

Галицький фаховий коледж імені В'ячеслава Чорновола  
відділення комп'ютерних технологій  
циклова комісія інформатики та комп'ютерних дисциплін

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач відділення

комп'ютерних технологій

Наталія СТЕФУРАК / \_\_\_\_\_ /

підпис

« \_\_\_\_ » \_\_\_\_\_ 2023 р.

## ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проєкту

освітньо-кваліфікаційного рівня «молодший спеціаліст»

зі спеціальності 122 «Комп'ютерні науки»

на тему: «Ігровий мобільний застосунок «CardMatch» з використанням  
методологію розробки RUP»

Студент групи КН-41

Зьомко В.А.

\_\_\_\_\_  
(підпис)

Керівник проєкту

Кузик В. М.

\_\_\_\_\_  
(підпис)

Консультанти:

з техніко-економічного  
обґрунтування

Меленчук Л.І.

\_\_\_\_\_  
(підпис)

нормоконтролер

Кульчинська Н.З.

\_\_\_\_\_  
(підпис)

Тернопіль - 2023

Галицький фаховий коледж імені В'ячеслава Чорновола  
відділення комп'ютерних технологій  
циклова комісія інформатики та комп'ютерних дисциплін

ЗАТВЕРДЖУЮ

Завідувач відділення

комп'ютерних технологій

Наталія СТЕФУРАК/\_\_\_\_\_/

підпис

«\_\_» \_\_\_\_\_ 2022 р.

## ЗАВДАННЯ

на дипломне проєктування  
на здобуття освітньо-кваліфікаційного рівня «молодший спеціаліст»  
студенту Зьомко Володимиру Андрійовичу  
(прізвище, ім'я та по-батькові студента)

1. Тема проєкту Ігровий мобільний застосунок «CardMatch» з використанням методології розробки RUP

затверджена наказом по коледжу від “16” грудня 2022р., № 1193-н

2. Термін здачі студентом завершеного проєкту “\_\_” \_\_\_\_\_ 2023 р.

3. Вихідні дані до проєкту \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

4. Перелік питань, які повинні бути розроблені в проєкті:

а) основна частина \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

б) техніко-економічне обґрунтування \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

5. Перелік графічного матеріалу \_\_\_\_\_

6. Консультанти проєкту: \_\_\_\_\_

Розділ	Консультанти	Підпис, дата	
		Завдання видано	Завдання прийнято
з техніко- економічного обґрунтування	<div>(вчена ступень, звання П.І.Б.</div> <div>консультанта)</div>		

**КАЛЕНДАРНИЙ ПЛАН**  
дипломного проєктування

№ п/п	Найменування етапу	Терміни	
		початку	завершення
1.	Вибір теми, ознайомлення з вимогами до дипломного проєктування	18.11.22 р.	16.12.22 р.
2.	Огляд типових рішень та написання відповідного розділу ПЗ	16.12.22 р.	06.02.23 р.
3.	Дослідження технологій реалізації та написання відповідного розділу ПЗ	26.01.23 р.	14.02.23 р.
4.	Розробка функціональних вимог до проєкту та робота над структурою програмного продукту. Написання відповідного розділу ПЗ.	14.02.23 р.	01.03.23 р.
5.	Встановлення та налаштування середовища реалізації та написання відповідного розділу ПЗ.	01.03.23 р.	15.03.23 р.
6.	Проектування програмного засобу (функціоналу, інтерфейсу, бази даних продукту) та написання відповідного розділу ПЗ	15.03.23 р.	18.04.23 р.
7.	Реалізація та налаштування програмного засобу та написання відповідного розділу ПЗ	18.04.23 р.	03.05.23 р.
8.	Доопрацювання модулів	03.05.23 р.	17.05.23 р.
9.	Тестування та налагодження програмного продукту та написання відповідного розділу ПЗ	17.05.23 р.	31.06.23 р.
10.	Опрацювання економічного розділу дипломного проєкту та оформлення спеціального розділу	20.05.23 р.	05.06.23 р.
11.	Робота над оформленням пояснювальної записки	05.06.23 р.	14.06.23 р.
12.	Оформлення пояснювальної записки	13.06.23 р.	14.06.23 р.
13.	Попередній захист дипломного проєкту	15.06.23 р.	16.06.23 р.
14.	Підготовка до захисту дипломного проєкту	17.06.23 р.	24.06.23 р.
15.	Захист дипломного проєкту	25.06.23 р.	26.06.23 р.

7. Дата видачі завдання “ ” \_\_\_\_\_ 2022 р.

Керівник \_\_\_\_\_ /

Завдання прийняв до виконання \_\_\_\_\_ /

## Реферат

Дипломний проєкт. «Ігровий мобільний застосунок «CardMatch» з використанням методології розробки RUP». 66 сторінок, 20 рисунків, 3 таблиць, 12 джерел.

Перед початком створення проєкту потрібно провести пошук та аналіз аналогічних мобільних ігрових додатків. Це дасть потрібну базу розуміння, як буде виглядати майбутній додаток та якою функціональністю потрібно його наділити. Для цього взято наступні мобільні додатки, які можна знайти в публічному доступі на маркеті PlayMarket:

- Matching Card;
- Matching King.

Мета проєкту – створення ігрового мобільного додатка для користувачів гаджетів з операційною системою Android.

Завдання проєкту – це проведення усіх етапів від проектування до тестування та надання повноцінної копії додатка, за обраною методологією проектування та розробки RUP.

Кінцевим результатом проєкту буде повністю функціональний ігровий додаток, що дасть користувачам можливість грати в знаходження пар однакових карточок. Також можливість внутрішньо ігрового вибору тематики ігрового процесу та його складності.

Для розробки визначеного вимогами додатка буде використовуватись середовище розробки Android Studio. Весь графічний інтерфейс для користувача буде спроектований та побудований згідно кращих практик UI/UX дизайну. Для реалізації програмного коду функціональності буде використано мову програмування Java.

МОБІЛЬНИЙ ДОДАТОК, ГРА, JAVA, UI, UX, ANDROID STUDIO,  
ОПЕРАЦІЙНА СИСТЕМА ANDROID

## Abstract

Diploma project. «Gaming mobile application «CardMatch» using the RUP development methodology». 66 pages, 20 pictures, 3 tables, 12 sources.

Before starting the project, you need to search and analyze similar mobile game applications. This will provide the necessary base of understanding of what the future application will look like and what functionality needs to be given to it. For this, the following mobile applications were taken, which can be found in the public access of the PlayMarket:

- Matching Card.
- Matching King.

The goal of the project is to create a gaming mobile application for users of gadgets with the Android operating system.

The task of the project is to carry out all stages from design to testing and providing a complete copy of the application, according to the chosen RUP design and development methodology.

The end result of the project will be a fully functional game application that will give users the opportunity to play matching cards. It is also possible to choose the theme of the gameplay and its complexity within the game.

The Android Studio development environment will be used to develop the application defined by the requirements. The entire graphical user interface will be designed and built according to the best practices of UI/UX design. The Java programming language will be used to implement the program code of the functionality.

MOBILE APP, GAME, JAVA, UI, UX, ANDROID STUDIO, ANDROID OPERATING SYSTEM

## ЗМІСТ

Скорочення та умовні позначки .....	6
Вступ.....	7
1 Аналіз та характеристика предметної області .....	8
1.1 Дослідження центрального елемента предметної області .....	8
1.2 Опис інших важливих елементів предметної області .....	11
1.3 Огляд та аналіз аналогічних додатків .....	17
1.4 Роль UI та UX в розробці мобільної гри.....	22
2 Проектування функціоналу та архітектури .....	27
2.1 Базовий функціонал та створення діаграм .....	27
2.2 Вибір методології розробки проекту .....	33
3 Розробка програмного коду та інтерфейсу .....	37
3.1 Наповнення класів програмним кодом .....	37
3.2 Використання xml розмітки для створення UI .....	46
4 Тестування функціональності та перевірка якості .....	49
4.1 Мета та суть тестування .....	49
4.2 Загальна перевірка розробленого додатка.....	49
5 Техніко-економічне обґрунтування .....	54
5.1 Огляд тенденцій цільового ринку мобільних ігрових додатків .....	54
5.2 Розрахунок витрат на проектування .....	56
5.3 Аналіз та вибір бізнес-моделі .....	57
Висновки .....	62
Перелік джерел посилання .....	63
Додатки.....	64

					ДП.КН 23.507.09.000 ПЗ			
Зм.	Арк.	№ докум.	Підпис	Дата				
Розроб.	Зьомко В. А.				Ігровий мобільний застосунок «CardMatch» з використанням методології RUP	Літ.	Арк.	Аркушів
Перевір.	Кузик В.М.						5	66
рецензент	Гавришків Н.Г.					ГФК.ВКТ.ЦКІКД КН-41		
Н.контр.	Кульчинська Н.З.							
Зав. від.	Стефурак Н.А.							

## СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ОС – операційна система.

ASDK – Android Software Development Kit.

UI – User Interface.

UX – User Experience.

UML – Unified Modeling Language.

QA – Quality Assurance.

POP-UP – Pop of popular – user interface

SCRUM – Sprint collaboration repetition and monitoring

RUP – Rational unified process

					ДП.КН 23.507.09.000 ПЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

## ВСТУП

Дипломний проєкт буде складатись з послідовних етапів осмислення вибраної теми, що дозволить створити базу для подальшого формування вимог, вибору методології та створення архітектури проєкту. На основі усіх цих етапів буде виконано створення програмного коду мобільного додатка та відповідно по завершенню протестовано.

Для аргументації обраної теми буде проведено аналіз кожної її складової. Проведення огляду ринку та мобільних застосунків та виокремлення переваг операційної системи Android не лише для кожного користувача окремо, а як величезної індустрії в сфері розвитку сучасних технологій. Також важливо визначити переваги обраної операційної системи та виду пристрою для розробки саме ігрового додатка:

- Великий ринок. Якщо створювати мобільні програми чи ігри для широкого загалу, доцільно орієнтуватися на платформу з найбільшою кількістю потенційних користувачів. Домінування на світовому ринку є очевидною перевагою ОС Android.

- Рентабельність. Якщо додати в Google Play, Amazon і китайські магазини програм, ОС Android заробляє приблизно стільки ж грошей, скільки Apple. Програми для Android можуть бути на одному рівні з програмами для iOS у більшості категорій, як щодо початкових завантажень програм, так і щодо покупок у програмі.

- Розробники додатків можуть використовувати Android Software Development Kit, не турбуючись про витрати на ліцензування.

- Більше простору для творчості та інновацій, ніж з iOS.

- Процес розробки програми може бути простішим і швидшим, що економить гроші.

					ДП.КН 23.507.09.000 ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		



# 1 АНАЛІЗ ТА ХАРАКТЕРИСТИКА ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Дослідження центрального елементу предметної області

Тенденції розширення сфери мобільних ігор. Міжнародний союз електрозв'язку оцінив, що було більше шести мільярдів абонентів мобільних телефонів у світі у 2018 р. Завдяки мініатюризації та можливості реалізації мобільних відеоігор сьогоднішній мобільні ігри стають дедалі помітнішою зростаючою сферою ігрового бізнесу та культури. Відбувається постійне розширення асортименту та збільшення кількості ігор, що виробляються та публікуються для кишенькових пристроїв, мобільних телефонів і планшетів. Варто відзначити розширення мобільних ігор також з точки зору якості. Багато нових інновацій, переваги яких принесли мобільні ігри – це специфічні характеристики мобільної медіа екосистеми, включно з цифровими каналами розповсюдження, новими модальностями інтерфейсу та сенсорні можливості, що доступні в сучасних мобільних пристроях.

Іншими важливими факторами мобільних ігор та ігор є щоденні контексти та практики, пов'язані із використанням мобільних додатків. Дослідження понад 4000 користувачів телефонів Android у 2017 році показало, що середній користувач отримував доступ до якоїсь програми чи іншої програми на телефоні близько 50 разів на день, загальною тривалістю більше однієї години на день. Однак середня сесія від відкриття сесії роботи з мобільним пристроєм до її закриття тривала лише 71 секунду. Незважаючи на те, що середні ігрові сеанси на мобільному пристрої, ймовірно, довші, розробка гри для швидких та коротких сеансів використання мобільних пристроїв відрізняється від створення типової комп'ютерної чи консольної відеоігри. Є лише кілька ігрових жанрів, унікальних для мобільних пристроїв; можна отримати доступ до більшості популярних жанрів відеоігор також як до мобільних версій.

Визначення мобільної гри, як сутності. Незважаючи на те, що це є досить поширеним терміном і явищем у сучасному технологічно інтенсивному суспільстві, не обов'язково само собою зрозуміло, що саме являє собою, і

					ДП.КН 23.507.09.000 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

визначає мобільну гру. Література про мобільні ігри часто технічно орієнтована і, як правило, рухається безпосередньо для обговорення впровадження ігор для мобільних телефонів та інших мобільних пристроїв, не уточнюючи самої ключової концепції. Однак існує кілька різних видів мобільних пристроїв, і навіть персональні комп'ютери сьогодні можна вважати “мобільними” через популярність невеликих і легких портативних комп'ютерів. Найпоширеніші способи розуміння мобільних ігор, однак, стосуються двох різних напрямків розробки та публікації ігор. Перший – це ігри на мобільних телефонах, а другий – на кишенькових електронних пристроях та консолях. Поза ринком споживчих товарів також є важлива робота, яка пов'язує мобільні ігри з мобільними обчисленнями та експериментами з доповненою реальністю, наприклад. Такі дослідження часто стимулювали інновації в побутовій електроніці та ігровій галузі.

