

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЇВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра мультимедійних систем

**Розробка Android додатку для вивчення англійської
лексики на платформі Unity**
Текстова частина до курсової роботи
за спеціальністю «Інженерія програмного забезпечення»

Керівник курсової роботи
ст. викладач Борозенний С. О.

(підпис)

“ ____ ” _____ 2024 р.

Виконав студент 3 р. н.

Коломійчук А. К.

“ ____ ” _____ 2024 р.

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра мультимедійних систем

ЗАТВЕРДЖУЮ

Зав. кафедри мультимедійних систем

доцент, к.ф.-м.н.

О. П. Жежерун

_____ 2024 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на курсову роботу

студенту Коломійчуку Артемію Костянтиновичу 3-го курсу факультету
інформатики

ТЕМА Розробка Android додатку для вивчення англійської лексики на
платформі Unity

Зміст ТЧ до курсової роботи:

Індивідуальне завдання

Календарний план виконання роботи

Анотація

Вступ

1 Аналіз предметної області

2 Проектування застосунку

3 Розробка

Висновки

Список використаних джерел

Дата видачі „___” _____ 2024 р. Борозенний О.С. _____

(підпис)

Завдання отримав _____

(підпис)

Тема: Розробка Android додатку для вивчення англійської лексики на платформі Unity

Календарний план виконання роботи:

№ п/п	Назва етапу	Термін виконання	Примітка
1.	Визначення теми курсової роботи	Жовтень 2023	
2.	Отримання завдання на курсову роботу	Жовтень 2023	
3.	Огляд літератури	Листопад – Грудень 2023	
4.	Написання теоретичної частини кваліфікаційної роботи	Січень 2024	
5.	Проектування застосунку	Лютий 2024	
6.	Розробка застосунку	Лютий - Квітень 2024	
7.	Написання текстової частини роботи	Квітень – Травень 2024	
8.	Захист курсової роботи	Травень 2024	

Студент Коломійчук А. К.

Керівник Борозенний О. С.

“ ”

ЗМІСТ

Стор.

АНОТАЦІЯ	5
ПЕРЕЛІК ЗАЛУЧЕНИХ СКОРОЧЕНЬ	6
ВСТУП	7
РОЗДІЛ 1: Аналіз предметної області	
1.1. Огляд застосунків для вивчення лексики.....	8
1.2. Аналіз ринку.....	11
РОЗДІЛ 2: Проектування застосунку	
2.1. Загальні вимоги.....	12
2.2. Опис запропонованих режимів тренувань.....	12
РОЗДІЛ 3: Розробка	
3.1. Засоби розробки.....	15
3.2. Специфіка розробки додатків із використанням Unity.....	15
3.3. Unity Gaming Services.....	17
3.4. Взаємодія із ОС Android.....	21
3.5. Досвід кінцевого користувача.....	22
Висновки	23
Використані джерела	24

Анотація

Курсова робота присвячена розробці мобільного застосунку для вивчення англійської лексики. Під час розроблення застосунку було досліджено підходи до вивчення лексики та особливості роботи із хмарними сервісами Unity.

Ключові слова: Unity, C#, англійська мова

Перелік залучених скорочень

API – прикладний програмний інтерфейс

ШІ – штучний інтелект

ОС – операційна система

Вступ

Вивчення лексики – один із ключових елементів вивчення будь-якої мови на будь-якому рівні, що є актуальним незалежно від того чи вивчається мова для читання книг або статей іноземною мовою, перегляду фільмів та серіалів мовою оригіналу, спілкування з носіями мови, чи з будь-якої іншої мети. Є значна кількість програм та інших засобів для полегшення вивчення та оцінювання засвоєння лексики як для персональних комп'ютерів, так і для мобільних пристроїв. Однак, на сьогоднішній день жодна з популярних програм для вивчення як виключно лексики, так і мови загалом, не надає зручний функціонал для вивчення різних значень слів та у контексті. Здебільшого слова в англійській мові має багато значень, кожне з яких так само може мати стилістичні нюанси або бути частиною ідіоми. Так само, не завжди зручно вивчати синоніми та антоніми, що є ключовим для збільшення виразності мовлення.

Метою цієї курсової роботи є розробка методів вивчення слів та відповідного додатку для вивчення англійської мови, а також аналіз переваг та недоліків розробки додатків за допомогою Unity.

Текстова частина курсової роботи містить у собі 3 основні розділи:

Перший розділ присвячено аналізу застосунків та підходів до вивчення лексики.

Другий розділ курсової роботи присвячений проектуванню кінцевого застосунку.

Третій розділ презентує засоби, використані в розробці застосунку, а також у ньому розглянуто переваги та недоліки розробки на платформі Unity.

Розроблений мобільний застосунок зможе допомогти користувачам у опануванні лексики та у переведенні пасивної лексики в активну, що є актуальним на будь-якому етапі вивчення мови.

