

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет інформаційно-аналітичних технологій та менеджменту

(повна назва)

Кафедра прикладної математики

(повна назва)

## КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)

Розробка навчальної математичної

гри засобами движка Unity

(тема)

Виконав:

здобувач 2 року навчання, групи САУМ-23-1

Пархоменко В.Г.

(прізвище, ініціали)

Спеціальність 124 Системний аналіз

(код і повна назва спеціальності)

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма Системний аналіз і управління

(повна назва освітньої програми)

Керівник проф. Сидоров М.В.

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ПМ

(підпис)

Сидоров М.В.

(прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет інформаційно-аналітичних технологій та менеджменту

Кафедра прикладної математики

Рівень вищої освіти другий (магістерський)

Спеціальність 124 Системний аналіз

(код і повна назва)

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма Системний аналіз і управління

(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри ПМ \_\_\_\_\_

(підпис)

“ 25 ” листопада 2024 р.

**ЗАВДАННЯ**  
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Пархоменку Владиславу Геннадійовичу  
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка навчальної математичної гри засобами движка Unity

затверджена наказом по університету від 22 листопада 2024 р. № 1228 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 6 січня 2025 р.

3. Вихідні дані до роботи ігрове програмне забезпечення для підготовки  
абітурієнтів до національного мультипредметного тесту

4. Перелік питань, що потрібно опрацювати в роботі \_\_\_\_\_

1. Системний аналіз предметної області

2. Вибір і обґрунтування методу розв'язання

3. Програмна реалізація

4. Реалізація системи

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій \_\_\_\_\_

1. Актуальність теми роботи \_\_\_\_\_

2. Постановка задачі \_\_\_\_\_

3. Системний аналіз предметної області \_\_\_\_\_

4. Технологія розробки програмного забезпечення \_\_\_\_\_

5. Демонстрація роботи мобільного додатку \_\_\_\_\_

### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Підбір та вивчення технічної літератури за темою роботи	25 листопада – 1 грудня 2024 р.	виконано
2	Вибір та обґрунтування методу	2 – 8 грудня 2024 р.	виконано
3	Розробка алгоритму і програми	9 – 22 грудня 2023 р.	виконано
4	Проведення аналітичних досліджень та розрахунків	23 – 29 грудня 2024 р.	виконано
5	Робота над текстом пояснювальної записки	30 грудня 2024 р. – 9 січня 2025 р.	виконано
6	Представлення роботи на рецензію в ЕК	10 січня 2025 р.	виконано

Дата видачі завдання 25 листопада 2024 р.

Здобувач \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_ проф. Сидоров М.В.  
(підпис) (посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка: 62 с., 7 табл., 13 рис., 10 джерел.

АВТОНОМНІСТЬ, АРХІТЕКТУРА ПРОЕКТУ, ДОСВІД КОРИСТУВАЧА, ІГРОВИЙ ДВИЖОК, ІНТЕГРОВАНЕ СЕРЕДОВИЩЕ РОЗРОБКИ, ІНТЕРФЕЙС КОРИСТУВАЧА, МОБІЛЬНИЙ ДОДАТОК, МУЛЬТИПЛАТФОРМЕННІСТЬ, СИСТЕМНИЙ АНАЛІЗ.

Об'єкт дослідження – підготовка абітурієнтів до НМТ з математики з урахуванням сьогоденних умов навчання.

Мета роботи – розробка інтерактивного програмного продукту для підготовки до НМТ, врахувавши необхідність його автономності і доступності для кожного учня.

Методи дослідження – ігровий движок Unity 6 для розробки навчального програмного забезпечення, який підтримує більшість існуючих платформ і має широкий функціонал для реалізації інтерактивних елементів.

У даній кваліфікаційній роботі ставиться задача розробити програмний продукт під мобільні платформи для підготовки абітурієнтів до НМТ. Проведено аналіз предметної області, в ході якого і визначили мобільну платформу як оптимальне рішення. Проведено огляд найбільш популярних на поточний час мобільних технологій, надано перевагу ігровому движку Unity 6. Розглянуто основні можливості даного движку.

Описано реалізацію програмного забезпечення. Приділено увагу архітектурі проекту. Наведено детальну інформацію про інтерфейс користувача і бізнес-логіку. Реалізовано функціональність відстеження активності користувачів і звітування про помилки та збої в роботі додатку. Описано перспективи розвитку програмного продукту і виклики, які постають перед їх реалізацією.

## ABSTRACT

Introductory note: 62 pages, 7 tables, 13 figures, 10 sources.

AUTONOMY, GAME ENGINE, INTEGRATED DEVELOPMENT ENVIRONMENT, MOBILE APPLICATION, MULTIPLATFORM SUPPORT, PROJECT ARCHITECTURE, SYSTEM ANALYSIS, USER EXPERIENCE, USER INTERFACE.

Object of research – preparation of applicants for the National Multisubject Test (NMT) in mathematics considering the current educational conditions.

Purpose of work – development of an interactive software product for NMT preparation, considering the need for its autonomy and accessibility for every student.

Methods of research – the Unity 6 game engine for the development of educational software, which supports most existing platforms and offers extensive functionality for implementing interactive elements.

This qualification work aims to develop a software product for mobile platforms to prepare applicants for the National Multisubject Test (NMT). An analysis of the subject area was conducted, identifying the mobile platform as the optimal solution. A review of the most popular mobile technologies to date was carried out, with preference given to the Unity 6 game engine. The main features of this engine have also been examined.

The implementation of the software has been described with particular attention given to the project architecture. Detailed information about the user interface and business logic is provided. Functionality for tracking user activity and reporting errors and crashes in the application has been implemented. The prospects for the software product's development and the challenges associated with their implementation have also been outlined.

## ЗМІСТ

	С.
Перелік скорочень, умовних познач, одиниць і термінів .....	8
Вступ .....	9
1 Системний аналіз предметної області та постановка задач дослідження .....	10
1.1. Системний аналіз задачі розробки навчальної математичної гри засобами движка Unity .....	10
1.2 Аналіз сценаріїв вирішення задачі розробки навчальної математичної гри засобами движка Unity .....	11
1.2.1 Кваліметрична модель аналізу проблеми .....	11
1.2.2 Оцінювання вектора пріоритетів незадоволеностей методом аналізу ієрархій .....	12
1.2.3 Модель вирішення проблеми .....	16
1.3 Змістовна та формальна постановка задачі .....	17
1.3.1 Змістовна постановка задачі .....	17
1.3.2 Фомальна постановка задачі .....	18
1.4 Постановка задач дослідження .....	19
2 Вибір та обґрунтування методу розв’язання .....	20
2.1 Огляд технологій розробки мобільних додатків .....	20
2.1.1 Нативна розробка мобільних додатків .....	20
2.1.2 Кросплатформенна розробка за допомогою технології Flutter .....	20
2.1.3 Мультиплатформенна розробка на React Native .....	21
2.1.4 Розробка мобільних браузерних додатків за допомогою Ionic .....	22
2.1.5 Створення мобільних додатків інструментами Multi-platform App UI .....	23
2.1.6 Мобільна розробка за допомогою Native Script .....	24
2.1.7 Kotlin Multiplatform як альтернатива нативній розробці .....	25
2.1.8 Розробка засобами ігрового движка Unity .....	26
2.2 Обґрунтування використання движка Unity 6 .....	26

	7
Висновки за розділом 2 .....	27
3 Програмна реалізація .....	29
3.1 Редактор рівнів Unity Editor 6000.0.32f .....	29
3.2 Редактор програмного коду JetBrains Rider .....	31
Висновки за розділом 3 .....	32
4 Реалізація системи .....	33
4.1 Архітектура проєкту .....	33
4.2 Реалізація головного меню .....	34
4.3 Реалізація процесу тестування .....	35
4.3.1 Структури даних і зв'язки між ними .....	35
4.3.2 Реалізація інтерфейсу користувача у процесі тестування .....	38
4.3.3 Функціональність допомоги користувачу .....	39
4.4 Збереження прогресу .....	41
4.5 Використання сервісів Unity для підтримки і підвищення якості продукту .....	42
4.5.1 Unity Analytics як інструмент збору статистики про використання програмного продукту .....	42
4.5.2 Усунення проблем роботи додатка засобами Cloud Diagnostics .....	45
4.6 Перспективи і виклики розвитку проєкту .....	47
Висновки за розділом 4 .....	49
Висновки .....	51
Перелік джерел посилання .....	52
Додаток А Лістинг програми .....	53

**ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАК, ОДИНИЦЬ І ТЕРМІНІВ**

HMT – національний мультипредметний тест;

API – application programming interface;

DI – dependency injection;

JSON – JavaScript Object Notation;

KMP – Kotlin multiplatform;

UI – user interface;

UX – user experience.

## ВСТУП

**Актуальність теми.** Освітня система України в теперішній час зіштовхується із серйозними викликами. Дистанційне навчання, обумовлене спочатку карантинном, а потім військовим станом, обмежує можливості здобувачів прямого спілкування з викладачами. Окрім того, через проблеми в енергетичній системі в окремих регіонах або на всій території України, застосовуються відключення електроенергії. Описані проблеми призводять до обмежень у навчанні для певних категорій населення, здобувачі знаходяться в нерівних умовах. Особливо вразливою групою є учні 10-11 класів, які знаходяться в перехідному періоді від навчання в школі до навчання в університеті. Через дистанційне навчання і перебої в роботі електроенергії, а отже, зв'язку і мережі Інтернет, стає актуальним питання розробки програмного продукту, який може допомогти із усуненням пробілів у знаннях учнів і підготовкою до здачі НМТ.

**Мета і завдання кваліфікаційної роботи.** Метою кваліфікаційної роботи є розробка інтерактивного програмного продукту для підготовки до НМТ, врахувавши необхідність його автономності і доступності для кожного учня. Для досягнення поставленої мети необхідно виконати наступні завдання:

- провести огляд і аналіз сучасного стану задачі «Розробка навчальної математичної гри засобами движка Unity»;
- обрати програмний засіб для розробки;
- створити базовий концепт додатку.

*Об'єктом дослідження* є процес підготовки абітурієнтів до НМТ з математики з урахуванням сьогоденних умов навчання.

*Предметом дослідження* є метод розробки програмного забезпечення для підготовки до НМТ.

**Методи дослідження.** У роботі використовується ігровий движок Unity 6 для розробки навчального програмного забезпечення, який підтримує більшість існуючих платформ і має широкий функціонал для реалізації інтерактивних елементів.

# 1 СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ

## 1.1 Системний аналіз задачі розробки навчальної математичної гри засобами движка Unity

Задача розробки програмного забезпечення для підготовки до НМТ з математики складається з двох підзадач. В першу чергу, власне розробка функціональності продукту. Окрім того, особливу увагу необхідно приділити умовам, у яких він буде використовуватись. Системний аналіз [1] цієї задачі охоплює кілька ключових аспектів, таких як: автономність, доступність і збереження здоров'я учнів.

Навчання українських школярів у нинішній час стикається із проблемою відключення світла і відсутності стабільного з'єднання з Інтернетом. Для забезпечення безперервного навчання учні повинні мати можливість використовувати всі основні функції додатка в офлайн-режимі. Локальне збереження навчальних матеріалів і прогресу дозволяє уникнути залежності від зовнішніх ресурсів, а оптимізація [3 – 5] енерговитрат забезпечує тривалу роботу на пристроях із низьким рівнем заряду. В той же час для користувача буде корисна опція збереження результатів у хмару для резервного копіювання.

Окрім автономності, гостро постає питання доступності програмного продукту для широкої аудиторії. Програмне забезпечення має підтримувати широкий спектр функцій для учнів із особливими освітніми потребами, таких як масштабування тексту, озвучення контенту та налаштування контрастності. Окрім того, додаток повинен підтримуватися навіть на застарілих або малопродуктивних пристроях, щоб забезпечити рівний доступ до навчальних матеріалів незалежно від технічних і фінансових можливостей користувачів.

