

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
Навчально-науковий інститут Інформаційних технологій
Кафедра інформаційних технологій

Пояснювальна записка

до бакалаврської роботи
на ступінь вищої освіти бакалавр

на тему: «РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ ДЛЯ ПІДГОТОВКИ ДО ЗНО НА
МОВІ ПРОГРАМУВАННЯ C#»

Виконав: студент 4 курсу, групи ПД-42
спеціальності

121 Інженерія програмного забезпечення

(шифр і назва спеціальності)

Гришаєнко В.І

(прізвище та ініціали)

Керівник Жебка В.В

(прізвище та ініціали)

Рецензент _____

(прізвище та ініціали)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

Навчально-науковий інститут Інформаційних технологій

Кафедра Інженерії програмного забезпечення _____

Ступінь вищої освіти - «Бакалавр» _____

Напрямок підготовки - 121 – Інженерія програмного забезпечення _____

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного забезпечення

О.В Негоденко

— ___ || _____ 2021 року

ЗАВДАННЯ НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ

Гришаєнко Владислав Ігорович

(прізвище, ім'я, по батькові)

1. Тема роботи: «Розробка мобільного додатку для підготовки до ЗНО на мові програмування C#»

Керівник роботи Жебка В.В, доцент , кандидат технічних наук

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від — «12» березня 2021 року № 65;.

2. Строк подання студентом роботи 01.06.2021

3. Вхідні дані до роботи:

Дані про процес проходження ЗНО;

Алгоритм дії навчальних додатків;

Розробка мобільного додатку для ОС Android;

Науково-технічна література, пов'язаних з розробкою програмного продукту на Unity для ОС Android;

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити).

4.1 Аналіз процесу ЗНО.

4.2 Аналіз додатків для навчання дистанційно.

4.3 Дослідження додатків для розробки програмного продукту для ОС Android.

4.4 Розробка додатку.

5. Перелік графічного матеріалу

1. _____

2. _____

3. _____

4. _____

6. Дата видачі завдання 19.04.2021

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			

Студент _____
(підпис) (прізвище та ініціали)

Керівник роботи _____
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Текстова частина бакалаврської роботи 47с., 33 рис., 15 джерел.

Об'єкт дослідження – підвищення знань в одному із трьох доступних предметів.

Предмет дослідження – додаток для підготовки, та перевірки знань до ЗНО.

Мета роботи – розробка програмного забезпечення для підготовки до ЗНО на мові програмування C#.

Методи дослідження – методи розробки програмного забезпечення, методи проектування та створення архітектури ПЗ.

Наукова новизна даного проекту полягає в наступному:

- розроблений додаток в якому одночасно можна підготуватися до трьох предметів ЗНО;
- встановлено, що працюючи з Unity, потрібно знати мову програмування C#, для створення гарного додатку;
- доведено, що мова програмування C# проста у використанні, та є конкурентоспроможна у середі розробки додатків для мобільних додатків;
- на основі проведених досліджень, були зроблені висновки та розроблено ПЗ;

Створений додаток допоможе не тільки учням, які збираються складати ЗНО, а і тим, хто просто хоче покращити свої навички в певному предметі.

Під час досліджень, та розробки додатку, був проведений аналіз схожих додатків на ОС Android через безкоштовний ресурс Play Market. Після цього були встановлені основні переваги, та недоліки таких додатків. В результаті, був створений список потреб, та функцій, які мають бути реалізовані в додатку.

Галузь використання – оскільки додаток розповсюджується безкоштовно, доступ до нього може отримати будь-хто, котрий має намір навчатися.

Ключові слова: UNITY, C#, ЗНО, GIT, API, VISUAL STUDIO, PHPMYADMIN, БД, ANDROID.

ЗМІСТ

ВСТУП	16
1. АНАЛІЗ ПРЕДМЕТІВ ТА ПРОЦЕСУ ПІДГОТОВКИ ДО ЗНО.....	10
1.1 Аналіз процесу ЗНО.....	10
1.2 Практичні та теоретичні знання під час підготовки до ЗНО.....	11
1.3 Огляд та аналіз існуючих додатків для підготовки до ЗНО	14
2. ВИБІР ПРОГРАМНИХ ЗАСОБІВ ДЛЯ РОЗРОБКИ ДОДАТКУ	22
2.1 Вибір середовища для розробки ПЗ.....	22
2.2 Вибір мови програмування для розробки ПЗ.....	26
2.3 Unity – основні компоненти та інтерфейс середовища.....	27
2.4 Вибір програми для створення дизайну для додатку.....	33
2.5 Контроль версій за допомогою Git та GitHub	36
3. ПРОЕКТУВАННЯ ТА РОЗРОБКА ДОДАТКУ.....	38
3.1 Діаграма варіантів використання	38
3.2 Діаграма діяльності додатку	40
3.3 Основні класи та функції додатку	41
3.3.1 Зчитування інформації	41
3.3.2 Клас для контролю типу навчання та предмета	44
3.3.3 Навігація по додатку.....	44
3.3.4 Генерація панелей.....	45
3.3.5 Заповнення панелей інформацією.....	47
3.3.6 Клас тестування	50
3.4 Створення БД та взаємодія з нею.....	51
ВИСНОВКИ.....	56
ЛІТЕРАТУРА	58
ДОДАТОК А.....	59
ДОДАТОК Б	65

УМОВНІ ПОЗНАЧЕННЯ

ПЗ – програмне забезпечення

БД – база даних

UML – уніфікована мова моделювання

UI – інтерфейс користувача

UX – досвід користувача

ЗНО – зовнішнє незалежне оцінювання

ВСТУП

Дистанційне навчання – це різноманітна сукупність нових технологій, за допомогою яких, можна навчатися онлайн. Перед усім використовуються настільні комп'ютери, ноутбуки, мобільні гаджети та інші технології. Перевагою перед звичайним навчання, є те, що не потрібно витратити багато часу на поїздки до начального закладу або додому, також кожний зможе в будь-який час отримати інформацію, яку він захоче, оскільки вона знаходиться у вільному доступі. Це надає перевагу навчатися з дому самостійно, навіть без вчителів. Таку перевагу надають додатки, для підготовки до ЗНО, або до конкретних предметів, в яких учень хоче поглибитись. Ми можемо самі обирати предмети, які ми хочемо вчити і тим самим найкращим шляхом підготуватися до екзаменів. Всі ці додатки оновлюються кожен рік, а той місяць, що дає можливість не пропустити найактуальнішу інформацію про нововведення у конкретному предметі, або про його зміни.

У зв'язку з теперішньою ситуацією в світі, такі додатки стають ще більш популярними, оскільки люди на дистанційному навчанні намагаються отримати, як найбільш корисної інформації, а саме такі ресурси можуть їм це надати.

Мета роботи – розробка програмного забезпечення для підготовки до ЗНО на мові програмування C#.

Об'єкт дослідження – підвищення знань в одному із трьох доступних предметів.

Предмет дослідження – додаток для підготовки, та перевірки знань до ЗНО.

Наукова новизна даної роботи полягає в:

- Розроблений додаток в якому одночасно можна підготуватися до трьох предметів ЗНО;

- Встановлено, що працюючи з Unity, потрібно знати мову програмування C#, для створення гарного та стабільного додатку;
- Доведено, що мова програмування C# проста у використанні, та є конкурентоспроможна у середі розробки додатків для мобільних додатків;
- На основі проведених досліджень, були зроблені висновки та розроблено ПЗ;

Практична значущість результатів дослідження, полягає у створенні програмного додатку для мобільних телефонів, яке буде допомагати учням перевіряти та вдосконалювати свої знання, в області трьох предметів:

- Українська мова;
- Українська література;
- Математика.

Результати досліджень:

1. На XI науково-технічній конференції студентів та молодих вчених навчально-наукового інституту інформаційних технологій Державного Університету Телекомунікацій «ОСОБЛИВОСТІ ДИСТАНЦІЙНОГО НАВЧАННЯ В 2020 РОЦІ»
2. На всеукраїнській науково-технічній конференції «ДОСЛІДЖЕННЯ МОБІЛЬНИХ ДОДАТКІВ ДЛЯ ПІДГОТОВКИ ДО ЗНО»

1. АНАЛІЗ ПРЕДМЕТІВ ТА ПРОЦЕСУ ПІДГОТОВКИ ДО ЗНО

1.1 Аналіз процесу ЗНО

ЗНО – це екзамен зовнішнього незалежного оцінювання, який починаючи з 2004 року і до цього моменту, складають всі учні одинадцятих класів, для того, щоб вступити до університету. З 2021 року, учні мають можливість обрати п'ять предметів для написання під час ЗНО, два з яких будуть обов'язковими, третій на вибір між Історією України та Іноземною мовою. Математика та Українська мова і література повинні будуть написані всіма абітурієнтами, інші три предмета, вибирає кожен для себе сам. Інші предмети обирає сам учень.

Також, оцінки за обов'язкові предмети, будуть конвертовані в оцінку ДПА, яка буде відображена в кінцевому атестаті, та буде впливати на остаточний бал під час вступу до університету. Інші два предмети, для ДПА, учень має вибрати сам, 3-й предмет: Історія України або Іноземна мова; 4-й предмет, будь-який інший, якого ще не було у списку.

У цьому році змінилися правила написання ЗНО з двох предметів [13]. Українська мова і література, тепер складається з двох рівнів, абітурієнт має вибрати один із них, перед початком написання:

- українська мова;
- українська література.

Абітурієнт, повинен обрати рівень, якого буде достатньо для його майбутньої спеціальності, рівень встановлюється університетом, в який поступає абітурієнт. Це надає перевагу тим, хто буде вступати на технічну спеціальність, бо вони просто можуть не здавати обов'язковий іспит з української літератури та зосередитись на профільних предметах.

Математика також тепер має два рівні:

- стандарт;

- поглиблений.

Учні, які будуть вступати до гуманітарних університетів, зможуть вибрати стандартний рівень, та без проблем написати тести. Також на кожному рівні передбачаються підказки, у вигляді основних формул, які будуть сприяти пришвидшенню написання тестів.

Після складання усіх екзаменів, всі мають чекати на перевірку завдань. Зазвичай результати будь відомі, через декілька тижнів. Після цього, абітурієнти можуть починати подавати заяви на вступ, до університетів. В 2021 році, на бюджет можна подати тільки 5 заяв, а на контракт до 30. Але треба бути уважним, подаючи заяви на бюджет, оскільки порядок університетів грає дуже велику роль. Абітурієнт може без проблем потрапляти на безоплатне навчання до престижного університету, але через його невдалий порядок у списку, він втратить це місце. Тому, подачу заяв, треба здійснювати дуже обережно. Після того, як заяви прийняті, всі повинні перевіряти списки, та дивитися, чи проходять вони на бюджет, у разі успіху, в конкретний день, який буде назначений Міністерством Освіти України, учень повинен привезти перелік документів, для остаточного вступу до університету. Без оригіналів, вступ не є можливим і воно буде віддане іншому учню. Якщо абітурієнт не проходить на бюджет, він може з легкістю вступити на контракт, тобто платне навчання. Але, потрібно мати на увазі, що кожний університет має свій пороговий бал для вступу на певну спеціальність, як для бюджету, так і для контракту, на контракті середній бал значно нижчий.

Основна сесія ЗНО 2021 року, буде проходити з 21 травня до 10 червня включно [13].

1.2 Практичні та теоретичні знання під час підготовки до ЗНО

Кожен учень має дуже гарно підготуватися до екзамену, оскільки від його результату, буде залежати вступ, до того університету, в який він

націлений. На тест, потрібно готуватися не тільки практично, тобто вчити літературу, формули та читати книжки, але і готуватися морально, бо це також дуже важлива складова. Є дуже багато прикладів, коли відмінники, які знають бездоганно предмет, можуть допускати помилки в елементарних завданнях, бо дуже хвилюються, тому потрібно розуміти, що це просто екзамен, такий самий як і в школі, тоді процес піде краще. Звісно в екстрених ситуаціях існують додаткові сесії для перездачі предмету, але це буде ще більший стрес для людини.

Але звісно, основна підготовка, це проходження матеріалу, по майбутнім предметам. Основна частина всіх екзаменів складається з тестів, які можна дуже легко вирішити, вибравши один, або декілька правильних варіантів, в залежності від типу завдання.

В українській мові і літературі, всі завдання, це тести різних видів з декількома, або одним правильним варіантом. Тільки останнє завдання потребує написанню власного висловлювання на певну тему, яка буде відома тільки під час самого тесту. Це дуже важливий розділ, оскільки за нього дають дуже багато балів, які не дуже важко отримати. Для того, щоб отримати максимальну кількість балів, звісно потрібно багато практики, бо втратити бали за орфографічну помилку не хочеться. Також бажано багато читати, навіть зарубіжної літератури, оскільки дуже важливо надати приклад з твору, це дасть можливість отримати більше балів за твір. Якщо, немає можливості навести приклад із твору, то можна згадати момент, з фільму, мультфільму, серіалу або власного життя. Якщо ж, зовсім нічого не приходить в голову, то можна схитрити, та вигадати такий момент у своєму житті, або житті своїх знайомих.

Тому, щоб підготуватися до цих двох предметів, достатньо пройтися по шкільній програмі, згадати всі правила, та виконувати практичні заняття, які схожі з тестами. Для підготовки до української літератури, потрібно прочитати всі твори та вірші, які є в програмі ЗНО, якщо немає часу на повні

твори, то можна читати скорочено, але при цьому втрачається велика кількість матеріалу, який може бути в тестах.

Оскільки, в цьому році екзамен по математиці дещо полегшили, і тепер не потрібно знати всі формули на пам'ять, бо вони вже будуть на бланку, можна більш зосередитися на останньому завданні. На відмінність від української літератури, останнє завдання з математики кожний раз має різних характер, тобто це може бути якась тема з алгебри, або геометрії. Для розв'язання таких завдань на максимальний бал, потрібно просто практикуватися, бо інакше ніяк не вийде його зробити. В додатку є багато посилань на онлайн ресурси, на яких можна подивитися приклади та способи їх вирішення.

Нові правила в математиці не означають, що не потрібно вчити формули зовсім, бо ви зможете використати ту формулу, якої немає в бланку, те тільки для тестів, а і останніх завдань. Щоб закріпити знання в математиці, потрібно багато практикуватися і робити купу завдань. Це можна робити за допомогою онлайн ресурсів або книжок, в яких є перелік завдань схожих із завданнями ЗНО.

На ЗНО також є багато заборон, не дотримання яких, може дуже багато коштувати для учня. Через порушення будь-якого з правил, можеш бути вигнаним з аудиторії, та всі завдання будуть анульовані. За останній рік з'явилися нові правила через карантинні обмеження, щодо носіння маски, та дотримування дистанції, але залишаються основні правила, які не можна порушувати.

Що категорично не можна робити під час написання тесту [13]:

- спілкуватися з іншими учасниками в аудиторії. Таким чином ви не тільки наражаєте себе на небезпеку, але і іншу людину, яке захоче вам допомогти;
- використовувати засоби зв'язку, або будь-яку іншу техніку, яка може дати змогу отримати відповіді на питання, або отримати додаткову інформацію;

- запізнюватися на тест. Найкраще бути в аудиторії заздалегідь. Допуск до кабінету забороняється, за десять хвилин до початку основної сесії, бо в цей час починається розповідь про час проведення тестування, правила та інші організаційні моменти.

1.3 Огляд та аналіз існуючих додатків для підготовки до ЗНО

За останній рік, ситуація в навчанні, та в світі в цілому, дуже змінилася. Зараз в основному всі знаходяться на дистанційному навчанні, що потребує використання нових технологій, програмного забезпечення та багато чого іншого. Тенденція в світі змінюється не тільки в сторону діджиталізації, а і до більшого використання мобільних телефонів, і все менше використання персональних комп'ютерів. По даним ресурсів “Hootsuite” [11] та “we are social” [12], можемо побачити, що кожний квартал доля мобільних гаджетів тільки зростає, за останні декілька років в середньому на 8,92%, що є дуже гарним показником, статистика зображена на рис. 1.1.