Аналіз предметної області

Огляд застосунків для вивчення лексики

Можна перерахувати сотні популярних та не дуже застосунків для вивчення англійської мови. Багато з них схожі між собою як за логікою, так і за функціоналом. Тим не менш, можна виділити декілька популярних застосунків для аналізу.

Найпопулярнішим застосунком для вивчення мов є Duolingo, побудований у вигляді послідовного курсу з завданнями різного типу. Хоча він і є зручним і навіть ефективним засобом для початківців, дуже складно виключно використовуючи цей застосунок досягти навіть середнього рівня володіння мовою. Використовується велика кількість режимів для вивчення, але контекстуальні особливості, синоніми та антоніми не беруться до уваги. Можливості вивчення обраних слів та вивчення лексики окремо від загального «курсу» немає. Вибірка лексики у курсах є дивною і мінливою, часто трапляються фрази, що не використовуються в повсякденному мовленні і загалом їхня структура є дуже базовою, що ставить під сумнів ефективність Duolingo як інструмента для вивчення лексики[1].

Іншим популярним застосунком є Quizlet – універсальний сервіс для вивчення будь-чого за допомогою карток. Сильна сторона сервісу – універсальність – можна вчити різні мови, шкільні та не лише предмети, і часто пропонуються варіанти автозаповнення для карток на основі введеного значення, що значно прискорює пошук і створення відповідних карток. Проте, саме для вивчення лексики це не завжди зручно, оскільки дані не завжди структуровані. Якщо додавати приклади використання слів, то картки стають дуже довгими, а сам застосунок не призначений для вивчення багатозначних слів та певних нюансів, бо для всього потрібно створювати картки окремо та в інших папках, оскільки мати картки з однаковими значеннями у одній папці не дозволено. Наявно лише 4 базових режими для вивчення – істина/хиба,

множинний вибір, співставлення та написання. Хоча ці режими і є зручними для вивчення лексики, вони швидко набридають через одноманітність та сталість матеріалів у картках.

