеність і варіативність часу виконання.

Аналіз функціональних точок (Function Point Analysis): Метод оцінки розміру програмного забезпечення на основі його функціональних вимог. Кожна функціональна вимога оцінюється за складністю, що дозволяє розрахувати загальну кількість функціональних точок і час, необхідний для їх реалізації.

В рамках цієї роботи основна увага буде приділена методу PERT та аналізу функціональних точок, оскільки вони дозволяють найбільш точно оцінити час розробки складних проектів, таких як гейміфіковані додатки для навчання.

Метод PERT

Метод PERT (Program Evaluation and Review Technique) є одним з найефективніших методів планування та оцінки часу виконання проектів, особливо коли існує велика невизначеність у часі виконання окремих завдань. Метод був розроблений у 1950-х роках для потреб військових проектів і з тих пір знайшов широке застосування у різних галузях.

Основні принципи методу PERT

Метод PERT використовує три оцінки для кожного завдання:

Оцінка оптимістична (O): Час, необхідний для виконання завдання за найсприятливіших умов.

Оцінка ймовірна (M): Час, який найбільш ймовірно знадобиться для виконання завдання.

Оцінка песимістична (P): Час, необхідний для виконання завдання за найгірших умов.

Формула для розрахунку очікуваного часу виконання (TE) виглядає так:

$$TE = \frac{O + 4M + P}{6}$$

Цей підхід дозволяє врахувати різні сценарії та отримати більш точну оцінку часу виконання завдання, ніж використання лише однієї оцінки.

Процедура оцінки за методом PERT

Збір та аналіз вимог: Визначення всіх завдань проекту та їхньої взаємозалежності.

Визначення трьох оцінок для кожного завдання: Отримання оптимістичної, ймовірної та песимістичної оцінок часу для кожного завдання.

Розрахунок очікуваного часу виконання (TE): Використання формули для кожного завдання.

Створення мережевого графіка проекту: Визначення послідовності виконання завдань та їхньої взаємозалежності.

Аналіз критичного шляху: Визначення критичного шляху проекту, тобто послідовності завдань, які визначають мінімальний час виконання всього проекту.

Переваги методу PERT

Врахування невизначеності: Метод дозволяє врахувати варіативність часу виконання завдань.

Точність оцінок: Використання трьох оцінок дозволяє отримати більш точні результати.

Аналіз критичного шляху: Метод допомагає визначити найбільш важливі завдання, що впливають на загальний час виконання проекту.

Метод PERT є особливо корисним для проектів з високою невизначеністю та складними взаємозв'язками між завданнями, такими як гейміфіковані додатки для навчання. Використання цього методу дозволяє ефективно планувати час розробки та уникати затримок, що підвищує шанси на успішне завершення проекту.

Дані для розрахунку

Завдання	Оптимістична оцінка (O), дні	Ймовірна оцінка (M), дні	Песимістична оцінка (P), дні	Залежність від інших завдань
Дизайн та інтерфейс	2	5	14	-
Система реєстрації та входу	3	6	12	Дизайн та інтерфейс
Розробка вікторини, математичної гри, гри на пам'ять	4	10	16	Дизайн та інтерфейс
Система досягень	2	4	7	Розробка вікторини, математичної гри, гри на пам'ять
Профіль, таблиця лідерів	2	4	7	Система реєстрації, система досягень
Тестування	1	3	6	Усі інші завдання

Розрахунок для кожної частини

Використовуємо формулу:

$$TE = \frac{O + 4M + P}{6}$$

Дизайн та інтерфейс:

$$TE_{\text{дизайн}} = \frac{2 + (4 * 5) + 14}{6} = \frac{2 + 20 + 14}{6} = \frac{36}{6} = 6 \text{ днів}$$

Система реєстрації:

$$TE_{\text{реєстрація}} = \frac{3 + (4 * 6) + 12}{6} = \frac{3 + 24 + 12}{6} = \frac{39}{6} = 6.5 \text{ днів}$$

Розробка ігор:

$$TE_{\text{розробкаГри}} = \frac{4 + (4 * 10) + 16}{6} = \frac{4 + 40 + 16}{6} = \frac{60}{6} = 10 \text{ днів}$$