Розподіл кишенькових відеоігор через продажі в роздрібних магазинах також сильно відрізняється від розповсюдження ігор для мобільних телефонів, які або попередньо встановлені, або встановлюються самим, використовуючи послуги мобільних даних. Деякі виробники телефонів експериментували з додатковими картками пам'яті як носієм розповсюдження ігор, але без великих успіхів.

Нові можливі напрямки в індустрії мобільних ігор. Успішна серія Angry Birds Rovio добре представляє основний світ мобільних ігор, розроблених для сучасних смартфонів. На основі попередніх надійних формул геймплею такі повсякденні ігри ефективно використовують як інтерфейс сенсорного екрану, так і аудіовізуальні сильні сторони смартфонів, процесор та можливості пам'яті. Багато з цих типів ігор спочатку випускаються як безкоштовні завантажуванні версії, потім вони спокушають гравців перейти на повну, платну версію програм. Альтернативний підхід, який називається моделлю “freemium”, покладається на придбання додатків “premium” таких функцій, як краще ігрове обладнання або додаткові рівні гри, які виходять з інакше безкоштовної гри за межі вбудованих обмежень. Незважаючи на комерційний успіх, такі методи

					ДП.КН 23.507.09.000 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

піддавалися критиці як гравцями, так і розробниками.

Гра з низькою складністю та без особливих зусиль, що характеризують повсякденні мобільні ігри, не обов'язково приваблюють геймерів, а деякі критики вважають стратегії монетизації, що застосовуються у “freemium” іграх, неетичними.

Окрім інновацій бізнес-моделі, мобільні ігри також стояли на передньому плані деяких технологічних експериментів. Існують режими гри, які доступні лише для ігор на мобільних пристроях, таких як ігри на основі локації. Хоча в мобільних та обчислювальних дослідженнях є кілька десятиліть історії, що також включає такі експерименти з грою, саме на початку 2000-х років були запуснені перші мобільні ігри на основі комерційного підґрунтя. Задовго до цього були різні види ігор у стилі пошуку скарбів, які згодом були перетворені на хобі “geocaching” з наявністю точних GPS-навігаційних пристроїв. Перші комерційні ігри на основі локації, такі як BotFighters використовував послуги менш точного місця розташування та SMS-повідомлення для передачі команд гри та інформації між гравцями та ігровим сервером. Збільшення фізичного, міського середовища завдяки можливостям віртуальних ігор, що призводить до росту потужності мобільних пристроїв, які використовуються в багатому діапазоні ігор альтернативної реальності. Такі складні форми мобільних ігор набувають все більшої популярності, але не досягли ніде рівня, яким користуються звичайні та усім привичні мобільні ігри.

Тисячі нових мобільних додатків щомісяця додаються до різних інтернет-магазинів, а ігри – найпопулярніша категорія серед їх сотень мільйонів користувачів. Отже, комерційне та культурне значення мобільних ігор значно розширилося від їх скромних початків у 90-х роках. Сьогодні ігри на мобільних пристроях серйозно ставлять під сумнів ігри на ПК та консолі, особливо якщо планшетні пристрої включені до категорії мобільних пристроїв. Мобільні ігри також стають все більш інтегрованими з популярними соціальними мережами, такими як Facebook. Звіти галузі вказують на більшість із мільярда користувачів Facebook, які активно користуються послугою зі своїми мобільними пристроями.

					ДП.КН 23.507.09.000 ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

Зростає кількість мобільних ігор, які забезпечують певний досвід онлайн-соціальних ігор, включаючи порівняння найкращих балів серед соціальної мережі або надсилання подарунків, або запрошення друзям зсередини мобільного ігрового додатка. Також слід зазначити, що в деяких галузевих дослідженнях незначна більшість мобільних соціальних геймерів є жінками. Тому пов'язування соціальних мереж з мобільними іграми приваблює нову аудиторію.

Дослідження і майбутнє мобільного геймінгу. Є кілька помітних напрямків дослідницької роботи, які стосуються цієї галузі. Зокрема, в Європі кілька дослідницьких центрів, що проводять постійну дослідницьку роботу по мобільних іграх. Одним із фонових факторів став Європейський Союз, який активно працював у підтримці досліджень та розробок мобільних ігор. Наприклад, “Mobile Entertainment and Industry and Culture” (MGAIN) дослідницький проект (2001 – 04) спрямований на розміщення мобільних ігор у більш широкому мобільному контексті та індустрії розваг і запропонував, щоб мобільні ігри продовжували зростати популярністю поряд з іншими мобільними програмами та послугами, такими як мобільна музика, послуги обміну повідомленнями, мультимедіа, азартні ігри, та послуги на основі місцезнаходження. Ще один великий європейський дослідницький проект, “Integrated Project in Pervasive Gaming” ( IPerG, 2004 – 08 ), орієнтований на новий художні, технологічні, та бізнес-можливості, пов'язані з тим, як нові мобільні технології дозволяють розширити ігровий досвід у просторовому, соціальному та часовому вимірах. IPerG випустив дві стипендії, які зацікавили частину простору розробників, відкритих мобільних технологій, а також вплинули на кілька прототипів ігор нових ігрових піджанрів, такі як мобільні скарби, міські пригодницькі ігри та масові багатокористувацькі мобільні ігри.

## 1.2 Опис інших важливих елементів предметної області

До основного елементу предметної області – мобільна гра, додаються інші дотичні елементи, без яких повна картина предметної області проекту не

					ДП.КН 23.507.09.000 ПЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

можлива. Таким елементом є і процеси, що визначають розробку мобільної гри, від зародження ідеї та мотиву до практичної реалізації та досягнення цілей.

Процес розробки мобільної гри ділиться на відповідні етапи, що своєю суттю відображаються процес розробки програмного застосунку в цілому. Технологічні вдосконалення підняли тенденцію мобільних ігор на новіші висоти, ніж традиційні ігри. І це відкрило багато областей для підприємств. Перетворення концепції розвитку гри в реальність є складним. Потрібно знати процес розробки мобільних ігор для створення застосунку, що може називатися мобільною грою. Він потребує планування, проектування, впровадження та розповсюдження гри в цільовому магазині додатків, якщо це комерційний проект. А використовуючи ексклюзивні технічні засоби, також потрібно знати різні кроки для побудови гри. Успішна гра передбачає деякі етапи перед виробництвом та після виробництвом. Краще розділити ці етапи на кроки, щоб більш доступно описати процес розробки мобільних ігор.

Не можливо заперечити, що ринок мобільних ігор є дуже прибутковим, широким та різноманітним. І мобільні ігри з'явилися протягом останніх кількох років. Відповідно до останнього звіту Statista, 3,24 мільярда людей грають у мобільні ігри по всьому світу. І можна очікувати, що ця кількість зросте в наступні роки.

Пандемія COVID-19 мала величезний вплив на тенденції мобільних ігор. За даними GameAnalytics, постачальника інструментів мобільної аналітики ігор, кількість мобільних геймерів зросла на 46% протягом першого кварталу 2020 року.

Згідно з останнім звітом Newzoo, індустрія мобільних ігор розвиватиметься зі складними річними темпами зростання в 11% і розраховуватиме досягти \$116,4 млрд. доходів до 2024 року. Діаграма тенденцій розвитку прибутковості сфери зображена на рисунку 1.1.

					ДП.КН 23.507.09.000 ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

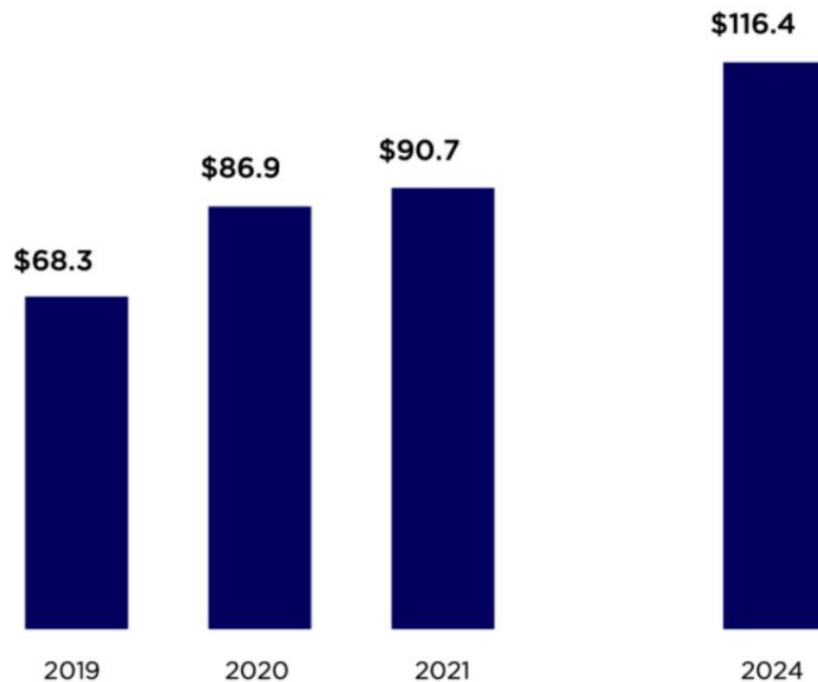


Рисунок 1.1 – Графік розвитку доходів сфери мобільних ігор

Щодня тисячі застосунків додаються до магазину додатків та ігрового магазину, але не всі ігри мають такий же потенціал, як знамениті Clash of Clans та PUBG. Успішний ігровий додаток слідує за твердою ідеєю, відповідною стратегією монетизації та містить захоплюючий та унікальний геймплей.

Щоб полегшити процес розробки мобільних ігор, ось покроковий посібник про те, як зробити ігровий мобільний додаток:

#### 1) Ідея гри.

Ідея – це основа будь-якої мобільної гри. Не потрібна дуже креативна чи інноваційна ідея. Можна почати з простої, але нестандартної ідеї, і перетворити її на чудову.

Ідея може бути абсолютно новою, або також можна взяти натхнення в вже відомій успішній грі. Це допоможе знайти винятковий спосіб надання зручного досвіду потенційним клієнтам майбутнього ігрового застосунку. Саме досвід гри спонукав мільйони геймерів у всьому світі повернутися до мобільних ігрових додатків.

## 2) Створення історії залучення.

Історії ігор відіграють ключову роль у процесі розробки мобільних ігор. Щоб повідомити користувачам програми про функції та мету гри, можна розказати історію, яка містить повний опис етапів, символів та нагород.

Використання концепції також має важливе значення в розробці мобільних ігрових додатків. Створити свою історію таким чином, щоб залучити цільову аудиторію та допомогти отримати переваги для розробленої мобільної гри.

Зрештою, саме ця історія змушує людей відчувати, що те, що вони дивляться, справжнє, а не вигадка, змушує їх відчувати себе пов'язаними з персонажем, і саме цього потрібно досягти.

Але для розробки ігор, таких як головоломки, не потрібно робити історії, оскільки немає необхідності в небажаних елементах.

## 3) Розробка плану гри.

Обов'язково ретельно створювати ігровий план для проекту розробки ігор. Це допомагає передати та виконати свої ідеї ефективніше. Стандартний ігровий план включає:

- графіка чи мистецтво;
- маркетинг;
- ігровий процес або дизайн;
- технічна структура та розвиток гри;

Рекомендується розглянути гнучкий метод розробки ігор, щоб можна було працювати з усіма своїми стратегічними елементами. Будь-які зміни, внесені в елемент плану, також можуть змінювати інші елементи.

## 4) Вирішення орієнтовної вартості розробки гри.

Щоб вирішити орієнтовний бюджет на розвиток мобільних ігор, потрібно врахувати деякі фактори.

Ці фактори допомагають зробити гру готовим до ринку продуктом з високою рентабельністю інвестицій.

					ДП.КН 23.507.09.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

Ось найважливіші фактори:

- мобільна операційна система;
- витрати на попередню розробку;
- витрати після розробки.

#### 5) Вибір Платформи розвитку гри.

Важливо визначити якою буде платформа для запуску ігрового додатка. Існує близько 73% користувачів смартфонів на базі Android, а кількість користувачів iOS величезна. Отже, це питання покладається на бюджет та власні уподобання.

Якщо вибрати розробку єдиної платформи, можна представити гру на обмеженому ринку. Але, якщо ціллю є збільшити площу охоплення програми та мати розумний бюджет, можна вибрати розробку на декілька платформ.