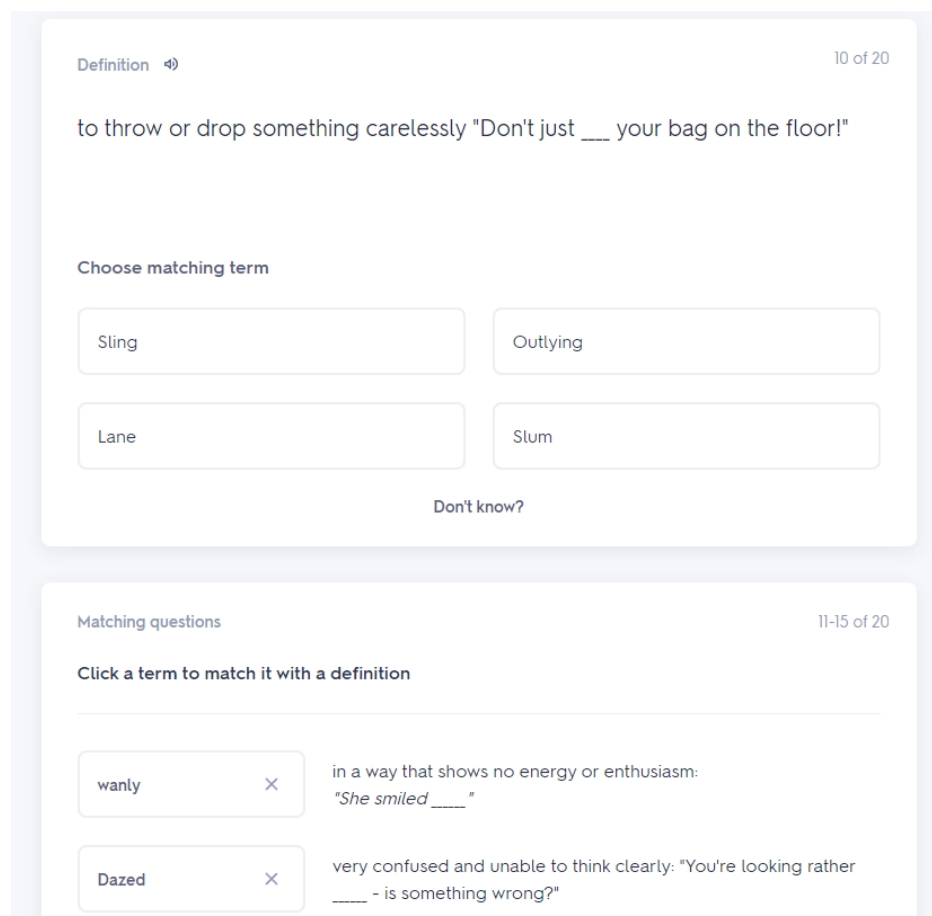


Рис. 1.1 – Інтерфейс тесту у Quizlet

На перший погляд Cambridge Dictionary +Plus є очевидним вибором для вивчення лексики, оскільки Кембриджський словник є одним з найбільш структурованих та об'ємних, містить багато контекстуальної інформації, до визначень часто вказані синоніми, антоніми та приклади використання з нюансами з точки зору граматики та використання службових слів, виділено багато ідіом та фразових слів. У додатку можна зручно переглядати цю інформацію у тому ж вигляді, в якому вона наявна на офіційному сайті словника. Проте, після додавання визначення у збірку для вивчення уся контекстуальна інформація, довідка, приклади використання, синоніми тощо видаляються і залишається лише визначення. Для вивчення пропонуються лише

три види тренувань – написання на основі визначення, перегляд карток та написання відповідно до вимови. Цих видів тренувань очевидно недостатньо для ефективного вивчення, а сам додаток не є надто зручним, містить багато реклами і не працює без підключення до інтернету.

WordUp є одним з найбільш функціональних та поціновуваних[2] застосунків. Він пропонує колекцію з наперед визначених слів із згенерованими ІІІ зображеннями. Застосунок надає декілька режимів тренувань та використовує інтервальний підхід до вивчення, що важливо якщо ціллю є саме довготривале вивчення. Під час вибору слів для вивчення надаються як різні значення для кожного слова, так і контекст із прикладами та порадами із використання, тим не менш під час вивчення залишається лише одна основна картка, що не є надто зручним. Особливістю застосунку також є наявність тренувань у режимі спілкування з ІІІ, що є ефективним елементом вивчення активної лексики. На жаль, у додатку немає гнучких налаштувань, але загалом він є доволі цілісним та зручним для вивчення якоїсь загальної лексики, а обрані слова вивчати не надто зручно.

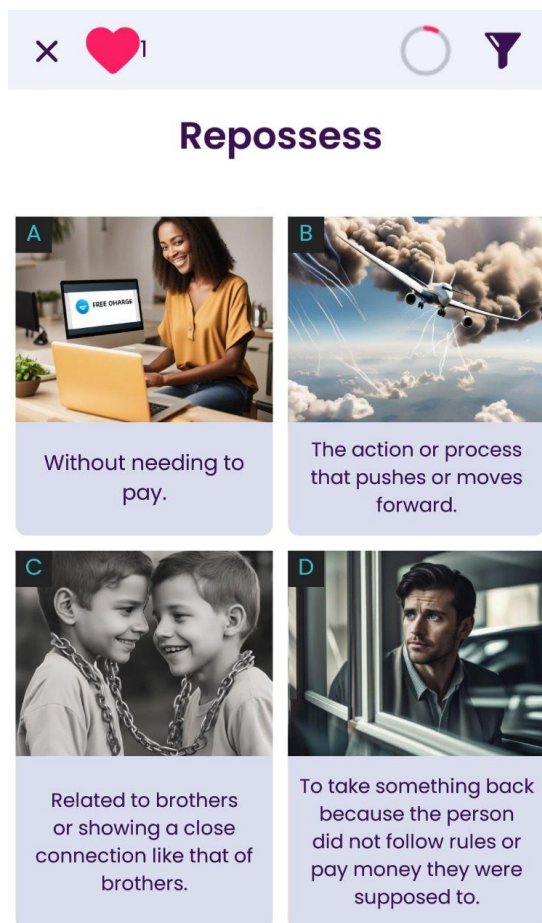


Рис. 1.2 – Інтерфейс множинного вибору у WordUp

Аналіз ринку

Хоча кількість додатків для вибору є значною, жоден з них не надає зручний функціонал для вивчення обраних користувачем слів із урахуванням багатозначності та контекстуальних особливостей. Навіть ті застосунки, що спеціалізуються виключно на допомозі у вивченні англійської лексики та мають базовий функціонал, фокусуються на вивченні загальних слів та фраз, не зважаючи на індивідуальні потреби користувачів. В результаті, для вивчення мови на просунутому рівні, користувачам може знадобитися декілька різних неструктурованих застосунків.

Проектування застосунку

Загальні вимоги

Основна ціль застосунку – дати змогу користувачу ефективно вивчати обрані ним слова з урахуванням багатозначності. Для цього важливо зберегти контекстуальну інформацію, синоніми, антоніми та приклади використання на етапі вивчення. З огляду на наявну інформацію, у картках, що позначають одне значення слова має зберігатися прогрес вивчення у трьох вимірах: сприйняття, використання та володіння синонімами/антонімами.

Опис запропонованих режимів тренувань

Оскільки проблемою багатьох програм є мала кількість режимів тренувань, ціллю також є розробка та впровадження великої їх кількості. Деякі з запропонованих режимів логічно та функціонально перетинаються між собою, що полегшує як розробку, так і сприйняття системи кінцевим користувачем.