Дистанційне навчання передбачає тривале використання персонального комп'ютера або мобільних пристроїв. Це може негативно впливати на зір і поставу, також призводити до гіподинамії. Важливо зазначити, що даний фактор

виключити повністю не є можливим. Натомість ставиться задача зведення його до мінімуму. В якості прикладу програмний продукт може нагадувати користувачам про необхідність робити перерви та пропонувати вправи для очей і короткі фізичні активності. Такий підхід допоможе запобігти втомі та підтримувати фізичне і психологічне благополуччя під час підготовки до НМТ.

## 1.2 Аналіз сценаріїв вирішення задачі розробки навчальної математичної гри засобами движка Unity

### 1.2.1 Кваліметрична модель аналізу проблеми

У якості цільової платформи платформи розглядаються наступні альтернативи: веб-додаток, мобільна платформа (iOS/Android), ПК і віртуальна реальність. При порівнянні даних платформ необхідно враховувати автономність, доступність та вплив на здоров'я користувачів. Мобільні додатки для iOS/Android забезпечують найвищу автономність, оскільки можуть працювати без з'єднання з Інтернетом, зберігаючи прогрес і навчальні матеріали локально. Також вони споживають порівняно мало енергії, що важливо в умовах обмеженого доступу до електроенергії. Окрім того, мобільні пристрої є необхідним засобом зв'язку і наявні у кожного учня. В той же час, маленькі екрани мають більш шкідливий вплив на зір у порівнянні із персональними комп'ютерами. Веб-додатки доступні для будь-якого пристрою з браузером і не споживають пам'яті пристрою за відсутності потреби встановлення. Однак вони зазвичай потребують постійного підключення до Інтернету, що свідчить про їхню недостатню автономність. Програмне забезпечення для ПК забезпечує більшу продуктивність і може бути корисним у стаціонарних умовах, але обмежує мобільність і доступність через потребу в постійному робочому місці. Якщо ж розглядати ноутбуки, то їхні батареї також обмежені, самі ж пристрої споживають порівняно багато ресурсу. Віртуальна реальність пропонує високий рівень інтера-

ктивності і має всі інструменти для того, щоб досягти максимальної реалістичності навчального процесу. В той же час мають місце суттєві обмеження у вигляді високої вартості обладнання, необхідності стабільного джерела живлення і значного навантаження на зір і вестибулярну систему користувачів, що може негативно впливати на здоров'я.

Оберемо оптимальну цільову платформу для розробки програмного продукту, виходячи з наступних критеріїв:

- критерій 1 (К1): автономність;
- критерій 2 (К2): оптимальне споживання ресурсу;
- критерій 3 (К3): доступність ціни;
- критерій 4 (К4): інтерактивність;
- критерій 5 (К5): мінімізація негативного впливу на здоров'я.

У якості цільових платформ розглянемо наступні альтернативи:

- альтернатива 1 (А1): веб-додаток;
- альтернатива 2 (А2): мобільна платформа;
- альтернатива 3 (А3): персональний комп'ютер;
- альтернатива 4 (А4): пристрій віртуальної реальності.

На рисунку 1.1 зображено ієрархічну модель проблеми вибору за умов багатокритеріальності.

### 1.2.2 Оцінювання вектора пріоритетів незадоволеностей методом аналізу ієрархій

Продовжуючи аналіз сценаріїв, побудуємо матриці попарних порівнянь [2] критеріїв, а також альтернатив за кожним з критеріїв. Матрицю попарних порівнянь критеріїв і обчислення з нею наведено у таблиці 1.1.

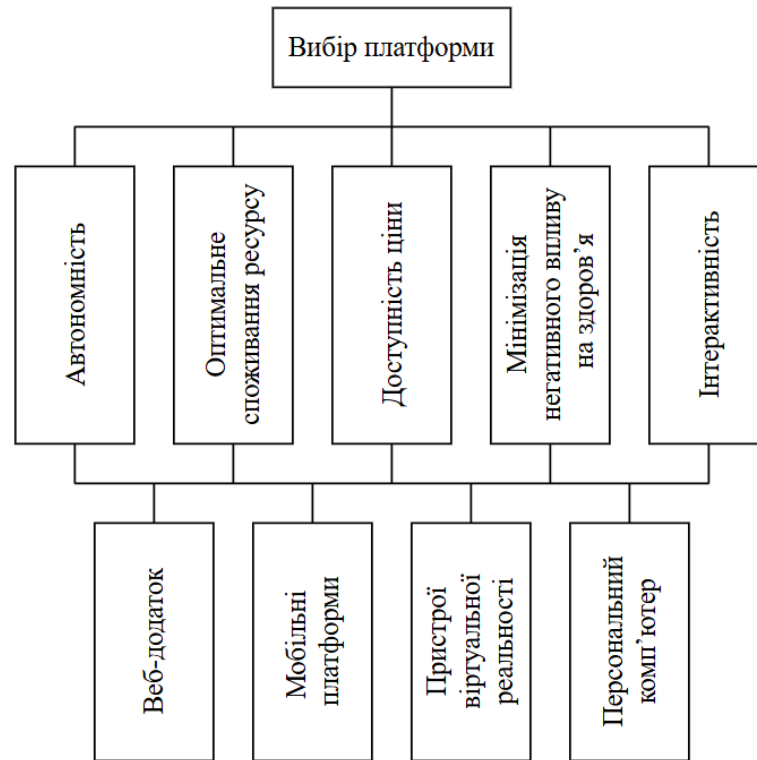


Рисунок 1.1 – Ієрархічна модель аналізу проблеми

Таблиця 1.1 – Матриця попарних порівнянь критеріїв

Критерії оцінювання	К1	К2	К3	К4	К5	Оцінки компонентів	Вектор пріоритетів	Величина значущості
К1	1	3	3	7	1	2,290	0,350	1,793
К2	$\frac{1}{3}$	1	$\frac{1}{3}$	7	$\frac{1}{3}$	0,763	0,117	0,657
К3	$\frac{1}{3}$	3	1	5	$\frac{1}{3}$	1,108	0,169	0,929
К4	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{1}{5}$	1	$\frac{1}{5}$	0,241	0,037	0,203
К5	1	3	3	5	1	2,141	0,327	1,719
Усього						6,543		5,301

Для таблиці 1.1 індекс узгодженості  $CI = \frac{5,301 - 5}{5 - 1} = 0,075$ , випадковий

індекс  $RI = 1,12$ , відносна узгодженість  $CR = \frac{CI}{RI} = \frac{0,075}{1,12} = 0,067$ .

Таблиця 1.2 – Порівняння за першим критерієм

K1	A1	A2	A3	A4	Власний вектор	Вектор пріоритетів	Величина значущості
A1	1	$\frac{1}{5}$	$\frac{1}{8}$	$\frac{1}{3}$	0,302	0,051	0,257
A2	51	1	7	5	3,637	0,618	2,794
A3	8	$\frac{1}{7}$	1	$\frac{1}{3}$	0,786	0,134	0,698
A4	3	$\frac{1}{5}$	3	1	1,158	0,197	0,875
Усього					5,883		4,625

Таблиця 1.3 – Порівняння за другим критерієм

K2	A1	A2	A3	A4	Власний вектор	Вектор пріоритетів	Величина значущості
A1	1	3	5	5	2,942	0,507	2,280
A2	$\frac{1}{3}$	1	7	7	2,010	0,346	1,544
A3	$\frac{1}{5}$	$\frac{1}{7}$	1	3	0,541	0,093	0,405
A4	$\frac{1}{5}$	$\frac{1}{7}$	$\frac{1}{3}$	1	0,312	0,054	0,236
Усього					5,806		4,465

Для прийняття рішення про використання методу необхідно провести порівняльний аналіз альтернатив. Оцінивши їх щодо кожного з критеріїв, отрима-

ємо дані, які представлені в таблицях 1.2 – 1.6. Випадковий індекс для матриць четвертого порядку дорівнює  $RI = 0,9$ .

Для таблиці 1.2 маємо  $CI = 0,208$ ,  $CR = 0,231$ . Для таблиці 1.3 маємо  $CI = 0,155$ ,  $CR = 0,172$ . Для таблиці 1.4 маємо  $CI = 0,078$ ,  $CR = 0,086$ . Для таблиці 1.5 маємо  $CI = 0,099$ ,  $CR = 0,110$ . Для таблиці 1.6 маємо  $CI = 0,112$ ,  $CR = 0,125$ .

Таблиця 1.4 – Порівняння за третім критерієм

К3	A1	A2	A3	A4	Власний вектор	Вектор пріоритетів	Величина значущості
A1	1	1	3	3	1,732	0,315	1,345
A2	1	1	7	9	2,817	0,512	2,156
A3	$\frac{1}{3}$	$\frac{1}{7}$	1	3	0,615	0,112	0,472
A4	$\frac{1}{3}$	$\frac{1}{9}$	$\frac{1}{3}$	1	0,333	0,061	0,260
Усього					5,497		4,233

Таблиця 1.5 – Порівняння за четвертим критерієм

К4	A1	A2	A3	A4	Власний вектор	Вектор пріоритетів	Величина значущості
A1	1	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{8}$	0,2666	0,048	0,199
A2	5	1	4	$\frac{1}{3}$	1,607	0,287	1,274
A3	5	$\frac{1}{4}$	1	$\frac{1}{3}$	0,803	0,144	0,627
A4	8	3	3	1	2,913	0,521	2,196
Усього					5,589		4,296

Таблиця 1.6 – Порівняння за п'ятим критерієм

K5	A1	A2	A3	A4	Власний вектор	Вектор пріоритетів	Величина значущості
A1	1	1	1	5	1,495	0,283	1,202
A2	1	1	5	8	2,515	0,477	2,111
A3	1	$\frac{1}{5}$	1	5	1,000	0,190	0,820
A4	$\frac{1}{5}$	$\frac{1}{8}$	$\frac{1}{5}$	1	0,266	0,050	0,205
Усього					5,276		4,337

### 1.2.3 Модель вирішення проблеми

Одержавши результати порівнянь критеріїв, як особа, що приймає рішення, здійснимо остаточні розрахунки і зробимо висновок. Згідно результатам, наведеним у таблиці 1.7, за оптимальну обираємо другу альтернативу, яка полягає у розробці навчального програмного продукту під мобільні платформи, такі як iOS та Android.

Таблиця 1.7 – Остаточні розрахунки

K A	K1	K2	K3	K4	K5	Узагальнені пріоритети
A1	0,051	0,507	0,315	0,048	0,283	1,472
A2	0,618	0,346	0,512	0,287	0,477	3,338
A3	0,134	0,093	0,112	0,144	0,190	0,941
A4	0,197	0,054	0,061	0,521	0,050	0,793

## 1.3 Змістовна та формальна постановка задачі

### 1.3.1 Змістовна постановка задачі

Задача даної роботи передбачає розробку навчального програмного продукту з урахуванням сучасних умов життя в Україні. Перед освітньою системою стоять складні економічні та соціальні виклики. Наслідки пандемії та військового стану значно вплинули на доступність та якість освіти. Велика кількість абітурієнтів не має стабільного доступу до навчальних ресурсів. Можливість підготовки з кваліфікованими викладачами також доступна не всім. Це ускладнює процес підготовки до Національного мультипредметного тесту (НМТ), і відповідно, ставить під загрозу вступ абітурієнта до закладу вищої освіти.