#### 6) Вибір мобільного ігрового двигуна.

Задній план ігрового додатка є дуже важливим, оскільки саме тут ігровий двигун стане в нагоді. Є багато двигунів на вибір для мобільних ігор, і можна вибрати той, який може відповідати ігровим потребам. Наприклад, перейти на 3D-двигун Unity для розширення можливостей 3D-ігор або вибрати Cry Engine, щоб підтримати функцію перетягування у грі.

Можна використовувати зворотні мови з різних варіантів, таких як C #, C + +, Node.js, Java, Laravel та Python для розробки ігор. Оскільки цей етап вирішить інфраструктуру додатка, розглядати потрібно найбільш відповідні бек- енд технології.

#### 7) Створення ігрового дизайн-документа

Після складання всієї концепції гри, відбувається перехід до документації щодо дизайну гри. Розробка дизайнерського документа необхідна для розробки гри. Цей живий та детальний документ містить усі деталі про гру, включаючи технології, елементи, дизайн, досвід, архітектуру тощо. Документ записує все, що має бути додано до додатка, і гарантує, що додаток тісно співставлений з кодом для розробки та запуску.

					ДП.КН 23.507.09.000 ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		



#### 8) Дизайн структури гри.

Приваблива ігрова інфраструктура допомагає збільшити продажі. Користувачі будуть грати в гру лише в тому випадку, якщо вона містить унікальну та привабливу графіку та сцени. Розробляючи архітектуру, обов'язково потрібно додавати такі елементи, як ігрове оточення, символи, предмети, текстури тощо.

Цей крок визначає кінцевий вигляд гри. Можна використати 2D або 3D-дизайн, щоб уточнити візуальні зображення гри. Крок передбачає багато кодування та проектування.

#### 9) Виготовлення відтворюваного прототипу.

Необхідно побудувати відтворюваний прототип для цільової ігрової платформи. Прототип повинен нагадувати найважливіші частини гри та включати всю істотну механіку. Хоча побудова прототипів може зайняти багато часу, важливо з'ясувати помилки дизайну гри.

Гравці можуть чітко поводитися, якщо управління грою не є спонтанним і деякі завдання важко виконати. Отже, можна легко виправити багато проблем на цьому етапі.

#### 10) Проектування та розробка мобільної гри.

Після затвердження прототипу починаються фактичні роботи з проектування гри. Навіть найменші речі тепер повинні бути виконані обережно. Тут буде продемонстровано творчість розробника, що зробить мобільну гру більш привабливою.

З'ясувавши все, слід почати будувати мобільну гру. Щоб створити винятковий бек-енд програми, варто враховувати всі функції, функціональні можливості, технології, досвід, навігацію тощо.

#### 11) Тестування мобільної гри.

Перш ніж повністю закінчити роботу над розробкою мобільної гри, їй потрібно тестування QA. Зрештою, гра буде представляти цілісну картину старань розробника, і потрібно переконатися в запуску ідеальної та без помилок версії. Отже, наступний крок є обов'язковим.

					ДП.КН 23.507.09.000 ПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

Професійні розробники тестують мобільні ігри, вводячи та граючи бета-версію на кількох пристроях. Він показує різні помилки, які може мати гра, а отже, можна їх виправити та створити найкращу версію гри.

Представлені етапи є загальним представленням розробки мобільної гри. Виходячи з цієї інформації надалі і буде виконуватись вибір конкретних елементів, з яких буде компонуватися мобільна гра для цього проекту. Тому в силу можливостей та поставленого завдання буде відбуватись вибір потрібного середовища розробки, мови програмування, підходу до дизайну.

### 1.3 Огляд та аналіз аналогічних додатків

Для подальшої розробки власного програмного додатка важливим є огляд подібних рішень, що може позитивно вплинути на прийняті рішення при безпосередній розробці чогось нового. Було обрано два схожих додатки, один з яких відносно нижче іншого, що дозволить побачити плюси та мінуси обох на контрасті один одного.

Першим додатком для огляду обрано гру Matching Card. Гра зустрічає користувача головним екраном, який зображено на рисунку 1.2.

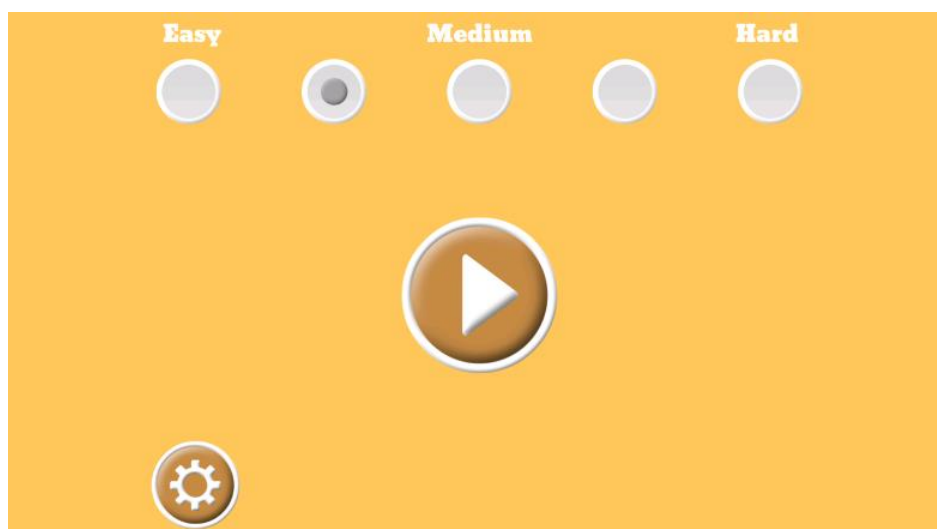


Рисунок 1.2 – Головний екран Matching Card

Головний екран максимально простий та не привабливий для користувача. Правильною практикою можна виділити наявність кнопки налаштувань, що дасть змогу користувачеві деяким чином вплинути на процеси

					ДП.КН 23.507.09.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

гри, які можуть йому не подобатись. Зазвичай це зміна налаштувань звуку та музики, якщо таке присутнє в додатку. На головному екрані користувачеві одразу ж надається можливість обрати рівень складності гри, що буде одразу ж початою після натискання кнопки в центрі екрану.

Отже далі переглянемо екран налаштувань (рисунок 1.3), натиснувши на інтуїтивно зрозуміло кнопку з шестернею.



Рисунок 1.3 – Екран налаштувань Matching Card

За допомогою налаштувань гра дає можливість мінімально змінити на власний розсуд елементи графічного оформлення заднього плану та зображень, які будуть розміщені на карточках під час гри. Також тут є доволі стандартні вимкнення звуку зовсім та вимкнення лише музики. Цікавою та не зовсім стандартною є можливість контролю часу відкритості карточок та кнопка встановлення усіх налаштувань до первісного виду.

Після зміни деяких налаштувань переходимо назад на головний екран натиснувши кнопку підтвердження внесених змін ОК. Далі встановлюємо бажаний рівень складності та переходимо безпосередньо до ігрового процесу за допомогою кнопки посередині головного екрану, яка за ігровою логікою відповідає своїм виглядом спонуканням до початку чогось, в нашому випадку до початку ігрового процесу. (рисунок 1.4).

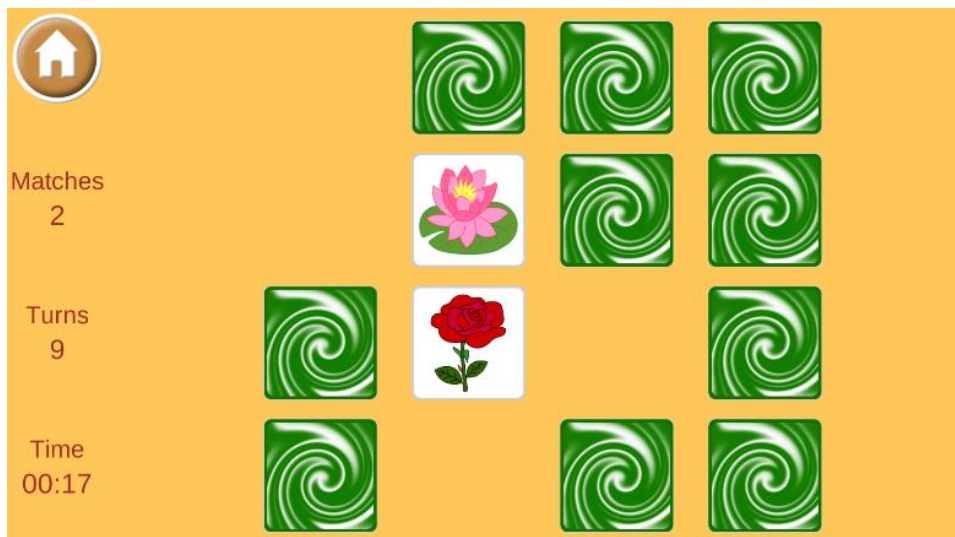


Рисунок 1.4 – Екран ігрового процесу Matching Card

Самий головний екран додатка містить в першу чергу карточки, які користувачеві потрібно буде відкривати та знаходити пари однакових. В лівому верхньому куті знаходиться кнопка повернення на головний екран, яка відповідає за маршрутизації між екранами додатка. Ігровий процес також виконує функцію підрахунку спроб відкриття карточок та кількість вже знайдених однакових пар. Важливим елементом екрану також є таймер, що спонукає гравця до зосередження та виконання цілі гри, як можна швидше. Карточки містять на зображення різного виду квітів, тварин та інше. Тобто якась класифікація відсутня. По закінченню знаходження пар, гра пропонує перейти на наступний рівень складності.

Друга порівняно краща в усіх аспектах гра Matching King. Головний екран (рисунок 1.5) вже максимально стилізований графічно та використано анімацію окремих елементів для більшої зацікавленості користувача.

На головному екрані зображені елементи для початку ігрового процесу, що містять для кращого розуміння і зображення і текст.

Усі три елементи по центру екрану мають легку анімацію, що приваблює додаткову увагу.



Рисунок 1.5 – Головний екран Matching King

Також розміщена кнопка для змінення налаштувань та кнопка виходу з додатка. Нижче по сторонах можна побачити кнопки відображені у вигляді хмаринок для перегляду досягнень користувачів гри. В самому низу розміщена кнопка для перегляду інших ігор розробника.

Переглянемо екран налаштувань додатка, що зображений на рисунку 1.6.

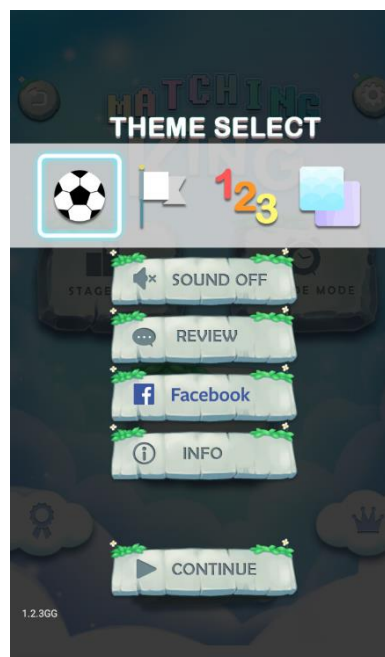


Рисунок 1.6 – Екран налаштувань Matching King

Налаштування пропонують користувачу обрати тематику гри, що вплине на зображення карточок ігрового процесу, можливість вимкнути звук додатка,

					ДП.КН 23.507.09.000 ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

інші кнопки відповідають за зворотній зв'язок чи отримання інформацію про розробників. Внизу розміщена кнопка для застосування змінених налаштувань та повернення на головний екран.

Переходячи до огляду ігрового процесу (рисунк 1.7) потрібно розібрати та описати його різні види. Matching King надає можливість вибору трьох різних видів ігрового процесу:

- Почергове проходження рівнів, які поступово стають складнішими.
- Аркадний режим без рівнів, суть якого – це швидке знаходження усіх пар серед карточок, після чого невеликий таймер оновлюється. Гра відбувається до тих пір поки новий етап гри не буде пройдено за виділений час.
- Два гравці – режим в якому по чергово два гравці на одному екрані виконують спроби знаходження однакових пар карточок.



Рисунок 1.7 – Екран ігрового процесу Matching King

Екран при ігровому процесі виглядає цікаво та яскраво, як і подобає гри. Зовнішня сторона карточок є прозорою з зображенням знаку питання та тінями для створення ілюзії об'ємності. Окрім головних елементів гри – карточок на екрані ігрового процесу присутні: назва поточного рівня, кількість спроб перевертання карточок, зірки, що відповідають за успішність проходження

					ДП.КН 23.507.09.000 ПЗ	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

рівня та кількість вже знайдених пар. Це що стосується ігрової інформації, а також у верхніх кутах розміщенні функціональні кнопки, одна для паузи гри та відображення ігрового меню, а інша для “підглядання” усіх карточок на екрані.

Також додаток Matching King при здійсненні певних дій, як вихід на головне меню, чи вихід з додатка та інші відображає спливаючі вікна, що є дуже хорошою UI/UX практикою в розробці мобільних ігор.