Система досягнень

$$TE_{\text{досягнення}} = \frac{2 + (4 * 4) + 7}{6} = \frac{2 + 16 + 7}{6} = \frac{25}{6} = 4.17 \text{ днів}$$

Профіль та таблиця лідерів:

$$TE_{\text{профіль}} = \frac{2 + (4 * 4) + 7}{6} = \frac{2 + 16 + 7}{6} = \frac{25}{6} = 4.17 \text{ днів}$$

Тестування:

$$TE_{\text{тестування}} = \frac{1 + (4 * 3) + 6}{6} = \frac{1 + 12 + 6}{6} = \frac{19}{6} = 3.17 \text{ днів}$$

Висновок:

За новими даними, оптимальний час на розробку кожної частини проекту методом PERT складає:

Дизайн та інтерфейс: 6 днів

Система реєстрації: близько 6.5 днів

Розробка вікторини, математичної гри та гри на пам'ять: 10 днів

Система досягнень та профіль та таблиця лідерів: близько 4.17 днів на кожен з них

Тестування: близько 3.17 днів

Загальний час створення додатку: $6 + 6.5 + 10 + (4.17 * 2) + 3.17 = 34.01$ днів

Отже, загальний час створення додатку складає приблизно 34.01 днів. Цей час включає в себе розробку всіх компонентів, від дизайну та реєстраційної системи до розробки ігор, систем досягнень, профілю та таблиці лідерів, а також тестування.

Цей розрахунок є орієнтовним і може змінюватись залежно від конкретних умов розробки, ресурсів та інших факторів, що можуть впливати на швидкість виконання завдань.

Ці оцінки дозволяють краще розуміти час, необхідний для кожної частини проекту, з урахуванням можливих варіацій у часі виконання завдань.

ВИСНОВКИ

У ході виконання роботи було розроблено програму "GamificationApp", яка спрямована на підвищення якості роботи програмістів через інтерактивні навчальні ігри. Основною метою розробки було створення зручного та ефективного інструменту, що допомагатиме користувачам вдосконалювати свої навички програмування у цікавій та мотивуючій формі.

Програма була спроектована з використанням модульної архітектури, що забезпечує легкість у підтримці та розширенні функціональності. Основні компоненти включають головну форму, математичну гру, гру на пам'ять та вікторину. Для розробки було обрано мову програмування C# та фреймворк Windows Forms, що забезпечує високу продуктивність та зручність у створенні користувацьких інтерфейсів. Серверна частина програми реалізована на ASP.NET Core, що забезпечує надійне управління даними та високу масштабованість.

Кожна гра у програмі включає кілька рівнів складності, що дозволяє користувачам поступово підвищувати свої навички. Використання системи досягнень та нагород забезпечує додаткову мотивацію для користувачів. Було розроблено механізм аутентифікації користувачів, що забезпечує безпеку та захист персональних даних. Програма дозволяє користувачам переглядати доступні курси, записуватися на них та виконувати завдання. Кожна гра пропонує користувачам різні завдання, включаючи математичні задачі, запам'ятовування послідовностей та відповіді на питання вікторини. Було реалізовано механізм оцінювання результатів завдань та надання зворотного зв'язку користувачам.

Програма дозволяє користувачам відстежувати свій прогрес у виконанні завдань та досягненнях, що сприяє підвищенню їхньої мотивації та ефективності навчання. Програма підтримує інтеграцію з популярними системами керування версіями, такими як GitHub та GitLab, що дозволяє

користувачам підключати свої репозиторії та проводити код-рев'ю. Крім того, забезпечується зручна інтеграція з системами управління проектами, такими як JIRA, що дозволяє користувачам відстежувати завдання та прогрес проектів.