Особливо актуальною проблемою постає підготовка з математики. Дана дисципліна вимагає регулярної практики розв'язання задач різного рівня складності. Також має місце потреба у розумінні фундаментальних концепцій. Мобільний додаток для підготовки до НМТ з математики має на меті забезпечити абітурієнтів інструментом, який надає доступ до навчальних матеріалів та інтерактивних тестів. Окрім того, корисною опцією може бути персоналізований фідбек, який би вказував на прогалини у знаннях абітурієнта з рекомендаціями додаткової практики з окремо взятих розділів. Особливої уваги потребує проблема обмеженого доступу до Інтернету та електроенергії. Варто зазначити, що запас енергії мобільних пристроїв можна порівняно легко поповнювати, використовуючи зарядні станції або портативні зарядні пристрої. При цьому необхідно пам'ятати, що запас заряду в поточній ситуації критично важливий, отже, потрібно забезпечити мінімальний рівень споживання ресурсу пристроїв. Завдяки мобільності, доступності та адаптивності програмний продукт може стати ефективною підтримкою для учнів, забезпечуючи їх навчальними ресурсами та підтримкою навіть за складних умов.

### 1.3.2 Формальна постановка задачі

Ставиться задача розробити мобільний додаток для підготовки до НМТ з математики. Функціональні вимоги до нього будуть наступними:

- доступ до навчальних матеріалів з математики, які включають основні теми, передбачені програмою НМТ;
- можливість проходження тестів та інтерактивних вправ з математики, адаптованих до формату НМТ;
- збереження прогресу користувача і надання аналітики щодо результатів тестування з включенням рекомендацій із подальшого навчання.

Також програмне забезпечення має відповідати нефункціональним вимогам:

- підтримка роботи в офлайн-режимі, що полягає в наданні доступу до основних функцій без підключення до Інтернету;
- інтуїтивний інтерфейс застосунку, зручний для користувачів різного віку та рівня технічної підготовки;
- сумісність додатка з різними операційними системами (Android, iOS) та його енергоефективність для роботи на пристроях із обмеженим ресурсом батареї.

Окрім того, необхідно брати до уваги наступні обмеження та припущення:

- використання додатку на пристроях середнього та нижчого класу, тобто необхідність врахування потенційних обмежень продуктивності та пам'яті;
- забезпечення інклюзивності, що полягає в реалізації елементів доступності, таких як контрастність, масштабування тексту, озвучення тощо;
- можливість подальшої масштабованості та розширення функціоналу, зокрема інтеграції нових тем і типів тестових завдань, розширення на інші платформи.

Розробка вважатиметься успішною, якщо додаток відповідає встановленим вимогам та пройде успішне тестування серед абітурієнтів, показавши підвищення рівня їхньої підготовки з математики та позитивні відгуки щодо зручності та функціональності.

#### 1.4 Постановка задач дослідження

Проведений системний аналіз предметної області показав, що оптимальним вибором з точки зору автономності, доступності і впливу на здоров'я учнів, є програмне забезпечення для мобільних платформ. В той же час не виключається можливість поширення на інші платформи в майбутньому.

Метою дослідження є розробка мобільного додатка для підготовки до НМТ з математики, який забезпечує інтерактивне навчання та тестування в офлайн-режимі, доступність для широкої аудиторії (включаючи учнів із різним рівнем технічних можливостей і особливими освітніми потребами) та сприяє збереженню фізичного і психічного здоров'я.

Для досягнення поставленої мети необхідно виконати наступні завдання:

- провести огляд і аналіз сучасного стану задачі «Розробка навчальної математичної гри засобами движка Unity»;
- обрати програмний засіб для розробки;
- створити базовий концепт додатку.

Виконання зазначених завдань передбачає реалізацію наступних етапів:

- провести системний аналіз предметної області;
- проаналізувати вимоги цільових користувачів програмного продукту;
- врахувати обмеження, з якими можуть мати місце у користувачів;
- обрати технологію та інструмент розробки програмного забезпечення;
- розробити функціональність і інтерфейс користувача;
- провести тестування на цільовій аудиторії, зібрати відгуки користувачів;
- обробити відгуки, внести зміни відповідно до побажань користувачів, усунути наявні дефекти;
- подальша підтримка програмного продукту, інтеграція додаткової функціональності.

## 2 ВИБІР ТА ОБҐРУНТУВАННЯ МЕТОДУ РОЗВ'ЯЗАННЯ

### 2.1 Огляд технологій розробки мобільних додатків

#### 2.1.1 Нативна розробка мобільних додатків

Розробка нативного [7] (рідного для платформи) мобільного додатка полягає у створенні програмного забезпечення, яке максимально інтегрується з операційною системою, такою як Android або iOS. Основною перевагою цього підходу є висока продуктивність додатка, адже він використовує всі можливості апаратного забезпечення пристрою. Окрім того, нативні додатки забезпечують покращений досвід користувача завдяки доступу до повного спектру функцій пристрою, таких як камера, GPS, сенсори руху тощо. Інтерфейс користувача в даному випадку відповідає стандартам і рекомендаціям конкретної платформи, це робить його зручним та інтуїтивно зрозумілим.

В той же час процес розробки нативних додатків має і певні недоліки. Основний із них – необхідність створення окремих версій для кожної платформи. Це може значно збільшити час і витрати на розробку. Наприклад, для платформи Android використовуються такі мови програмування, як Java або Kotlin, а для iOS – Swift або Objective-C. Частіше за все це вимагає залучення спеціалізованих команд розробників для підтримки кожної версії, що може ускладнити процес оновлення та випуску нових функцій. Високі витрати на підтримку нативних додатків роблять їх менш привабливими для невеликих проєктів або команд із обмеженими ресурсами.

#### 2.1.2 Кросплатформенна розробка за допомогою технології Flutter

Використання Flutter дозволяє створювати додатки для різних операційних систем, використовуючи єдину кодову базу. Flutter підтримує не лише мо-

більні платформи Android та iOS, але й веб-додатки, десктопні програми для Windows, macOS і Linux, що робить його універсальним інструментом для мультиплатформенної розробки. Основною мовою програмування є Dart, яка забезпечує високу продуктивність і зручний синтаксис для розробників. Також до ключових переваг Flutter можна віднести використання власного графічного рушія, який дозволяє створювати візуально привабливі інтерфейси з плавними анімаціями, які виглядають однаково на всіх підтримуваних платформах.

Попри значні переваги, Flutter має і свої обмеження. Хоча продуктивність додатків, створених за допомогою цієї технології, достатньо висока, вона може поступатися нативним додаткам у сценаріях, які вимагають інтенсивного використання апаратних ресурсів. Реалізація деяких специфічних функцій платформи може вимагати написання додаткового коду на нативних мовах, таких як Java/Kotlin для Android або Swift/Objective-C для iOS. Залежність від технології також несе певні ризики у разі зміни його підтримки або оновлень.

### 2.1.3 Мультиплатформенна розробка на React Native

Facebook пропонує кросплатформенну технологію React Native, яка аналогічно Flutter, дозволяє створювати мобільні додатки для різних платформ з використанням однієї кодової бази. Основною мовою програмування в React Native є JavaScript, завдяки чому полегшується робота для розробників, які мають досвід у веб-розробці. В основі React Native лежить компонентний підхід. Даний підхід спрощує процес створення, повторного використання коду та управління інтерфейсом користувача. Зручним інструментом даної технології є живе перезавантаження (Hot Reloading), яке прискорює процес розробки, дозволяючи одразу відстежити зміни в коді без необхідності повторного запуску додатка. Велика спільнота розробників створює численні бібліотеки, плагіни та документацію, що спрощує розв'язання багатьох задач.

Однак React Native має і певні недоліки. Як для будь-якої кросплатфор-

менної технології, додатки, створені його засобами, можуть поступатися нативним у продуктивності. Особливо гостро проблема постає в сценаріях зі складними графічними або обчислювальними задачами. Крім того, доступ до деяких функцій платформи може вимагати написання нативного коду окремо для iOS і Android, що частково знижує переваги кросплатформенності. Також залежності від JavaScript-движка та бібліотек призводять до того, що розмір додатків на React Native зазвичай більший. Таким чином, у великих проєктах можуть виникати труднощі з масштабуванням і оптимізацією.

#### 2.1.4 Розробка мобільних браузерних додатків за допомогою Ionic

Ionic – це кросплатформенна технологія для розробки мобільних додатків, яка базується на веб-технологіях, таких як HTML, CSS і JavaScript. Вона дозволяє створювати додатки для iOS, Android і веб-платформ. Завдяки використанню аналогічного коду для веб-розробки, Ionic доволі легко інтегрується з фреймворками Angular, React або Vue. Таким чином, можна відзначити зручність у роботі для розробників із відповідним досвідом. Для відображення інтерфейсу користувача використовується WebView, що робить додатки схожими на нативні, але фактично вони працюють через браузерний механізм пристрою. Завдяки використанню веб-технологій розробники можуть швидко створювати та змінювати інтерфейси, використовуючи звичні інструменти HTML, CSS і JavaScript. Завдяки широкій бібліотеці готових компонентів спрощується створення UI, забезпечуючи консистентність дизайну між платформами. Таким чином, розробка додатків за допомогою даної технології коштує дешево. Крім того, Ionic має велику спільноту користувачів і добре документовану екосистему, що скорочує час розробки.

Проте Ionic має і свої недоліки. Оскільки додатки працюють через WebView, їхня продуктивність може бути нижчою порівняно з нативними рішеннями. Це стає особливо помітним у в сценаріях із великими графічними або

обчислювальними навантаженнями. Залежність від браузерного механізму також може викликати затримки у взаємодії, що негативно впливає на плавність роботи додатка. Крім того, для доступу до деяких нативних функцій платформи може знадобитися використання спеціальних плагінів або написання нативного коду. Якщо ставиться задача розробки складних додатків із високими вимогами до продуктивності або інтеграції з платформами, варто розглянути інші підходи.

### 2.1.5 Створення мобільних додатків інструментами Multi-platform App UI

Порівняно новим інструментом виступає .NET MAUI (Multi-platform App UI) – кросплатформенна технологія від Microsoft, яка дозволяє створювати додатки для iOS, Android, Windows і macOS. MAUI є наступником технології Xamarin.Forms, яка на поточний час застаріла, і базується на .NET, забезпечуючи високу інтеграцію з екосистемою Microsoft. Основною мовою програмування для .NET MAUI є C#, що робить цю технологію зручною для розробників, які вже мають досвід з .NET. Тісна інтеграція з .NET і підтримка сучасних інструментів розробки, таких як Visual Studio, є однією з головних переваг. Це забезпечує зручний і продуктивний робочий процес для розробників, включаючи доступ до потужного набору бібліотек і функцій для роботи з базами даних, веб-службами, графікою тощо. завдяки кросплатформенності можливо суттєво скоротити час і витрати на розробку, зберігаючи можливість створення додатків із нативним виглядом і продуктивністю. Ще одним важливим аспектом є підтримка адаптивного інтерфейсу, тобто оптимізація дизайну для різних пристроїв і екранів.

Серед недоліків .NET MAUI можна виділити те, що екосистема плагінів та бібліотек на даний час не настільки розвинута, як у інших популярних фреймворків, таких як React Native або Flutter. Це обумовлено тим, що технологія є порівняно новою. Крім того, продуктивність може бути трохи нижчою порівня-

но з нативними додатками, особливо для складних завдань, що потребують інтенсивного використання апаратних ресурсів. Підтримка платформ теж має деякі обмеження. Наприклад, під Linux технологія наразі не адаптована, що може бути критичним для певних проєктів.

### 2.1.6 Мобільна розробка за допомогою Native Script

NativeScript – це кросплатформенна технологія для створення мобільних додатків, яка дозволяє використовувати одну кодову базу для розробки додатків для мобільних платформ. Основною особливістю NativeScript є можливість безпосередньо працювати з нативними API платформ, використовуючи мови програмування JavaScript і TypeScript. Окрім того, є доступною підтримка Angular. Це дозволяє отримати продуктивність, близьку до нативних додатків, зберігаючи зручність використання веб-технологій для розробки. Однією з головних переваг NativeScript є можливість прямого доступу до нативних компонентів платформи без необхідності використання спеціальних плагінів. Це дає змогу розробникам викликати функції операційної системи, створювати користувацькі компоненти та інтегрувати додаток з нативними функціями пристрою, такими як камера, GPS або датчики. Також зручною опцією є те, що вихідний код даної платформи знаходиться у відкритому доступі. Таким чином, кожен розробник має можливість змінювати його під свої потреби.