#### 1.4 Роль UI та UX в розробці мобільної гри

Остання важлива складова предметної області розробки мобільних ігор – це UI та UX дизайн, що стоїть поруч з ігровим двигуном по важливості. У процесі ігрового дизайну дизайнери та художники відповідають за те, щоб зробити ігри казковими та мають на меті забезпечити найвищу якість ігрового процесу, так звану відтворюваність. Дизайн мобільної гри включає UI (інтерфейс користувача) та UX (досвід користувача). Здебільшого, як дизайнери UI, так і UX розробляють стиль всієї гри, від кольорів до типографії, віджетів та інших інтерактивних елементів, які гравці використовуватимуть.

В ідеалі дизайн інтерфейсу для мобільних ігор спрямований на створення бездоганного та приємного геймплею. Користувачі хочуть, щоб гра була легкою для розуміння, інтуїтивно зрозумілою, переконливою та привабливою. Саме це робить дизайн складним процесом. Ще одним викликом для художників є боротьба з невеликим розміром екрана, який передбачає набагато менше місця, на відміну від ігор для ПК або планшета. Також успіх ігрового інтерфейсу значно залежить від дизайну UX. Щоб краще розуміти матеріал, що буде представлено надалі потрібно визначити різницю між двома процесами дизайну ігор – UI та UX, які часто плутаються та використовуються взаємозаміно.

Кожен початківець дизайну мобільних ігор повинен знати, що UI для мобільних ігор та UX – це не одне і те ж. По суті, UI – це макет елементів інтерфейсу, тоді як UX відповідає за те, як працює загальна гра, і користувачі взаємодіють з нею.

					ДП.КН 23.507.09.000 ПЗ	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

Винятковий художник UI має глибокі навички малювання, має видатне бачення естетики дизайну та знає різні методи правильного функціонування інтерфейсу. Художники UI розробляють усі візуальні компоненти гри, включаючи кнопки, піктограми, символи, оточення, графіку тощо.

Завдання хорошого дизайнера UX – дослідити поведінку користувачів за допомогою програми, створити прототипи, провести тестування. Тож дизайнер UX буде дуже детальний кадр взаємодії та реакції, який повинен задовольняти гравців під час гри.

Дизайн мобільних ігор: найкраща практики UI та UX. Існує багато принципів дизайну ігор, які допомагають дизайнерам у виготовленні якісної продукції. Важливо проаналізувати найкращі поради та ідеї для кращого UI та UX:

#### 1) Спочатку дослідження

На початку проекту важливо з'ясувати, хто цільова аудиторія, які їх переваги та як створити таку гру, яка б залучила якомога більше користувачів. Деякі дизайнери зазнають невдачі через когнітивні упередження – вони думають, що гравцям сподобається те, що вони малюють, не досліджуючи ринок клієнтів. Ось чому дослідження та розуміння цільової аудиторії є запорукою успіху.

Вивчення ігор та портфоліо конкурентів. Пошук подібних мобільних ігор допоможе розробити кращі програми.

Визначення потенційних користувачів, щоб створити дизайн, орієнтований на користувача.

Створення супер простих прототипів своїх ідей, щоб отримати якомога більше цінних відгуків. Оброблення зворотного зв'язку і підтвердження прототипів, щоб нарешті отримати ідею, яка буде працювати для потенційних користувачів.

#### 2) Забезпечення доступності для всіх

Дизайнери повинні мати на увазі, що гравці різні, отже, різні стилі ігор та жанри. Наприклад, у людей можуть бути погані зір, рухові та слухові розлади.

					ДП.КН 23.507.09.000 ПЗ	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		



Важливо всебічно розробляти свої мистецькі рішення. Завдяки всебічному підходу до дизайну, гравці з порушеннями або без них однаково насолоджуватимуться однією і тією ж грою.

### 3) Розгляд відволікання користувача

Дизайнери повинні мати на увазі, що користувачі часто відволікаються від ігор з різних причин – вхідні дзвінки, низький рівень акумулятора, погане підключення до Інтернету тощо. Завдання дизайнера тут - подумати про те, яким має бути UX у цій справі, і як тримати користувачів у подальшій грі.

Наприклад, деякі ігри не закриваються і не призупиняються, коли хтось телефонує гравцеві. У верхній частині екрана з'являється невелике вікно з двома кнопками – Прийміть або відхиліть дзвінок. Таким чином, користувач, а не додаток чи смартфон, вирішує, продовжувати грати чи починати розмову.

### 4) Додати лише необхідні функції

Щоб створити мобільний ігровий інтерфейс як красиво, так і функціонально, дизайнери UI та UX повинні прагнути знайти баланс між компонуванням гри та її особливостями. Намагання додати занадто багато варіантів може відволікти гравців і перетворити їх досвід на жах. Новачки в грі можуть застрягти і не знають, як почати гру, рухатися далі або навіть виходити з неї. Найпопулярніші ігри на ринку мають певний набір функцій, орієнтованих на допомогу користувачам у виконанні лише необхідних дій.

Щоб покращити ігровий досвід, потрібно проаналізувати найбільш використовувані функції та уточнити їх, щоб зробити дизайн функціональним та непомітним за один раз.

### 5) Спростити інтерфейс

Безладний дизайн інтерфейсу для мобільних ігор зазвичай переповнений надмірною інформацією та деталями. Додаткові кнопки, піктограми та текст створюють безлад на екрані.

Безлад – це катастрофа для мобільних додатків взагалі, оскільки у них набагато менше місця, ніж на екранах ПК. Важливо видалити все, що робить дизайн складним і заплутаним для гравців.

					ДП.КН 23.507.09.000 ПЗ	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		

Для цього:

- Створити простий у використанні дизайн для виняткової роботи користувачів.
- Визначити пріоритетні кнопки, що спонукають до дії.
- Уникати заповнення екрана небажаним вмістом, що негативно впливає на взаємодію користувача.
- Переконатись, що текст, зображення, піктограми та інші елементи чіткі та зрозумілі.
- Подумати про розумне використання білого простору.
- Не перестаратись з кольорами – це відволікає гравця від ігрового процесу.

#### б) Уточнення ігрової навігації

Серед інших принципів дизайну ігор, основним є налагодження навігації в грі. Гравці повинні легко переходити через інтерфейс. Для цього:

- Переконатись, що панель навігації, піктограми та кнопки виглядають інтуїтивно зрозумілими для користувачів. Гравці відчують себе задоволеними, коли вільно переходять через гру.
- Організувати інформацію в грі послідовно, щоб гравці виконували якомога менше дій, щоб досягти кінцевої точки.
- Навігація повинна засвоюватися загальним дизайном інтерфейсу мобільної гри та виглядати як природна частина вмісту додатка. З одного боку, це повинно бути видно, а з іншого – не повинно вражати очей.
- Повідомити гравців про їх поточне місцезнаходження всередині додатка, щоб вони могли безперешкодно переходити до гри.
- Переконатись, що параметри навігації завжди знаходяться в одному місці на різних екранах. Якщо переміщувати їх туди-сюди з місця на місце, користувачі будуть плутатись і дезорієнтуватись.
- Забезпечити логічне управління навігацією з точки зору гравців. Таким чином вони швидко знайдуть те, що шукають.

					ДП.КН 23.507.09.000 ПЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

#### 7) Розмістити UI належним чином

Якщо врахувати, що більшість гравців мобільних ігор є правшами, дизайнери повинні мати на увазі, що інтерфейс повинен бути адаптований з урахуванням областей, що натискаються на екрані. Пристрій може мати незручні та важкодоступні зони.

#### 8) Надати зворотній зв'язок користувачам відносно дій

У ідеальному ігровому світі об'єкти реагують на наші дії. Навіть уявний світ ігор повинен робити те саме. Гравці очікують, що елементи UI відповідають їхнім діям. Мета дизайнерів – забезпечити отримання користувачем зворотного зв'язку після дії. Якщо елементи гри не дають зворотного зв'язку, гравець може спантеличитись, чи те, на що користувач натиснув не працює, чи це він не натиснув на нього.

#### 9) Використовувати спливаючі вікна з розумом

POP-UP – це функціональний компонент гри, який використовується для передачі додаткової інформації гравцям. Підключити видиму частину інтерфейсу користувача до невидимої за допомогою анімації: після того, як гравець натискає на елемент, що викликає спливаюче вікно, краще оживити запуск вікна. Наприклад, збільшити його, коли він відкриється, і мінімізувати вікно після закриття. Це допоможе користувачам краще зрозуміти походження спливаючого вікна.

#### 10) Завжди тестувати

Зазвичай навіть візуально досконалий і продуманий дизайн виявить свої невидимі недосконалості при введенні у експлуатацію. Дизайн зазвичай виглядає ідеально на великому екрані розробника, але він не виглядає однаково на смартфоні користувача. Ось чому важливо перевірити гру з реальними гравцями на різних пристроях, щоб переконатися, що вона відповідає очікуванням.

					ДП.КН 23.507.09.000 ПЗ	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дата		

## 2 ПРОЕКТУВАННЯ ФУНКЦІОНАЛУ ТА АРХІТЕКТУРИ

Етап проектування розроблюваної програмної системи є одним з найважливіших в усьому життєвому циклі додатка. Завдяки цьому етапові будуть окреслені основні функціональні залежності та сутності. Наступний етап розробки та реалізації буде повністю залежати від проектування.

Діаграми надають візуальний огляд робочого процесу та/або системи, спрощуючи розуміння та спілкування між командами розробників і користувачами. Діаграми використовуються, щоб представити або створити спільне розуміння того, як зараз працює робочий процес користувача та як він може працювати, як з ділової, так і з технічної сторони.

Діаграми дозволяють обробити все, що потрібно користувачам, і відображати ці потреби їм у організованому, споживаному вигляді. Коли кожен може візуально побачити процес або запропоновану систему – розробник витрачає менше часу на пояснення та більше часу на розбудову та вдосконалення процесів. Для представлення усього описаного вище будуть використовуватись різного виду UML діаграми та їх опис.

### 2.1 Базовий функціонал та створення діаграм

Діаграми варіантів використання зазвичай створюються на початковому етапі проектування. Вони використовуються для зображення функцій системи та учасників, залучених до використання цих функцій. Вони зображені за допомогою спеціального позначення, яке гарантує відсутність двозначності в тому, що намагаються передати. Вони мають перевагу перед текстовим описом, який може бути відкритим для тлумачення, що може спричинити проблеми з наданням вибраного рішення.

Діаграма варіантів використання зображена на рисунку 2.1.

					ДП.КН 23.507.09.000 ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

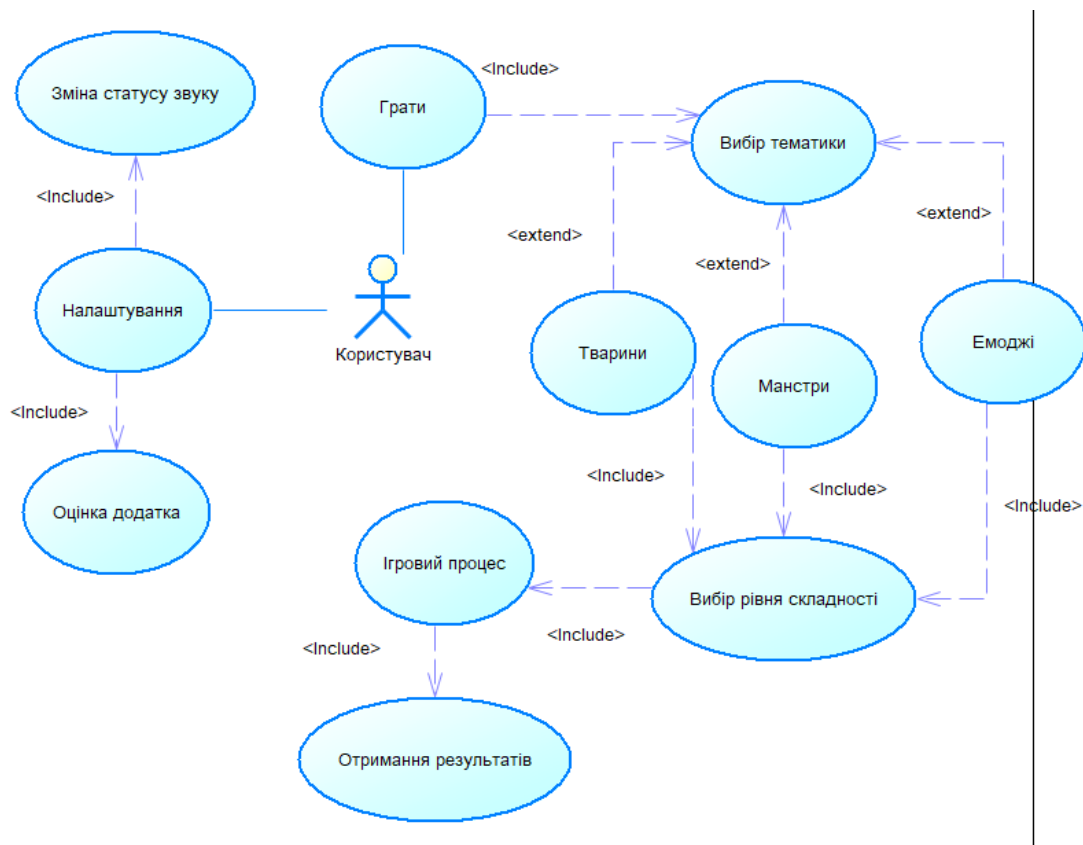


Рисунок 2.1 – Діаграма варіантів використання

Проектований додаток буде отримувати зовнішній вплив лише від одного актора – Користувача, який являється персоною, хто буде користуватись розробленим додатком. Усі необхідні для збереження дані не будуть великими, тому їх усіх буде збережено всередині системи без застосування особливих баз даних. Усі варіанти використання визначенні зв'язком:

- асоціації, якщо вони первинні та виходять напряду від користувача;
- зв'язком залежності типу “include”, який виражає залежність унікальних нерозширюваних варіантів використання;
- зв'язком залежності типу “extend”, який виражає залежність декількох не унікальних розширюваних варіантів використання.