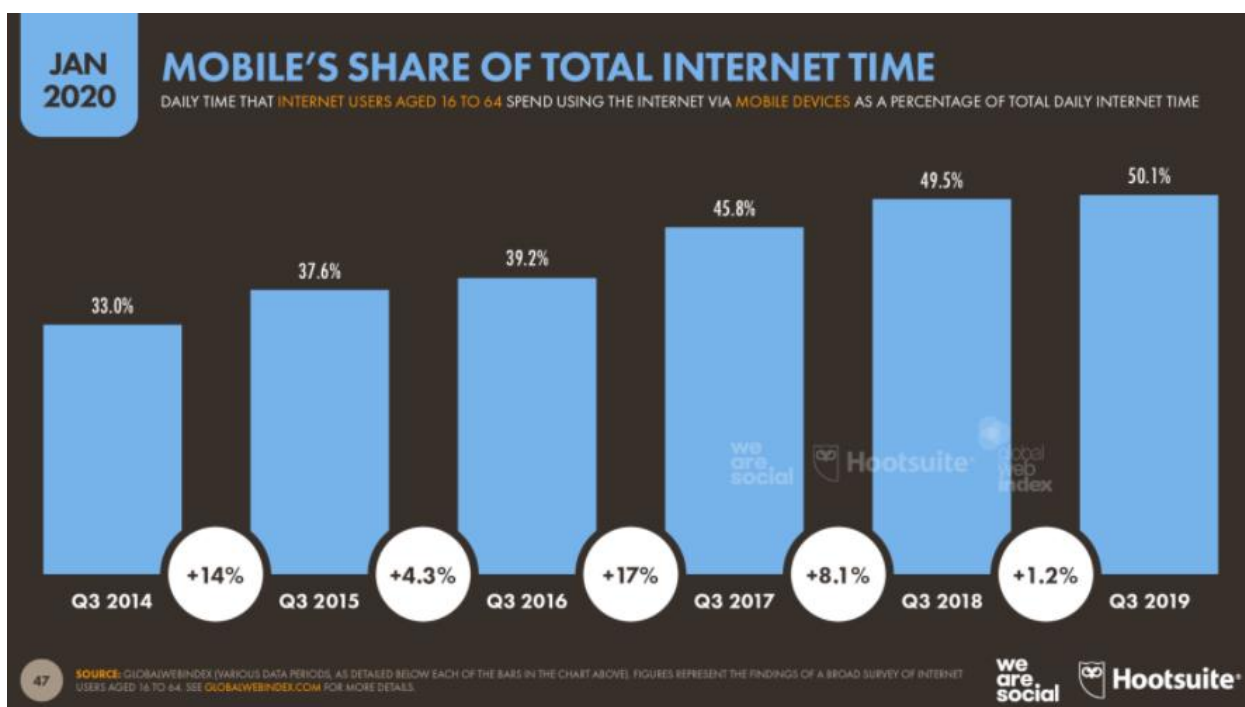


Рисунок 1.1 — Статистика мобільних гаджетів по рокам

Якщо дивитися статистику по всім додаткам, то на даний момент мобільні гаджети використовують 53,3% користувачів, для того щоб користуватися інтернетом, цю статистику можна побачити на рис. 1.2.

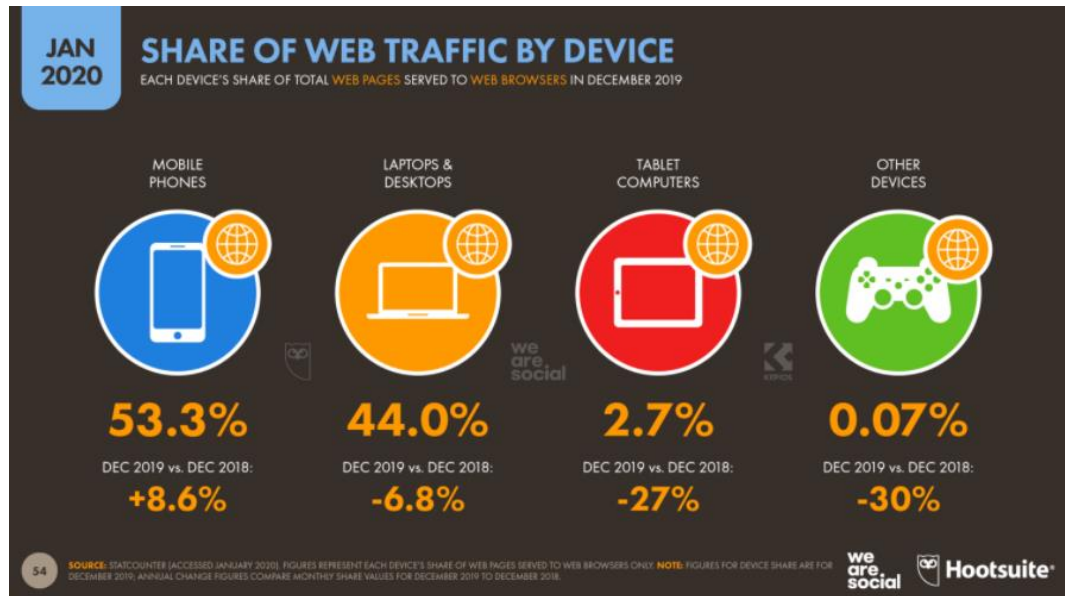


Рисунок 1.2 — Статистика по всім гаджетам

Оскільки мій додаток створений для допомоги, щодо підготовки до ЗНО, то я буду порівнювати його зі схожими аналогами, які я буду шукати на безкоштовному ресурсі Google Play Market. Тут можна дуже легко знайти схожі додатки, та подивитися їх статистику.

Проаналізувавши, можна виділити п'ять додатків, які ми будемо порівнювати по різним параметрам:

- “ЗНО 2020 тести: Українська література”;
- “ЗНО 2021. Історія України”;
- “ЗНО 2021 на 190+. Мова і література”;
- “ZNO”;
- “Складу ЗНО”.

В основу більшості таких додатків, лежить можливість підготуватися тільки до одного предмету, що робить не зручним користування, бо потрібно завантажити велику кількість додатків, щоб вивчати матеріал. В кожному з

них, можна виконувати завдання, проходити різноманітні тести, читати корисну інформацію. В цілому, всі додатки такого типу мають позитивні оцінки, в моєму списку, це також видно. Середня оцінка, 4 бали та більше. Лише один додаток має нижчу оцінку, це додаток “ZNO”, його оцінка лише 2.7 балів.

Статистика завантажень, також дуже гарна, для такого виду додатків, оскільки аудиторія досить не велика. В середньому, гарний додаток має від 50 до 100 тисяч завантажень, але тенденція в тому, що цифри тільки збільшуються.

Статистика проаналізованих додатків:

- “ЗНО 2020 тести: Українська література”:

Додаток має 50+ тисяч завантажень та більше ніж 1000 відгуків. “В цьому додатку можна знайти дуже гарну систему тестів, які мають дуже обширне покриття темами, таким чином засвоїти матеріал буде ще набагато простіше. Але єдиний мінус цього додатку, в тому, що він дає змогу підготуватися тільки до одного предмету.

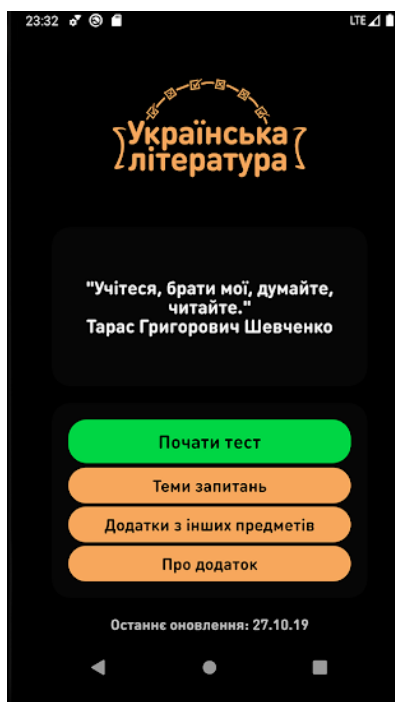


Рисунок 1.3 — Інтерфейс додатку “ЗНО 2020 тести: Українська література”

- “ЗНО 2021 на 190+. Мова і література”:

Даний додаток має 100+ тисяч завантажень і більше 1500 позитивних відгуків. В цьому додатку дуже гарно зроблено вивчення української мови, тут можна слухати наголоси, вивчати фразеологізми, постійно йде повторення пройденого матеріалу та аудіо завдання. Також додаток має досить стильний дизайн, та зрозумілий інтерфейс для всіх користувачів.

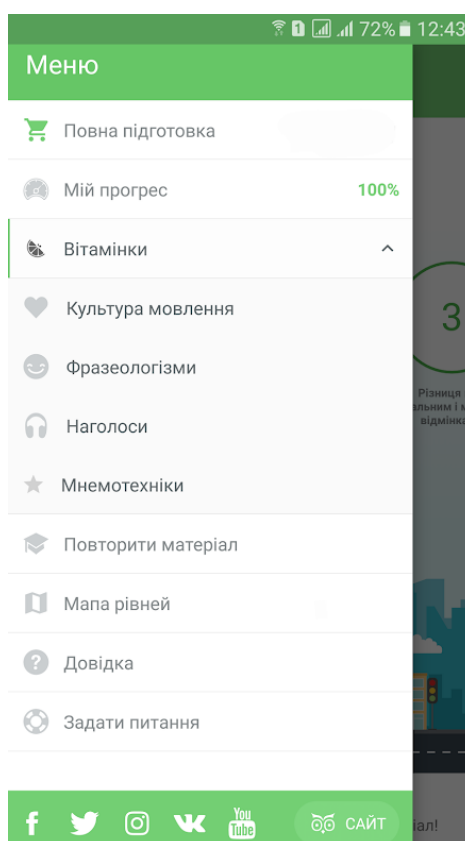


Рисунок 1.4 — Інтерфейс додатку “ЗНО 2021 на 190+. Мова і література”

- “ЗНО 2021. Історія України”:

Має більше 100 тисяч завантажень і 1000+ відгуків. Даний додаток має добре підібрані теоретичні джерела, все описано по дрібницям і не потрібно шукати додаткові книжки або ресурси, щоб доповнити тему, додатка повністю достатньо, для того, щоб вивчити предмет. Також тести по історичним датам, дадуть змогу якнайшвидше їх запам’ятати. Дизайн

складається тільки з двох кольорів, що дає змогу зосередитись на навчанні і не відволікатися від тестів.



Дата проведення ЗНО - червень 2021 р.

Рисунок 1.5 — Інтерфейс додатку “ЗНО 2021. Історія України”

- “ZNO”:

Цей додаток має більше 10000 тисяч завантажень. Перевага в цього додатку, в тому, що він дає можливість підготуватися до декількох предметів одночасно. Але у той же час, цей список предметів дуже малий, і матеріалу не достатньо для повної підготовки. Також є проблеми з оптимізацією на деяких девайсах, тому у нього найнижча оцінка серед списку додатків.



Рисунок 1.6 — Інтерфейс додатку “ZNO”

- “Складу ЗНО”:

Має більше тисячі завантажень. Як і в минулому додатку тут можна підготуватися до декількох предметів, але тут список вже дуже значущий. Хоч додаток і має дуже простий дизайн, ним легко користуватися. Додаток працює майже завжди стабільно і не має суттєвих критичних помилок. Мінус тільки в тому, що даний додаток має трохи застарілий дизайн, але він не грає головну роль в додатку.



Рисунок 1.7 — Інтерфейс додатку “Складу ЗНО”

Проаналізувавши цю статистику, можна порахувати, який відсоток користувачів залишають відгуки після завантаження і середню кількість завантажень. В нашому випадку, це 1.5%, що є дуже гарним результатом, оскільки такі додатки в основному отримують 1-2% відгуків від своїх користувачів. Середня кількість завантажень 50 тисяч, що є гарним показником серед навчальних додатків в секторі підготовки до ЗНО.

Основний функціонал всіх цих додатків – це тести, які для кожного предмету є унікальним та теоретична інформація. Проходячи такі тести, учень, може запам’ятовувати інформацію, правильні відповіді, та використовувати їх в майбутньому. Більш прогресивні додатки, дають змогу

купити підписку, щоб мати більше різноманітних функцій. Вони отримають більше тестів, підтримку, яка буде допомагати з питаннями, та можливо якісь додаткові унікальні можливості. Але все це, не дуже впливає на процес навчання, і не коштує своїх грошей.

В моєму ж додатку будуть всі ці функції, вони дадуть можливість більш краще та професійно підготуватися до предметів. Функції які реалізовані в моєму додатку:

- унікальні тести для кожного з трьох предметів;
- посилання на відео, в яких розповідається, про конкретну тему, яка буде на ЗНО;
- інформація про ЗНО в цілому, процес його проходження, правила та корисні підказки;
- корисні посилання, для кожного предмету;
- веб-ресурси, на яких можна підготуватися додатково до кожного з трьох предметів;
- приклади власного висловлювання в розділі з української мови;
- посилання на твори та вірші;
- рейтинг основних університетів України, котрий дасть змогу абітурієнту вибрати університет, в який він хоче вступити [15];
- уривки із текстів, в розділі української літератури;
- список формул для розділу математики.

2. ВИБІР ПРОГРАМНИХ ЗАСОБІВ ДЛЯ РОЗРОБКИ ДОДАТКУ

2.1 Вибір середовища для розробки ПЗ

Для вибору середовища для розробки ПЗ, будемо розглядати три основні методи: Unity, Android Studio та Xamarin.

Android Studio – це інтегроване середовище, для розробки ПЗ та мобільних додатків для платформ Android від компанії Google. Дану програму можна встановити на MacOS, Linux та Windows.

Перевага даного середовища:

- постійні оновлення, та підтримка від Google;
- інтерфейс дуже простий, в якому дуже легко розібратися;
- для того, щоб додати елемент до додатка, потрібно просто знайти його в списку, та перетягнути;
- всі елементи мають велику кількість налаштувань, їх повністю можна змінити під свій дизайн;
- в даній програмі можна використовувати такі мови програмування як: Java та Kotlin. Якщо використовувати Kotlin можна в великих об'ємах зекономити кількість коду;
- комфортний редактор коду;
- розробляти додатки можна не тільки для смартфонів та планшетів на Android, а і для портативних комп'ютерів, Android TV, Android Wear та багато чого іншого.

Недоліки:

- емулятори працюють не завжди коректно, а іноді просто зависають, або не запускають програму;
- складність тестування додатку;
- велика кількість застарілої інформації, та документації, навіть на офіційних сайтах від Google;

- потрібна велика продуктивність комп'ютера на якому розробляється додаток.