До недоліків можна віднести те, що продуктивність додатків, створених на основі цієї технології, може поступатися нативним рішенням, особливо в сценаріях із високими вимогами до графіки чи обчислювальних ресурсів. Для роботи з нативними API розробники повинні мати впевнені знання платформ Android та iOS, що може стати викликом для новачків. Крім того, хоча екосистема NativeScript активно розвивається, вона все ще менша порівняно з такими технологіями, як React Native або Flutter.

### 2.1.7 Kotlin Multiplatform як альтернатива нативній розробці

Хоча мова програмування Kotlin в самому початку передбачалася для нативної розробки під платформу Android в якості альтернативи мові Java, не так давно її використання почало поширюватися і на інші операційні системи. Розроблена компанією JetBrains, технологія для кросплатформенної розробки Kotlin Multiplatform дозволяє використовувати одну кодову базу для створення додатків для різних платформ, таких як Android, iOS, веб і десктоп. Основною мовою програмування є Kotlin, яка забезпечує зрозумілий синтаксис, схожий на Java, високий рівень безпеки типів і зручність у використанні. КМР дозволяє спільно використовувати логіку додатка (наприклад, роботу з API, обчислення, обробку даних), при цьому залишаючи розробку інтерфейсу користувача окремою для кожної платформи, щоб забезпечити нативний вигляд і функціональність. Таким чином, головною перевагою Kotlin Multiplatform можна вважати гнучкість у підході до розробки. Це дозволяє досягти балансу між повторним використанням коду та підтримкою нативного вигляду додатків. Kotlin Multiplatform також інтегрується з екосистемами Android і iOS, що забезпечує доступ до їхніх SDK та інших інструментів. Ще одним плюсом є сумісність із популярними бібліотеками, такими як Ktor для роботи з мережею чи Kotlinx Serialization для обробки даних.

Серед недоліків Kotlin Multiplatform можна виділити наступні. Незважаючи на тісну інтеграцію з цільовими платформами, розробка інтерфейсу користувача все ще потребує написання окремого коду для кожної платформи, що може збільшити час і витрати на реалізацію. Також оскільки технологія перебуває в стадії активного розвитку, можливі зміни в її API або обмежена документація для окремих сценаріїв використання, тобто неможливо гарантувати стабільність роботи додатків. Крім того, початок роботи з КМР може бути складнішим для розробників без досвіду роботи з Kotlin

### 2.1.8 Розробка засобами ігрового движка Unity

Кросплатформенна розробка за допомогою ігрового движка Unity дозволяє створювати інтерактивні додатки та ігри для широкого спектра платформ, таких як мобільні (Android, iOS), стаціонарні (Windows, macOS, Linux), веб (WebGL), а також ігрові консолі (PlayStation, Xbox, Nintendo Switch) і пристрої віртуальної та доповненої реальності (AR/VR). Основною мовою програмування в Unity є C# (C-Sharp), що забезпечує гнучкість у розробці та зручний синтаксис. Також C# має вбудований механізм «прибирання сміття», завдяки якому зникає необхідність ручного очищення пам'яті. Ігровий рушій Unity пропонує потужні інструменти обробки графіки, анімації, фізики та інших складних обчислень. Окрім цього, доступний візуальний редактор для створення сцен та інтерфейсу користувача, що значно полегшує розробку достатньо складних проєктів.

Водночас на розробку за допомогою засобів Unity накладаються певні обмеження. Програмне забезпечення, створене інструментами цього движка, потребують додаткової оптимізації порівняно з нативними рішеннями, особливо на мобільних пристроях із обмеженими ресурсами. Окрім того, остаточний розмір продукту може бути більшим, що важливо враховувати для платформ із обмеженим обсягом пам'яті.

### 2.2 Обґрунтування використання движка Unity 6

Останньою версією движка на поточний час є Unity 6 [8], яка містить суттєві покращення і нововведення у порівнянні з попередніми. Одним із ключових удосконалень є впровадження наскрізних багатокористувацьких робочих процесів, тобто забезпечує повну інтеграцію всіх компонентів, необхідних для роботи з багатокористувацьким функціоналом, починаючи від налаштування мережевої інфраструктури і закінчуючи обробкою дій користувачів у реально-

му часі. Нова система мультиплеєрної підтримки значно спрощує створення додатків з онлайн-функціоналом, зменшує час інтеграції та ітерацій, що особливо важливо для мобільних платформ. Це дозволяє швидше запускати проекти на ринок, зберігаючи їхню якість.

Визначною зміною є оптимізація графічного движка. Unity 6 представила такі інструменти, як GPU Resident Drawer та Render Graph для Universal Render Pipeline (URP), які дозволяють перенести частину обчислень з центрального процесора на графічний. Це забезпечує значне підвищення продуктивності, особливо в ресурсомістких додатках, які вимагають високої деталізації та плавної графіки.

Особливої уваги заслуговує інструмент Sentis, який дозволяє легко інтегрувати штучний інтелект у додатки. Це відкриває нові можливості для створення адаптивного контенту, що реагує на дії користувача в реальному часі. Завдяки Sentis можна розробляти інтелектуальні сценарії, які роблять ігровий процес або взаємодію з додатком більш динамічною та персоналізованою. Це є суттєвим кроком вперед у напрямку інтеграції сучасних технологій штучного інтелекту.

Якщо говорити про сумісність зі старими версіями, то перенесення проєктів на Unity 6 проходить достатньо легко. Більшість функціональності, розробленої на попередній версії, підтримується і на поточній. Даний факт є важливим, оскільки Unity має велику спільноту розробників і доступ до широкої екосистеми плагінів та бібліотек. Це дозволяє швидко знаходити рішення для типових задач і скорочує час на реалізацію нестандартного функціоналу.

## Висновки за розділом 2

Даний розділ було присвячено аналізу різних технологій для розробки мобільних додатків. Умовно їх можна поділити на дві групи – нативні та кросплатформенні, деякі з них можна вважати гібридними. У ході дослідження розглянули як сильні, так і слабкі сторони кожної з технологій.

З боку оптимальної роботи на мобільних пристроях найкраще себе позиціонують нативні додатки. Вони демонструють найвищу продуктивність, забезпечують швидкий відгук користувацького інтерфейсу та доступ до всіх функцій мобільних пристроїв. В той же час нативна розробка займає багато часу, потребує залучення більшої кількості фахівців для розробки програмного продукту, а в подальшому – його підтримки. З цього також випливає більша вартість роботи.

З оглядом на обмеженість в часі, було обрано кросплатформенне рішення, а саме ігровий движок Unity. Не на останньому місці стоїть на меті створення програмного продукту з ігровими елементами, для чого Unity найбільше підходить серед всіх розглянутих варіантів. Цей інструмент дозволяє розробляти кросплатформенні додатки з високим рівнем інтерактивності та багатими можливостями для впровадження ігрових механік. Завдяки цьому можна забезпечити оптимальний баланс між продуктивністю, зручністю розробки та відповідністю потребам проєкту.

## 3 ПРОГРАМНА РЕАЛІЗАЦІЯ

### 3.1 Редактор рівнів Unity Editor 6000.0.32f

Станом на поточний момент часу, останньою стабільною версією движка є 6000.0.32f. Інтерфейс редактора Unity є інтуїтивно простим. Відлагодження гри можливе прямо всередині нього без необхідності запуску на цільовому пристрої. Движок підтримує написання скриптів мовою програмування C#. Раніше підтримувалися також Boo (діалект Python, підтримку прибрати у 5-ій версії) і модифікація JavaScript, відома як UnityScript (підтримку припинено у версії 2017.1). Розрахунки фізики здійснює фізичний движок PhysX від NVIDIA для 3D фізики і Box2D для 2D фізики. Функції графічного API виконують DirectX і OpenGL.

Проект гри в Unity можна поділити на сцени (рівні) – окремі файли, які містять в собі ігрові світи зі своїм набором об'єктів, сценаріїв і налаштувань. Сцени можуть містити в собі як, власне, ігрові об'єкти (моделі), так і порожні ігрові об'єкти. Кожен об'єкт у свою чергу містить набори компонентів, з якими взаємодіють скрипти. Скрипти, створені розробником, аналогічно прикріплюються у вигляді компонентів. Також об'єкти мають назву, при цьому назва абсолютно не валідується – може містити будь-яку комбінацію символів, бути порожньою або дублювати назву іншого об'єкта. Для зручності об'єктам можна призначати тег (мітку) і фізичний шар, на якому він може відобразитися. Так, кожен об'єкт на сцені обов'язково містить компонент Transform, який зберігає в собі координати місцезнаходження (прямокутна декартова система), повороти (кути Ейлера) і розмірів об'єкту по всіх трьох координатних осях. На рисунку 3.1 зображено приклад розміщення об'єктів на сценах.

В редакторі наявна система наслідування об'єктів; дочірні об'єкти повторюють всі зміни позиції, повороту і масштабу батьківського об'єкта. При цьому координати дочірнього об'єкта вказуються відносно безпосереднього об'єкта-батька. Об'єкти, які знаходяться в корені сцени, мають абсолютні координати.

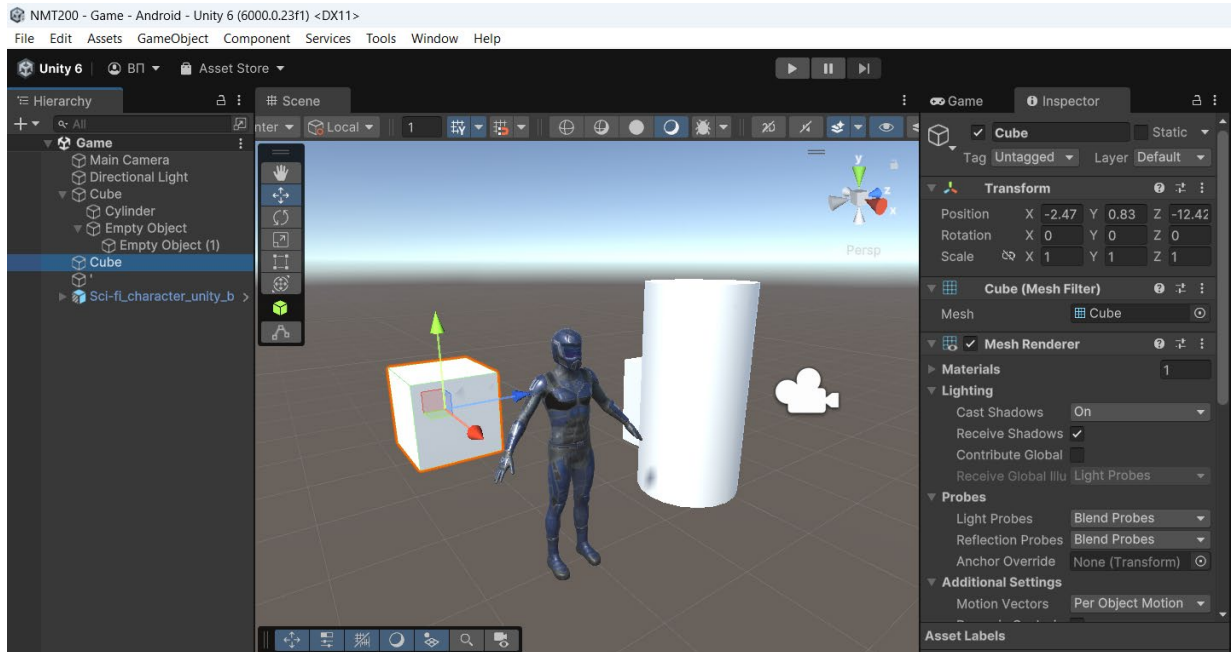


Рисунок 3.1 – Розміщення об’єктів на сцені Unity

Також Unity підтримує фізику твердих тіл і тканин, а також фізику типу Ragdoll (ганчіркова лялька).