Загалом було сформульовано концепцію 21 режиму:

- Флеш-картка – користувачу відображається ідентична до наявної в колекції картка для нагадування та запам'ятовування
- Флеш-картка+ – те саме, що і флеш-картка, але також користувачу пропонується скласти словосполучення із запропонованим значенням слова для розвитку активної лексики
- Правда/хиба – користувачу відображається пара слово-визначення, і він повинен вказати чи вірно вони співставлені
- Правда/хиба+ – те саме, що і правда/хиба, але у разі відповіді «хиба» користувач має вписати правильне слово до запропонованого визначення
- Множинна відповідь I – користувачу надаються декілька слів, і він повинен вибрати те, яке відповідає запропонованому визначенню

- Множинна відповідь II - користувачу надаються декілька визначень, і він повинен вибрати те, яке відповідає запропонованому слову
- Множинна відповідь III - користувачу надається перелік слів до запропонованого слова, і він повинен вибрати правильний синонім
- Множинна відповідь IV - користувачу надається перелік слів до запропонованого слова, і він повинен вибрати правильний антонім
- Множинна відповідь V - користувачу надається приклад використання в якому пропущене слово, і він повинен вибрати правильне слово для доповнення
- Множинна відповідь VI - користувачу показується зображення, і він повинен вибрати правильне слово з переліку
- Множинна відповідь VII – користувачу надається слово та варіанти контекстуальних тегів (напр. формальне, неформальне, у переносному значенні, США тощо) з наперед визначених наборів, і він повинен вибрати єдиний правильний
- Співставлення I - користувачу надається список слів разом з їх визначеннями, і він повинен правильно співставити кожне слово з його відповідним значенням
- Співставлення II - користувачу надається список слів разом з їх синонімами, і він повинен правильно співставити кожне слово з його відповідним синонімом
- Співставлення III - користувачу надається список слів разом з їх антонімами, і він повинен правильно співставити кожне слово з його відповідним антонімом
- Співставлення IV - користувачу надається список речень або фраз, в яких пропущене слово, і він повинен співставити їх із відповідними словами
- Співставлення V - користувачу надається список слів які він повинен співставити із запропонованими тегами

- Написання I – користувачу надається визначення і він повинен вписати відповідне слово
- Написання II – користувачу надається слово і він повинен написати речення із використанням цього слова
- Написання III – користувачу надається аудіо фрагмент із вимовою, і він повинен вписати відповідне слово
- Написання IV – користувачу надається зображення, і він повинен вписати відповідне слово
- Написання V – користувачу надається приклад використання в якому пропущене слово, яке він повинен вписати

Розробка

Засоби розробки

Основним засобом для розробки було обрано Unity – ігровий рушій, що є найпопулярнішим[3] засобом для розробки мобільних ігор та, відповідно, має усі засоби необхідні для розробки в тому числі й мобільних застосунків. Програмний код було написано на вже застарілій[4-5] на момент написання роботи C# 9 та платформі .Net 4. Застарілість зумовлена тим, що під час створення кінцевого застосунку код написаний на C# конвертується у C++, який потім компілюється у нативний код для збільшення швидкодії[6]. Ця та інші оптимізації унеможливають використання певних засобів як мови, так і деяких NuGet пакунків[7]. Цю проблему частково вирішує Unity Package Manager, який дає змогу зручно додавати офіційно підтримувані пакунки.

У проєкті використана база даних SQLite, оскільки вона є найпопулярнішою[8], і чи не єдиною, яку можна зручно використовувати з Unity на Android. Головною її перевагою є простота – базу даних можна ініціалізувати напряму з класів, що зберігають відповідні дані, а підключення відбувається до локального файлу без необхідності підключення до інтернету та без використання сторонніх засобів для створення локального серверу.

Для контролю версій було використано PlasticSCM, адже він є частиною сервісів Unity та, відповідно, нативно підтримується у Unity Editor. Головною відмінністю від git є спрямованість на першочергове використання системи у вигляді графічного додатку, а не консольного та швидкодія у разі роботі з великими файлами.