Для однозначного тлумачення зображених варіантів використання на рисунку 2.1 до них надається опис:

- Налаштування – перехід до екрану налаштувань системи.
- Зміна статусу звуку – перша функція екрану налаштувань, що дозволяє встановити значення звуку системи на позитивне, чи негативне.

– Оцінка додатка – друга функція екрану налаштувань, яка дасть можливість зворотного зв'язку користувача та розробника. Ймовірно не буде розроблена в першому релізі додатка.

– Грати – перехід до екрану вибору тематики гри.

– Вибір тематики – застосування одного з трьох розширюваних варіантів використання Тварини, Монстри, Емоджі, які визначають тематику зображень на карточках під час ігрового процесу.

– Вибір рівня складності – застосування бажаного рівня складності ігрового процесу, що визначить кількість карточок, серед яких потрібно буде знайти пари. В грі будуть присутні шість рівнів складності: Початківець, Легко, Середній, Важко, Важче, Майстер. З розмірами ігрового поля відповідно: 2x3, 3x4, 3x6, 4x7, 4x8, 5x10.

– Ігровий процес – головна активність додатка, що передбачає собою запуск визначеного рівнем складності таймеру та розміщення карточок для знаходження серед них однакових пар. Тут важливо описати саму суть гри – знаходження серед представлених карточок двох з однаковими зображеннями, після чого вони будуть пропадати з екрану користувача. Паралельно буде відраховуватись таймер для мотивації користувача.

– Отримання результатів – екран з спливаючим повідомленням про успішне проходження рівня вибраної складності. Тут будуть відображені успішність проходження у вигляді зірочок від однієї до трьох. Після цього можна перейти до головного екрану, звідки все починається заново, або перейти на наступний рівень складності, якщо це можливо.

Наступною для проектування системи додатка буде описана діаграма класів, що буде відображати функціональність зсередини. Діаграма класів належить до сімейства структурних діаграм – графічне зображення статичної структурної моделі.

Класи можуть бути пов'язані один з одним різними способами: асоційованими (пов'язаними один з одним), залежними (один клас залежить/використовує інший клас), спеціалізованими (один клас є підтипом

					ДП.КН 23.507.09.000 ПЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

іншого класу) або упакованими (згрупованими разом як одиниця – упаковка). Діаграма класів не виражає нічого конкретного щодо зв'язків даного об'єкта, але вона абстрактно описує потенційні зв'язки одного об'єкта з іншими об'єктами.

Діаграму класів можна використовувати для:

- ілюструвати як прості, так і складні моделі інформаційних систем;
- розуміти загальний огляд схем програми;
- візуально виражати конкретні потреби системи;
- виділити конкретний код, який потрібно запрограмувати та впровадити в структуру;
- забезпечують незалежний від реалізації опис типів, що використовуються в системі;
- поділитися відповідною інформацією із зацікавленими сторонами;
- виявляти помилки на ранніх стадіях життєвого циклу.

Повна діаграма класів зображена в Додатку В. Далі будуть описані класи з встановленням їх використання чи функціональності. При розробці класи будуть наповнені атрибутами та функціями, згруповані за призначенням.

Опис спроектованих класів:

- `GameApplication` – первинний клас системи, який відповідає за запуск додатка.
- `FontLoader` – клас для завантаження потрібних шрифтів, які не є стандартними середовища розробки.
- `MainActivity` – клас первинної активності системи для створення необхідних для роботи додатка змінних та встановлення заднього фону.
- `EventBus` – клас, який буде отримувати, обробляти та передавати події між компонентами користувацького інтерфейсу та ігрового двигуна.
- `ScreenController` – клас для обробки отриманого запиту зміни активності на іншу. Відповідає встановлення потрібного екрану додатка.
- `MenuFragment` – клас фрагмент головного екрану.

					ДП.КН 23.507.09.000 ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

- DifficultySelectFragment – клас фрагмент екрану вибору складності ігрового процесу.
- GameFragment – клас фрагмент екрану ігрового процесу.
- ThemeSelectFragment – клас фрагмент екрану вибору тематики гри.
- DifficultyView – клас розміщення елементів на екрані вибору складності та обробки вибраної складності.
- BoardView – клас відповідальний за виконання анімацій карточок на ігровому полі. Також виконує створення потрібної кількості рядків та стовпців карток.
- TileView – клас, що визначає вигляд та вміст кожної карточки ігрового поля.
- Game – клас, що відповідає за початок ігрового процесу – розміщення карточок вибраної тематики та компонування їх на ігровому полі.
- BoardConfiguration – клас, який містить інформацію про усі можливі компонування ігрового поля. Встановлює потрібну інформацію для ігрового процесу – час, кількість карточок, стовпців та рядків.
- BoardArrangement – клас, що запускає побудову ігрового поля через ігровий двигун до початку гри для користувача.
- GameState – клас даних, що містить можливі стани ігрового процесу для правильної роботи ігрового двигуна.
- Engine – клас двигун ігрового процесу, виконує встановлення всіх потрібних параметрів та запуск подій для початку ігрового процесу.
- Event – інтерфейс, який буде реалізовуватись абстрактним класом для подальшого утворення подій системи.
- AbstractEvent – абстрактний клас подій системи, реалізовує інтерфейс Event.
- EventObserver – інтерфейс переглядач подій системи.
- EventObserverAdapter – клас, що завершує ланцюжок інтерфейсів та абстрактного класу і безпосередньо реалізовує їх створюючи потрібні події.



Далі класи-події поділяються на UI та Engine. Це означає, що UI події відповідають за виконання процесів для графічного інтерфейсу користувача, а Engine події за процеси ігрового двигуна. Також окрему будуть виділені три POP-UP класи, що відповідають за спливаючі вікна.

UI події:

- ResetBackgroundEvent – клас виконувач події повернення до головного екрану, що тим самим спростовує роботу усіх подій, що тривали за їх непотрібності.
- NextGameEvent – клас для запуску наступного рівня складності для продовження гри після проходження минулого.
- FlipCardEvent – клас реагує на подію перевертання карточки на ігровому полі.
- StartEvent – клас події початку гри з головного екрану.
- ThemeSelectedEvent – клас події вибору тематики ігрового процесу.
- BackGameEvent – клас, що дозволяє повертатись назад з інших екранів.
- DifficultySelectedEvent – клас події вибору рівня складності гри.

Engine події:

- HidePairCardsEvent – клас для приховання пари однакових карточок, що були знайдені користувачем.
- FlipDownCardsEvent – клас для перевертання відкритих користувачем карточок вниз, якщо вони не однакові.
- GameWonEvent – клас запуску події, що означає проходження рівня користувачем.

POP-UP класи:

- PopupManager – клас управління спливаючими вікнами системи.
- PopupSettingsView – клас відображення спливаючого вікна для налаштувань гри.
- PopupWonView – клас відображення спливаючого вікна проходження користувачем рівня ігрового процесу.

Також для представлення маршрутизації всередині системи розроблена діаграма послідовності (рисунок 2.2). Діаграма послідовності в UML зображує співпрацю об'єктів і визначає порядок, у якому відбуваються події між об'єктами для досягнення мети. Діаграма послідовності також відома як тимчасова діаграма, діаграма подій або сценарій подій.

Взаємодія об'єктів більш поширена у верхній частині фігури. Об'єкти взаємодіють один з одним за допомогою повідомлень як на вертикальній, так і на горизонтальній осях, як показано горизонтальними стрілками та мітками повідомлень. Початкове повідомлення діаграми послідовності знаходиться зазвичай зліва. Повідомлення в діаграмах послідовності можна класифікувати та розділяти за призначенням. В якості об'єктів, що будуть спілкуватись між собою на діаграмі виступатимуть екрани додатка: Налаштування, Головний екран, Вибір тематики, Вибір складності, Ігровий процес.

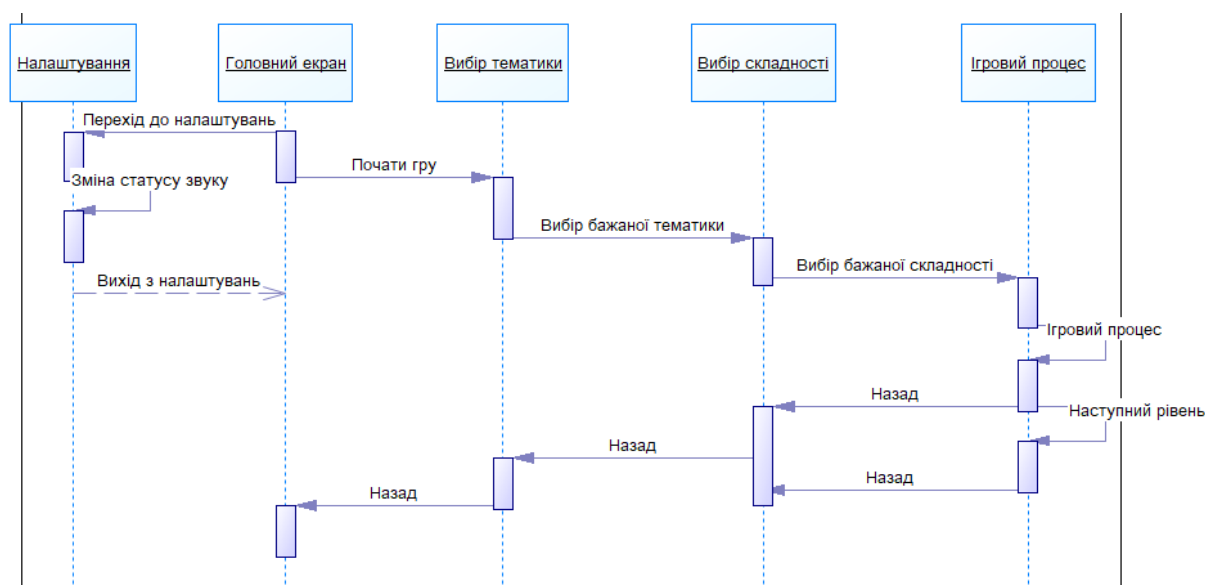


Рисунок 2.2 – Діаграма послідовності проектового додатка

## 2.2 Вибір методології розробки проекту

При виборі методології з багатьох існуючих було обрано дві найбільш підходящі для даного проекту – це SCRUM та RUP. Для остаточного вибору єдиної методології буде проведено їх аналіз та порівняння.

Однією з фундаментальних цілей Rational Unified Process є надання моделі, яка допоможе в ефективній реалізації різних підходів для розробки. Метою RUP є адаптація проекту, команди, а також організації. Є кілька

фундаментальних ідей, від яких залежав процес. Однією з таких ідей є будівельні блоки, які визначають всю процедуру розробки, яка відбуватиметься, наприклад, коли та який розвиток відбуватиметься в конкретному проекті із застосуванням RUP.

Є кілька факторів, які повинні враховуватися членами команди проекту при виборі застосування RUP для розробки проекту. Деякі з факторів, які необхідно враховувати, це:

RUP – це методологія бізнес-моделювання, яка ділить проект розробки на 4 розділи, а саме:

Початок – це перша фаза розробки проекту, на якій включається вивчення запропонованого рішення, аналіз ресурсів, необхідних для завершення проекту.

Розробка. На цьому етапі проект повторно оцінюється, і для визначення архітектури застосовується модель прецеденту. Він включає використання всього програмного забезпечення, яке можливе для виконання оцінки ризиків щодо перешкод, виявлених на першому етапі.

Будівництво. Ця фаза визначила побудову системи, починаючи з менших компонентів, щоб інтегрувати їх у формування повного проектного рішення.

Перехід – після розробки проекту; потім він передається від розробника до клієнта. Коригування можна зробити пізніше відповідно до рівня задоволеності клієнта. Для застосування RUP потрібне різне нове та вдосконалене програмне забезпечення. Таким чином, ця методологія є обов'язковою для використання у випадку великих підприємств, які можуть інвестувати великі суми інвестицій для придбання програмного забезпечення та технологій. Існує шість найкращих практик:

- Розробка програмного забезпечення інтерактивно.
- Керування вимогами.
- Використання компонентної архітектури.
- Програмне забезпечення для візуальної моделі.
- Перевірка якості програмного забезпечення.
- Контроль змін програмного забезпечення.