При імпортуванні текстур в Unity можливо згенерувати альфа-канал (можливість надання прозорості об’єкта), карту нормалей, карту світла, карту відображень. При цьому саму текстуру безпосередньо приєднати до моделі неможливо. Необхідно створити матеріал, до якого буде призначено шейдер. Далі цей матеріал прикріплюється до моделі. Редактор Unity підтримує написання і редагування шейдерів. Unity має також компонент для створення анімації. При цьому кращим підходом є створення анімацій у спеціалізованих 3D-середовищах для моделювання, таких як Blender або 3D Max, потім готову анімацію можна імпортувати в Unity проект разом із моделлю.

Unity підтримує систему рівнів деталізації. Її сутність полягає в тому, щоб на далекій відстані від гравці високодеталізовані моделі замінюються на низькодеталізовані, і навпаки. Таким чином, оптимізується навантаження на графічний процесор. Окрім того, корисним є механізм вибракування оклюзії, який полягає в тому, що для об’єктів, які не потрапляють в поле зору камери, не візуалізуються геометрія і колізії, що також оптимізує проект.

При компіляції проєкту створюється папка, яка містить виконуючий файл гри (.exe), якщо мова йде про Windows. Для платформи Android можна створити файл з розширенням .apk для встановлення безпосередньо на мобільний пристрій, або .aab-файл для публікації в Google Play. Для iOS, створюється папка нативного проєкту, яку в подальшому необхідно запустити в інтегрованому середовищі розробки Xcode, яка доступна виключно для операційної системи macOS.

Движок підтримує велику кількість популярних форматів медіа. Моделі, звуки, відео, текстури, матеріали і скрипти можна запакувати в спеціальні файли формату .unitypackage і передавати іншим розробникам або розповсюджувати у вільному доступі. Такий самий формат використовується у внутрішньому магазині Unity Asset Store, в якому розробники можуть безкоштовно або за гроші публікувати в загальний доступ різні елементи, які можуть бути корисними при створенні ігор.

### 3.2 Редактор програмного коду JetBrains Rider

Незважаючи на широкий інструментарій Unity редактора, всередині нього не підтримується написання програмного коду. Для цієї задачі використовуються такі інтегровані середовища розробки, як наприклад, Microsoft Visual Studio, Visual Studio Code або JetBrains Rider.

В якості зовнішнього редактору коду обрано JetBrains Rider. Rider пропонує спеціалізовані функції для розробників, що працюють із C#-кодом, які включають швидкий рефакторинг, підсвічування помилок у режимі реального часу, інструменти для автоматичного завершення коду, а також інтеграцію з Unity Debugger для відлагодження.

Однією з відмінних рис даного інтегрованого середовища розробки є найповніша інтеграція з движком Unity. Rider дозволяє швидко перемикатися між Unity і кодом, автоматично оновлюючи проєкт після збереження змін. Крім

того, середовище має інтуїтивно зрозумілий інтерфейс і підтримує різні платформи, включаючи Windows, macOS і Linux. Завдяки оптимізації для великих проектів, Rider працює швидше та споживає менше ресурсів у порівнянні з багатьма іншими IDE. Також зручним інструментом є можливість перевірки посилянь в коді на об'єкти на сценах Unity редактора.

На відміну від Microsoft Visual Studio або Visual Studio Code, Rider є платним програмним забезпеченням. Це може накладати певні обмеження на його використання. В той же час JetBrains надає безкоштовну ліцензію для студентів вищих навчальних закладів, що підвищує продуктивність навчання здобувачів вищої освіти.

### Висновки за розділом 3

В даному розділі наведено огляд функціональності ігрового движка Unity, а саме робота з графікою, фізикою, моделями і анімаціями. Зручна компонентна система спрощує розробку архітектури проекту. Вбудовані засоби оптимізації проектів є незамінними при створенні ігор, оскільки економлять час розробників, який було б необхідно витратити для ручної реалізації оптимальної роботи. Також завдяки широкій спільноті розробників, легко інтегрувати різні плагіни через Unity Asset Store.

В якості інтегрованого середовища розробки прийнято рішення обрати JetBrains Rider через повну інтеграцію з ігровим движком Unity. Також, оскільки доступна безкоштовна ліцензія для студентів, даний продукт ідеально підходить для використання при написанні кваліфікаційної роботи.

## 4 РЕАЛІЗАЦІЯ СИСТЕМИ

### 4.1 Архітектура проєкту

Архітектуру проєкту [9] можна умовно поділити на три рівні (шари):

- рівень доступу до даних (Data Access Level) – скрипти, які працюють безпосередньо з базою даних;
- рівень бізнес-логіки (Business Logic Level) – безпосередньо основна логіка роботи додатку;
- рівень інтерфейсу (View) – обробка запитів через ввід користувача, відображення їх на екран.

Для забезпечення взаємодії між шарами, використовується такий шаблон (паттерн) проєктування, як впровадження залежностей (Dependency Injection). Він полягає в тому, що залежності (сервіси) впроваджуються, або передаються за посиланням в залежний об'єкт (клієнт) і стають частиною клієнтського стану. Таким чином, відокремлюється створення залежностей клієнта від його власної логіки, що дозволяє компонентам бути слабко зв'язаними і дотримуватися таких важливих принципів проєктування, як принцип інверсії залежностей і принципу єдиного зв'язку. На рисунку 4.1 видно, що взаємодія між рівнями даних, які не є сусідніми, завжди здійснюється тільки через проміжний рівень.

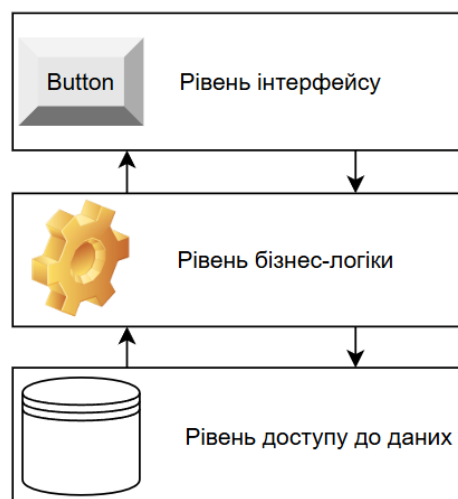


Рисунок 4.1 – Схема трирівневої архітектури проєкту

## 4.2 Реалізація головного меню

Інтерфейс програмного забезпечення достатньо простий та інтуїтивний. При першому відкритті користувачеві пропонується вибір режиму (рисунок 4.2). Це може бути тестування за окремо взятими розділами шкільного курсу математики [10]. Таким чином, користувач має можливість попрактикуватися в розв'язанні задач на певні теми. Режим тестування симулює реальну процедуру проведення НМТ, охоплюючи всі розділи математики.

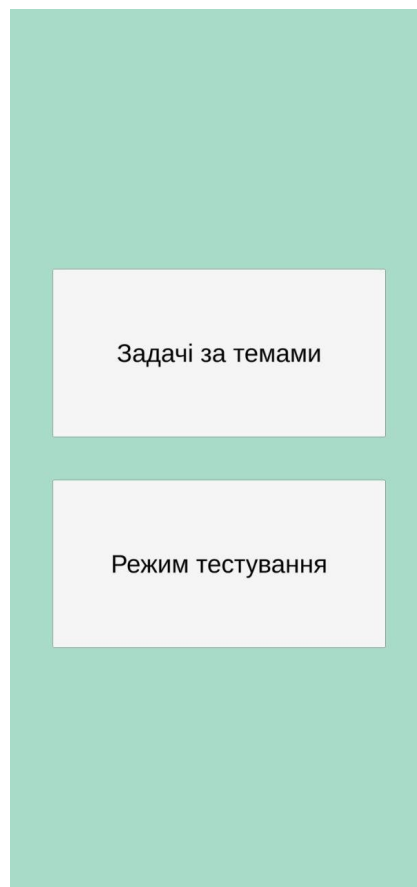


Рисунок 4.2 Скріншот вибору режиму

При виборі режиму «Задачі за темами» відкривається ієрархічний список всіх розділів, який можливо вертикально прокручувати. Скріншот списку тем зображено на рисунку 4.3. В даному меню користувач може або обрати тему, або повернутись до попереднього меню, натиснувши кнопку назад у верхньому лівому куті.

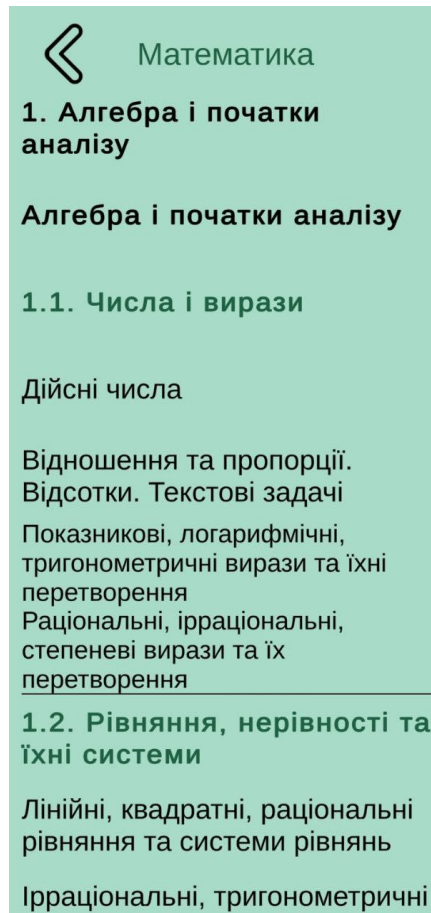


Рисунок 4.3 – Список розділів шкільного курсу математики

При виборі теми користувач переходить на екран тестування. Аналогічний перехід відбувається і при виборі режиму тестування. Бізнес-логіка для обох режимів реалізована ідентично і буде описана далі.

### 4.3 Реалізація процесу тестування

#### 4.3.1 Структури даних і зв'язки між ними

Кожне тестове питання є структурою даних, яка складається з наступних полів:

- Id – ідентифікатор питання, який однозначно визначає тестове питання;
- AnswerVariants – список доступних варіантів відповіді;

- CorrectAnswer – правильний варіант (буква) відповіді;
- QuestionSprite – спрайт (текстура), на якій зображено текст питання;
- ExplanationSprite – текстура, на якій зображено пояснення до розв’язання даної задачі.

Структура даних, яка зберігає у собі інформацію про варіант відповіді, складається всього з двох полів:

- AnswerLetter – буква, яка позначає варіант відповіді;
- AnswerSprite – текстура, яка містить текст варіанту відповіді.

Використання текстур для наповнення змісту питань, варіантів відповіді розв’язань пояснюється тим, що текстові елементи інтерфейсу користувача не підтримують математичні формули.

Самі ж питання до тесту додатково обгортаються в структуру даних QuestionsSet, яка містить у собі заголовок теми і список тестових питань.