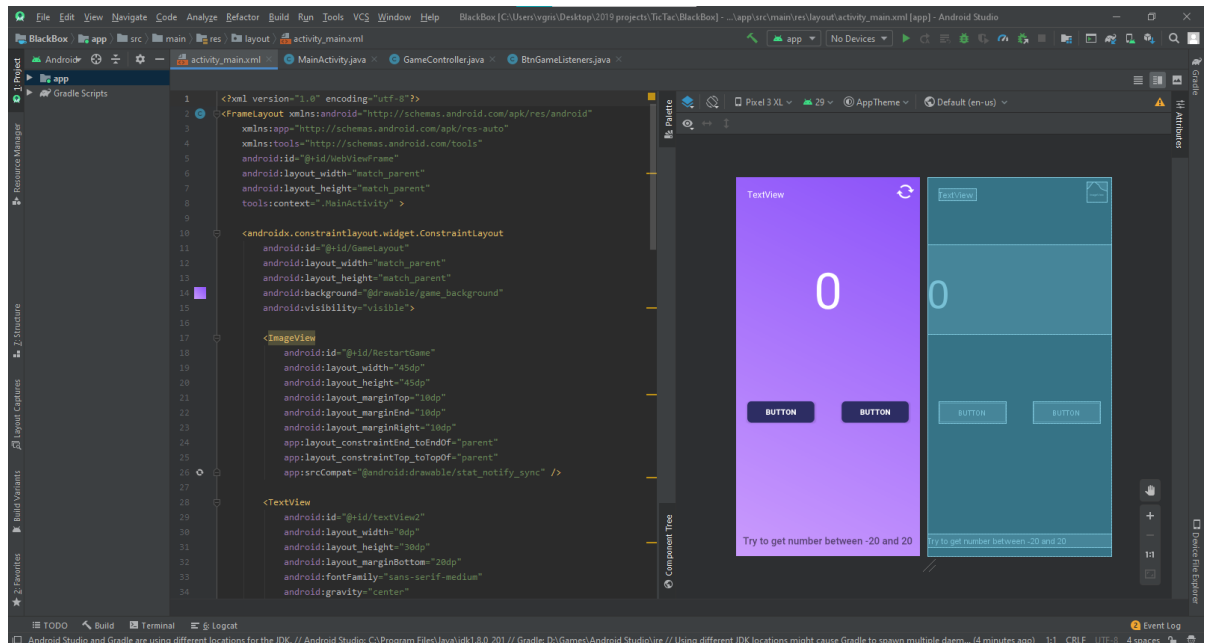


Рисунок 2.1 — Інтерфейс Android Studio

Unity – це міжплатформова середовище розробки ПЗ. Найчастіше цю програму використовують, як готовий рушій гри, але на ньому можна розробляти і нативні додатки, за допомогою легкої розробки інтерфейсу. Основна мова програмування, яка використовується, це C#.

Плюси даного способу розробки [3, с.21]:

- можливість розробити додаток на 25 різних платформах;
- не складний, та зрозумілий інтерфейс;
- дуже простий процес розробки UI, для того, щоб додати елемент в проект, потрібно просто вибрати потрібний елемент зі списку, та додати його.
- сайт с документацією для кожної версії програми;
- стабільні версії з довгим строком підтримки;
- також великим плюсом, є магазин асетів, в якому можна знайти дуже багато безкоштовних плагінів для ваших проектів.

- можливість одразу побачити, як буде виглядати додаток на телефоні;
- повне редагування всіх елементів;
- велика, розвинена екосистема, що дає можливість швидко вирішити свою проблему.

Мінуси даної програми [3, с.24]:

- складність розробки великих проектів;
- велика кількість елементів на сцені, дуже погано впливає на оптимізацію додатку, після цього потрібно додатково витратити час, на доопрацювання, або видаляти елементи із додатку;
- велика вага проекту.

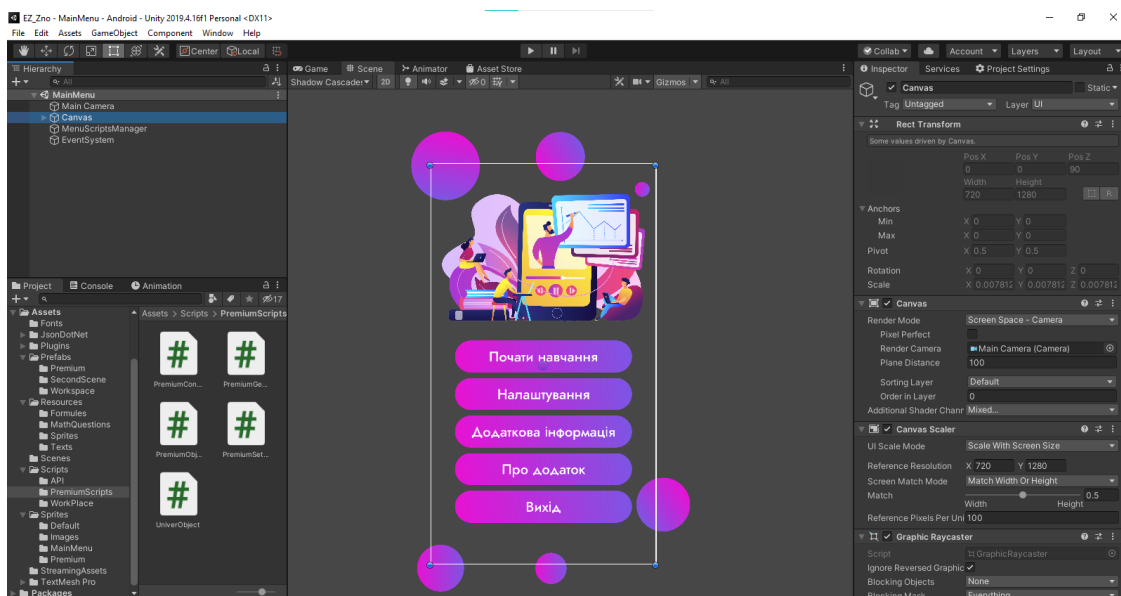


Рисунок 2.2 — Інтерфейс Unity

Xamarin – це платформа, яка дає можливість розробляти кросс-платформенні додатки. За допомогою Xamarin та мов програмування C#, або F# можна створювати додатки на такі основні платформи, як: Android, iOS, UWP, Tizan та MacOS, WPF, GTK#. Цей спосіб розробки має багато плюсів:

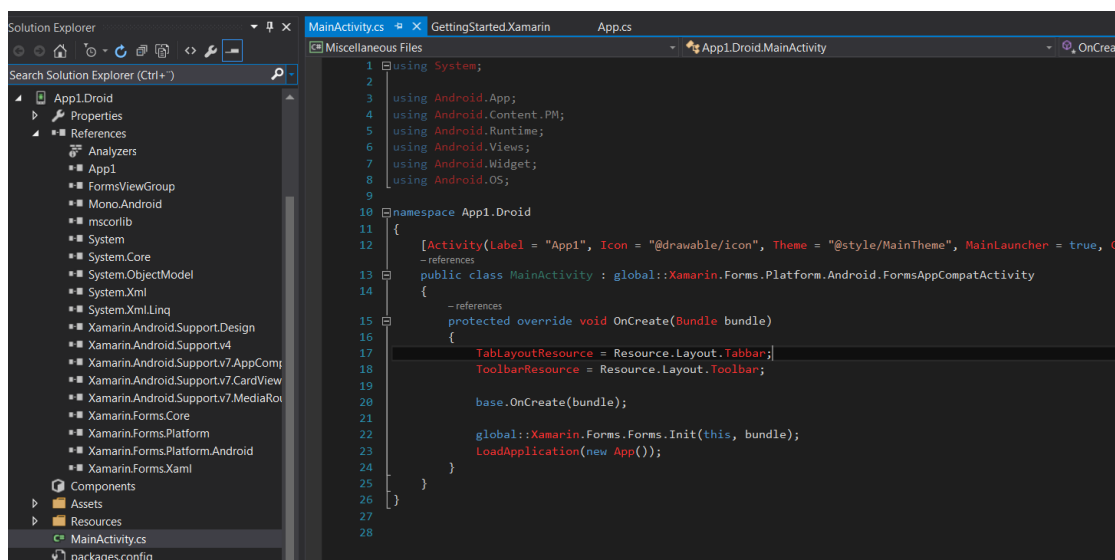
- єдиний стек технологій, який дає можливість зберігати основні частини коду, та не переписувати їх під певну платформу;

- велика продуктивність;
- дуже легко підтримувати, та оновлювати код;
- нативний UI;
- сумісність з обладнанням використовуючи різні плагіни та основні функції різних пристроїв, та різних платформ;
- проект з відкритим кодом.

Мінуси даного методу розробки:

- повільні оновлення для платформ;
- мала популярність. Для новачків буде дуже складно знайти розв'язання своєї проблеми в інтернеті, на форумі, оскільки їх екосистема не дуже розвинена;
- Xamarin не підходить для розробки додатків з великою кількістю анімацій та графіки;
- велика вага додатку, навіть тільки створений проект може важити 20МБ і все це через велику кількість бібліотек.

Для того, щоб почати користуватися Xamarin, потрібно завантажити Visual Studio і вже за допомогою цієї IDE та Xamarin почати розробляти додаток.



```

1  using System;
2
3  using Android.App;
4  using Android.Content.PM;
5  using Android.Runtime;
6  using Android.Views;
7  using Android.Widget;
8  using Android.OS;
9
10 namespace App1.Droid
11 {
12     [Activity(Label = "App1", Icon = "@drawable/icon", Theme = "@style/MainTheme", MainLauncher = true, Co
13     -references
14     public class MainActivity : global::Xamarin.Forms.Platform.Android.FormsAppCompatActivity
15     {
16         -references
17         protected override void OnCreate(Bundle bundle)
18         {
19             TabLayoutResource = Resource.Layout.Tabbar;
20             ToolbarResource = Resource.Layout.Toolbar;
21
22             base.OnCreate(bundle);
23
24             global::Xamarin.Forms.Forms.Init(this, bundle);
25             LoadApplication(new App());
26         }
27     }
28 }

```

Рисунок 2.3 — Приклад коду додатка на Xamarin

Проаналізувавши ці три програми, для створення ПЗ, я вибрав Unity. Оскільки він має простий у використанні інтерфейс, стабільні версії, з якими можна працювати і не боятися, що колись вона зламається. Велике ком'юніті, яке може допомогти вирішити проблему. Велика кількість налаштувань основних елементів проекту, та ассетів, які обов'язково допоможуть у розробці. Тому, розробка проекту, буде відбуватися за допомогою Unity.

2.2 Вибір мови програмування для розробки ПЗ

Unity дуже добре працює з багатьма мовами програмування, що є дуже гарною новиною, бо можна вибрати ту мову, яка найбільш сподобається. Але є лише декілька мов, які досі є актуальними, та на яких можна успішно програмувати та створювати ПЗ.

C# - це об'єктно-орієнтована мова програмування. Це основна мова для написання скриптів для Unity. Це дуже проста та продуктивна мова програмування, також її перевага в тому, що основні класи Unity, а саме Mono, всі створені на мові програмуванні C#. Тобто, якщо ти знаєш C#, тобі буде легше працювати з Unity.

Воо – також дуже проста та продуктивна мова, яка підтримує .NET та Mono. Але від неї поступово позбуваються, бо вона не популярна серед програмістів. Це дуже великий мінус, оскільки при наявності проблеми, яку ми не в змозі вирішити самі, ми не зможемо знайти вирішення, або допомогу в інтернеті.

C/C++ - найшвидші та найпродуктивніші мови програмування. Але вони є більш складнішими ніж C#. Тому, якщо ви новачок, то не слід одразу починати з цих мов. Розвинена екосистема, є велика кількість форумів та навчальних відео.

Lua – це мова програмування суто для ігор, або розробки модифікувань для них. Ця мова не може бути основною, вона має свій інтерпритатор, який дає змогу підключати деякі скрипти написані на Lua.

Проаналізувавши усі ці мови програмування, я вибрав C#, оскільки вони є дуже популярною. Можна знайти багато мануалів, та форумів з допомогою. Та основні уроки, інструкції та документація на писані саме на цій мові програмування. Тому, розробка проекту буде відбуватися на цій мові програмування.

IDE, тобто інтегрована середовище розробки, яку я буду використовувати для написання програмного коду, це Visual Studio.

2.3 Unity – основні компоненти та інтерфейс середовища

На перший погляд Unity має незрозумілий інтерфейс, який дуже складно зрозуміти, але це не так. Потрібно лише декілька годин, для того, щоб зрозуміти всі аспекти. На рис. 2.4 представлений основний інтерфейс Unity, на якому зображено:

- ієрархія компонентів – це список, в якому знаходяться всі елементи певної сцени, які ми додали на головну робочу середу. Основними елементами, є камера та канвас [4, с.106];
- список папок, це ієрархія самого проекту, а не тільки однієї сцени. В цих папках можуть знаходитися скрипти, спрайти, текстові файли та файли від певних завантажених ассетів із магазину;
- головна робоча середа, на яку, ми додаємо наші елементи. Для того, щоб додати елемент, треба просто вибрати його зі списку, і він одразу з'явиться на робочій середі і ієрархії сцени. На малюнку 2.4 ми можемо побачити сцену головного меню, і елементи головного меню на робочому місці;
- інспектор – це розділ, за допомогою якого, ми маємо можливість налаштувати елемент, за допомогою стандартних параметрів, або

за допомогою скриптів, які були написані нами. На рис. 2.4 ми бачимо інспектор канвасу;

- верхня панель робочої середи. Тут знаходяться кнопки для запуску та паузи сцени. Також, за допомогою кнопок “Game” та “Scene”, ми можемо переключатися між двома вікнами. Вікно “Scene” відповідає за редагування елементів сцени. Тільки там ми можемо вибрати елемент, та відреагувати його, в вікні “Game” ми можемо побачити результат редагування, змінити розмір екрану, та протестувати, запустивши проект;
- останнє, це елементи редагування, які знаходяться над ієрархією сцени. Вони дають змогу дуже легко змінювати розміри, напрямок елементів.