Таким чином, розроблена програма "GamificationApp" успішно реалізує поставлені завдання та забезпечує користувачам зручний та ефективний інструмент для підвищення своїх навичок програмування. Використання інтерактивних навчальних ігор та системи досягнень сприяє підвищенню мотивації користувачів та забезпечує їм цікаве та корисне навчання. Модульна архітектура програми забезпечує легкість у підтримці та розширенні функціональності, а інтеграція з іншими інструментами дозволяє ефективно використовувати програму у поєднанні з існуючими робочими процесами. Програма "GamificationApp" є ефективним засобом для вдосконалення навичок програмістів та сприяє їхньому професійному зростанню у цікавій та мотивуючій формі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Макаревич О. О. Гейміфікація як невід’ємний чинник підвищення ефективності елементів дистанційного навчання. Молодий вчений, 2015. С. 275-278.
2. Побризгаєва В. Г., Наливайко О. О. ГЕЙМІФІКАЦІЯ НАВЧАННЯ У ПОЧАТКОВИХ КЛАСАХ В УМОВАХ ДИСТАНЦІЙНОГО НАВЧАННЯ. Електронне наукове фахове видання “ВІДКРИТЕ ОСВІТНЄ Е-СЕРЕДОВИЩЕ СУЧАСНОГО УНІВЕРСИТЕТУ”, (16), 2024. С. 134-149.
3. Переяславська С., Смагіна, О. Гейміфікація як сучасний напрям вітчизняної освіти. Відкрите освітнє е-середовище сучасного університету, 2019. С. 250-260.
4. ТКАЧЕНКО О. Гейміфікація освіти: формальний і неформальний простір. Актуальні питання гуманітарних наук, 2015. С. 303-309.
5. Кривонос О. М., Горобець С. М., Кривонос М. П., Нехаєнко К. О. Використання стратегій гейміфікації для мотивації школярів до вивчення інформатики. Вісник Житомирського державного університету імені Івана Франка. Педагогічні науки, (115), 2023. С. 65-74.
6. Duolingo URL: <https://uk.duolingo.com/> (дата звернення: 06.06.2024).
7. Inc E. Edmodo - офіційна програма в Microsoft Store. Програми Microsoft. URL: <https://apps.microsoft.com/detail/9wzdncrdsmjn?hl=uk-ua&gl=UA> (дата звернення: 06.06.2024).
8. Codecademy Review: Worth The Hype? - Learnopoly. Learnopoly. URL: <https://learnopoly.com/codecademy-review/> (date of access: 06.06.2024).
9. Leaderboard Зображення – огляд 70,407 Стокові фото, векторні зображення й відео. Adobe Stock.

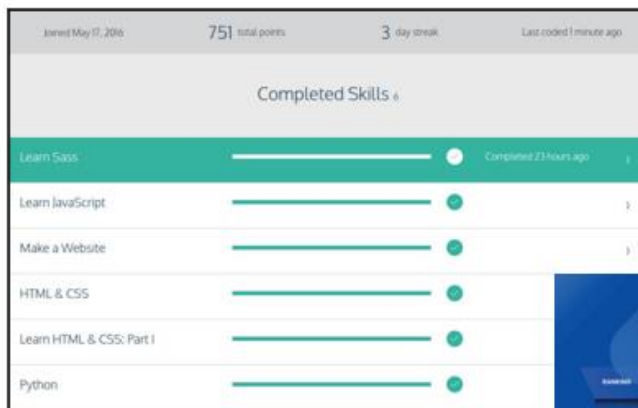
- URL: <https://stock.adobe.com/ua/search?k=leaderboard> (дата звернення: 06.06.2024).
10. Mojang. Minecraft Education – Додатки в Google Play. Android Apps on Google Play.
URL: <https://play.google.com/store/apps/details?id=com.mojang.minecraftedu&hl=uk&pli=1> (дата звернення: 06.06.2024).
 11. All About Labster Short Simulations. Labster | Virtual Labs for Universities and High Schools. URL: <https://www.labster.com/blog/about-short-virtual-labs> (date of access: 06.06.2024).
 12. The Beer Game by Zensimu - Supply chain simulation. The Beer Game by Zensimu - Supply chain simulation. URL: <https://beergameapp.com/> (date of access: 06.06.2024).
 13. Codecademy. <https://www.codecademy.com/> (date of access: 06.06.2024).
 14. Coursera | Degrees, Certificates, & Free Online Courses. Coursera. URL: <https://www.coursera.org/> (date of access: 06.06.2024).
 15. Freecodecamp. URL: <https://www.freecodecamp.org/> (дата звернення: 06.06.2024).
 16. Learn the Latest Tech Skills; Advance Your Career | Udacity. Learn the Latest Tech Skills; Advance Your Career | Udacity. URL: <https://www.udacity.com/> (date of access: 06.06.2024).
 17. Build new skills. Advance your career. edX. URL: <https://www.edx.org/> (date of access: 06.06.2024).
 18. Sololearn: Learn to Code. SoloLearn: Programmieren lernen. URL: <https://www.sololearn.com/en/> (date of access: 06.06.2024).
 19. HackerRank - Online Coding Tests and Technical Interviews. HackerRank. URL: <https://www.hackerrank.com/> (date of access: 06.06.2024).
 20. Liberty J. Programming C#: building .NET applications with C. " O'Reilly Media, Inc, 2005. С. 672