Специфіка розробки додатків із використанням Unity

Оскільки першочергово Unity – це ігровий рушій, він надає функціонал необхідний саме для розробки ігор. Через це у порівнянні з іншими підходами до розробки на Android, як Jetpack Compose та PWA, деякі елементи робити

значно легше, а інші майже повністю доводиться робити вручну. Так, наприклад, хоча значно легше працювати з фізичними симуляціями та спецефектами, робота з інтерфейсом, що має вигляд форм передбачає ручне створення, керування та оновлення. Це дає змогу створювати унікальний та функціональний інтерфейс, проте значно уповільнює розробку та негативно впливає на якість кінцевого продукту

Поріг входження у розробку на Unity є порівняно низьким, оскільки C# підтримує різні парадигми та має синтаксис подібний до інших об'єктно-орієнтованих мов програмування, але часто надає більше вбудованих можливостей та «синтаксичного цукру», а Unity Editor надає інструменти для роботи зі сценою у візуальному редакторі без необхідності в специфічних навичках та знаннях. Простота розробки зумовила певну неструктурованість у підходах до розробки на Unity: деякі інструменти через свою зручність використовуються майже всюди, але загального підходу до архітектури немає і проекти на Unity у своєму технічному виконанні можуть значно відрізнятися. Також не характерним є використання автоматичних тестів, через що кількість засобів для їхнього проведення є малою, а якість – низькою.

Користувацький інтерфейс можна створювати як з коду, так і в графічному інтерфейсі Unity Editor. Часто використовуються префаби – об'єкти-заготовки які можуть формувати ієрархію із наслідуванням; після зміни префаба змінюються всі його нащадки та використані у сценах екземпляри. Зазвичай інтерфейс в іграх є сталим і такий підхід є досить ефективним. Коли доводиться створювати інтерфейс з коду так само використовуються префаби, але керування є не надто зручним, оскільки потрібно налаштовувати розміри та позиції усіх об'єктів та враховувати позицію кожного об'єкта у ієрархії. Це особливо незручно у разі роботи з формами, бо доводиться контролювати розмір полів, додавати нові, розширювати їх та батьківські контейнери.

Unity Gaming Services

Для надання серверних функцій було використано Unity Gaming Services – комплексну платформу для розробки, підтримування та розвитку застосунків. Хоча її і зручно використовувати саме з Unity Editor через інтеграції відповідних сервісів, здебільшого сервіси UGS можна використовувати незалежно від рушія Unity. Вагомою перевагою платформи є її гнучкість – новачки можуть використовувати сервіси «із коробки», а просунуті користувачі мають доступ до безлічі розширених налаштувань та можуть комунікувати з API напряду. UGS використовується як у малих проектах, так і в іграх великих корпорацій[9], демонструючи реальну гнучкість та масштабованість сервісів, що надаються.

Player Authentication – один із ключових сервісів, що дає змогу налаштувати автентифікацію користувачів, як анонімно так і через постачальників ідентифікаційних даних. Підтримуються як загальновідомі постачальники: Unity, Google, Apple, Facebook, Steam, Oculus, так і назадовні або OpenID. Оскільки Unity може використовуватися для мультиплатформенної розробки, для різних типів пристроїв можна задати різні способи автентифікації або різних провайдерів.

Analytics – сервіс збору даних про користувача та відповідний пристрій. Без додаткових налаштувань доступні дані про кількість, довжину та періодичність сесій, а також фінансові дані у разі мобільної розробки. Можна створювати власні події із назадовними параметрами та групувати користувачів за певними критеріями у групи. Це й же сервіс надає можливості для аналізу та візуалізації даних через обрані вручну метрики у графічному інтерфейсі або ж використовуючи SQL запити. Оскільки в багатьох країнах діють закони з захисту прав споживачів, Analytics залежно від регіону зазначає обмеження та вимоги до розробника щодо збору даних та погодження від користувачів. У застосунку сервіс було використано для збору даних про найпопулярніші слова, які потім можна запропонувати користувачам для вивчення.