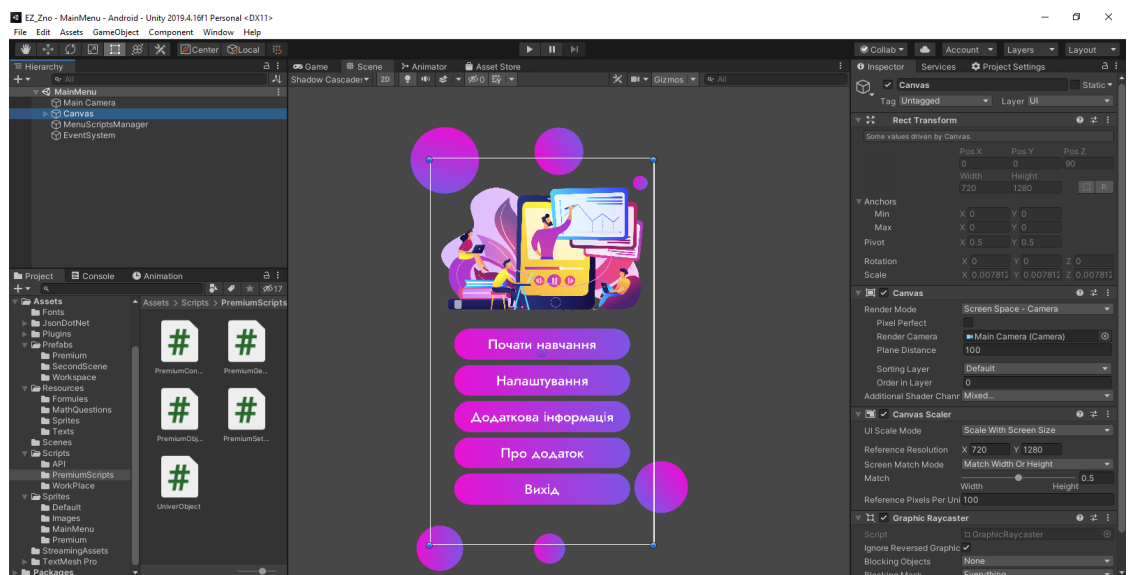


Рисунок 2.4 — Інтерфейс Unity

Основні функції та переваги Unity:

- дуже легкий редактор сцен по принципу (Drag-and-Drop). Тобто, для того, щоб додати будь-який новий елемент на сцену, потрібно просто вибрати його із готового списку і перетягнути на робоче місце;

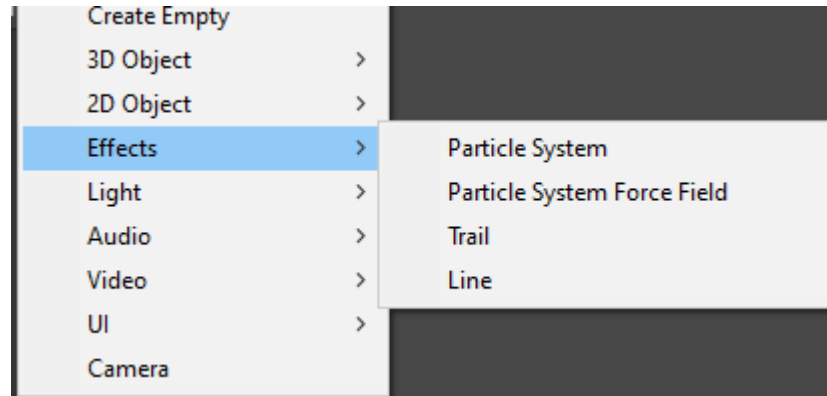


Рисунок 2.5 - Список елементів

- в Unity можна дуже легко створити нову сцену, та одразу почати працювати з нею. Саме за допомогою сцен відбувається навігація серед всіх розділів;
- в розділі “Game” можна вибрати потрібний розмір екрану, і одразу побачити, як приблизно буде виглядати додаток при такому розширенні;
- повне налаштування всіх компонентів. Всі елементи можна налаштувати під себе. Змінити їх параметри та зовнішній вигляд, додати скрити які будуть доповнювати або змінювати роботу цих елементів;

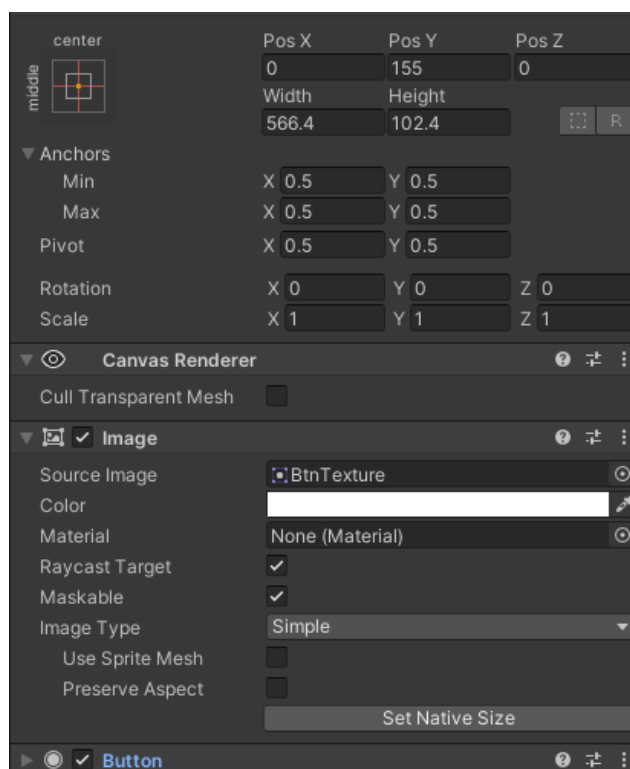
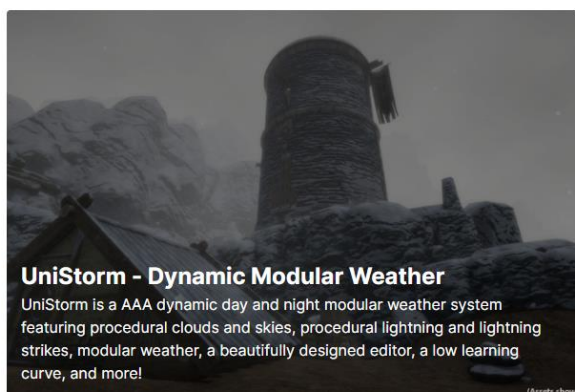


Рисунок 2.6 — Налаштування елемента “Button”

- величезний магазин асетів, в якому можна знайти як платні, так і безкоштовні плагіни, які значно полегшать процес роботи над проектом;

Home > Tools

Tools



Camera Controller
Feature rich, the Camera Controller supports off-center 3rd person views, advanced view obstruction detection, character fading, camera shakes, and lots more.

Refine by [clear filters](#)

All Categories

- 3D (37973) v
- 2D (8623) ^
- Characters (1573)
- Environments (1147)
- Fonts (35)
- GUI (2474) v
- Textures & Materials (3352) v
- Add-Ons (50) v
- Audio (6430) v
- Essentials (50) v
- Templates (2670) v
- Tools (9000) v

Рисунок 2.7 — Магазин асетів

- відображення всіх помилок у консолі, що дає змогу одразу виправити помилки. Також можна використовувати пошук по консолі, по ключовим словам. Використовуючи мову програмування C#, можна виводити в консоль не тільки помилки, але і слова для перевірки певних даних, використовуючи різні кольори для певних змінних;
- оскільки для свого проекту я використовував спрайти формату PNG, то для мене дуже корисна річ, це атласи із спрайтів. Вони дають змогу об'єднати декілька рисунків в один атлас, який буде набагато менше займати пам'яті і при цьому не втратить якість. Unity дуже просто працювати з такими видами елементів, тому це дає велику перевагу під час розробки;
- за допомогою Unity одразу можна розробляти проект на декілька платформ, додаючи тільки незначні зміни в програмний код;

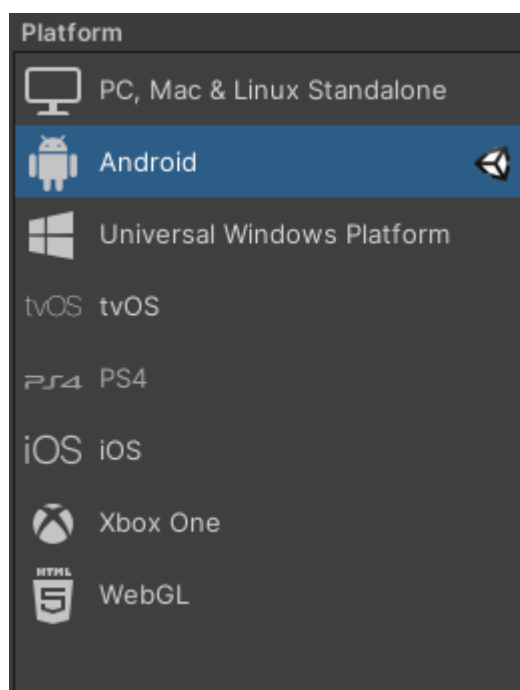


Рисунок 2.8 — Список платформ для розробки

- якщо включити “Device Simulator”, можна отримати можливість одразу в Unity протестувати, як буде виглядати додаток на

певному девайсі. На рис. 2.9 зображено інтерфейс “Device Simulator”, для мобільного телефону Xiaomi Mi 5. Також ми можемо побачити детальну інформацію про цей телефон, його операційну систему, процесор, графічний процесор, розмір екрану, та інші налаштування.

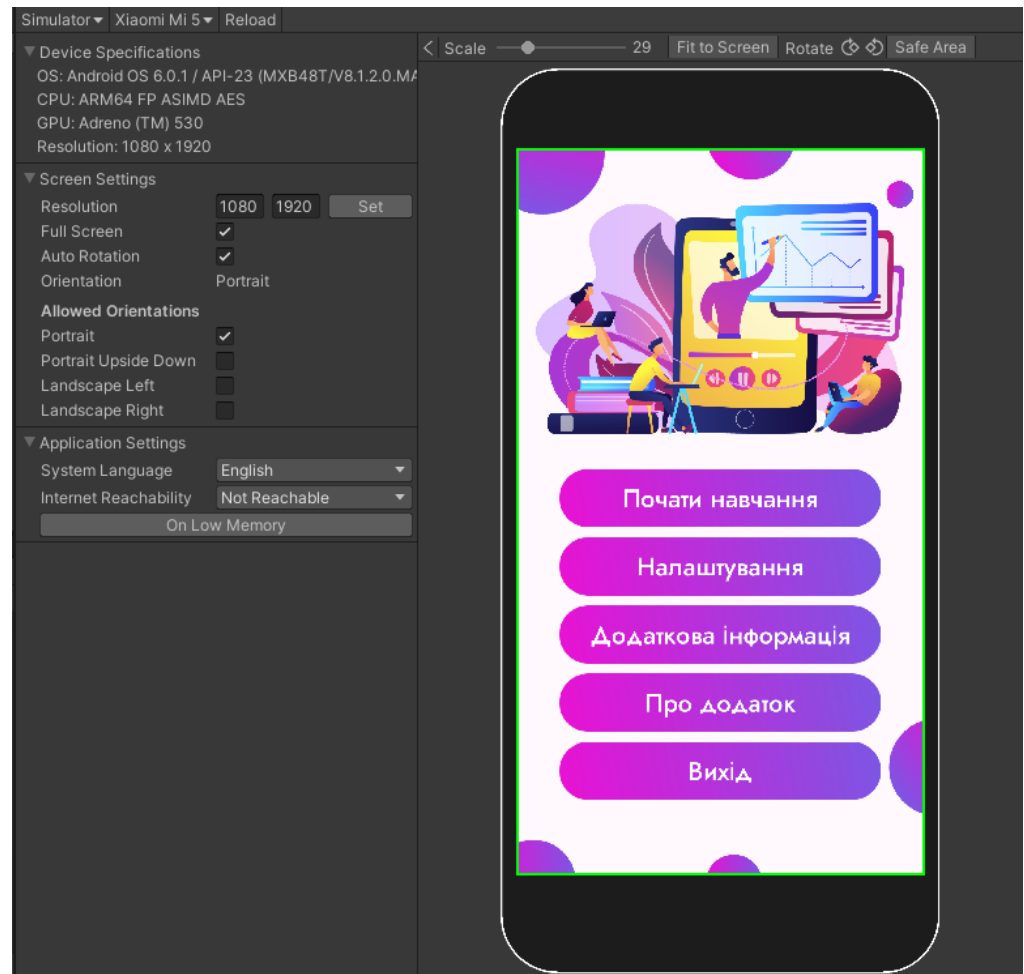


Рисунок 2.9 — Інтерфейс “Device Simulator”

Щоб отримати ще більше функцій, можна отримати доступ до Unity Services. Для цього, потрібно зареєструватися на офіційному сайті та увійти в свій аккаунт в Unity. Після цього в проект можна буде додати багато різноманітного функціоналу по відстеженню статистики користувачів, показ реклами від Unity, метрику та доступ до внутрішніх покупок. Також буде доступна можливість додати онлайн складову, тобто взаємодію між

користувачами та хмарні технології. Вся статистика буде відображатися на сайті у вигляді різноманітних графіків.

2.4 Вибір програми для створення дизайну для додатку

Перед початком основної розробки, потрібно розробити хоча б початковий дизайн та інтерфейс, з який ми будемо працювати на початку. В майбутньому його можна буде допрацювати з усіма правилами UI/UX дизайну.

UX – це User Experience, або досвід користувача, який він отримує від процесу користування додатком. Цей розділ відповідає за простоту дизайну, щоб він дуже легко запам'ятовувався людиною. Найважливіше, щоб після довгого періоду, протягом якого людина не користується додатком, він дуже швидко згадав весь інтерфейс і не мав в майбутньому проблем.

UI – це User Interface, або інтерфейс користувача, він відповідає за налаштування самого дизайну та інтерфейсу, як він буде виглядати, які елементи буде мати в собі, кольори, та наскільки зручно буде користуватися саме цим дизайном. Перед усім, це дослідження, після якого можливі зміни, якщо будуть помітні незручності для користувача.

Для розробки інтерфейсу, та дизайну в цілому існує дуже багато сайтів, та програм, але є декілька, які ми зараз розглянемо, між якими був вибір:

- Figma [10] - це безкоштовний веб-сайт, для створення дизайну та прототипування інтерфейсів мобільних додатків, сайтів та інших програмних продуктів. Все можна налаштувати під свій смак, змінити колір, висоту, ширину [1, с.30] та багато іншого. Для людей, які створюють веб-сайти, є велика перевага, Figma надає CSS код, що дуже пришвидшує верстку веб-сайтів. Також дуже просто реалізований імпорт всіх елементів, що дає змогу одразу після створення інтерфейсу, перенести його в Unity. Цей ресурс підтримує всі основні розширення, такі як: PNG, JPEG та SVG;

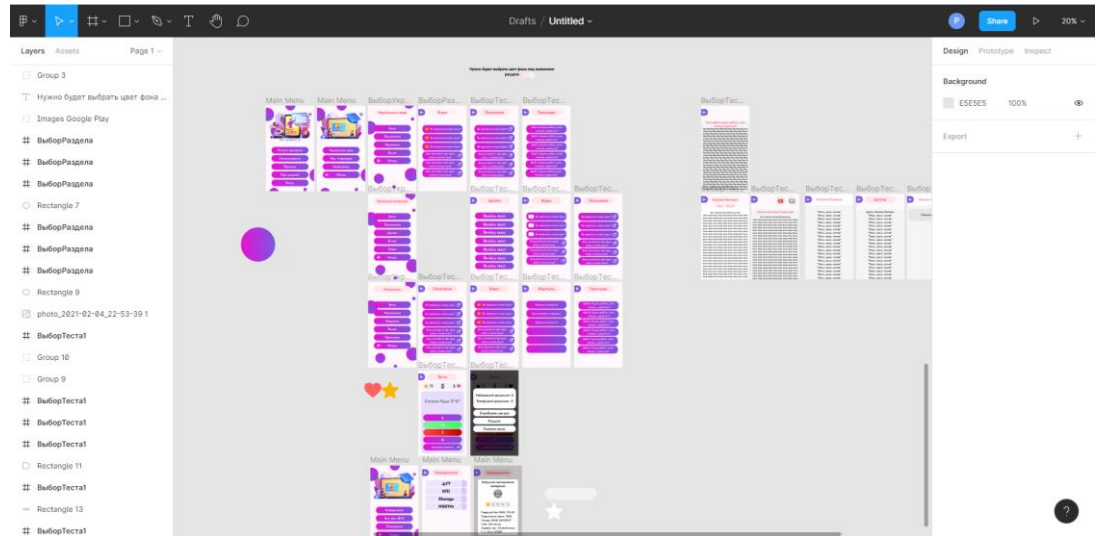


Рисунок 2.10 — Інтерфейс Figma

- Adobe XD – це програмний продукт, створений компанією Adobe. В основному ця програма використовується тільки для створення інтерфейсів. Як і Figma, ця програма підтримує векторну графіку. Можна дуже легко редагувати елементи, створювати маски та сітки. Підтримує багато плагінів, та має мобільний додаток, через який можна переглядати розроблений дизайн. Але є і мінуси, в тому що, це платний додаток. Він не підтримує експорт в CSS, тільки через додатковий плагін. Постійне оновлення є як плюсом, так і мінусом, оскільки бувають нестабільні патчі;

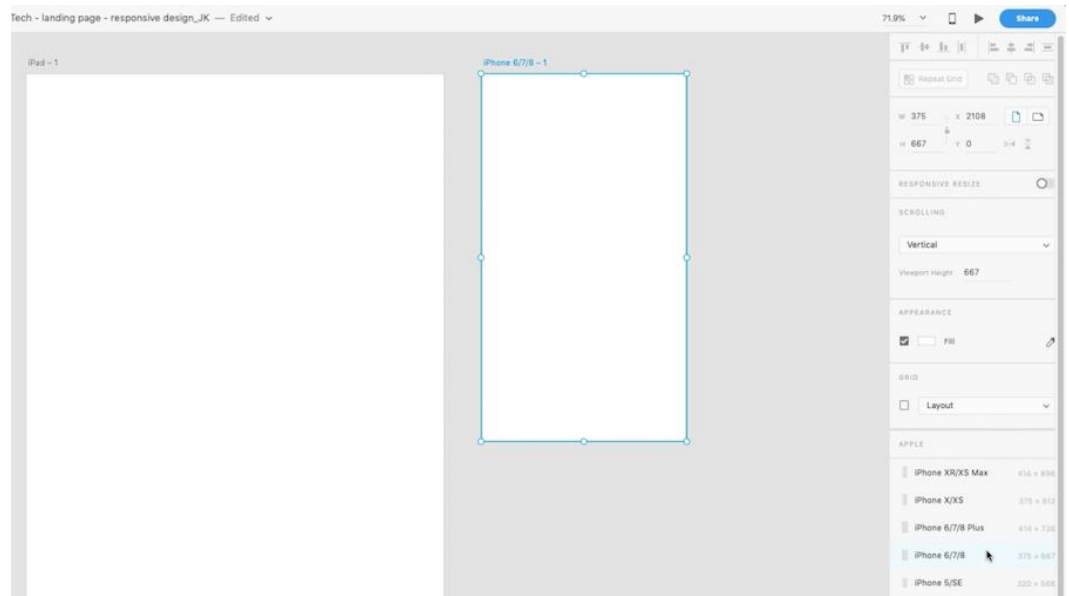


Рисунок 2.11 — Интерфейс Adobe XD

- Photoshop – це програма для багатофункціонального редагування малюнків, та графіки в цілому. Вона була також створена компанією Adobe. Великою перевагою, є те, що тут можна зробити абсолютно все, із зображенням. Тільки встановивши програму, ми отримуємо величезний арсенал функціоналу, який допоможе в розробці. Мінус даної програми, тільки в тому, що вона платна. Та складний для новачків інтерфейс.