Рисунок 4.4 демонструє взаємозв’язки між структурами даних, які зберігають інформацію про тестування

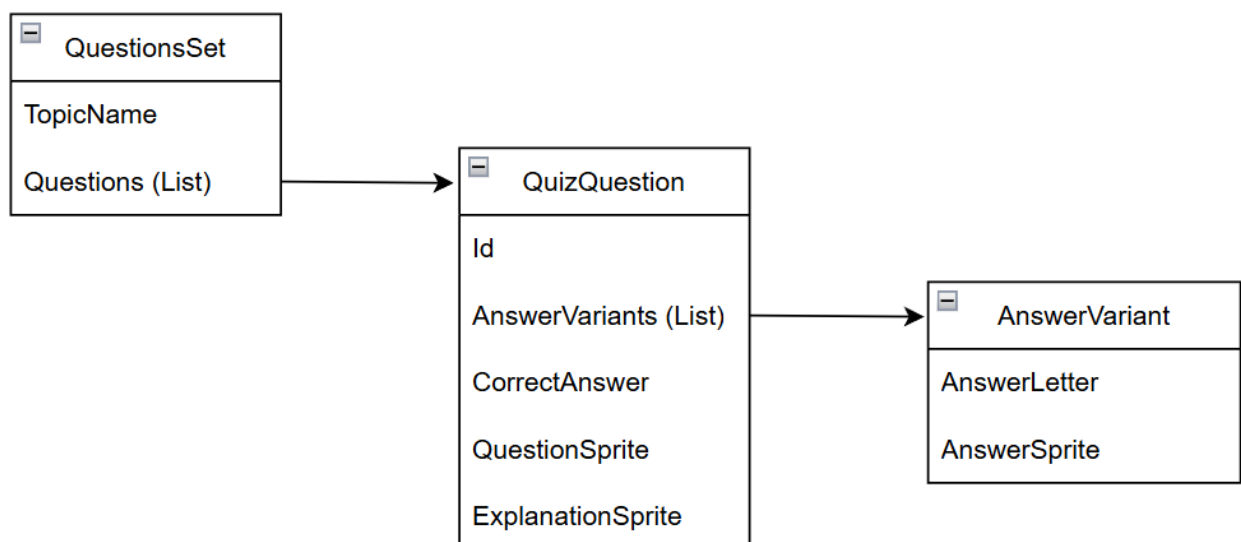


Рисунок 4.4 – Взаємозв’язки між основними структурами даних

Для зберігання і редагування контенту тестів використовуються спеціальні скрипти, конфігурацій, які наслідують від вбудованого в Unity типу даних ScriptableObject. Дані скрипти дозволяють створювати об’єкти, в яких зручно

записувати і редагувати дані. Приклад набору тестових питань, який відповідає темі «Дійсні числа» зображено на рисунку 4.5. Дані зручно подаються у вигляді деревоподібної ієрархічної структури, їх можна легко згорнути і розгорнути. Зручною опцією також є те, що ці дані є статичними і розробник має до них доступ навіть без необхідності запуску ігрового режиму.

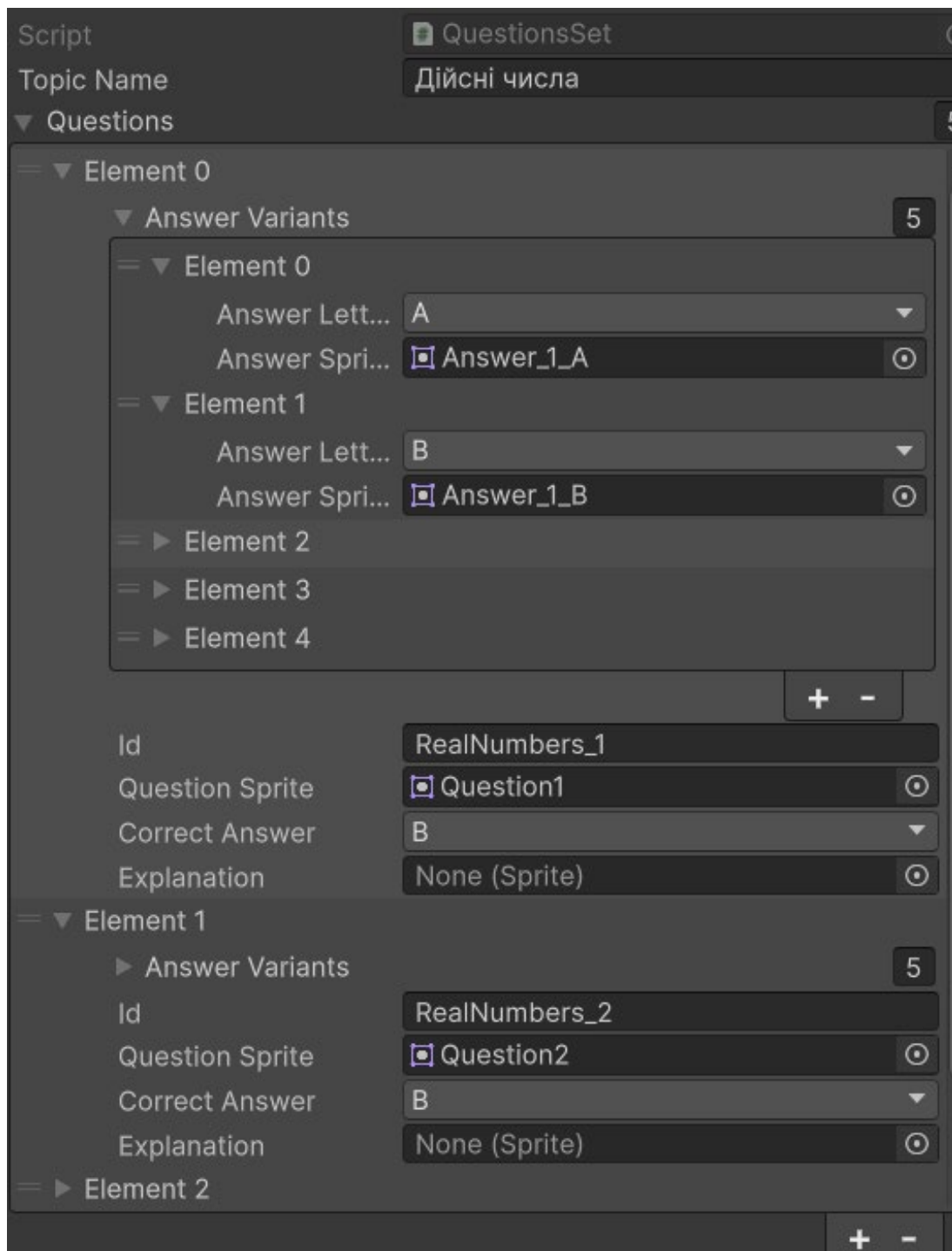


Рисунок 4.5 – Набір тестових питань за розділом «дійсні числа»

### 4.3.2 Реалізація інтерфейсу користувача у процесі тестування

Інтерфейс тестування є екраном, на якому вгорі відображається тема, за якою проходить тест (якщо це режим «Тестування за темами»). Відображається номер питання та їхня загальна кількість. Далі йде зображення, на якому знаходиться текст питання. Більшу частину екрану займає список варіантів відповіді. Для кожного вказано букву, зображення і прапорець, за допомогою якого можна відмітити даний варіант відповіді. Нижче розташовані кнопки навігації, за допомогою яких можна перемикатися вперед і назад. Кнопка «Пропустити» також перенаправляє на наступне питання. Коли варіант відповіді обрано, її текст міняється на «Відповісти». На рисунках 4.6 і 4.7 відповідно зображено стан екрану питання, коли жодну відповідь не обрано, або є обраний варіант.

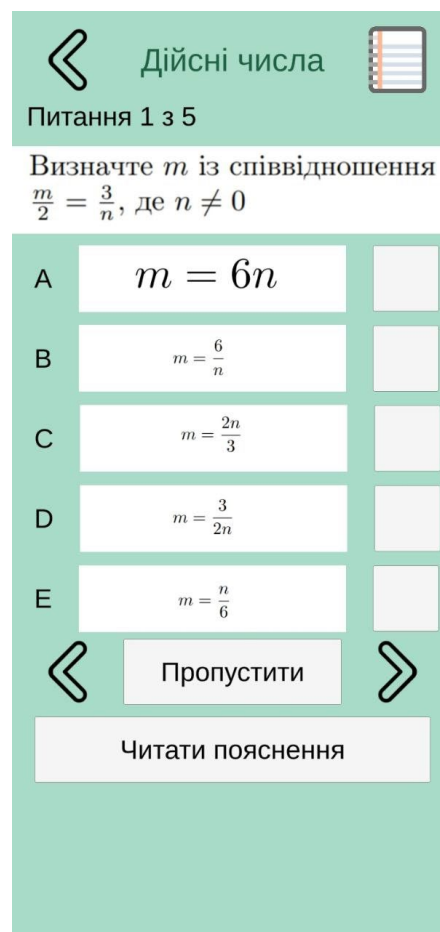


Рисунок 4.6 – Екран тестового питання без обраної відповіді

Дійсні числа

Питання 1 з 5

Визначте  $m$  із співвідношення  $\frac{m}{2} = \frac{3}{n}$ , де  $n \neq 0$

A  $m = 6n$

B  $m = \frac{6}{n}$

C  $m = \frac{2n}{3}$

D  $m = \frac{3}{2n}$

E  $m = \frac{n}{6}$

Відповісти

Читати пояснення

Рисунок 4.7 – Екран тестового питання з обраною відповіддю

#### 4.3.3 Функціональність допомоги користувачу

В процесі проходження НМТ передбачено, що кожен абітурієнт буде забезпечений аркушем паперу в якості чернетки і ручкою, таким чином надаючи можливість детально розписати процес розв’язання і перевірити свою відповідь на правильність. Оскільки використання даного продукту не є реальною процедурою проходження тестування, користувач не завжди може мати в наявності чернетку. Для цього зручним рішенням є реалізація інструменту малювання. Для економії часу на розробку, було використано готовий безкоштовний плагін

FreeDraw з магазину Unity Asset Store. Він є доволі легким в інтеграції до існуючого проєкту. Використання на мобільних також достатньо зручним, за замовчуванням доступні червоний, синій і зелений кольори та ластик. Викликається при натисканні на кнопку «блокнот» у верхньому правому куті екрану тестування. Приклад використання дошки для малювання як альтернативу паперовій чернетці наведено на рисунку 4.8.

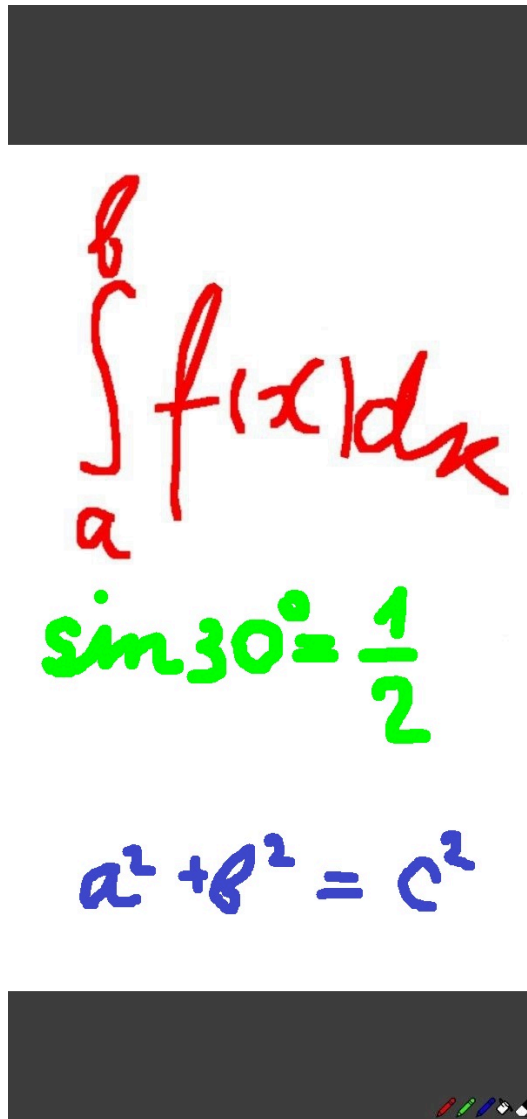


Рисунок 4.8 – Використання інтегрованої дошки для малювання

Оскільки програмне забезпечення в першу чергу є навчальним, доступна функція відображення повного розв’язання задачі. Його можна відкрити, натиснувши на кнопку «Читати пояснення».

## 4.4 Збереження прогресу

З метою зручнішого використання додатку, реалізовано логіку збереження поточного прогресу, досягнутого користувачем. Таким чином, дані про відповіді на питання за різними темами зберігаються між сесіями в межах одного мобільного пристрою.