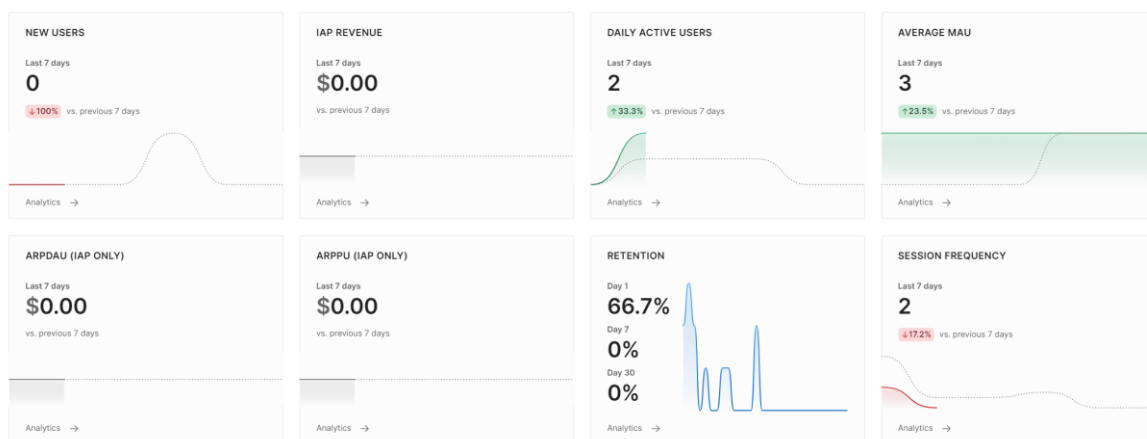


Рис. 3.1 – Дані Analytics, що відображаються за замовчуванням

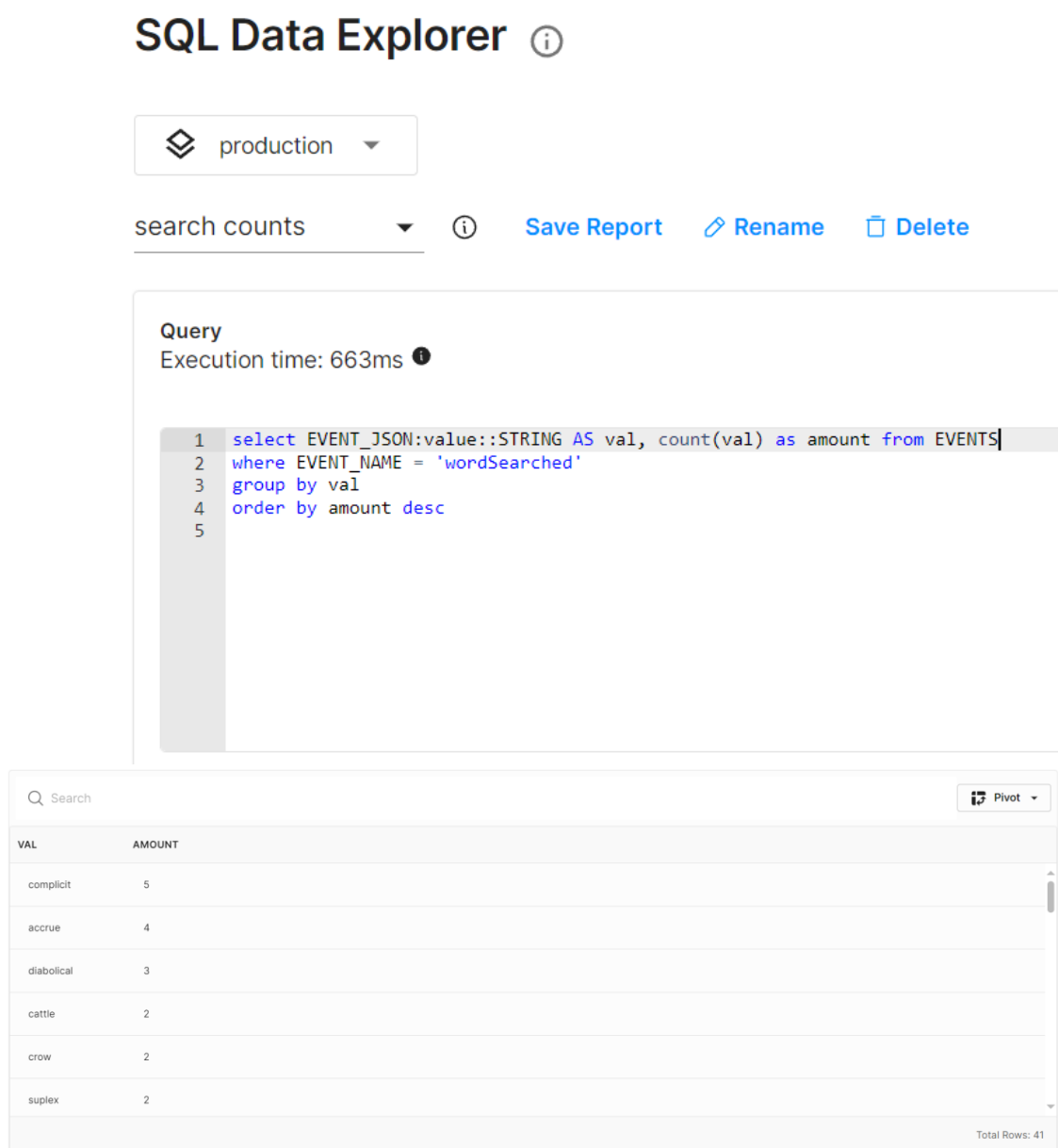


Рис. 3.2 – SQL запит та відповідний результат виконання

Cloud Code – сервіс для виконання у хмарі. Він дає змогу використовувати модулі на C# та JavaScript функції незалежно від клієнта. Це дає змогу змінювати відповідну логіку без необхідності оновлення застосунку на пристроях користувачів. У розробленому застосунку цей сервіс використовується для викликів до API словників незалежно від клієнта, а отже без необхідності зберігати ключі до API напряму у застосунку.