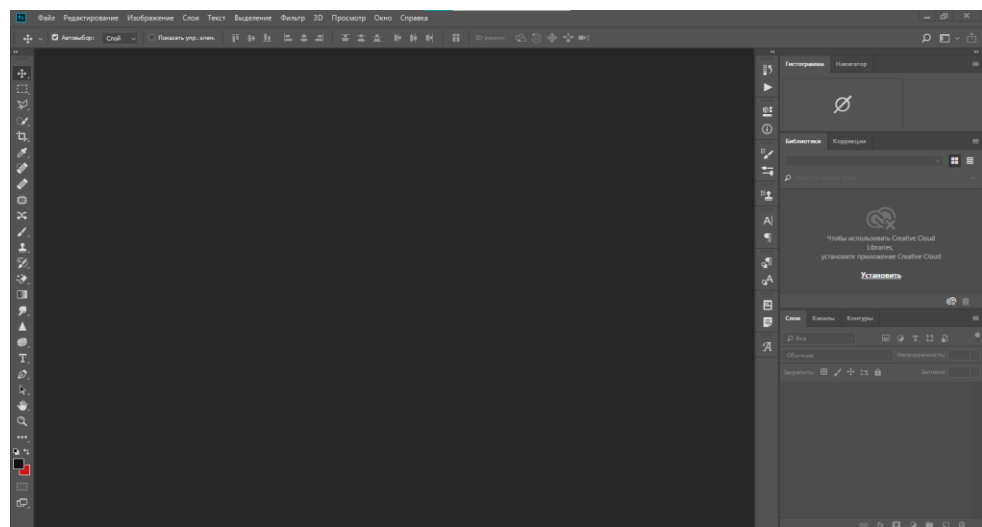


Рисунок 2.12 — Интерфейс Photoshop

Проаналізувавши всі ці додатки, було обрано Figma, оскільки її повністю достатньо для розробки мого простого інтерфейсу [2, с.46]. Ціна на Adobe XD занадто висока, а така велика кількість функціоналу, як в Photoshop просто не потрібна для такого проекту. Figma є простою в користуванні, та безкоштовною, ці фактори стали основними у виборі.

2.5 Контроль версій за допомогою Git та GitHub

Контроль версій за допомогою сторонніх програм, або сайтів дає дуже гарну можливість зберігати свої напрацювання без страху їх втратити. Якщо в майбутньому потрібно буде подивитися зміни, які були здійснені, або вже давно видалений код, це можна все зробити за допомогою контролю версій. Також такі системи дають можливість працювати над проектом одразу декільком працівникам, які потім зможуть об'єднати свої проекти в один, і завантажити це як єдине ціле. Дуже часто такі системи використовують як систему для зберігання проектів з відкритим кодом, що дає можливість іншим програмістам запозичити вже готовий код.

Git – це система для управління версіями проекту, також вона є розподіленою. Управляти нею можна через консоль, тобто готовими запитамми, які потрібно знати за заздалегідь. За допомогою її можна створити репозиторій, тобто сам початковий проект, в який ми будемо надалі зберігати нашу інформацію, та завантажувати наші проекти.

Github – це онлайн веб-сервіс для зберігання своїх проектів, як приватник, так і з відкритим кодом. Цей сервіс підтримує всі функції Git та розширює їх в декілька разів. Загрузивши свій проект, ми можемо побачити його на сайті. Дуже зручно переглядати код, який підсвічується, якщо він був зміненим, або редагованим після останнього завантаження. Якщо користувач робить коміт, можна залишити коментар, для інших розробників. Також можна створити тестову вітку, на яку в подальшому будуть завантажені не

стабільні версії додатку, а потім, вона буде об'єднана з основною, при відсутності критичних багів. Приклад коміту можна побачити на рис. 2.13.

```

17  app/build.gradle
@@ -8,7 +8,7 @@ android {
8      targetSdkVersion 28
9      versionCode 1
10     versionName "1.2"
11     -   testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
12     +   testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
13     }
14     buildTypes {
15         release {
@@ -21,14 +21,15 @@ android {
21     dependencies {
22         implementation fileTree(dir: 'libs', include: ['*.jar'])
23         //noinspection GradleCompatible
24     -   implementation 'com.android.support:appcompat-v7:28.0.0'
25     -   implementation 'com.android.support.constraint:constraint-layout:1.1.3'
26     -   implementation 'com.android.support:design:28.0.0'
27     +   implementation 'androidx.appcompat:appcompat:1.0.0'
28     +   implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
29     +   implementation 'com.google.android.material:material:1.0.0'
30     +   implementation 'com.rubik11:slidableactivity:2.0.6'
31     }
32     testImplementation 'junit:junit:4.12'
33     -   androidTestImplementation 'com.android.support.test:runner:1.0.2'
34     -   androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
35     -   implementation 'com.android.support:recyclerview-v7:28.0.0'
36     -   implementation 'com.android.support:cardview-v7:28.0.0'
37     +   androidTestImplementation 'androidx.test:runner:1.1.0'
38     +   androidTestImplementation 'androidx.test.espresso:espresso-core:3.1.0'
39     +   implementation 'androidx.recyclerview:recyclerview:1.0.0'
40     +   implementation 'androidx.cardview:cardview:1.0.0'
41     +   implementation 'com.google.android.gms:play-services-ads:18.2.0'
42     }

```

Рисунок 2.13 — Приклад коміту на Github

Також, для більшої зручності, можна встановити додаток Github Desktop, який працює на ПК, та завантажувати за допомогою його. Це також дуже зручно, оскільки можна створити репозиторій, і в ньому одразу проект, над котрим будете працювати. Програма одразу буде бачити всі зміни в додатку, і тільки натиснувши одну кнопку, можна буде завантажити новий коміт.

3. ПРОЕКТУВАННЯ ТА РОЗРОБКА ДОДАТКУ

3.1 Діаграма варіантів використання

Для того, щоб чітко розуміти, що може робити користувач в додатку, потрібно проаналізувати додаток та створити UML діаграму варіантів використання. Вона дає змогу побачити всі функції, якими може користуватися користувач. UML-діаграма зображена на рис. 3.1.

UML – це уніфікована мова моделювання, розроблена для опису об'єктного моделювання, в основному вона використовується в області програмного забезпечення, для моделювання бізнес-процесів, системного проектування та організації різноманітних структур.

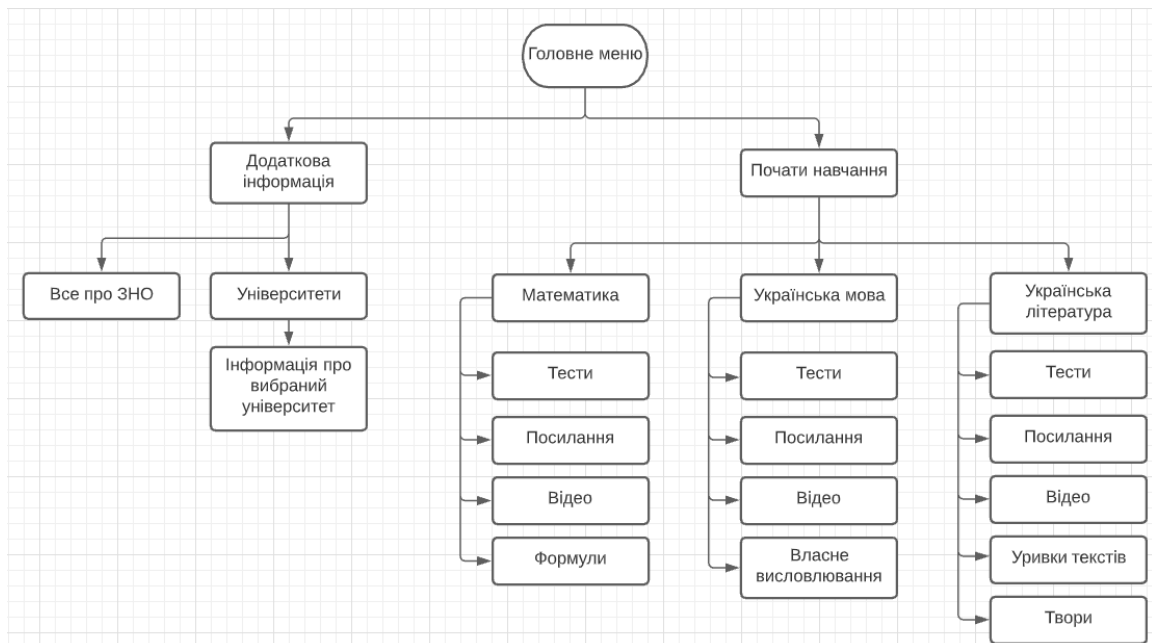


Рисунок 3.1 — UML-діаграма додатку

На малюнку 3.1, ми можемо побачити детальну реалізацію UML-діаграми для мого додатку. За допомогою її, ми можемо побачити основні розділи в додатку:

- додаткова інформація:

В цьому розділі знаходяться два підрозділи. Один з них, це “Все про ЗНО”, в ньому учень зможе знайти інформацію про процес ЗНО, коли потрібно подавати документи, як потрібно здавати тести, та правила проведення. Інший буде націлений на допомогу до вступу, він називається “Університети”. В цьому розділі знаходиться список із 50 університетів України, які були зібрані по певним критеріям [15]. Користувач зможе ознайомитися з основною інформацією, подивитися їх оцінку в інтернеті, кількість бюджетних місць, адресу, телефон та веб-сайт, та обрати для себе більш підходящий.

- почати навчання:

Цей розділ має в собі основні три підрозділи, в яких вже буде проходити підготовка до самих предметів, це: “Українська мова”, “Українська література” та “Математика”. В кожному із цих розділів є як унікальні та схожі методи навчання:

- тести;
- посилання;
- відео;
- формули;
- власне висловлювання;
- уривки творів;
- твори.

Всі ці розділи допомагають підготуватися до трьох предметів. Підрозділ з формулами є унікальним для розділу з математикою; власне висловлювання є унікальним для української мови; такі розділи як, уривки текстів та твори, це підрозділи, які зустрічаються тільки в розділі української літератури.

3.2 Діаграма діяльності додатку

Діаграма діяльності додатку, це графічне зображення, можливих шляхів розвитку використання додатку. Тут відображаються можливі вхідні та вихідні дані, та варіанти для певних ситуацій користувачів.

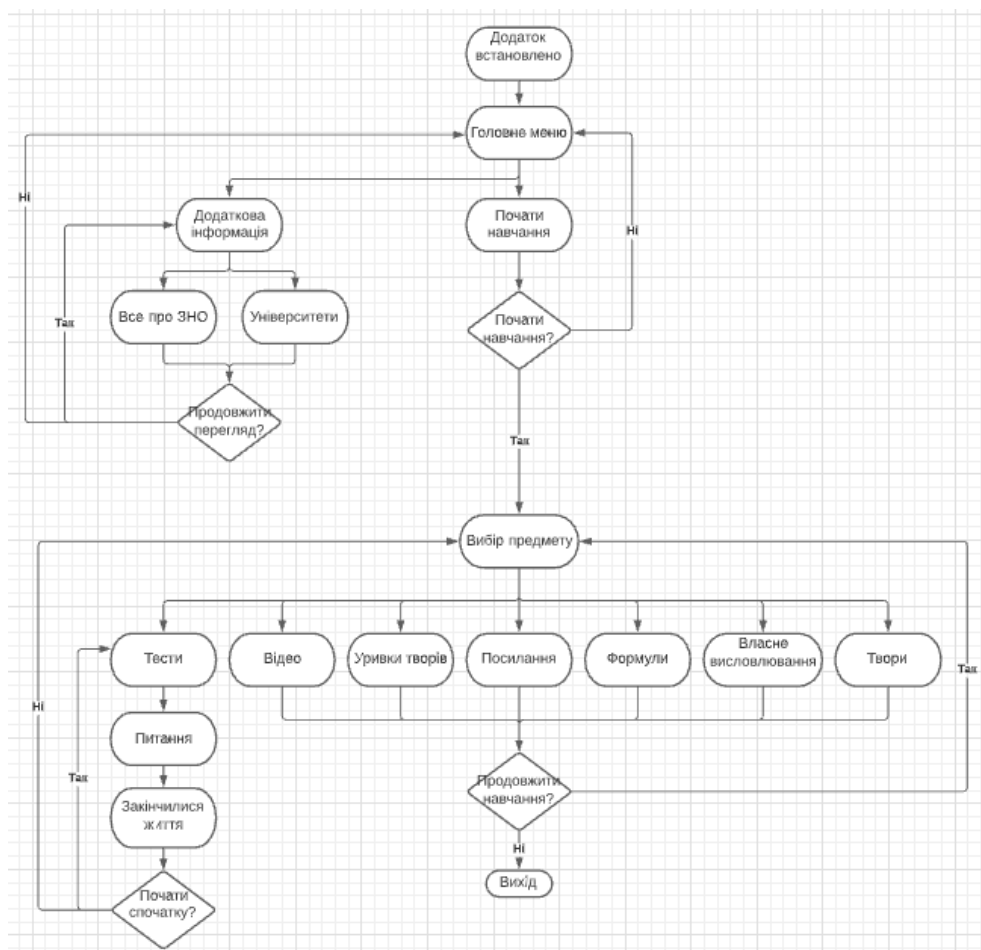


Рисунок 3.2 — Діаграма діяльності додатку

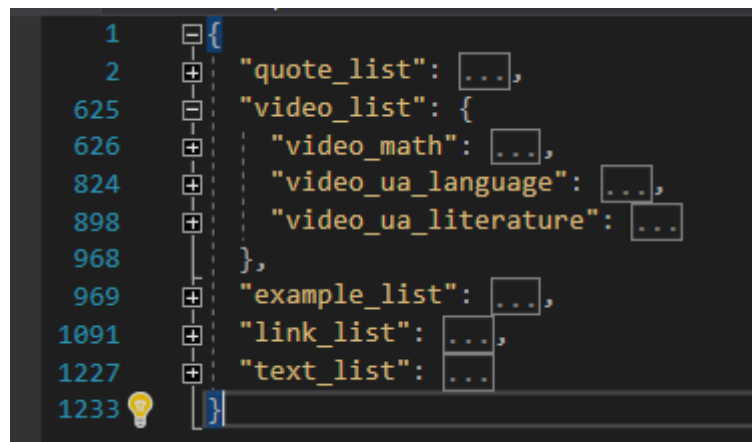
На діаграмі діяльності, яка зображена на рис. 3.2 прописані всі можливі варіанти користування додатком, від початку завантаження та входу, до кінцевого виходу із додатку.