21. C# and Windows Forms. URL: <https://metanit.com/sharp/windowsforms/1.1.php> (date of access: 06.06.2024).
22. Microsoft Visual Studio Documentation – офіційна документація з використання Visual Studio. URL: <https://docs.microsoft.com/en-us/visualstudio/> (date of access: 06.06.2024).
23. Microsoft Docs – офіційна документація Microsoft з розробки на C# та .NET. URL: <https://docs.microsoft.com/en-us/dotnet/csharp/> (date of access: 06.06.2024).
24. Sells C. Windows Forms Programming in C#. Addison-Wesley Professional, 2003. 848 с.
25. Stack Overflow. URL: <https://stackoverflow.com/questions/> (date of access: 06.06.2024).
26. Troelsen A., Japikse P. Pro C# 9 with .NET 5: Foundational Principles and Practices in Programming. Apress, 2021. 1376 с.
27. Unit Testing in .NET Core and .NET Standard. URL: <https://docs.microsoft.com/en-us/dotnet/core/testing/> (date of access: 06.06.2024).
28. W3Schools tutorials. URL: <https://www.w3schools.com/> (date of access: 06.06.2024).
29. Hejlsberg A., Torgersen M., Wiltamuth S., Golde, P. The C# programming language. Pearson Education, 2008. C. 784.
30. Sharp J. Microsoft Visual C# 2013 Step by Step. Pearson Education, 2013. 82 с.
31. Troelsen A., Olsen A. Pro C# 5.0 and the .NET 4.5 Framework (Vol. 6). New York: Apress, 2012. 554 с.
32. Albahari J. C# 10 in a Nutshell : O'Reilly Media, Inc., 2022. 1060 с.
33. Amann S., Proksch S., Nadi S., Mezini M. A study of visual studio usage in practice. In 2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER), IEEE, 2016. C. 124-134.

34. Sells C., Weinhardt M. Windows forms 2.0 programming : Pearson Education, 2006. 1296 c.
35. Griffiths I., Adams M. NET Windows forms in a nutshell : O'Reilly Media, Inc., 2003. 879 c.
36. Freeman A. Windows Forms. In Introducing Visual C# 2010 (pp. 1033-1067). Berkeley, CA: Apress, 2010. C. 10.
37. Mamone M. Creating Windows Forms. Practical Mono, 2006. C. 133-178.
38. Microsoft documentation SQLite. URL: <https://learn.microsoft.com/dotnet/standard/data/sqlite/?tabs=netcore-cli> (date of access: 06.06.2024).
39. Learning C#. URL: <https://dotnet.microsoft.com/en-us/learn/csharp> (date of access: 06.06.2024).
40. C# concepts. URL: <https://learn.microsoft.com/uk-ua/dotnet/csharp/programming-guide/concepts/> (date of access: 06.06.2024).
41. C# tutorial. URL: <https://www.w3schools.com/cs/index.php> (date of access: 06.06.2024).
42. Troelsen A. C# and the. NET Platform : Apress, 2001. 1008 c.
43. Troelsen A., Japikse P. C# 6.0 and the. NET 4.6 Framework : Apress, 2015 1625 c.
44. Liberty J. Programming C#: building. NET applications with C : O'Reilly Media, Inc., 2005 672 c.