					ДП.КН 23.507.09.000 ПЗ	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дата		

Rational Unified Process. Scrum – це методологія гнучкого управління проектами, яка використовується для проекту, що передбачає розробку програмного забезпечення. Метою цієї методології є забезпечення можливостей створення нового програмного забезпечення протягом 2-3 тижнів. Це наголошує на командній роботі керівництва проекту.

Складність проекту вирішується за допомогою методології Scrum, оскільки вона допомагає зробити інформацію про проект прозорою. Існує три елементи методології:

Ролі. У цій методології є чотири основні ролі: Scrum Team, Scrum Master, Product Owner і Development Team.

Артефакти. Це різноманітні матеріальні результати роботи, якими є інформація продукту, інформація спринту, мета спринту, прирости та визначення зробленого.

Події. Це один із важливих елементів Scrum, оскільки забезпечує регулярність у методології. Деякі з подій, включених до методології, це спринт, планування спринту, огляд спринту та ретроспектива спринту.

Є деякі фактори, які необхідно враховувати членам команди проекту під час застосування методології Scrum, а саме:

- Структура організації.
- Процес, залучений до проекту.
- Люди, які беруть участь у проекті.
- Технології, які використовуються в проекті.
- Дизайн проекту.

Порівняння RUP і Scrum. У різних дослідженнях було виявлено, що застосування RUP є складним, тоді як із застосуванням Scrum можна вирішити складність, пов'язану з конкретним проектом. Методологія Scrum може бути використана в організаціях, які використовують різні нові та передові технології. Однак застосування RUP не потребує передових технологій. З цієї причини методологію Основна увага Scrum зосереджена на людях та їх потенціалі, а також навичках.

					ДП.КН 23.507.09.000 ПЗ	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

За допомогою застосування Scrum можна як покращити, так і підвищити рівень продуктивності продукту та його якість. Рівень задоволеності різних зацікавлених сторін організації можна підвищити за допомогою Scrum. Однак було виявлено, що бувають випадки, коли застосування Scrum в організації призводить до розповзання обсягу робіт через відсутність конкретної дати завершення конкретного проекту.

Було виявлено, що основна увага RUP зосереджена на необхідності точної документації для розробки проекту. Різноманітний потенціал інтеграції поширюється співробітниками організації. Із застосуванням RUP різні зміни, які відбуваються в середовищі організації, можуть бути адаптовані її співробітниками належним і ефективним чином. Однак одним із найбільших недоліків RUP є те, що його успіх залежить від досвідчених членів команди проекту, що вимагає найму певних експертів всередині організації. Це не завжди можливо для малого підприємства.

Було обрано для розробки проекту методологію RUP, через переваги підходящі для поточного проекту: в процесі розробки створюється супровідна документація, весь проект розробляється покроково однією великою ітерацією, високий рівень використання вже розробленого функціоналу повторно.

					ДП.КН 23.507.09.000 ПЗ	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		

## 3 РОЗРОБКА ПРОГРАМНОГО КОДУ ТА ІНТЕРФЕЙСУ

Для розробки програмного коду проекту було обрано середовище розробки Android Studio. Вибір достатньо очевидний через лідерство на ринку середовищ розробки мобільних додатків. Підтримує розробку на обраній мові програмування Java та інші нові чи старіші мови. Надає влаштовану систему створення та кастомізації віртуальних мобільних девайсів для швидких та зручних тестів. Обрана мова програмування Java, через її багаторічну популярність, що веде за собою велику інформаційну базу для створення потрібних для проекту рішень.

### 3.1 Наповнення класів програмним кодом

Розділ передбачає створення програмного коду спроектованих класів та реалізацію запланованого функціоналу. Буде представлено вирізки з діаграми класів системи з Додатку В, наповнені атрибутами та функціями. Певні частини коду, що визначають важливу та унікальну для проекту бізнес-логіку будуть представлені в лістингах та відповідно описані.

Секція класів фрагментів функціонування різних екранів додатка та сам клас контролер цих екранів зображено на рисунку 3.1.

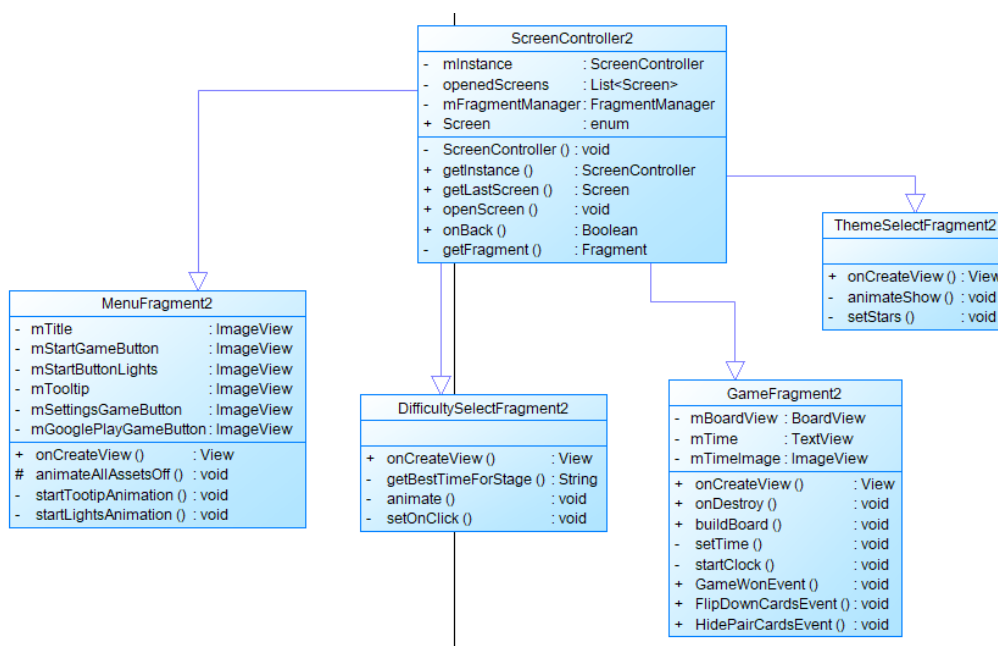


Рисунок 3.1 – Класи фрагменти екранів

Опис функцій класу ScreenController:

ScreenController getInstance() – функція для отримання сутності поточного екрану для обробки.

Screen getLastScreen() – функція для отримання інформації про поточний відкритий екран, що відображається для користувача.

void openScreen(Screen screen) – функція відображення потрібного переданого екрану для користувача.

boolean onBack() – функція перевірки можливості повернення назад з поточного екрану користувача.

Fragment getFragment(Screen screen) – відповідно до отриманої назви екрана створює його клас, що викликає запуск цього екрану для користувача вже за допомогою функціоналу цього класу. Код функції наведено в лістингу 3.1.

#### Лістинг 3.1 – Код функції getFragment(Screen screen)

```
private Fragment getFragment(Screen screen) {  
    switch (screen) {  
        case MENU:  
            return new MenuFragment();  
        case DIFFICULTY:  
            return new DifficultySelectFragment();  
        case GAME:  
            return new GameFragment();  
        case THEME_SELECT:  
            return new ThemeSelectFragment();  
        default:  
            break;  
    }  
    return null;  
}
```

Опис функцій класу MenuFragment:

View onCreateView() – функція виконавець усього, що потрібно при створенні об'єкта класу. Встановлює функціональність для присутніх на екрані кнопок.

void animateAllAssetsOff(AnimatorListenerAdapter adapter) – функція створення та програвання анімацій елементів на головному екрані додатка.

void startTooltipAnimation() – функція для створення та анімації підказки для кнопки початку гри головного екрану.

					ДП.КН 23.507.09.000 ПЗ	Арк.
						38
Змн.	Арк.	№ докум.	Підпис	Дата		

`void startLightsAnimation()` – функція встановлення зображення та його анімації для створення світіння навколо кнопки початку гри головного екрану. Код функції наведено в лістингу 3.2.

### Лістинг 3.2 – Код функції `startLightsAnimation()`

```
private void startLightsAnimation() {
    ObjectAnimator animator = ObjectAnimator.ofFloat(mStartButtonLights,
"rotation", 0f, 360f);
    animator.setInterpolator(new AccelerateDecelerateInterpolator());
    animator.setDuration(6000);
    animator.setRepeatCount(ValueAnimator.INFINITE);
    mStartButtonLights.setLayerType(View.LAYER_TYPE_HARDWARE, null);
    animator.start();
}
```

Секція класів, що будує та виконує найголовніший функціонал додатка – ігровий процес, зображена на рисунку 3.2.

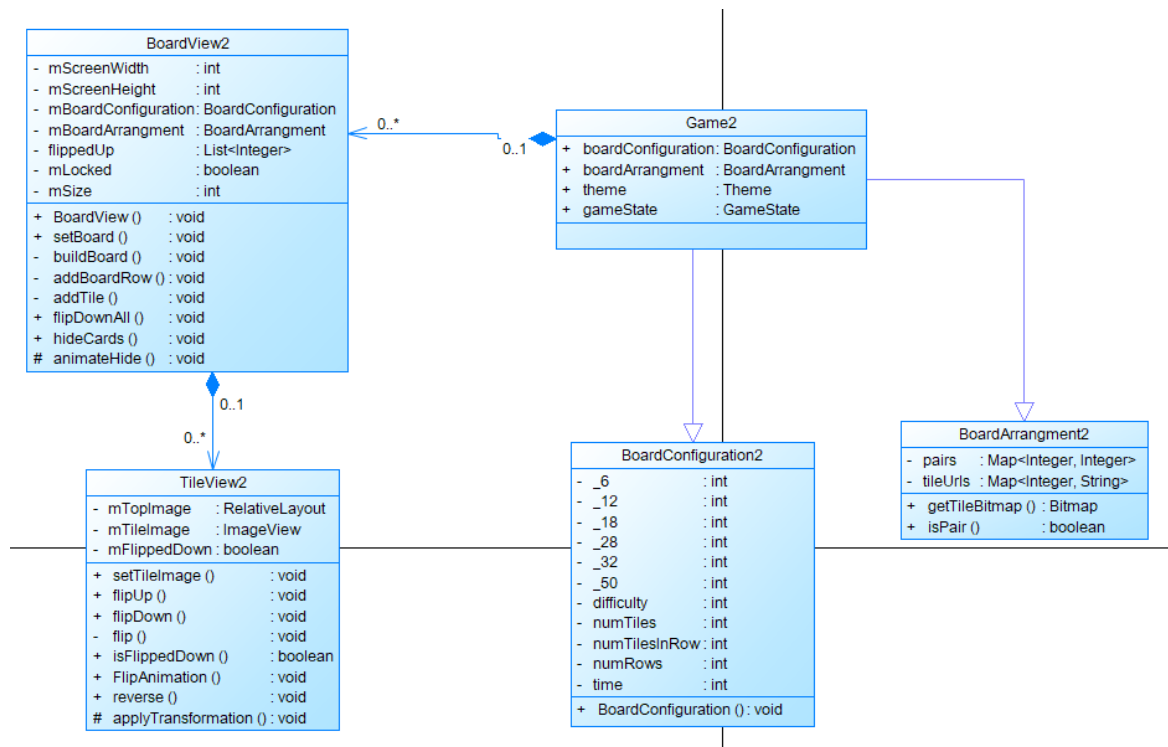


Рисунок 3.2 – Класи ігрового процесу додатка

Клас `Game` виконує роль посередника, між класом, який буде викликати функціональність ігрового процесу та класами з цієї функціональністю. Містить перелік об'єктів класів основного функціоналу ігрового процесу.

Опис функцій класу `BoardView`:

`BoardView(Context context, AttributeSet attributeSet)` – конструктор класу, в якому виконується розрахунок та встановлення параметрів ігрового поля.



`void setBoard(Game game)` – функція для початку побудови ігрового поля, виконує розрахунок висоти та ширини деяких об’єктів. Після проведення необхідних обчислень викликає наступну функцію для створення усіх елементів ігрового поля на графічному інтерфейсі користувача.

`void buildBoard()` – та сама функція побудови ігрового поля, виконує створення рядів карточок.

`void addBoardRow(int rowNum)` – функція безпосереднього створення рядів карточок відповідно до отриманого параметра кількості рядів.

`void addTile(final int id, ViewGroup parent)` – функція створення найменшого елемента ігрового поля за розмір, но найбільшого по важливості – карточок. Розміщує елементи карточки, наповнює їх картинками та визначає функціональність натискання для карточок (лістинг 3.3).

### Лістинг 3.3 – Код функції `onClick()` для карточки ігрового поля

```
tileView.setOnClickListener(new View.OnClickListener() {  
  
    @Override  
    public void onClick(View v) {  
        if (!mLocked && tileView.isFlippedDown()) {  
            tileView.flipUp();  
            flippedUp.add(id);  
            if (flippedUp.size() == 2) {  
                mLocked = true;  
            }  
            Shared.eventBus.notify(new FlipCardEvent(id));  
        }  
    }  
});
```

`void flipDownAll()` – функція для встановлення усіх створених карточок у положення перевернутих вниз.