3.3 Основні класи та функції додатку

3.3.1 Зчитування інформації

Для більш детального розуміння, як працює додаток, нижче будуть приведені основні класи та функції, які створюють основний функціонал програми. Основні компоненти інформацію у моєму додатку зберігаються у вигляді файлів розширення JSON.

JSON – це текстовий формат для зберігання та обміну інформацією, у файлі знаходиться тільки текст. Процес заповнення файлу такого виду називається серіалізацією [5, с.347], тобто перетворення послідовності байтів в текстову інформацію, а процес отримання інформації, називається десеріалізацією [5, с.347], тобто все навпаки. Приклад оформлення JSON файлу, зображений на рис. 3.3.



```

1      {
2      "quote_list": [...],
625    "video_list": {
626      "video_math": [...],
824      "video_ua_language": [...],
898      "video_ua_literature": [...]
968    },
969    "example_list": [...],
1091   "link_list": [...],
1227   "text_list": [...]
1233 }

```

Рисунок 3.3 — Оформлення JSON файлу

Для того, щоб зручно отримувати, та записувати інформацію в такий формат файлу, потрібно мати клас, який буде з ним працювати, тому для цього був створений клас `JsonDeserializer`. Цей клас має в собі чотири методи, кожний з яких десереалізує [5, с.347] отриманий файл по посиланню, а потім повертає об'єкт класу, в якому буде знаходитися інформація, з якою ми зможемо працювати в майбутньому. В цьому класі знаходяться такі методи, як:

- `GetDeserializedUniverObj` - метод повертає об'єкт класу `UniverObject`. Цей клас взаємодіє тільки з підрозділом “Університети”, за допомогою цього класу, ми отримуємо інформацію про університет із бази даних, а потім виводимо цю інформацію на екран;
- `GetDeserealizedRootObj` - метод повертає об'єкт класу `RootObjects`. Цей клас є одним із основних в додатку, оскільки зберігає в собі всю основну інформацію розділів, з якими взаємодіє користувач. В ньому знаходяться всі посилання на відео, додаткові ресурси, уривки творів та багато іншої інформації, яку користувач може побачити в додатку;
- `GetDeserializedTestObj` - метод повертає об'єкт класу `TestObject`. Цей клас повністю відповідає за розділ з тестами. В ньому ми можемо знайти всі запитання та відповіді, які будуть задані користувачу, якщо він захоче пройти ці тести.;
- `GetDeserializedCompObj` - метод повертає об'єкт класу `CompositionObject`. Цей клас зберігає в собі посилання на всі твори, та вірші які потрібно прочитати для вдалого складання ЗНО. Приклад оформлення методу зображений на рис. 3.4.

```

public CompositionObject GetDeserializedCompObj(string fileName)
{
    string json;

    if (Application.platform == RuntimePlatform.Android)
    {
        TextAsset file = Resources.Load(fileName) as TextAsset;
        json = file.ToString();
        return CompositionObject.CreateFromJSON(json);
    }
    else
    {
        string path;
        path = Path.Combine(Application.dataPath, "Resources/" + fileName + ".json");
        json = File.ReadAllText(path);

        return CompositionObject.CreateFromJSON(json);
    }
}

```

Рисунок 3.4 — Оформлення методу `GetDeserializedCompObj`

Як ми бачимо на рис. 3.4 у нас є перевірка платформи на якій ми зараз знаходимось, Android або будь-яка інша. Це робиться для того, щоб можна було тестувати додаток на комп'ютері, оскільки на телефоні посилання на файл дуже відрізняються і для них потрібний окремий виняток.

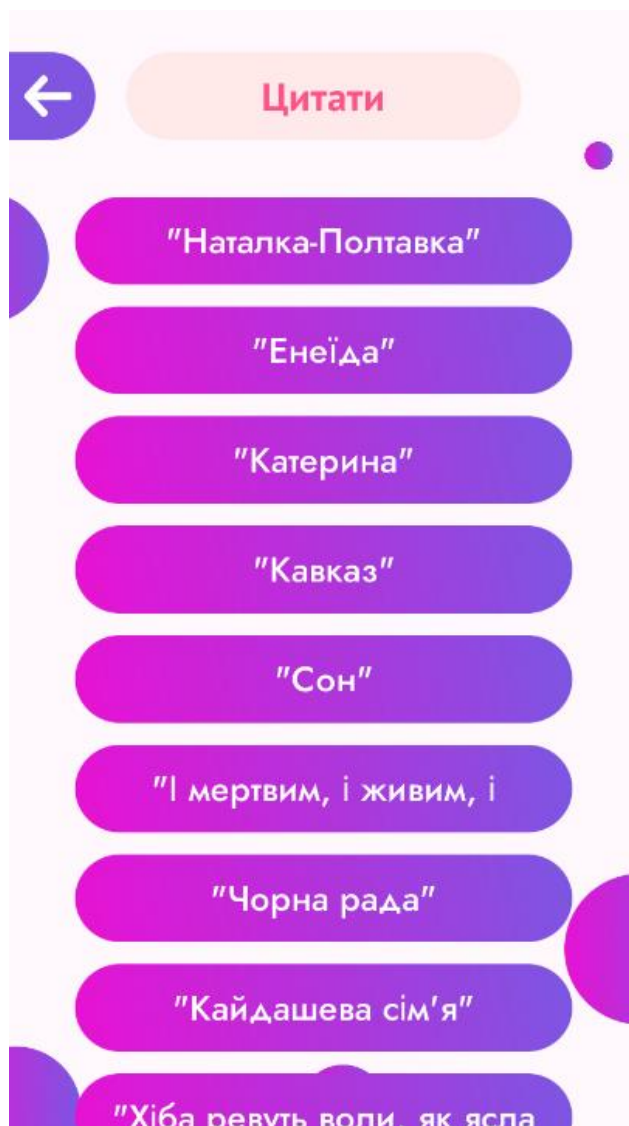


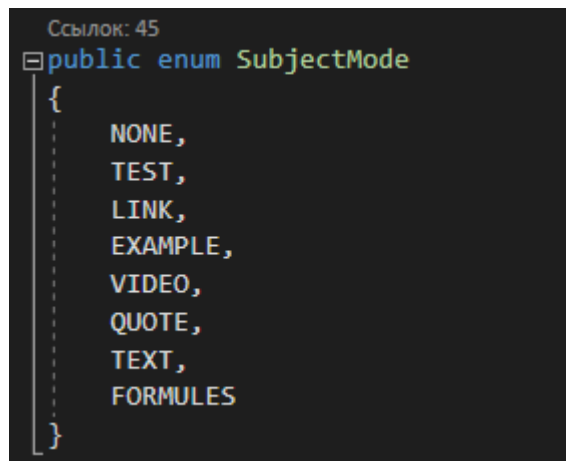
Рисунок 3.5 — Розділ з цитатами

Але принцип роботи частини коду для комп'ютера, не відрізняється. Ми отримуємо посилання на JSON файл, який потрібно прочитати, потім читаємо його і після цього створюємо об'єкт класу, та повертаємо його.

3.3.2 Клас для контролю типу навчання та предмета

Для контролю, який предмет, або тип навчання вибрав користувач, був створений клас `ComponentsController`, який мав в собі два ENUM класи, вони були названі `SubjectMode` та `Subject`. `SubjectMode` відповідав за зберігання типу навчання, а `Subject` за вид предмету відповідно.

ENUM – це тип перерахованих змінних, якщо просто, то це список змінних з певними назвами. Таким списком дуже легко користуватися, якщо щось має декілька режимів, і потрібно мати спосіб легко з ним працювати, в нашому випадку, це тип навчання, та предмети. В класі було створено дві змінні, цих типів, для зберігання інформації. Приклад оформлення ENUM класу можна побачити на рис. 3.6.



```
Ссылка: 45
public enum SubjectMode
{
    NONE,
    TEST,
    LINK,
    EXAMPLE,
    VIDEO,
    QUOTE,
    TEXT,
    FORMULES
}
```

Рисунок 3.6 — Оформлення ENUM класу

3.3.3 Навігація по додатку

Для навігації по всьому додатку, був створений клас `NavigationButtons`, який тримає в собі функції для навігації та переходом по сценам додатку. Список методів, які знаходяться в класі:

- `GoToWorkPlace` – метод, за допомогою якого, можна перейти на головну робочу сцену;
- `GoToTempScene` – метод, за допомогою якого, можна перейти на тимчасову сцену, на якій виводиться допоміжна інформація;

- GoToMainMenu – метод, який запускає сцену головного меню;
- GoToPickLesson – функція, яка дає змогу перейти на сцену вибору типу навчання та предмет;
- GoToUniversList – метод, за допомогою якого, ми переходимо в розділ зі списком університетів;
- GoToExample – метод, який переключає сцену, на сцену із прикладами власного висловлювання;
- ExitGame – метод, який виключає додаток;
- SetTestProperties – метод, який змінює налаштування змінної, яка зберігає параметри типу навчання;
- SetSubject – метод, який змінює налаштування змінної, яка зберігає параметри предмету навчання.

```

public class NavigationButtons : MonoBehaviour
{
    [SerializeField]
    private SubjectMode subjectMode;
    [SerializeField]
    private Subject subject;

    Ссылка: 0
    public void GoToWorkPlace() // Перехід на робочу сцену (WorkPlace)...

    Ссылка: 1
    public void GoToTempScene() // Перехід на тимчасову сцену, на якій виводиться допоміжна інформація (TempScene)...

    Ссылка: 0
    public void GoToMainMenu() // Перехід на сцену головного меню (MainMenuScene)...

    Ссылка: 0
    public void GoToPickLesson() // Перехід на сцену, для вибору предмета та типу навчання (PickLessonScene)...

    Ссылка: 0
    public void GoToUniversList() // Перехід на сцену із додатковою інформацією про ЗНО, та списком університетів (PremiumScene)...

    Ссылка: 1
    public void GoToExample() // Перехід на сцену, з прикладами власних висловлювань (ExampleScene)...

    Ссылка: 0
    public void ExitGame() // Вихід із додатку...

    Ссылка: 0
    public void SetTestsProperties() // Метод для вибору типу навчання...

    Ссылка: 0
    public void SetSubject() // Метод для вибору предмету навчання...
}

```

Рисунок 3.7 — Оформлення класу NavigationButtons

3.3.4 Генерація панелей

Оскільки основний елемент додатку, це ScrollView [4, с.132], тобто список, який можна гортати, при цьому переміщати інформацію, яка міститься в ньому, в гору або вниз. Для цього потрібно було створити клас,

який би міг корегувати такий `ScrollView` [4, с.132], для того, щоб він міг працювати з будь-якою кількістю панелей, які будуть знаходитися в ньому.

`FormsGenerator` став таким класом, він не тільки контролює роботу списку елементів, але і відповідає за створення панелей в ньому. В постійному циклі методу `FixedUpdated` проходить постійна перевірка і в залежності від результату ми корегуємо положення панелей. Для деяких розділів існує перевірка, для того, щоб положення панелей було трохи інакшим, оскільки розмір деяких панелей відрізняється.

В цьому ж класі, існує метод `InstantiatePanels()` в якому ми створюємо панель, та викликаємо метод із класу `PanelsInfo`, для того, щоб заповнити панель інформацією. Для того, щоб створити певну панель, було створено два масиви, які зберігають в собі вже заздалегідь створенні об'єкти, які нам потрібно створити, та заповнити інформацією. Після ініціалізації сцени, ми отримуємо інформацію, який предмет, та тип навчання вибрав користувач, в залежності від цього, ми вибираємо панелі з масиву. Отримавши індекс панелі, ми створюємо в `ScrollView` [4, с.132] цю панель. Цей скрипт був доданий на дочірній елемент `ScrollView` [4, с.132], а саме `Content`, тому саме в ньому будуть знаходитися всі створені елементи. Під час створення панелі, ми одразу додаємо її в новий масив `Blocks`, для можливого майбутнього редагування. Весь код методу `InstantiatePanels()`, можна побачити на рис. 3.8.

```

public void InstantiatePanels() // Метод, який створє панелі
{
    GameObject panel = null;

    if (ComponentsController.testInfo.Equals(Test.NO)) // Перевірка, з якого масиву потрібно брати готові панелі
        panel = PrefabsWorkPlaceArray[workPlaceIndex];
    else if (ComponentsController.testInfo.Equals(Test.YES))
        panel = PrefabsTempSceneArray[tempSceneIndex];

    float panelWidth = panel.GetComponent<RectTransform>().rect.width;
    for (int i = 0; i < BlocksCount; i++) // Цикл, який поступово створює панелі
    {
        if (ComponentsController.testInfo == Test.YES &&
            ComponentsController.subjectMode == SubjectMode.QUOTE)
        {
            RectTransform rect = panel.transform.GetComponent<RectTransform>();
            rect.sizeDelta = new Vector2(panelWidth, panelsInfo.CheckPanelQuoteSize(i)); // Змінюємо розмір панелі
        }
        if(ComponentsController.subjectMode == SubjectMode.FORMULES)
        {
            RectTransform rect = panel.transform.GetComponent<RectTransform>();
            rect.sizeDelta = panelsInfo.SetFormulesPanelInfo(panel, i);
        }

        Blocks[i] = Instantiate(panel, transform, false); // Створення панелі GameObject - Panel
        panelsInfo.SetPanelsInformation(Blocks[i], i); // Заповнюємо панель інформацією

        if (i == 0)
            continue;

        Blocks[i].transform.localPosition = new Vector2(Blocks[i].transform.localPosition.x,
            Blocks[i - 1].transform.localPosition.y - Blocks[i - 1].transform.GetComponent<RectTransform>().sizeDelta.y / 2
            - panel.GetComponent<RectTransform>().sizeDelta.y / 2 - panOffset);

        BlocksPositions[i] = -Blocks[i].transform.localPosition;
    }
}

```

Рисунок 3.8 — Метод InstantiatePanels()

3.3.5 Заповнення панелей інформацією

В Unity є дуже зручний елемент, який має назву Prefab [8, с.112]. Функціонал Unity дає змогу створити макет, або форму, яка буде мати всі потрібні нам поля для вводу, кнопки, текстові поля, та багато чого іншого. Кожний такий елемент буде мати однакові назви цих полів, що дає змогу дуже легко отримати до них доступ, через стандартні функції Mono. Кожний такий елемент є унікальним, та має свої налаштування, тобто, якщо ми будемо змінювати налаштування одного елемента, всі вони не будуть змінюватися, хоч і були створені на подібні одного макету. За допомогою такого способу були створені всі панелі, в кожному розділі. Вони генеруються, та поміщаються в Content, елементу ScrollView [4, с.132], потім можуть бути додатково оброблені різноманітними скриптами.