За збереження даних відповідає скрипт `SaveManager`. На момент запуску додатку, цей скрипт завантажує в пам'ять всі дані, які було збережено в минулій сесії. Коли користувач повторно відкриває тест, який проходив у минулі рази, відповіді, відмічені раніше, відобразяться. При зміні або першій відповіді, `SaveManager` зберігає дані у файл в кеші. Дані в цьому файлі записані у форматі JavaScript Object Notation (JSON), який є компактним і не використовує багато пам'яті, а також дозволяє з високою швидкістю ці дані зчитувати і записувати. Окрім того, структуру даних у форматі JSON зручно прочитати, що є корисним при розробці і тестуванні збереження і завантаження. Наприклад, із рисунку 4.9 можна побачити, що збережено три відповіді на питання за темою «Дійсні числа». На перше і третє питання було дано варіант відповіді «А», на друге – варіант «Д». При цьому, користувач не набрав балів за останні відповіді.

З урахуванням необхідності роботи програмного продукту офлайн, збереження даних відбувається локально. Таким чином, при очищенні даних додатку, весь прогрес буде втрачено. У перспективі буде корисно реалізувати можливість збереження прогресу на сервері.

```
[
  {
    "Topic": "Дійсні числа",
    "RecordedAnswers": {
      "RealNumbers_1": 1,
      "RealNumbers_2": 5,
      "RealNumbers_3": 1
    },
    "PointsCount": 0
  }
]
```

Рисунок 4.9 – Збережені дані про відповіді у форматі JSON

## 4.5 Використання сервісів Unity для підтримки і підвищення якості продукту

### 4.5.1 Unity Analytics як інструмент збору статистики про використання програмного продукту

Розробники будь-якого проєкту зацікавлені в тому, щоб залучити велику аудиторію користувачів і утримувати її. Це неможливо без збору статистики про дії користувачів у додатку.

Наведемо перелік основних даних, які необхідно збирати в процесі роботи даного додатку:

- кількість встановлень і видалень у певні відрізки часу;
- кількість заходів у додаток і тривалість сесій;
- повернення користувачів на наступний день і тиждень після встановлення;
- найбільш і найменш популярні теми;
- найбільш популярні режими проходження тестів;
- правильні і неправильні відповіді;
- бали, які одержують абітурієнти в процесі тестування;
- звернення до повного пояснення завдання та використання дошки для малювання;
- категорія користувача (абітурієнт, батько, вчитель тощо).

Для збору даних існують різні інструменти, такі як Firebase Analytics (інструмент Google Analytics для мобільних платформ), Facebook Analytics, AppsFlyer тощо. Дані сервіси містять готовий функціонал, але потребують додаткової інтеграції. Сторонні бібліотеки, як правило, значно збільшують розмір додатку. Також розповсюдженою проблемою є несумісність версій, що призводить до нестабільної роботи програмного забезпечення або взагалі збоїв.

Для того, щоб не стикатися з подібними проблемами, надамо перевагу вбудованому інструменту Unity Analytics. Цей потужний інструмент дозволяє

розробникам зібрати, обробити та проаналізувати дані про поведінку користувачів у мобільних, десктопних або консольних додатках. Його використання в проєкті надає низку важливих переваг, які роблять процес розробки та вдосконалення продукту ефективнішим.

Інтегрувати Unity Analytics у проєкт достатньо просто. Цей інструмент не потребує складного налаштування. Розробники можуть активувати його безпосередньо через Unity редактор, що значно скорочує час впровадження. Завдяки вбудованій відправці стандартних подій дозволяє швидко почати збір даних, таких як рівень залученості користувачів, частота використання додатку та інші метрики, важливі для оцінки ефективності програмного продукту.

Ще однією перевагою Unity Analytics є доступність попередньо налаштованих інформаційних панелей, які забезпечують розробників візуально зрозумілими звітами. Ці панелі дозволяють відстежувати ключові показники, такі як кількість активних користувачів, тривалість сеансів, а також ігрові події, які відбуваються в додатку. Наприклад, розробник може побачити, скільки часу користувач витрачає на певне тестове питання або які теми викликають найбільший інтерес. Ця інформація є основою для прийняття рішень щодо оптимізації контенту або геймплею. Приклад записаних подій в інформаційній панелі зображено на рисунку 4.10.

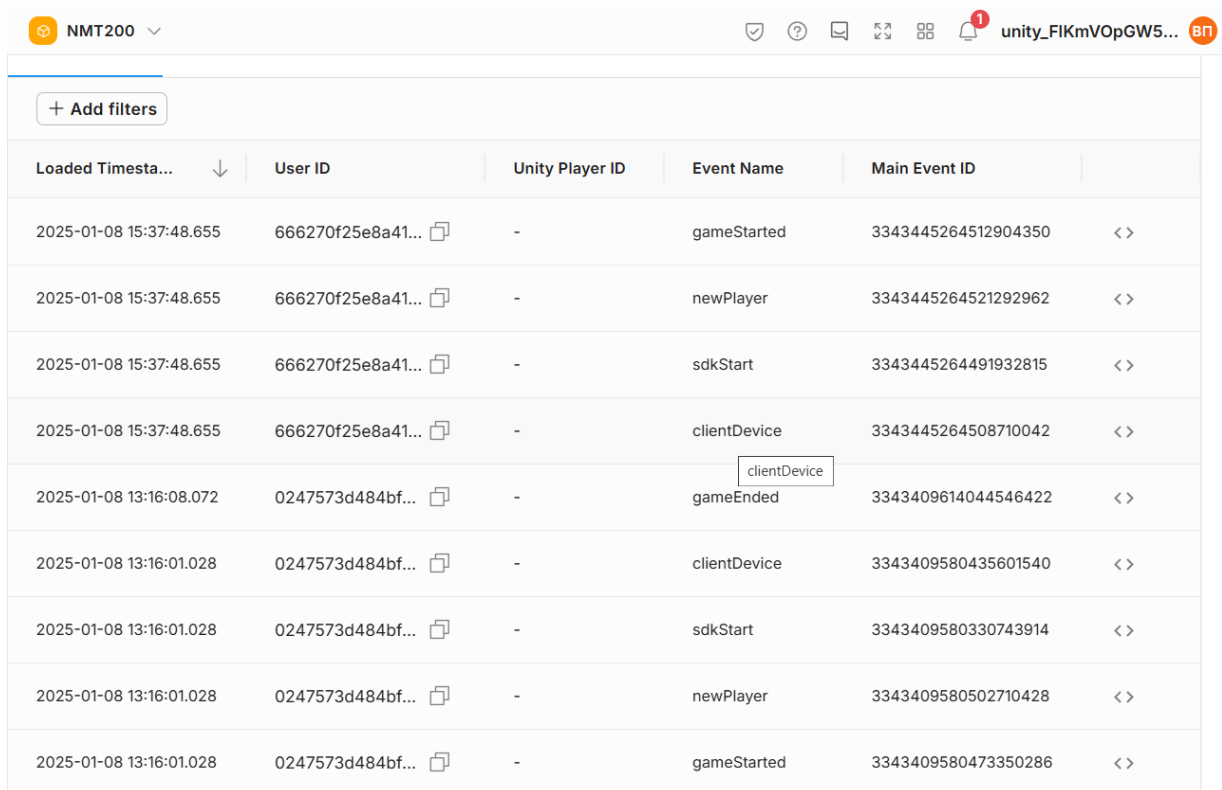
Unity Analytics також забезпечує підтримку індивідуальних ігрових подій, що дозволяє збирати дані, які відносяться до конкретного проєкту. Це дає змогу розробникам адаптувати аналітику до цілей проєкту, відстежуючи специфічні аспекти користувацької поведінки. Наприклад, у рамках навчальної гри можна аналізувати, які математичні задачі викликають труднощі у користувачів, або оцінювати, наскільки ефективно працюють ігрові елементи, спрямовані на навчання.

Важливо також зазначити, що Unity Analytics пропонує інструменти для сегментації аудиторії. Це означає, що розробники можуть розділяти користувачів на групи за різними критеріями, такими як вік, рівень активності, географічне положення тощо. Такий підхід дозволяє краще розуміти потреби окремих

сегментів і адаптувати продукт до їхніх очікувань. У контексті навчального програмного продукту сегментація може бути використана для визначення, які аспекти гри найбільше впливають на успішність учнів із різним рівнем підготовки.

Також корисною опцією Unity Analytics є можливість працювати із зібраними даними в режимі реального часу. Це дає змогу швидко реагувати на проблеми, що виникають, або впроваджувати зміни для покращення досвіду користувача. Наприклад, якщо аналітика показує, що певна частина гри є надто складною для користувачів. Таким чином, розробники можуть оперативно внести корективи, щоб зробити її більш доступною.

Таким чином, використання Unity Analytics у проєкті не лише сприяє підвищенню ефективності процесу розробки, а й дозволяє значно поліпшити користувацький досвід. Завдяки широким можливостям збору та аналізу даних, цей інструмент є незамінним для створення продуктів, які відповідають потребам та очікуванням аудиторії.



Loaded Timesta...	User ID	Unity Player ID	Event Name	Main Event ID
2025-01-08 15:37:48.655	666270f25e8a41...	-	gameStarted	3343445264512904350
2025-01-08 15:37:48.655	666270f25e8a41...	-	newPlayer	3343445264521292962
2025-01-08 15:37:48.655	666270f25e8a41...	-	sdkStart	3343445264491932815
2025-01-08 15:37:48.655	666270f25e8a41...	-	clientDevice	3343445264508710042
2025-01-08 13:16:08.072	0247573d484bf...	-	gameEnded	3343409614044546422
2025-01-08 13:16:01.028	0247573d484bf...	-	clientDevice	3343409580435601540
2025-01-08 13:16:01.028	0247573d484bf...	-	sdkStart	3343409580330743914
2025-01-08 13:16:01.028	0247573d484bf...	-	newPlayer	3343409580502710428
2025-01-08 13:16:01.028	0247573d484bf...	-	gameStarted	3343409580473350286

Рисунок 4.10 – Інформаційна панель подій в Unity Analytics

#### 4.5.2 Усунення проблем роботи додатка засобами Cloud Diagnostics

Після публікації гри у відповідних магазинах (Google Play для Android та App Store для iOS) стає актуальним питання підтримки стабільної роботи додатка. В процесі розробки і тестування дефекти виявляються і усуваються. Однак далеко не завжди можливо передбачити всі випадки, коли програмний продукт працює некоректно. Помилки в роботі поділяються на два типи: необроблені виключення (exceptions) і збої програми (crashes). У випадку виключень додаток продовжує свою роботу, хоча і поведінка відрізняється від очікуваної. Степені важкості подібних дефектів встановлюються розробниками. Під збоєм мається на увазі раптове завершення програми. Це може статися, наприклад, через переповнення пам'яті або через нативну проблему. Критично важливо запобігти відтоку користувачів через некоректну роботу додатку. Для виявлення таких проблем існують сторонні бібліотеки, подібні до тих, що використовуються для аналітики. Однак для сторонніх плагінів залишається та сама проблема складності інтеграції і оптимізації.

Unity пропонує вбудований інструмент Cloud Diagnostics. Аналогічно Unity Analytics, він так само легко інтегрується в проект. Однією з головних переваг Unity Cloud Diagnostics є автоматичний збір звітів про помилки (crash reports). Ця функція дозволяє відстежувати всі випадки, коли додаток несподівано завершує свою роботу, і надавати розробникам точні технічні деталі. Наприклад, розробник може отримати інформацію про конфігурацію пристрою користувача, стан пам'яті та інші ключові параметри, що спричинили збій. Такий підхід дозволяє швидко ідентифікувати проблеми та виправляти їх у наступних оновленнях.