`void hideCards(int id1, int id2)` – функція для приховання пари карточок – це для подальшого використання у функціональності знаходження однакової пари карточок.

`void animateHide(final TileView v)` – функція реалізації анімації приховання карточок.

Опис функцій класу `TileView`:

`void setTileImage(Bitmap bitmap)` – функція встановлення зображення для карточки. `Bitmap` – клас з бібліотеки класів Java та Android Studio, який

					ДП.КН 23.507.09.000 ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

дозволяє зручну та швидку роботу з зображеннями представляючи їх у двійковому коді.

`void flipUp()` – функція перевертання карточки вверх.

`void flipDown()` – функція перевертання карточки вниз.

`void flip()` – функція перевертання карточки з поточного положення в протилежне.

`boolean isFlippedDown()` – функція для перевірки, чи карточка перевернута вниз.

`void FlipAnimation(View fromView, View toView)` – функція створює 3D анімацію для перевертання карточки. Ефект 3D у плоскому просторі створюється завдяки класу з бібліотеки класів Java та Android Studio – Camera, що дозволяє утворити та маніпулювати матрицею значень положень об'єктів в просторі.

`void reverse()` – функція зміни значень при перевертанні карточки.

`void applyTransformation(float interpolatedTime, Transformation t)` – функція технічної обробки перевертання карточки. Код цієї функції наведено в лістингу 3.4.

Лістинг 3.4 – Код функції `applyTransformation(float interpolatedTime, Transformation t)`

```
// Кут навколо осі Y повороту за визначений час
// обрахований в радіанах та градусах
final double radians = Math.PI * interpolatedTime;
float degrees = (float) (180.0 * radians / Math.PI);

// Коли досягнуто середину анімації, потрібно приховати
// первинний вигляд і показати вигляд призначення. Також потрібно змінити
// кут на 180 градусів, щоб призначення не перевернулось
if (interpolatedTime >= 0.5f) {
    degrees -= 180.f;
    fromView.setVisibility(View.GONE);
    toView.setVisibility(View.VISIBLE);
}

if (forward)
    degrees = -degrees; // визначає напрямок повороту коли
                        // перевертання починається

final Matrix matrix = t.getMatrix();
camera.save();
camera.rotateY(degrees);
```

					ДП.КН 23.507.09.000 ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

```

camera.getMatrix(matrix);
camera.restore();
matrix.preTranslate(-centerX, -centerY);
matrix.postTranslate(centerX, centerY);

```

Клас BoardConfiguration не містить лише конструктор, в якому функціонал для отримання ззовні даних про параметри ігрового поля(складність, кількість карточок, кількість карточок в одному рядку, кількість рядків, час) та виконати повернення параметрів розміщення елементів ігрового поля відповідно до отриманих. Реалізовано це логічною структурою switch case.

Опис функцій класу BoardArrangement:

Bitmap getTileBitmap(int id, int size) – функція передачі зображень для карточок ігрового поля.

boolean isPair(int id1, int id2) – логічна функція для перевірки, чи дві карточки отримані ззовні є парою, тобто містять однакові картинки на передній частині.

Секція класів, що активує та відображає на графічному інтерфейсі спливаючі вікна системи, зображена на рисунку 3.3.

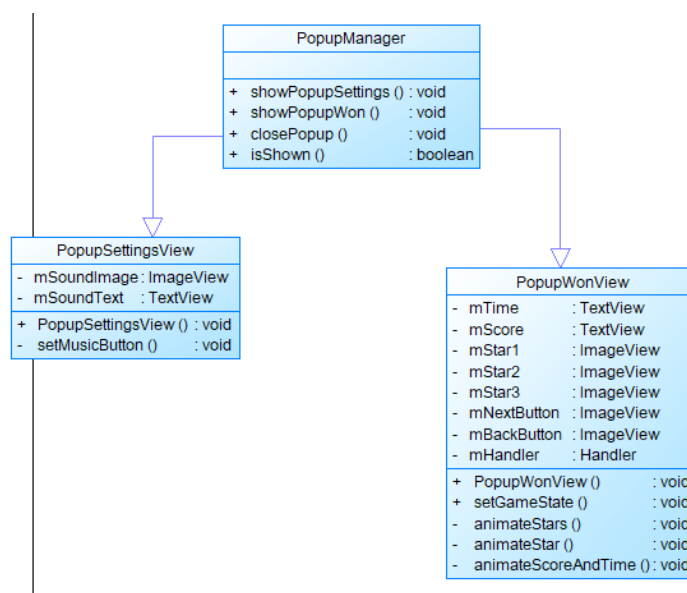


Рисунок 3.3 – Класи спливаючих вікон системи

Опис функцій класу PopupManager:

`void showPopupSettings()` – функція для виклику та анімації спливаючого вінка екрану налаштувань.

`void showPopupWon(GameState gameState)` – функція для виклику та анімації спливаючого вінка інформації при проходженні рівня ігрового процесу. Приймає параметр стану ігрового процесу, якщо він рівний стану завершення рівня, то спливаюче вікно буде продемонстроване.

`void closePopup()` – функція для закриття будь-якого з спливаючих вікон. Виконує приховання елементів, що знаходиться в сутності графічного інтерфейсу, який відповідає за відображення спливаючих вікон.

`boolean isShown()` – функція перевірки, чи відображається будь-яке спливаюче вікно.

Опис функцій класу PopupSettingsView:

`PopupSettingsView(Context context, AttributeSet attrs)` – конструктор класу виконує встановлення потрібних зображень на спливаючому вікні налаштувань та утворює функціональне натискання для них.

`void setMusicButton()` – функція для зміни статусу кнопки налаштувань звуку.

Опис функцій класу PopupWonView:

`PopupWonView(Context context, AttributeSet attrs)` – конструктор класу встановлює потрібні дані на спливаючому вікні при проходженні рівня.

`void setGameState(final GameState gameState)` – функція для обробки даних завершеного ігрового процесу для їх подальшого відображення на спливаючому вікні.

`void animateStars(int start)` – функція для анімації появи потрібної кількості зірочок за проходження рівня гри на спливаючому вікні.

`void animateStar(final View view, int delay)` – функція самої анімації появи зірочок на спливаючому вікні проходження рівня гри.

`void animateScoreAndTime (final int remainedSeconds, final int achievedScore)` – функція анімації часу, що залишився та тексту успішності на

					ДП.КН 23.507.09.000 ПЗ	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		

спливаючому вікні. Всередині цієї функції реалізовані окремі дві функції для поставленої мети, код яких наведено в лістингу 3.5.

Лістинг 3.5 – Код функції animateScoreAndTime (final int remainedSeconds, final int achievedScore)

```
final int totalAnimation = 1200;
Clock.getInstance().startTimer(totalAnimation, 35, new OnTimerCount() {
    @Override
    public void onTick(long millisUntilFinished) {
        float factor = millisUntilFinished / (totalAnimation * 1f); // 0.1
        int scoreToShow = achievedScore - (int) (achievedScore * factor);
        int timeToShow = (int) (remainedSeconds * factor);
        int min = timeToShow / 60;
        int sec = timeToShow - min * 60;
        mTime.setText(" " + String.format("%02d", min) + ":" +
String.format("%02d", sec));
        mScore.setText("" + scoreToShow);
    }

    @Override
    public void onFinish() {
        mTime.setText(" " + String.format("%02d", 0) + ":" +
String.format("%02d", 0));
        mScore.setText("" + achievedScore);
    }
});
```

Останнім описаним класом з його функціональністю буде ігровий двигун гри Engine. Наповнений атрибутами та функціями клас двигун зображено на рисунку 3.4.

Engine2	
- mInstance	: Engine
- mPlayingGame	: Game
- mFlippedId	: int
- mToFlip	: int
- mScreenController	: ScreenController
- mSelectedTheme	: Theme
- mBackgroundImage	: ImageView
- mHandler	: Handler
+ Engine ()	: void
+ start ()	: void
+ stop ()	: void
+ onEvent ()	: void
- arrangeBoard ()	: void
+ getActiveGame ()	: Game
+ getSelectedTheme ()	: Theme
+ setBackgroundImageView ()	: void

Рисунок 3.4 – Клас двигуна гри

Engine() – конструктор класу, створює сутність класу керувань екранами ScreenController для початку роботи.

`void start()` – функція запуску ігрового двигуна для початку ігрового процесу. Виконує надсилання усіх потрібних для гри подій EventBus.

`void stop()` – функція зупинки ігрового двигуна при закінченні ігрового процесу. Виконує вилучення усіх подій з EventBus та зупинку працюючих процесів.

Клас містить низку функцій з однаковою назвою, но ця функція відповідальна за виконання функціональності при тій чи іншій ігровій події. Її унікальність визначається не іменем функції, а параметрами, які вона приймає, а саме – певні події ігрового процесу.

`void onEvent(ResetBackgroundEvent event)` – обробник події встановлення потрібного заднього фону.

`void onEvent(ThemeSelectedEvent event)` – обробник події вибору тематики ігрового процесу, що впливає на відповідний встановлений задній фон.

`void onEvent(DifficultySelectedEvent event)` – обробник подій протікання подальших операцій ігрового процесу згідно обраної складності.

`void arrangeBoard()` – первинна функція розміщення елементів поля ігрового процесу. Код наведено в лістингу 3.6. Виконує послідовно наступні операції:

- створює потрібну кількість пар карточок;
- переміщує їх для забезпечення завжди різного розміщення на полі;
- розміщення карточок на полі за допомогою низки циклічних структур.

#### Лістинг 3.5 – Код функції `arrangeBoard()`

```
BoardConfiguration boardConfiguration = mPlayingGame.boardConfiguration;
BoardArrangement boardArrangement = new BoardArrangement();
// створення пар
// result {0,1,2,...n} // n-number of tiles
List<Integer> ids = new ArrayList<Integer>();
for (int i = 0; i < boardConfiguration.numTiles; i++) {
    ids.add(i);
}
// перемішування
// result {4,10,2,39,...}
Collections.shuffle(ids);
// розміщення поля
List<String> tileImageUrls = mPlayingGame.theme.tileImageUrls;
Collections.shuffle(tileImageUrls);
boardArrangement.pairs = new HashMap<Integer, Integer>();
```

					ДП.КН 23.507.09.000 ПЗ	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дата		

```
boardArrangment.tileUrls = new HashMap<Integer, String>();
int j = 0;
for (int i = 0; i < ids.size(); i++) {
    if (i + 1 < ids.size()) {
        // {4,10}, {2,39}, ...
        boardArrangment.pairs.put(ids.get(i), ids.get(i + 1));
        // {10,4}, {39,2}, ...
        boardArrangment.pairs.put(ids.get(i + 1), ids.get(i));
        // {4,
        boardArrangment.tileUrls.put(ids.get(i), tileImageUrls.get(j));
        boardArrangment.tileUrls.put(ids.get(i + 1), tileImageUrls.get(j));
        i++;
        j++;
    }
}
mPlayingGame.boardArrangment = boardArrangment;
```

Game getActiveGame() – функція для отримання стану ігрового процесу.

Theme getSelectedTheme() – функція для отримання вибраної тематики ігрового процесу.

void setBackgroundImageView(ImageView backgroundImage) – встановлення картинки для фону гри.

Код, що не був розібраний та описаний в цьому розділі розміщено в Додатку А. Найбільш важливі класи для функціонування усього додатку було наведено в розгорнутому вигляді. Встановлено їх атрибути та функції, більшість функцій описано та наведено деякі фрагменти коду з них. Загалом цей розділ вмістив в собі пряму розробку коду додатка.

### 3.2 Використання xml розмітки для створення UI

При розробці графічного інтерфейсу використано розмітку xml з відповідними функціональними тегами для побудови скелету усіх екранів для відображення користувачу.

За кожним екраном, спливаючим вікном та елементом з особливими властивостями в інтерфейсі користувача стоїть xml документ з тегами та їх правильною структурою. Ці файли називаються в Android Studio – layout, що в прямому перекладі означає макет. Опис створених макетів для додатка:

– activity\_main – макет для головного екрану додатка, що демонструється одразу при запуску.

– board\_view – макет поля для ігрового процесу.

					ДП.КН 23.507.09.000 ПЗ	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		

- difficult\_view – макет екрану вибору складності.
- difficulty\_select\_fragment – макет окремого елемента складності гри, всього на екрані вибору складності їх створюється шість.
- game\_fragment – макет екрану, де відбувається ігровий процес.
- popup\_settings\_view – макет спливаючого вікна для екрану налаштувань.
- popup\_won\_view – макет спливаючого вікна при проходженні рівня ігрового процесу.
- theme\_select\_fragment – макет екрану користувача для вибору тематики ігрового процесу.
- tile\_view – макет для карточки ігрового поля, яких буде створено стільки скільки цього вимагає рівень складності ігрового процесу.

Для розуміння вигляду макету зсередини код наведено в лістингу 3.6.