Для того, щоб не переповнювати основний клас створення панелей, та зробити код більш зрозумілий для читання, всі методи для заповнення інформацією були винесені в окремий клас під назвою PanelsInfo. В класі

FormsGenerator, ми створюємо екземпляр класу PanelsInfo, та передаємо йому екземпляр класу RootObjects, який створюємо за допомогою методу із класу JsonSerializer, для того, щоб отримати інформацію, з якою будемо працювати. В цьому класі створені змінні:

- subject – вона зберігає в собі предмет навчання;
- subjectMode – зберігає в собі предмет навчання;
- rootObjects – зберігаю майже всю інформацію, яку буде отримувати додаток;
- blocksCount – кількість панелей, які потрібно створити, та заповнити інформацією;
- mainIndex – головний індекс для головних масивів JSON файлу;
- preIndex – індекс, за допомогою якого, ми взаємодіємо з дочірніми елементами головних масивів;
- secondSceneIndex – індекс, для другого розділу уривків із творів.

Для перевірки всіх цих параметрів існує метод CheckTestsProperties(), в залежності від типу навчання та вибраного предмету він читає екземпляр класу RootObjects та заповнює всі змінні потрібною нам для роботи інформацією.

Для того, щоб заповнити панелі інформацією, був створений метод SetPanelsInformation(GameObject panel, int i), він отримує саму панель, та її індекс під час створення. В залежності від типу навчання, та предмету, цей метод викликає інші методи, які будуть заповнювати панель інформацією. Список всіх додаткових методів класу:

- SetQuotePanelInformation – цей метод налаштовує панелі в розділі уривків творів. Дає назву панелі, та дає можливість при натисканні на неї, відкривати нову сцену, з усіма уривками;
- SetExamplePanelInformation – цей метод налаштовує панелі, для прикладів власного висловлювання, а також запам'ятовує індекс

цієї панелі, за допомогою методу `PlayerPrefs`. Приклад оформлення методу зображено на рис. 3.9;

- `SetInfoTextAndUrl` – цей метод дає можливість при натиску на панель відкривати посилання в браузері;
- `SetTvirInformation` – цей метод змінює картинку панелі, та також дає можливість відкрити посилання в браузері після натиску на кнопку;
- `SetFormulesPanelInfo` – цей метод працює із картинками формул. Він змінює їх розміри, та повертає нове оброблене зображення;
- `GetBlocksCount` – метод, який повертає змінну `blocksCount`;
- `GetMainIndex` – метод, який повертає змінну `mainIndex`;
- `GetPreIndex` - метод, який повертає змінну `preIndex`;
- `GetSecondSceneIndex` - метод, який повертає змінну `secondSceneIndex`.

```
public void SetExamplePanelInformation(GameObject panel, string PanelText, int index)
{
    panel.transform.Find("ExampleText").GetComponentInChildren<TextMeshProUGUI>().text = PanelText;
    panel.transform.GetComponent<Button>().onClick.AddListener(delegate {
        new NavigationButtons().GoToExample();
        PlayerPrefs.SetInt("ExampleID", index);
    });
}
```

Рисунок 3.9 — Оформлення методу `SetExamplePanelInformation()`

Оскільки всі панелі однакові, та створені по одному макету [8, с.112], і лише інформація, якою ми заповнюємо їх, відрізняється, ми можемо використати методи `Find()` та `GetComponent()` [9, с.99]. За допомогою цього методу, ми можемо знайти потрібний нам компонент в панелі, та змінити його параметри.

3.3.6 Клас тестування

Для того, щоб проводити тестування користувачів, був створений клас Tests, який повністю контролював процес постановки питань, та отримання відповідей. За допомогою класу JsonSerializer, ми зчитуємо файл “tests” та отримуємо всі запитання і відповіді для трьох предметів. Для цього, в цьому класі були створені такі зміни:

- lives – контролює кількість спроб, коли кількість спроб дорівнює нулю, тестування закінчується і все потрібно почитати спочатку;
- buttons – це масив, який зберігає в собі кнопки, за допомогою яких відповідає користувач;
- score – кількість поточних балів;
- answerIndex – це індекс кнопки, яка має правильну відповідь;
- testObject – зберігає в собі екземпляр класу TestObject, який має в собі всі запитання та відповіді, я якими в подальшому буде працювати клас;

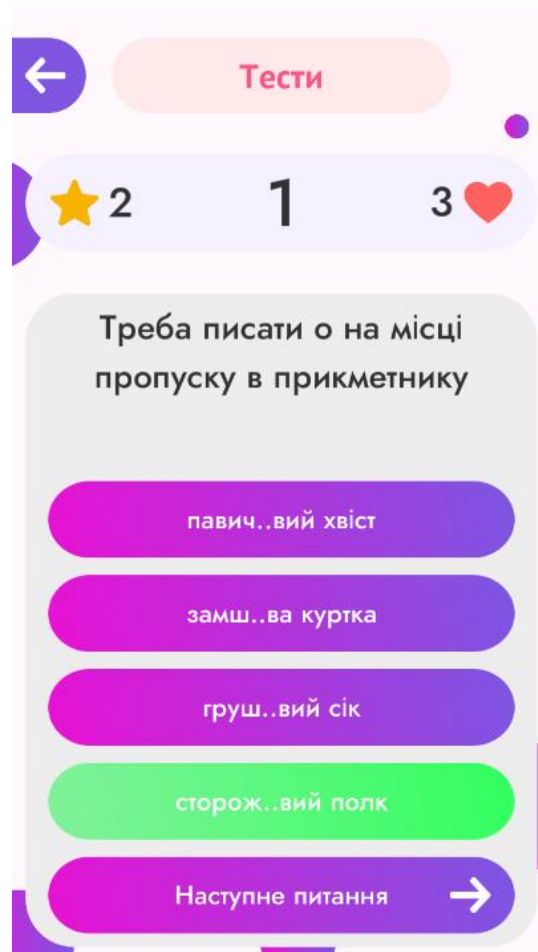


Рисунок 3.10 – Розділ тестування

На рис. 3.10 можна побачити приклад проходження тесту. Також треба зазначити, що всі запитання показуються випадковим способом, тобто кожний раз порядок запитань, буде відрізнятися. Вони зберігаються на телефоні, що дає змогу користуватися розділом будь-коли, навіть якщо немає доступу до інтернету.

3.4 Створення БД та взаємодія з нею

Для того, щоб створити розділ зі списком університетів, потрібно було десь зберігати всю інформацію, яка в подальшому буде виводитися на екран користувачу у вигляді списку. Для цього можна було знов використати файл JSON, в якому у вигляді масиву будуть зберігатися назви університетів, із додатковою інформацією про них. Але в подальшому, цей спосіб був

відкинутий, бо для того, щоб оновити інформацію про один університет потрібно було оновлювати весь додаток та створювати новий APK файл, що робило процес дуже довгим.

Для цього була створена БД, яка зберігалася на сервері, та доступ до якої можна було отримати за допомогою API. Якщо в цьому випадку, потрібно було додати, або видалити якесь поле, то це можна було швидко зробити в самій БД і не втручатися в сам додаток, та створювати новий APK.

API – це програмний інтерфейс додатку, або опис способів, методів, функцій, за допомогою яких, один програмний продукт, або додаток співпрацює з іншою.

За допомогою системи управління базою даних “phpmyadmin” була створена БД під назвою “zno_main” та таблиця в ній “Univers”. Ця таблиця має в собі такі поля:

- id – унікальний номер елемента, який не повторюється в таблиці;
- logo_name – скорочена назва університету на англійській мові, для отримання в майбутньому логотипу;
- short_name – скорочена назва, або ж аббревіатура університету, яка показується при генерації списку;
- full_name – повна назва університету;
- score – середня оцінка, яку поставили учні в інтернеті, на різноманітних форумах;
- middle_score – середня оцінка за останній рік, для вступу на бюджет;
- licenses_amount – кількість місць на бюджет;
- phone – телефон приймальної комісії;
- web_site – веб-сайт університету;
- adress – фізична адреса університету;

- geo – гео-мітка на гугл картах, це поле зроблене для того, щоб при натиску на фізичний адрес, відкривалися гугл карти з відкритим фізичним адресом.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно	Действие
<input type="checkbox"/>	1 id	int(11)			Нет	Нет	AUTO_INCREMENT	Изменить Удалить Первичный Уникальный Индекс Пространственный Ещё
<input type="checkbox"/>	2 logo_name	varchar(255)	utf8_general_ci		Нет	Нет		Изменить Удалить Первичный Уникальный Индекс Пространственный Ещё
<input type="checkbox"/>	3 short_name	varchar(255)	utf8_general_ci		Нет	Нет		Изменить Удалить Первичный Уникальный Индекс Пространственный Ещё
<input type="checkbox"/>	4 full_name	varchar(255)	utf8_general_ci		Нет	Нет		Изменить Удалить Первичный Уникальный Индекс Пространственный Ещё
<input type="checkbox"/>	5 score	varchar(255)	utf8_general_ci		Нет	Нет		Изменить Удалить Первичный Уникальный Индекс Пространственный Ещё
<input type="checkbox"/>	6 middle_score	varchar(255)	utf8_general_ci		Нет	Нет		Изменить Удалить Первичный Уникальный Индекс Пространственный Ещё
<input type="checkbox"/>	7 licenses_amount	varchar(255)	utf8_general_ci		Нет	Нет		Изменить Удалить Первичный Уникальный Индекс Пространственный Ещё
<input type="checkbox"/>	8 phone	varchar(255)	utf8_general_ci		Нет	Нет		Изменить Удалить Первичный Уникальный Индекс Пространственный Ещё
<input type="checkbox"/>	9 web_site	varchar(255)	utf8_general_ci		Нет	Нет		Изменить Удалить Первичный Уникальный Индекс Пространственный Ещё
<input type="checkbox"/>	10 adress	varchar(255)	utf8_general_ci		Нет	Нет		Изменить Удалить Первичный Уникальный Индекс Пространственный Ещё
<input type="checkbox"/>	11 geo	varchar(255)	utf8_general_ci		Нет	Нет		Изменить Удалить Первичный Уникальный Индекс Пространственный Ещё

Рисунок 3.11 — Оформлення БД

Для того, щоб отримати інформацію із створеної БД, потрібно було створити API, тобто функціонал, який дав би можливість працювати зі створеною базою даних. Для цього був спроектований клас API_Controller, в якому була створена стала змінна link, яка зберігає в собі посилання, за допомогою якого ми будемо отримувати список наших університетів, та метод GetUniversList(). Цей метод, за допомогою змінної link, робить запит в API, після вдалої відповіді, отримує список університетів у вигляді JSON.

В подальшому за допомогою класу PremiumGenerator створюється список університетів, а з методу класу JsonSerializer створюється екземпляр класу UniverObject та проходить заповнення інформацією новостворених панелей.

```

public class API_Controller : MonoBehaviour
{
    private string link = "https://ez-zno.xyz/universities/";

    ссылка: 1
    public IEnumerator GetUniversList(Action<string> action)
    {
        var request = UnityWebRequest.Get(link);
        request.downloadHandler = new DownloadHandlerBuffer();
        request.SetRequestHeader("Content-Type", "application/json");
        yield return request.SendWebRequest();
        if (request.isNetworkError || request.isHttpError)
        {
            Debug.Log(request.error);
        }
        else
        {
            action(request.downloadHandler.text);
        }
    }
}

```

Рисунок 3.12 — Клас API_Controller

Для того, щоб не зберігати велику кількість зображень логотипів, вони також були перенесені на сервер, це дало змогу звільнити багато пам'яті, та зменшити вагу програмного додатку. Для того, щоб отримати цей логотип, в базі даних існує змінна "logo_name", за допомогою його, та сталого посилання на папку з логотипами, ми робимо запит до створеного посилання, та отримуємо наш текстуру нашого логотипу, та створюємо спрайт. Після цього, ми змінюємо стандартне зображення в префабі [8, с.112], на зображення логотипу, яке ми завантажили.



Рисунок 3.13 — Приклад завантаження логотипу

Саме в методі `GetUniverLogo(GameObject univerPanel)`, ми отримуємо панель, в якій потрібно змінити логотип на завантажений, зробивши запит та створивши новий спрайт, для цього у телефона має бути доступ до інтернету. На рис. 3.13 ми можемо побачити результат завантаження логотипу, та присвоєння його до раніше створеної інформаційної панелі про університет.

ВИСНОВКИ

В результаті проведеного дослідження створений додаток для підготовки до ЗНО, на мові програмування C# та для ОС Android. Підсумовуючи всі аналізи та дослідження, які були проведені під час розробки можна отримати основні моменти:

- в наш час технології дуже швидко розвиваються і можна чітко прослідити тенденції змін. З кожним роком телефони все більше стають розповсюдженими, популярними та використовуються, як основний інструмент для комунікації, навчання та просто використання інтернету. Також це просліджується серед додатків для дистанційного навчання;
- враховуючи теперішню ситуацію в світі триває діджиталізація всього сектору навчання. З'являється все більше додатків для навчання. Проведені аналізи доводять, що розробка таких додатків, є дуже актуальними, а аналіз аналогів, показав, які функції є основними, для такого типу додатків;
- були описані програмні засоби, додатки та технології, якими потрібно володіти, для успішної розробки додатку на ОС Android. Визначено, що для розробки програмного продукту на Unity потрібно обов'язково знати мову програмування C# та знати основні моменти інтерфейсу, функціоналу Unity;
- в результаті розробки ПЗ було визначено, що Unity є досить легким для засвоєння, та має багато корисного функціоналу. Його перевага в великому ком'юніті та великій кількості можливостей налаштування внутрішніх елементів;
- встановлено, що мова програмування C# дуже добре підходить для розробки програмних продуктів для ОС Android, оскільки є дуже розповсюдженою, та має гарну екосистему в інтернеті.

Отже, в наш час дистанційне навчання стає все більш звичним і для того, щоб було комфортно навчатися та отримувати знання, потрібно мати гарну технічну базу із додатків, для розробки яких, можна використовувати Unity та мову програмування C#.

ЛІТЕРАТУРА

1. Eugene F. Designing in Figma: The complete guide to designing with reusable components and styles in Figma / Fedorenko Eugene., 2020. –164с.
2. Леві Д. UX-стратегія. Чого хочуть користувачі і як їм це дати. / Д. Леві., 2017. – 304 с.
3. Joseph H. Unity in Action, Second Edition, 2018. – 400 с.
4. Kishan T. Beginning Unity Android Game Development: From Beginner to Pro, 2020. – 270 с.
5. Прайс М. С# 8 и .NET Core. Розробка та оптимізація., 2021. – 816 с.
6. Патерни об'єктно-орієнтованого програмування / Е.Гамма, Р. Хелм, Р. Джонсон, Д. Вліссідес., 2020. – 448 с.
7. Dirk S. Getting Started with Visual Studio 2019: Learning and Implementing New Features / Strauss Dirk., 2019. – 283 с.
8. Thomas F. Learning Unity Android Game Development / Finnegan Thomas., 2015. – 338 с.
9. Alan T. Mastering Unity Scripting / Thorn Alan., 2015. – 380 с.
10. Figma [Електронний ресурс] – Режим доступу до ресурсу: <https://www.figma.com/>.
11. Hootsuite [Електронний ресурс] – Режим доступу до ресурсу: <https://www.hootsuite.com/>.
12. wearesocial [Електронний ресурс] – Режим доступу до ресурсу: <https://wearesocial.com/>.
13. osvita.ua [Електронний ресурс] – Режим доступу до ресурсу: <http://osvita.ua/consultations/>.
14. Український центр оцінювання якості освіти [Електронний ресурс] – Режим доступу до ресурсу: <https://testportal.gov.ua/>.
15. Освіта.ua [Електронний ресурс] – Режим доступу до ресурсу: <https://vstup.osvita.ua/>.