Ще одна важлива функція – моніторинг користувацьких звітів (user-reported issues). Користувачі можуть надсилати зворотній зв'язок про проблеми, які вони помічають під час використання додатка. Ця інформація зберігається у вигляді звітів, які об'єднуються з автоматично зібраними даними, надаючи розробникам повну картину про можливі недоліки. У контексті навчальної гри, це

може допомогти оперативно виявити, наприклад, некоректну роботу певних завдань чи помилки у логіці гри.

Unity Cloud Diagnostics дозволяє відстеження проблем на різних етапах життєвого циклу продукту. Інструмент дозволяє аналізувати дані не лише від фінальної версії додатку, а й від тестових збірок. Це означає, що потенційні помилки можуть бути виявлені та виправлені ще до того моменту, коли програмний продукт стане доступним для остаточних користувачів. Такий підхід значно підвищує якість програмного забезпечення та зменшує кількість проблем після релізу.

Завдяки хмарній архітектурі Unity Cloud Diagnostics забезпечує швидкий доступ до звітів і детальної інформації незалежно від місця знаходження команди розробників. Звіти про помилки надходять до хмари менше, ніж за хвилину. На рисунку 4.11 ми можемо побачити звіт про помилку, виявлений на пристрої Xiaomi Redmi 7 з операційною системою Android 10, час надходження з точністю до мілісекунд. Проект зібрано на версії Unity 6000.0.23f1. Особливо важливо інформацією є трасування стеку виключення, за допомогою якого розробники можуть виявити місце в програмному коді, де виникла помилка. Таким чином, стає можливим відтворити дану ситуацію і усунути дефект. Це далеко не повна інформація про помилку, але її зазвичай достатньо для вирішення проблеми в роботі програмного забезпечення.

**NullReferenceException: Object reference not set to an instance of an object.** < Jan 8, 2025, 22:48:03.237 > [JSON](#)

NMT200.UI.Menu.MainMenuScreen.SubscribeAll () (at <00000000000000000000000000000000>:0)

Incident count	Device	OS version	CPU	GFX	Unity version
1	Xiaomi/Redmi 7/onc	Android OS 10 / API-29 (QQQ1.191008.001/V11.0.2.0.QFLMIXM)	ARM64 FP ASIMD AES	Adreno (TM) 506	6000.0.23f1

**STACK TRACE**    METADATA    LOGS

Managed Stack Trace:

```
NMT200.UI.Menu.MainMenuScreen.SubscribeAll () (at <00000000000000000000000000000000>:0)
```

Рисунок 4.11 – Приклад звіту про помилку в роботі додатку

Попри низку позитивних якостей інструментів аналітики і діагностики, вони мають серйозне обмеження – робота функціоналу відправлення звітів можлива тільки при з'єднанні з мережею Інтернет. Таким чином, завжди ми маємо на увазі, що не будемо знати про всі випадки, коли в роботі додатку наявні помилки. Однак вважатимемо подібні втрати даних незначними, тобто проаналізувати вибірку наявних даних буде цілком можливо

#### 4.6 Перспективи і виклики розвитку проекту

Проект навчальної математичної гри, створений засобами Unity, має значний потенціал для подальшого розвитку. Удосконалення функціональних можливостей, збагачення контенту та підвищення зручності для користувачів є ключовими напрямками, які визначають перспективи проекту. Проте ці напрями також пов'язані з певними викликами, які потрібно врахувати.

Однією із найважливіших задач на майбутнє є вдосконалення інтерфейсу користувача. Простий, інтуїтивно зрозумілий та естетично привабливий інтерфейс є важливим фактором успіху будь-якого програмного продукту. Серед викликів тут можна виділити необхідність забезпечення адаптивності інтерфейсу для різних пристроїв, а також інтеграції функцій, які спрощують взаємодію з програмним продуктом, таких як підказки чи інтерактивні елементи. Окрім того, припускається залучення професійного UI/UX дизайнера, який може створити максимально дружній до користувача інтерфейс.

Ще однією перспективною функцією є можливість збереження прогресу користувачів на сервері. Звичайно, в першу чергу гра позиціонує себе як програмне забезпечення, для якого з'єднання з інтернетом не є обов'язковим. Однак раніше вже описувалася ситуація, що прогрес може бути втрачений у випадку очищення даних додатку, втрати або виходу з ладу мобільного пристрою. Таким чином, корисною опцією буде можливість створення акаунту для збереження даних на сервері. Це дозволить гравцям відновлювати дані з будь-якого

пристрою, забезпечуючи більшу гнучкість у використанні додатку. Проте реалізація такої функціональності вимагає створення безпечного механізму збереження даних, що враховує питання конфіденційності та захисту особистої інформації користувачів. Для інтеграції логіки взаємодії з сервером може виникнути необхідність залучення розробників сервера або в якості альтернативи можна використати готові серверні рішення, такі як Firebase або Playfab.

На даному етапі гра містить мінімум контенту, який необхідний для практики розв'язання задач за програмою НМТ. Всі файли і дані зберігаються локально. Однак необхідно врахувати те, що мобільний додаток може вмістити лише обмежений обсяг таких даних. Рішенням даної проблеми є створення серверного сховища для навчального контенту. Це дозволить додавати нові задачі, навчальні модулі чи тести без необхідності оновлення клієнтського додатку. Такий підхід забезпечує динамічність контенту і знижує витрати на підтримку продукту. Серед викликів реалізації серверного сховища можна виділити забезпечення його надійності, масштабованості та відповідності вимогам захисту даних.

Окрему увагу слід приділити додаванню ігрових елементів, які сприяють підвищенню залученості користувачів. Наприклад, у гру можна інтегрувати систему досягнень, яка мотивує користувачів завершувати певні завдання, або таблицю лідерів, яка сприяє здоровій конкуренції між учасниками. Також серед для мотивації регулярного відкриття гри можна реалізувати функцію щоденних задач і систему Push-сповіщень.

Необхідно врахувати, що розробники програмного забезпечення не є фахівцями зі створення контенту. Таким чином, необхідно залучати спеціалістів, які будуть працювати над збалансованим підбором задач. Також даний проєкт задуманий для вступу не в одному навчальному році. Оскільки з кожним роком програма НМТ зазнає змін, необхідно коректно підлаштовувати її під актуальні вимоги.

Отже, розвиток проєкту пов'язаний із широкими перспективами вдосконалення функціональності та підвищення досвіду користувача. Однак кожен з нових

напрямів потребує ретельного планування та розв'язання технічних викликів, що робить цей процес не лише захопливим, але й складним. Успішна реалізація цих функцій дозволить створити якісний програмний продукт, який буде відповідати сучасним освітнім вимогам і задовольняти потреби користувачів.

#### Висновки за розділом 4

У даному розділі було детально розглянуто реалізацію навчального програмного продукту. Було проведено огляд інтерфейсу користувача, описано навігацію і відповідальність елементу за його функцію. Проведено аналіз основного функціоналу додатка, описано шлях користувача від першого відкриття і до проходження тестування. Реалізовано функціонал допомоги користувачу, такий як дошка для малювання, що заміняє аркуш паперу і ручку, таким чином забезпечивши всім необхідним для проведення розрахунків. Окрім того, можливо переглянути повне розв'язання задачі. Важливою є функція збереження і завантаження прогресу користувача. Таким чином, стан проходження тестування між сесіями зберігається.

Для програмного продукту передбачено можливість підтримки її функціоналу і усунення наявних проблем. Завдяки інструменту Unity Analytics, можливий збір даних про такі активності користувачів, як встановлення або видалення додатку, тривалість сесій, найбільш і найменш популярні розділи математики. Ця інформація допоможе зробити додаток більш збалансованим, утримати і залучити якомога більшу аудиторію, і що найголовніше – вдосконалити якість підготовки до тестування. Важливим також є виявлення і усунення дефектів і збоїв, які не вдалося усунути в процесі розробки і тестування, але мали місце серед остаточних користувачів. Для даної задачі використовується інструмент Unity Cloud Diagnostics, який в автоматичному режимі збирає детальну інформацію про кожен збій в роботі додатку, таким чином спрощуючи розробникам процес усунення помилок і збоїв програми.

Даний проєкт має подальші перспективи розвитку. В першу чергу, необхідно удосконалити інтерфейс користувача таким чином, щоб він був максимально зручним. Наприклад, корисною опцією буде можливість збереження прогресу на сервері, що запобігатиме втраті даних. Також для урізноманітнення задач необхідне серверне сховище, щоб поступово завантажувати нові матеріали. Далі, необхідно розробити механіки утримання і залучення гравців. Перед реалізацією кожної з цих задач стоять виклики, і майже на кожному етапі є необхідним залучення спеціалістів із суміжних галузей.

## ВИСНОВКИ

Під час написання кваліфікаційної роботи виконано наступні задачі:

– проведено огляд і аналіз доцільності розробки навчальної математичної гри засобами движка Unity;

– обрано ігровий движок Unity 6 в якості програмного засобу для розробки.

В результаті виконання кваліфікаційної роботи створено прототип навчальної математичної гри для мобільних платформ, яка має на меті допомогу до підготовки абітурієнтів до НМТ. В першу чергу враховано умови, в яких на даний момент часу здійснюється підготовка для певної частини учнів, а саме перебої в роботі електроенергії, відсутність або низька швидкість інтернету.

Дана робота несе велику наукову, науково-технічно і соціально-економічну значущість, оскільки має на меті покращення якості та доступності освіти для молоді, яка далі вступатиме до закладів вищої освіти і здобуватиме професію, при цьому займаючись науковою діяльністю. В той же час сам додаток буде абсолютно безкоштовним і не міститиме в собі платної функціональності.

Даний проєкт є перспективним, оскільки є велика кількість ідей, які можна реалізувати в грі для покращення її якості і залучення користувачів. У подальшому є необхідність залучення фахівців із різних галузей, ведення комунікації з вчителями, методистами НМТ. Також можливо розробити аналогічне програмне забезпечення для вивчення інших дисциплін або інтегрувати їх в існуюче. Всі ці ідеї потребують багато часу і ресурсів, та їхня реалізація виходить за межі виконання кваліфікаційної роботи магістра. Отже, одержані результати можна використовувати в подальшому для вдосконалення програмного продукту і якості освіти в Україні в цілому.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Тевяшев А. Д., Лямец В. И. Системный анализ. Вводный курс. Харків : ХНУРЕ, 2003. 448 с.
2. Saaty T. L. Decision making with the analytic hierarchy process. *International Journal of Services Sciences*. 2008. Vol. 1, No. 1. P. 83 – 98.
3. Люшенко Л. А., Хіцко Я. В. Розробка та аналіз вимог до програмного забезпечення. Київ : КПІ ім. Ігоря Сікорського, 2020. 63 с.
4. Цибульник С. О., Барандич К. С. Технології розроблення програмного забезпечення. Частина 1. Життєвий цикл програмного забезпечення. Київ : КПІ ім. Ігоря Сікорського, 2022. 270 с.
5. Супруненко О. О. Аналіз вимог до програмного забезпечення. Черкаси : Вид. від. Черкаського національного університету імені Богдана Хмельницького, 2012. 24 с.
6. Розробка мобільних додатків від А до Я: повний гайд. URL: <https://dan-it.com.ua/uk/blog/rozrobka-mobilnih-dodatkiv-vid-a-do-ja-povnij-gajd> (дата звернення: 26.11.2024).
7. ТОП 6 фреймворків для розробки кросплатформних додатків. URL: <https://web-academy.ua/blog/junior/top-6-most-popular-cross-app-dev-frameworks> (дата звернення: 02.01.2025).
8. Unity 6 вийшов! URL: <https://uk.sharpcoderblog.com/blog/unity-6-is-out> (дата звернення: 01.12.2024).
9. Авраменко В.С., Авраменко А.С. Проектування інформаційних систем. Черкаси: Черкаський національний університет ім. Б. Хмельницького, 2017. 434 с.
10. Тести ЗНО/НМТ онлайн з математики – сайт ЗНО.Освіта.UA. URL: <https://zno.osvita.ua/mathematics> (дата звернення: 10.11.2024).