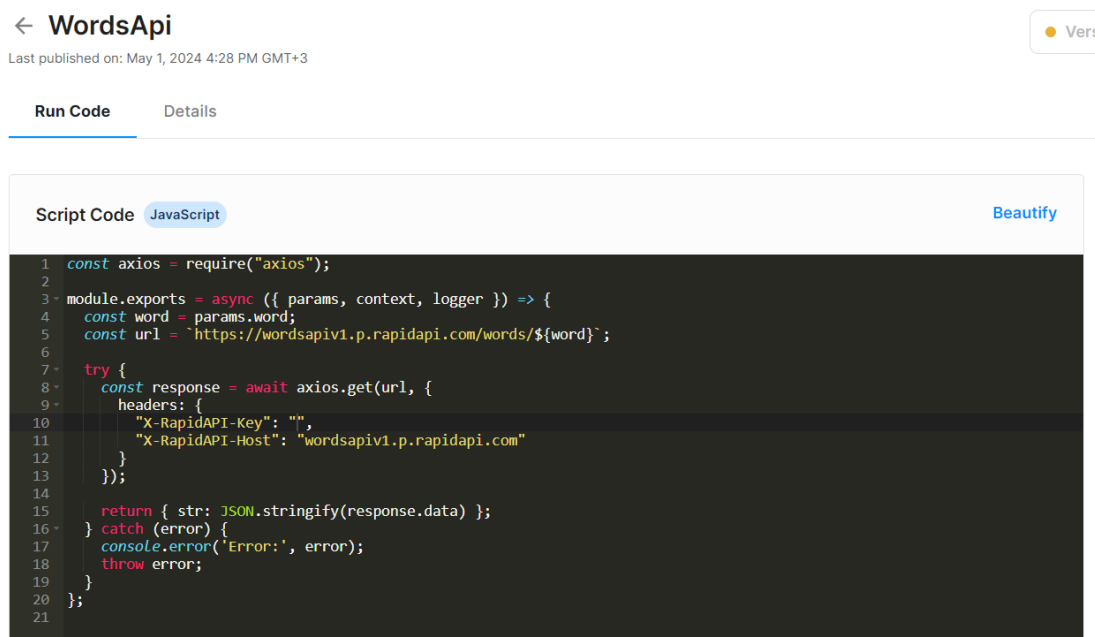


Рисунок 3.3 – Скрипт-посередник для викликів API

Cloud Save – сервіс для збереження стійких даних користувача незалежно від пристрою. Хоча сервіс насамперед і призначений для зручності користувачів, він також надає розробнику можливість відстеження прогресу користувачів і об'єднує дані із іншими доступними через Analytics. Крім того сервіс можна використовувати для збереження та оновлення спільних даних: стан світу, гільдії, досягнення тощо.

```
private async static Task<bool> CloudVersionIsNewer(string tempPath)
{
    try
    {
        byte[] file = await CloudSaveService.Instance.Files.Player.LoadBytesAsync("db");
        File.WriteAllBytes(tempPath, file);
        byte[] timerFile = await CloudSaveService.Instance.Files.Player.LoadBytesAsync("time");
        long cloudTime = BitConverter.ToInt64(timerFile, 0) - 1_000_000;
        long localTime = long.Parse(File.ReadAllText(Application.persistentDataPath + "/time.txt"));
        return cloudTime > localTime;
    }
    catch (Exception ex)
    {
        Debug.LogError(ex.Message);
        Debug.Log("No cloud db");
        return false;
    }
}
```

Рисунок 3.4 – Завантаження даних користувача з хмари

Remote Config – сервіс для зміни налаштувань застосунку без необхідності у оновленні застосунку на пристроях користувачів. Цей сервіс щільно інтегрований із іншим - Game Overrides, що дає змогу робити індивідуальні налаштування для різних категорій користувачів. Використовуючи дані з Analytics можна прицільно налаштовувати застосунок для користувачів з різних регіонів або інших визначених груп. Також сервіс дає змогу проводити А/В тести для порівняння того як різні параметри впливатимуть на взаємодію користувача із застосунком.

Schedule

Your schedule dictates when and for how long your changes will be in place.

Edit Schedule

Start Date

May 01 2024, 09:17 UTC

End Date

Run indefinitely

Priority

Medium (default)

Audience

These are the players that are eligible to enter your experience.

Edit Audience

Audience Type

Audiences

Audience Detail

New Players, Non-Spenders

Rollout Percentage

100% of selected audience

Content

The content of your experience.