ДОДАТОК А

Повний код класу FormsGenerator

```
using UnityEngine;
using UnityEngine.UI;

public class FormsGenerator : MonoBehaviour
{
    private PanelsInfo panelsInfo;

    private int workPlaceIndex; // Першочерговий індекс json масива, який
    позначає індекс головного List'a , котрий зберігає в собі -> (quote_list,
    video_list, example_list, formule_list, link_list, text_list)

    private int tempSceneIndex; // Індекс, який позначає внутрішні елементи
    List'ов , котрі прописані угорі

    private Test testInfo;

    public GameObject[] PrefabsWorkPlaceArray = null; // Масив, який зберігає
    префаби для сцени WorkPlace

    public GameObject[] PrefabsTempSceneArray = null; // Масив, який зберігає
    префаби для сцени TempScene

    public GameObject contentExampleScrollView = null;

    private int BlocksCount;
        [Range(1, 500)]
    public int panOffset;
        [Range(0f, 20f)]
    public float snapSpeed;

    private int selectedPanID;
    private bool isScrolling = false;
```

```

private RectTransform contentRect;
private Vector2 contentVector;
public ScrollRect scrollRect;

private GameObject[] Blocks;
private Vector2[] BlocksPositions;

public void Start()
{
testInfo = ComponentsController.testInfo;
contentRect = GetComponent<RectTransform>();

string fileName;
if (ComponentsController.subjectMode.Equals(SubjectMode.TEXT))
{
contentRect.anchorMin = new Vector2(.5f, .72f);
contentRect.anchorMax = new Vector2(.5f, .72f);
}
else if
(ComponentsController.subjectMode.Equals(SubjectMode.EXAMPLE))
{
contentRect.anchorMin = new Vector2(.5f, .4f);
contentRect.anchorMax = new Vector2(.5f, .4f);
}

fileName = "publications";

RootObjects rootObjects = new
JsonDeserializer().GetDeserializedRootObj(fileName); // Десеріалізуємо об'єкт
та записуємо його в змінну rootObjects

panelsInfo = new PanelsInfo(rootObjects);

```

```

BlocksCount = panelsInfo.GetBlocksCount();
workPlaceIndex = panelsInfo.GetMainIndex();
tempSceneIndex = panelsInfo.GetPreIndex();

BlocksPositions = new Vector2[BlocksCount];
Blocks = new GameObject[BlocksCount];

InstantiatePanels();
}

private void FixedUpdate()
{
    if (contentRect.anchoredPosition.y >= BlocksPositions[0].y && !isScrolling ||
contentRect.anchoredPosition.x <= BlocksPositions[BlocksPositions.Length - 1].y
&& !isScrolling)
        scrollRect.inertia = false;

    float nearestPos = float.MaxValue;
    for (int i = 0; i < BlocksCount; i++)
    {
        float distance = Mathf.Abs(contentRect.anchoredPosition.y -
BlocksPositions[i].y);
        if (distance < nearestPos)
        {
            nearestPos = distance;
        }
    }

    if (testInfo.Equals(Test.YES) ||
        ComponentsController.subjectMode == SubjectMode.FORMULES ||

```

```

        ComponentsController.subjectMode == SubjectMode.TEXT)
    {
        if (i > BlocksCount - 2)
            selectedPanID = i - 1;
        else
            selectedPanID = i;
    }
    else if (testInfo.Equals(Test.NO))
    {
        if (i <= 2 && BlocksCount > 4) // Фокусує екран на одній панелі,
        котру обираємо за допомогою змінної selectedPanID
            selectedPanID = i + 2;
        else if (i >= BlocksCount - 2 && BlocksCount > 4)
            selectedPanID = i - 2;
        else
            selectedPanID = i;
    }
}
}
}

```

```
float scrollVelocity = Mathf.Abs(scrollRect.velocity.y);
```

```
if (scrollVelocity < 2000 && !isScrolling) scrollRect.inertia = false;
```

```
if (isScrolling || scrollVelocity > 400) return;
```

```
contentVector.y = Mathf.SmoothStep(contentRect.anchoredPosition.y,
BlocksPositions[selectedPanID].y, snapSpeed * Time.fixedDeltaTime);
```

```
contentRect.anchoredPosition = contentVector;
```

```
}
```

```
public void Scrolling(bool scroll)
```

```
{
```

```

    isScrolling = scroll;
    if (scroll) scrollRect.inertia = true;
}

public void InstantiatePanels() // Метод, який створює панелі
{
    GameObject panel = null;

    if (ComponentsController.testInfo.Equals(Test.NO)) // Перевірка, з якого
масиву потрібно брати готові панелі
        panel = PrefabsWorkPlaceArray[workPlaceIndex];
    else if (ComponentsController.testInfo.Equals(Test.YES))
        panel = PrefabsTempSceneArray[tempSceneIndex];

    float panelWidth = panel.GetComponent<RectTransform>().rect.width;
    for (int i = 0; i < BlocksCount; i++) // Цикл, який поступово створює
панелі
    {
        if (ComponentsController.testInfo == Test.YES &&
            ComponentsController.subjectMode == SubjectMode.QUOTE)
        {
            RectTransform rect =
panel.transform.GetComponent<RectTransform>();
            rect.sizeDelta = new Vector2(panelWidth,
panelsInfo.CheckPanelQuoteSize(i)); // Змінюємо розмір панелі
        }
        if(ComponentsController.subjectMode == SubjectMode.FORMULES)
        {
            RectTransform rect =
panel.transform.GetComponent<RectTransform>();
            rect.sizeDelta = panelsInfo.SetFormulesPanelInfo(panel, i);
        }
    }
}

```

```
        Blocks[i] = Instantiate(panel, transform, false); // Створення панелі
GameObject - Panel

        panelsInfo.SetPanelsInformation(Blocks[i], i); // Заповнюємо панель
інформацією

        if (i == 0)
            continue;

        Blocks[i].transform.localPosition = new
Vector2(Blocks[i].transform.localPosition.x,
        Blocks[i - 1].transform.localPosition.y - Blocks[i -
1].transform.GetComponent<RectTransform>().sizeDelta.y / 2
        - panel.GetComponent<RectTransform>().sizeDelta.y / 2 - panOffset);

        BlocksPositions[i] = -Blocks[i].transform.localPosition;
    }
}
}
```


ДОДАТОК Б



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ ДЛЯ ПІДГОТОВКИ ДО ЗНО НА МОВІ ПРОГРАМУВАННЯ C#

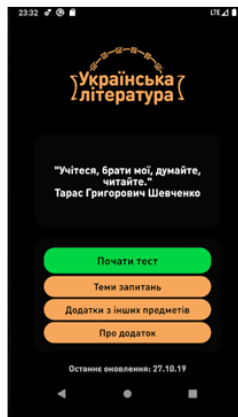
Виконав студент 4 курсу
групи ПД-42
Гришаєнко Владислав Ігорович
Керівник роботи
Жебка В.В

Київ – 2021

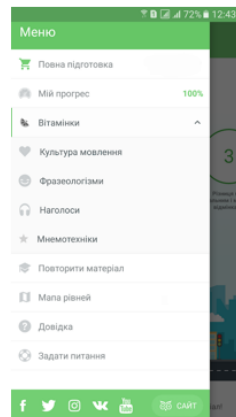
МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

- **Мета роботи** – розробка програмного забезпечення для підготовки до ЗНО на мові програмування C#.
- **Об'єкт дослідження** – підвищення знань в одному із трьох доступних предметів.
- **Предмет дослідження** – додаток для підготовки, та перевірки знань до ЗНО.

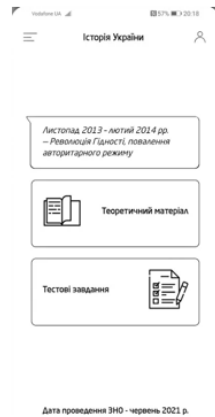
АНАЛОГИ



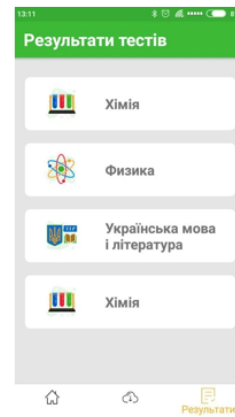
ЗНО 2020 тести:
Українська література



ЗНО 2021 на 190+.
Мова і література



ЗНО 2021.
Історія України



ZNO



Складу ЗНО

3

АНАЛОГИ

Назва додатку	Переваги	Недоліки
ЗНО 2020 тести: <u>Українська література</u>	<ul style="list-style-type: none"> - Гарна система тестів - Велика кількість матеріалу 	<ul style="list-style-type: none"> - Доступний тільки один предмет
ЗНО 2021 на 190+. <u>Мова і література</u>	<ul style="list-style-type: none"> - Різноманітність матеріалу - Стильний дизайн 	<ul style="list-style-type: none"> - Платна підписка - Доступний тільки один предмет
ЗНО 2021. <u>Історія України</u>	<ul style="list-style-type: none"> - Гарна теоретична інформація 	<ul style="list-style-type: none"> - Доступний тільки один предмет
ZNO	<ul style="list-style-type: none"> - Декілька предметів для вивчення 	<ul style="list-style-type: none"> - Проблеми з оптимізацією - Бувають підвисання додатку
Складу ЗНО	<ul style="list-style-type: none"> - Великий список предметів 	<ul style="list-style-type: none"> - Іноді не завантажуються тести

4

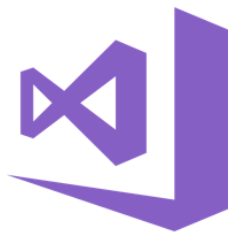
ТЕХНІЧНІ ЗАВДАННЯ

- *проаналізувати процес проходження ЗНО;*
- *проаналізувати додатки для підготовки до ЗНО;*
- *дослідити програмні продукти, для розробки додатків для ОС Android;*
- *спроектувати та розробити мобільний додаток.*

5

ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ

- *Unity;*
- *Figma;*
- *Visual Studio;*
- *[phpmyadmin](#).*



6

МЕТОДИ ТА КЛАСИ ПРОГРАМИ

Основні реалізовані класи у додатку:

- [API_Controller](#) – клас для роботи з API;
- [JsonDeserializer](#) – клас для десеріалізації файлу, та отримання інформації с нього;
- [PanelsInfo](#) – клас для заповнення панелей інформації;
- [NavigationButtons](#) – клас для навігації по додатку;
- [FormsGenerator](#) – клас для генерації панелей, та їх контролю;
- [PremiumGenerator](#) – клас для генерації списку університетів.
- [Tests](#) – клас, який відповідає за тестування.

7

База даних

Структура БД в [rhrmuadmin](#):

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно	Действие
1	id	int(11)			Нет	Нет	AUTO_INCREMENT	Изменить Удалить Первичный Уникальный Индекс Пространственный Ещё
2	logo_name	varchar(255)	utf8_general_ci		Нет	Нет		Изменить Удалить Первичный Уникальный Индекс Пространственный Ещё
3	short_name	varchar(255)	utf8_general_ci		Нет	Нет		Изменить Удалить Первичный Уникальный Индекс Пространственный Ещё
4	full_name	varchar(255)	utf8_general_ci		Нет	Нет		Изменить Удалить Первичный Уникальный Индекс Пространственный Ещё
5	score	varchar(255)	utf8_general_ci		Нет	Нет		Изменить Удалить Первичный Уникальный Индекс Пространственный Ещё
6	middle_score	varchar(255)	utf8_general_ci		Нет	Нет		Изменить Удалить Первичный Уникальный Индекс Пространственный Ещё
7	licenses_amount	varchar(255)	utf8_general_ci		Нет	Нет		Изменить Удалить Первичный Уникальный Индекс Пространственный Ещё
8	phone	varchar(255)	utf8_general_ci		Нет	Нет		Изменить Удалить Первичный Уникальный Индекс Пространственный Ещё
9	web_site	varchar(255)	utf8_general_ci		Нет	Нет		Изменить Удалить Первичный Уникальный Индекс Пространственный Ещё
10	adress	varchar(255)	utf8_general_ci		Нет	Нет		Изменить Удалить Первичный Уникальный Индекс Пространственный Ещё
11	geo	varchar(255)	utf8_general_ci		Нет	Нет		Изменить Удалить Первичный Уникальный Индекс Пространственный Ещё

Для взаємодії з нею створений клас [API_Controller](#), та метод [GetUniversList\(\)](#).

8

Розгляд готового додатку

Приклад роботи класу JsonDeserializer можна побачити на рис. 1. Цей клас серіалізує дані, та створює екземпляр класу. Після цього клас FormsGenerator створює список цитат за допомогою створеного екземпляра класу.

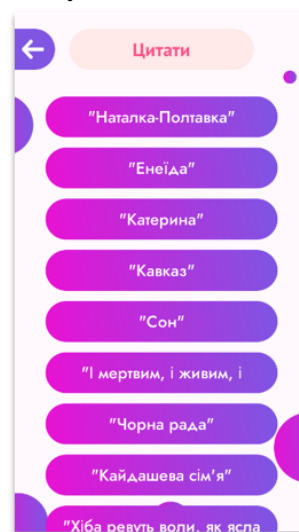


Рисунок 1 – Розділ з цитатами

9

На рис. 2 можна побачити реалізацію списку університетів. За допомогою класу JsonDeserializer та FormsGenerator створюється сам список університетів. За допомогою класу API_Controller отримуємо інформацію з бази даних та в подальшому заповнюємо нею панельі.



Рисунок 2 – Список університетів

10

Розділ тестування зображений на рис. 3. Для цього розділу був створений окремий клас Tests, який контролює запитання та відповіді. За допомогою класу JsonDeserialzer ми отримуємо екземпляр класу з всією інформацією, яка потрібна для тестування. У користувача є 3 можливості відповісти неправильно, якщо він витратить всі спроби, накопичування очок та тестування, почнеться спочатку.

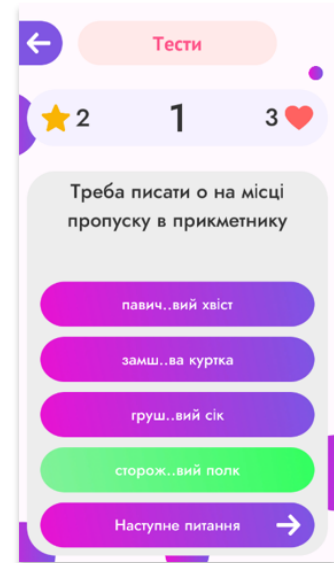


Рисунок 3 - Тестування

11

АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

- Гришаєнко В.І. Особливості дистанційного навчання в 2020 році // Сучасні інфокомунікаційні технології: Матеріали XI НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ СТУДЕНТІВ ТА МОЛОДИХ ВЧЕНИХ. Збірник тез. 11.12.2020, К.ДУТ, 2020 – 91с.
- Гришаєнко В.І. Дослідження мобільних додатків для підготовки до ЗНО // Застосування програмного забезпечення в інфокомунікаційних технологіях: Матеріали ВСЕУКРАЇНСЬКОЇ НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ ЗАСТОСУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ В ІНФОКОМУНІКАЦІЙНИХ ТЕХНОЛОГІЯХ. Збірник тез. 12.02.2021, К.: ДУТ, 2021 – 129с.

12

ВИСНОВКИ

1. Все більше людей вибирають телефон, як основний інструмент для навчання та комунікації.
2. З кожним роком дистанційне навчання стає все більш розповсюдженим та повсякденним.
3. Були проаналізовані аналоги, та визначені переваги, та недоліки.
4. Було визначено, що мова програмування C# підходить для розробки додатку на ОС Android.
5. Був розроблений додаток на мові програмування C# та за допомогою програми Unity.

Зараз на фоні карантину, та дистанційного навчання, такі додатки мають великий шанс стати успішними. Потрібно лише мати великий список предметів для підготовки, та гарний матеріал для навчання.