ДОДАТОК А ПРЕЗЕНТАЦІЯ

Гейміфікація у системі навчання:
аналіз, моделювання та
програмна реалізація



Приклади рішень

LEADERBOARD

RANKING	NAME	MATCHES	ALL POINTS	PLAYER POINTS	TOTAL POINTS	WIN
01	Player Name	05	20	17	09	03
02	Player Name	05	20	17	09	03
03	Player Name	05	20	17	09	03
04	Player Name	05	20	17	09	03
05	Player Name	05	20	17	09	03

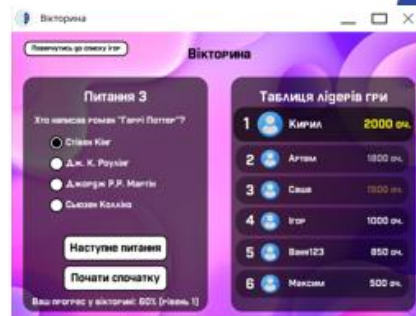
ЗАСОБИ РОЗРОБКИ

**WINDOWS
FORMS**

АРХІТЕКТУРА СИСТЕМ



РЕЗУЛЬТАТИ РОЗРОБКИ



ДОДАТОК Б ЛІСТИНГ КОДУ

```
using System;
using System.Windows.Forms;

namespace GamificationApp
{
    public partial class LoginForm : Form
    {
        public LoginForm()
        {
            InitializeComponent();
            CenterToScreen();
            StyleForm();
        }

        private void btnLogin_Click(object sender, EventArgs e)
        {
            string username = txtUsername.Text;
            string password = txtPassword.Text;

            if (UserManager.AuthenticateUser(username, password))
            {
                MainForm mainForm = new MainForm(username);
                mainForm.Show();
                this.Hide();
            }
            else
            {
                MessageBox.Show("Невірне ім'я користувача або пароль.");
            }
        }

        private void btnRegister_Click(object sender, EventArgs e)
        {
            RegistrationForm registrationForm = new RegistrationForm();
            registrationForm.Show();
        }
    }
}
```

```
private void StyleForm()
{
    this.BackColor = System.Drawing.Color.LightBlue;
    this.txtUsername.BorderStyle = BorderStyle.FixedSingle;
    this.txtPassword.BorderStyle = BorderStyle.FixedSingle;
    this.btnLogin.BackColor = System.Drawing.Color.LightGreen;
    this.btnRegister.BackColor = System.Drawing.Color.LightCoral;
    this.btnLogin.FlatStyle = FlatStyle.Flat;
    this.btnRegister.FlatStyle = FlatStyle.Flat;
    this.btnLogin.Cursor = Cursors.Hand;
    this.btnRegister.Cursor = Cursors.Hand;
}
}
}
using System;
using System.Windows.Forms;

namespace GamificationApp
{
    public partial class MainForm : Form
    {
        private string _username;

        public MainForm(string username)
        {
            _username = username;
            InitializeComponent();
            CenterToScreen();
        }

        private void btnQuiz_Click(object sender, EventArgs e)
        {
            QuizForm quizForm = new QuizForm(_username);
            quizForm.Show();
        }
    }
}
```

```

private void btnMemoryGame_Click(object sender, EventArgs e)
{
    MemoryGameForm memoryGameForm = new MemoryGameForm(_username);
    memoryGameForm.Show();
}

private void btnMathGame_Click(object sender, EventArgs e)
{
    MathGameForm mathGameForm = new MathGameForm(_username);
    mathGameForm.Show();
}
}
}
using System;
using System.Windows.Forms;

namespace GamificationApp
{
    public partial class MathGameForm : Form
    {
        private Random rand = new Random();
        private int num1;
        private int num2;
        private int correctAnswer;
        private int correctAnswersCount = 0;
        private int totalQuestions = 3;
        private int currentQuestion = 0;
        private int currentLevel = 1;
        private string _username;

        public MathGameForm(string username)
        {
            _username = username;
            InitializeComponent();
            GenerateQuestion();
            CenterToScreen();
            LoadUserProgress();
        }
    }
}