Edit Content

Variant 1 - (15%)

Variant 2 - (85%)

🔗 Config Overrides

Key Name

Type

Value

TagsTuples

json

[[{"Formal","Informal"},{"Approving","Disapproving"}, {"Figuratively","Literally"}, {"Old","Modern"}, {"Old","Slang"}, {"Informal","Specialized"}]]

Рисунок 3.5 – Game Override із A/B тестуванням для значення TagTuples у Remote Config

Взаємодія із ОС Android

Оскільки застосунок призначений для допомоги у вивченні слів які обирає користувач, було вирішено додати можливість зберігати слова з текстів. Для цього в ОС Android використовується Activity з Intent PROCESS_TEXT. Загалом процес додавання активності не відрізняється від аналогічного у разі розробки на Java чи Kotlin, але для створення активності потрібно використовувати сторонні засоби, що без контексту застосунку не відображають помилки у коді і їх доводиться обробляти на етапі побудови застосунку у Unity Editor.

```

public class ActionTextActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        CharSequence text = getIntent().getCharSequenceExtra(Intent.EXTRA_PROCESS_TEXT);
        String input = text.toString();
        try {
            File file = new File(getExternalFilesDir(null), child:"bookmarked.txt");
            FileOutputStream fileout = new FileOutputStream(file, append:true);
            OutputStreamWriter outputWriter = new OutputStreamWriter(fileout);
            outputWriter.write(str:"\n");
            outputWriter.write(input);
            outputWriter.close();
        } catch (Exception e) {
            e.printStackTrace();
            Log.e("ActionTextActivity", "Error writing to file");
        }
        finish();
    }
}

```

Рисунок 3.6 – Код активності написаний на Java

```

<activity
    android:name="com.kolomiichuk.meanintor.ActionTextActivity"
    android:label="Bookmark"
    android:exported="true"
    android:theme="@android:style/Theme.Translucent.NoTitleBar">
    <intent-filter>
        <action android:name="android.intent.action.PROCESS_TEXT" />
        <category android:name="android.intent.category.DEFAULT" />
        <data android:mimeType="text/plain" />
    </intent-filter>
</activity>

```

Рисунок 3.7 – Визначення активності у AndroidManifest

Досвід кінцевого користувача

Основною відмінністю у кінцевому досвіді користувача є порівняно довгий час запуску застосунку. Хоча інтуїтивно додаток на Unity не має працювати ефективно, під час тестування застосунку не було виявлено особливої різниці у енергоспоживанні у порівнянні із іншими подібними застосунками.

Висновки

У процесі розробки було проаналізовано різні застосунки для вивчення англійської мови та досліджено різні підходи до вивчення лексики. Було визначено основні недоліки наявних засобів та розроблено нові режими для тренувань лексики.

Було оглянуто хмарні сервіси Unity та їхнє практичне використання. Завдяки використанню цих сервісів було впроваджено додаткові можливості для покращення досвіду кінцевого користувача.

У процесі створення графічного інтерфейсу було виявлено значні недоліки інструментів Unity Editor для створення форм, через які розробка значно уповільнилася. Для створення застосунків, що містять багато форм або інших змінних елементів інтерфейсу Unity не є ефективним засобом розробки.

Для подальших досліджень можна виконати аналіз Unity UI Toolkit та хмарних сервісів AWS та Azure.

Використані джерела

- [1] It's a free app loved by millions. Is Duolingo wasting your time? [Електронний ресурс]. – Режим доступу: <https://www.fluentlanguage.co.uk/blog/duolingo-review-fluentlanguage> (дата звернення: 14.05.2024). – Назва з екрана.
- [2] The 13 Best English Vocabulary Apps for Language Learners [Електронний ресурс]. – Режим доступу: <https://www.fluentu.com/blog/english/best-english-vocabulary-app/> (дата звернення: 14.05.2024). – Назва з екрана.
- [3] Creating games for Android [Електронний ресурс]. – Режим доступу: <https://unity.com/solutions/mobile/android-game-development> (дата звернення: 15.05.2024). – Назва з екрана.
- [4] .NET and .NET core support policy [Електронний ресурс]. – Режим доступу: <https://dotnet.microsoft.com/en-us/platform/support/policy/dotnet-core#cadence> (дата звернення: 15.05.2024). – Назва з екрана.
- [5] C# compiler [Електронний ресурс]. – Режим доступу: <https://docs.unity3d.com/Manual/CSharpCompiler.html> (дата звернення: 15.05.2024). – Назва з екрана.
- [6] IL2CPP overview [Електронний ресурс]. – Режим доступу: <https://docs.unity3d.com/Manual/IL2CPP.html> (дата звернення: 15.05.2024). – Назва з екрана.
- [7] Scripting restrictions [Електронний ресурс]. – Режим доступу: <https://docs.unity3d.com/Manual/ScriptingRestrictions.html> (дата звернення: 15.05.2024). – Назва з екрана.
- [8] Most widely deployed SQL database engine [Електронний ресурс] – Режим доступу: <https://www.sqlite.org/mostdeployed.html> (дата звернення: 15.05.2024). – Назва з екрана.

[9] Unity gaming services: live game development platform [Електронний ресурс] – Режим доступу: <https://unity.com/solutions/gaming-services#case-studies> (дата звернення: 16.05.2024). – Назва з екрана.