```

```

    }

    private void GenerateQuestion()
    {
        if (currentQuestion < totalQuestions)
        {
            lblQuestionCounter.Text = $"{currentQuestion + 1}/{totalQuestions}
питання (Рівень {currentLevel})";
            num1 = rand.Next(1, 20);
            num2 = rand.Next(1, 20);
            lblQuestion.Text = $"{num1} + {num2} = ?";
            correctAnswer = num1 + num2;
            currentQuestion++;
        }
        else
        {
            MessageBox.Show($"Рівень {currentLevel} завершено! Ви відповіли правильно
на {correctAnswersCount} з {totalQuestions} запитань.");
            UpdateProgressBar();
            btnRestart.Visible = true;
            btnCheckAnswer.Enabled = false;

            if (currentLevel < 3)
            {
                currentLevel++;
                currentQuestion = 0;
                correctAnswersCount = 0;
                btnCheckAnswer.Enabled = true;
                btnRestart.Visible = false;
                GenerateQuestion();
            }
            else
            {
                MessageBox.Show("Вітаємо! Ви завершили всі рівні.");
                UpdateUserProgress();
            }
        }
    }
}

```

```
private void btnCheckAnswer_Click(object sender, EventArgs e)
{
    if (int.TryParse(txtAnswer.Text, out int userAnswer))
    {
        if (userAnswer == correctAnswer)
        {
            correctAnswersCount++;
            MessageBox.Show("Правильно!");
        }
        else
        {
            MessageBox.Show("Неправильно!");
        }

        GenerateQuestion();
        txtAnswer.Text = string.Empty;
    }
    else
    {
        MessageBox.Show("Введіть коректне число.");
    }
}

private void btnRestart_Click(object sender, EventArgs e)
{
    currentQuestion = 0;
    correctAnswersCount = 0;
    currentLevel = 1;
    btnCheckAnswer.Enabled = true;
    btnRestart.Visible = false;
    GenerateQuestion();
}

private void LoadUserProgress()
{
    var progress = UserManager.GetProgress(_username);
```

```

        lblProgress.Text = $"Ваш прогрес у математичній грі:
{progress.mathGameProgress}%";
    }

    private void UpdateUserProgress()
    {
        var progress = UserManager.GetProgress(_username);
        UserManager.UpdateProgress(_username, progress.quizProgress,
progress.memoryGameProgress, progressBar.Value);
        LoadUserProgress();
    }

    private void UpdateProgressBar()
    {
        int progressValue = (int)((double)correctAnswersCount / totalQuestions *
100);
        progressBar.Value = Math.Min(progressValue, progressBar.Maximum);
    }
}

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace GamificationApp
{
    public partial class MemoryGameForm : Form
    {
        private List<string> sequence = new List<string>();
        private int currentStep = 0;
        private int correctAnswers = 0;
        private int currentLevel = 1;
        private string _username;

        public MemoryGameForm(string username)
        {

```

```
        _username = username;
        InitializeComponent();
        CenterToScreen();
        LoadUserProgress();
    }

    private void btnStart_Click(object sender, EventArgs e)
    {
        sequence.Clear();
        currentStep = 0;
        correctAnswers = 0;
        btnRestart.Visible = false;
        AddToSequence();
        ShowSequence();
    }

    private void AddToSequence()
    {
        Random rand = new Random();
        sequence.Add(rand.Next(1, 5).ToString());
    }

    private async void ShowSequence()
    {
        DisableButtons();
        var sequenceCopy = sequence.ToList();
        foreach (var step in sequenceCopy)
        {
            lblDisplay.Text = step;
            await Task.Delay(1000);
            lblDisplay.Text = string.Empty;
            await Task.Delay(500);
        }
        EnableButtons();
    }

    private void btn1_Click(object sender, EventArgs e)
```

```
{
    CheckStep("1");
}

private void btn2_Click(object sender, EventArgs e)
{
    CheckStep("2");
}

private void btn3_Click(object sender, EventArgs e)
{
    CheckStep("3");
}

private void btn4_Click(object sender, EventArgs e)
{
    CheckStep("4");
}

private void CheckStep(string step)
{
    if (sequence[currentStep] == step)
    {
        currentStep++;
        correctAnswers++;
        if (currentStep == sequence.Count)
        {
            MessageBox.Show("Правильно! Наступний раунд.");
            UpdateProgressBar();
            DisableButtons();
            currentStep = 0;
            AddToSequence();
            ShowSequence();
        }
    }
    else
    {
```

```
        MessageBox.Show("Неправильно! Гра закінчена.");
        DisableButtons();
        UpdateProgressBar();
        btnRestart.Visible = true;
        UpdateUserProgress();
    }
}

private void EnableButtons()
{
    btn1.Enabled = true;
    btn2.Enabled = true;
    btn3.Enabled = true;
    btn4.Enabled = true;
}

private void DisableButtons()
{
    btn1.Enabled = false;
    btn2.Enabled = false;
    btn3.Enabled = false;
    btn4.Enabled = false;
}

private void btnRestart_Click(object sender, EventArgs e)
{
    btnStart_Click(sender, e);
}

private void LoadUserProgress()
{
    var progress = UserManager.GetProgress(_username);
    lblProgress.Text = $"Ваш прогрес у грі на пам'ять:
{progress.memoryGameProgress}%";
}

private void UpdateUserProgress()
```

```

        {
            var progress = UserManager.GetProgress(_username);
            UserManager.UpdateProgress(_username, progress.quizProgress,
progressBar.Value, progress.mathGameProgress);
            LoadUserProgress();
        }

private void UpdateProgressBar()
{
    int progressValue = (int)((double)correctAnswers / sequence.Count * 100);
    progressBar.Value = Math.Min(progressValue, progressBar.Maximum);
}
}
}
using System;
using System.Windows.Forms;

namespace GamificationApp
{
    static class Program
    {
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new LoginForm());
        }
    }
}
using System;
using System.Windows.Forms;

namespace GamificationApp
{
    public partial class QuizForm : Form
    {

```

```

private int currentQuestionIndex = 0;
private int correctAnswers = 0;
private int currentLevel = 1;
private string _username;

private string[][] questions = new string[][]
{
    new string[] { "Столиця Франції?", "Париж", "Лондон", "Берлін", "Мадрид" },
    new string[] { "Скільки буде 5 * 6?", "30", "25", "20", "35" },
    new string[] { "Столиця України?", "Київ", "Львів", "Одеса", "Харків" },
    new string[] { "Хто написав 'Кобзар'?", "Шевченко", "Франко", "Леся
Українка", "Куліш" },
    new string[] { "Яка найбільша ріка в світі?", "Амазонка", "Ніл", "Міссісіпі",
"Волга" }
};
private int[] correctAnswersIndices = { 0, 0, 0, 0, 1 };

public QuizForm(string username)
{
    _username = username;
    InitializeComponent();
    LoadQuestion();
    CenterToScreen();
    LoadUserProgress();
}

private void LoadQuestion()
{
    if (currentQuestionIndex < questions.Length)
    {
        lblQuestionCounter.Text = $"{currentQuestionIndex + 1}/{questions.Length}
питання (Рівень {currentLevel})";
        lblQuestion.Text = questions[currentQuestionIndex][0];
        rbOption1.Text = questions[currentQuestionIndex][1];
        rbOption2.Text = questions[currentQuestionIndex][2];
        rbOption3.Text = questions[currentQuestionIndex][3];
        rbOption4.Text = questions[currentQuestionIndex][4];
    }
}

```

```

else
{
    MessageBox.Show($"Рівень {currentLevel} завершено! Ви відповіли правильно
на {correctAnswers} з {questions.Length} запитань.");
    UpdateProgressBar();
    btnRestart.Visible = true;
    btnNext.Enabled = false;

    if (currentLevel < 3)
    {
        currentLevel++;
        currentQuestionIndex = 0;
        correctAnswers = 0;
        btnNext.Enabled = true;
        btnRestart.Visible = false;
        LoadQuestion();
    }
    else
    {
        MessageBox.Show("Вітаємо! Ви завершили всі рівні.");
        UpdateUserProgress();
    }
}

private void btnNext_Click(object sender, EventArgs e)
{
    int selectedAnswer = -1;
    if (rbOption1.Checked) selectedAnswer = 0;
    if (rbOption2.Checked) selectedAnswer = 1;
    if (rbOption3.Checked) selectedAnswer = 2;
    if (rbOption4.Checked) selectedAnswer = 3;

    if (selectedAnswer == correctAnswersIndices[currentQuestionIndex])
    {
        correctAnswers++;
        MessageBox.Show("Правильно!");
    }
}

```

```

    }
    else
    {
        MessageBox.Show("Неправильно!");
    }

    currentQuestionIndex++;
    LoadQuestion();
}

private void btnRestart_Click(object sender, EventArgs e)
{
    currentQuestionIndex = 0;
    correctAnswers = 0;
    currentLevel = 1;
    btnNext.Enabled = true;
    btnRestart.Visible = false;
    LoadQuestion();
}

private void LoadUserProgress()
{
    var progress = UserManager.GetProgress(_username);
    lblProgress.Text = $"Ваш прогрес у вікторині: {progress.quizProgress}%";
}

private void UpdateUserProgress()
{
    var progress = UserManager.GetProgress(_username);
    UserManager.UpdateProgress(_username, progressBar.Value,
    progress.memoryGameProgress, progress.mathGameProgress);
    LoadUserProgress();
}

private void UpdateProgressBar()
{
    int progressValue = (int)((double)correctAnswers / questions.Length * 100);

```

```
        progressBar.Value = Math.Min(progressValue, progressBar.Maximum);
    }
}
}
using System;
using System.Windows.Forms;

namespace GamificationApp
{
    public partial class RegistrationForm : Form
    {
        public RegistrationForm()
        {
            InitializeComponent();
            CenterToScreen();
            StyleForm();
        }

        private void btnRegister_Click(object sender, EventArgs e)
        {
            string username = txtUsername.Text;
            string password = txtPassword.Text;

            if (UserManager.RegisterUser(username, password))
            {
                MessageBox.Show("Реєстрація успішна. Тепер ви можете увійти.");
                this.Close();
            }
            else
            {
                MessageBox.Show("Користувач з таким ім'ям вже існує.");
            }
        }

        private void StyleForm()
        {
            this.BackColor = System.Drawing.Color.LightBlue;
        }
    }
}
```

```

        this.txtUsername.BorderStyle = BorderStyle.FixedSingle;
        this.txtPassword.BorderStyle = BorderStyle.FixedSingle;
        this.btnRegister.BackColor = System.Drawing.Color.LightGreen;
        this.btnRegister.FlatStyle = FlatStyle.Flat;
        this.btnRegister.Cursor = Cursors.Hand;
    }
}
}
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;

namespace GamificationApp
{
    public static class UserManager
    {
        private static readonly string usersFilePath = "users.txt";
        private static readonly string progressFilePath = "progress.txt";

        static UserManager()
        {
            // Ensure the files exist
            if (!File.Exists(usersFilePath))
            {
                File.Create(usersFilePath).Close();
            }
            if (!File.Exists(progressFilePath))
            {
                File.Create(progressFilePath).Close();
            }
        }

        public static bool RegisterUser(string username, string password)
        {
            var users = File.ReadAllLines(usersFilePath);
            if (users.Any(u => u.Split(',')[0] == username))

```

```

        {
            return false; // User already exists
        }

        File.AppendAllText(usersFilePath, $"{username},{password}\n");
        File.AppendAllText(progressFilePath, $"{username},0,0,0\n"); // Initialize
progress
        return true;
    }

    public static bool AuthenticateUser(string username, string password)
    {
        var users = File.ReadAllLines(usersFilePath);
        return users.Any(u => u.Split(',')[0] == username && u.Split(',')[1] ==
password);
    }

    public static void UpdateProgress(string username, int quizProgress, int
memoryGameProgress, int mathGameProgress)
    {
        var progressList = new List<string>(File.ReadAllLines(progressFilePath));
        var userProgress = progressList.FirstOrDefault(p => p.Split(',')[0] ==
username);

        if (userProgress != null)
        {
            progressList.Remove(userProgress);

progressList.Add($"{username},{quizProgress},{memoryGameProgress},{mathGameProgress}");
            File.WriteAllLines(progressFilePath, progressList);
        }
    }

    public static (int quizProgress, int memoryGameProgress, int mathGameProgress)
GetProgress(string username)
    {
        var progressList = File.ReadAllLines(progressFilePath);
        var userProgress = progressList.FirstOrDefault(p => p.Split(',')[0] ==
username);

```

```
    if (userProgress != null)
    {
        var parts = userProgress.Split(',');
        return (int.Parse(parts[1]), int.Parse(parts[2]), int.Parse(parts[3]));
    }

    return (0, 0, 0);
}
}
```