#### Лістинг 3.6 – Код xml розмітки макету tile\_view.xml

```
<?xml version="1.0" encoding="utf-8"?>
<ui.TileView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

    <ImageView
        android:id="@+id/image"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="@drawable/tile"
        android:scaleType="centerCrop" />

    <RelativeLayout
        android:id="@+id/image_top"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="@drawable/tile" >

        <ImageView
            android:layout_width="50dp"
            android:layout_height="wrap_content"
            android:adjustViewBounds="true"
            android:layout_centerInParent="true"
            android:scaleType="centerCrop"
            android:src="@drawable/tile_back_star" />
    </RelativeLayout>

</ui.TileView>
```

Для створення макетів використано такі теги:

- ImageView – тег для розміщення зображення.

					ДП.КН 23.507.09.000 ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		



- `FrameLayout` – найпростіший вид розмітки, зазвичай пустий простір для розміщення в ньому дочірніх елементів.
- `RelativeLayout` – тег, що дозволяє позиціонувати елементи розміщені в ньому відносно інших дочірніх чи батьківських елементів.
- `TextView` – тег для розміщення тексту.
- `LinearLayout` – тег для розміщення дочірніх йому елементів горизонтально або вертикально та ніяк інакше.

					ДП.КН 23.507.09.000 ПЗ	Арк.
						48
Змн.	Арк.	№ докум.	Підпис	Дата		

## 4 ТЕСТУВАННЯ ФУНКЦІОНАЛЬНОСТІ ТА ПЕРЕВІРКА ЯКОСТІ

### 4.1 Мета та суть тестування

Щоб бачення кінцевого продукту втілювалося в життя, воно повинно працювати за планом. Дотримання вимог до продукту певною мірою є обов'язковим, оскільки це допомагає отримати бажані кінцеві результати.

Продукти завжди обслуговують користувачів певним чином, тому дуже важливо, щоб вони приносили обіцяну цінність, отже, вони повинні працювати згідно зазначених вимог, щоб забезпечити чудовий досвід роботи з користувачами. Наприклад, розробка програми включає в себе багато процесів, і тестування отримує уявлення про кожну частину – воно перевіряє, чи правильно вирівняна графіка додатків, перевіряє основну функціональність, перевіряє, чи є меню інтуїтивно зрозумілим тощо. Іноді після того, як розробники виправляють проблеми, виникають проблеми. Інша проблема може несподівано з'явитися десь в іншому місці, саме так іноді проходить тестування. Тому важливо знайти ці проблеми, які потрібно вирішити і надати користувачеві остаточну версію якісного продукту.

Сумісність пристрою також перевіряється для продукту, наприклад, сумісність програми з багатьма пристроями, що можуть мати різні версії операційної системи. Компанії, що проводять тестування, зазвичай мають багато пристроїв та інструментів, щоб охопити якомога більше сценаріїв тестування, якщо це передбачено для системи. В нашому ж випадку для такого тестування буде використовуватись емулятор та справжні пристрої на операційній системі Android.

### 4.2 Загальна перевірка розробленого додатка

Пункти якості додатка, які будуть перевірені в ході тестування:

- Простий у використанні та навігації – вміст, який використовується в додатку, має мати логічну структуру. Це має бути на всіх наданих сторінках.
- Клавіатура та сенсорний простір – потрібно переконатися, що програма

					ДП.КН 23.507.09.000 ПЗ	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

забезпечує більше сенсорного простору порівняно з клавіатурним простором. Незважаючи на це, користувачі не повинні напружуватися, коли справа доходить до друку, якщо він присутній.

– Включає відгуки користувачів – погано, коли користувачі не можуть поділитися своїм досвідом у додатку. Щоб вирішити цю проблему, додаток має містити опцію зворотного зв'язку.

– Швидке завантаження сторінки – користувачі програми, як відомо, нетерплячі, коли справа доходить до часу відповіді. Додаток має бути розроблений таким чином, щоб час завантаження сторінки становив не більше 20-22 секунди.

Цей пункт тестування передбачає ознайомлення з письмовою документацією – визначеними вимогами в перших етапах створення проекту, щоб підтвердити, що встановлені вимоги були належним чином виконані. Це має вирішальне значення для зустрічної перевірки, якщо якийсь крок або фрагмент відсутній у коді, який використовується для написання програми.

Тестування Android-дodatка проводилось вручну, як на вбудованому емуляторі середовища розробки Android Studio, так і на фізичному мобільному пристрої для фінальної версії ігрового додатка. Далі буде проведено та описано послідовність перевірки функціональності розробленої системи з наданням скріншотів. Додаток працює в горизонтальному положенні.

При запуску гри одразу відкривається головний екран (рисунок 4.1) з можливостями початку гри та зміни налаштувань.



Рисунок 4.1 – Головний екран гри

					ДП.КН 23.507.09.000 ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

Головний екран представлено з відповідним приємним для користувача заднім фоном та назвою гри. На екрані розміщені три кнопки: кнопка таблиці рекордів, яка поки що не буде доступна, кнопка для початку гри з відповідною анімовано підказкою, та зліва кнопка налаштувань. Екран налаштувань зображено на рисунку 4.2.



Рисунок 4.2 – Екран налаштувань

На екрані налаштувань відображається спливаюче вікно з кнопками для зміни стану звуку додатка та кнопки оцінки додатка, яка ще не працює. Після змінення налаштувань вони застосовуються одразу ж. Для виходу з налаштувань достатньо натиснути функціональну кнопку налаштувань пристрою, що поверне користувача до головного екрану. Екран вибору тематики ігрового процесу після натискання кнопки гри головного екрану зображено на рисунку 4.3.

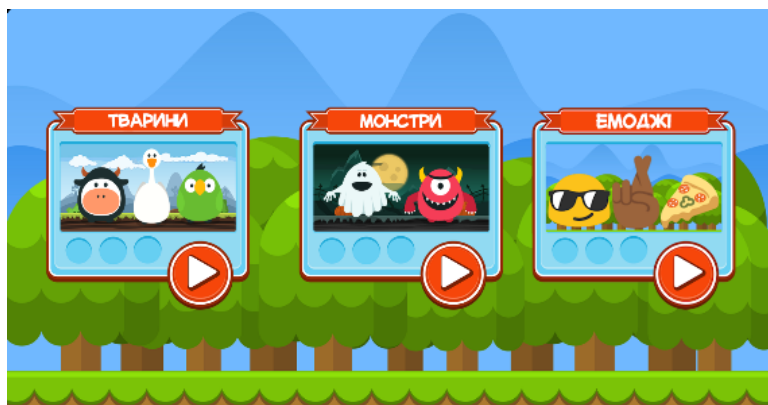


Рисунок 4.3 – Екран вибору тематики ігрового процесу

					ДП.КН 23.507.09.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		51

На цьому екрані розміщені три тематики на вибір для користувача. Цей вибір вплине на зображення, що будуть розміщені на карточках ігрового процесу. Вибравши одну з трьох тематик користувачеві відображається наступний екран вибору складності гри, що зображено на рисунку 4.4.



Рисунок 4.4 – Екран вибору тематики ігрового процесу

На екрані вибору складності відображається шість фрагментів з різною градацією складності. Кожен фрагмент складності містить рамку з надписом класифікації складності, кількість зірок, що значить результат вже пройденого рівня складності, та напис з кращим результатом проходження рівня. Якщо рівень жодного разу не пройдено на пристрої зірки результативності та час не мають відображатись. Для тестування буде перевірено функціональність Середнього рівня складності, що зображено на рисунку 4.5.

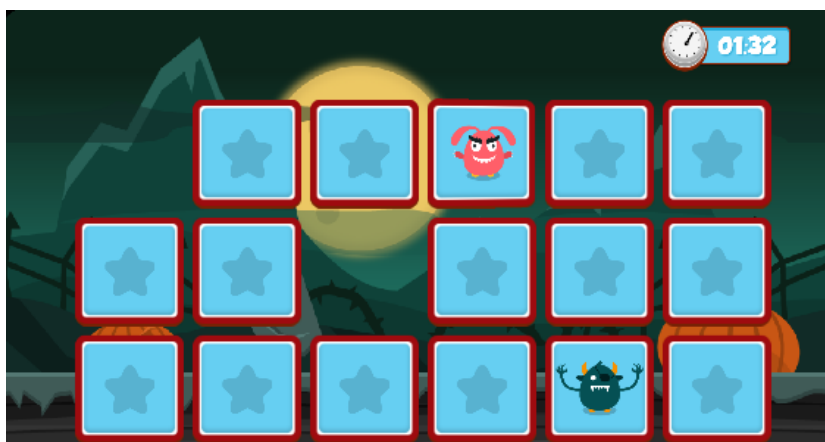


Рисунок 4.5 – Екран вибору тематики ігрового процесу

На екрані у відповідності з вибраною складністю розміщені 18 карточок. При натисканні на карточку вона відкривається, а після відкриття двох карточок, у випадку якщо вони однакові то карточки пропадають. Якщо ж карточки різні вони перевертаються назад та приховуються. Також задній фон встановлений у відповідності до вибраної раніше тематики ігрового процесу. У правому верхньому куті починається відраховуватись таймер з часом встановленим для проходження рівня відповідної складності. Коли для всіх карточок буде знайдено однакові пари та усі вони зникнуть з екрану буде відображено спливаюче вікно пройденого рівня(рисунок 4.6).



Рисунок 4.6 – Спливаюче вікно проходження рівня

На спливаючому вікні проходить анімація відображення спливання часу до нуля, розрахунок оцінки та встановлення відповідної кількості зірочок. На ньому також розміщено дві кнопки, повернення до вибору складності та кнопка переходу до наступного рівня складності.

Загалом було перевірено усі можливі випадки розвитку подій при використанні гри користувачем. Перевірено усі тематики та усі рівні складності. Пройдено усі рівня для перевірки правильності відображення елементів ігрового поля та елементів, що змінюються після проходження рівнів на інших екранах.

## 5 ТЕХНІКО-ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ

### 5.1 Огляд тенденцій цільового ринку мобільних ігрових додатків

Обсяг світового ринку мобільних ігор у 2021 році становив 93 мільйона доларів США, і очікується, що протягом 2021–2030 років він зростатиме на 12,2% у середньому. Зростаюче впровадження інноваційних технологій для створення ігор, а також збільшення використання смартфонів є важливими факторами зростання ринку.

Крім того, зростаюча популярність кіберспорту є головною рушійною силою ринку. Крім того, у міру, як технологія 5G буде використовуватися все ширше, користувачі отримають доступ до більш високих швидкостей передачі даних, таким чином розширюючи ринок мобільних ігор. Крім того, зі збільшенням наявного доходу люди можуть купувати смартфони, що призводить до збільшення кількості гравців.

Ринок ігор зростає в усьому світі, і смартфони відіграють у ньому велику роль. Ігровий бізнес набув масштабності в результаті появи мобільних ігор. Такі платформи, як Facebook та Instagram, почали пропонувати мобільні ігри, щоб використовувати привабливі ігри для вдосконалення тактики реклами.

Оскільки більшість країн у всьому світі перебувають у стані підвищеної готовності, щоб запобігти поширенню коронавірусу, впроваджуючи локдауни та накази залишатися вдома для них, відбулося значне зростання попиту на цифрові розваги, особливо на ігри для смартфонів. Великі видавці мобільних ігор помітили величезний сплеск завантажень на ринках, уражених COVID-19.

Так, Всесвітня організація охорони здоров'я (ВООЗ) рекомендувала громадянам залишатися вдома та грати в ігри під час COVID-19. Ігрова кампанія #PlayApartTogether стала вірусною завдяки підтримці ВООЗ і агресивній розробці продуктів кількома ігровими компаніями, зокрема Activision Blizzard і Glu Mobile.

					ДП.КН 23.507.09.000 ПЗ	Арк.
						54
Змн.	Арк.	№ докум.	Підпис	Дата		

Можливість: проникнення в хмару та ігри з підтримкою доповненої та віртуальної реальності

Хмарні ігри перетворюють бізнес мобільних ігор, пропонуючи кращі варіанти розвитку для постачальників мобільних ігор. Хмарні ігри були прийняті багатьма технологічними компаніями та хмарними геймерами, оскільки вони зменшують потребу в пам'яті смартфонів для розблокування гри.

Крім того, AR дозволяє користувачам взаємодіяти з аудіо- та візуальною інформацією гри в більш реальних умовах. Роблячи ще один крок вперед, VR пропонує гравцям першу точку огляду та 360-градусне бачення з захоплюючим досвідом реального світу.

Крім того, виробники смартфонів випускають ігрові телефони, які дають геймерам кращий досвід, таким чином підтримуючи зростання світового ринку мобільних ігор.

Мобільні ігри в основному завантажуються з Google Play. У 2021 році Google Play займав найбільшу ринкову частку понад 60%. Продажі смартфонів під керуванням ОС Android розширюються на ринках, що розвиваються, таких як Індія, Бразилія та Китай, що сприяє завантаженню ігор через Google Play Store.

Прийняття бізнес-моделі Freemium сприятиме зростанню ринку. Очікується, що бізнес-модель freemium матиме найвищий CAGR (Compound annual growth rate – загальний середньорічний темп росту) понад 10% протягом 2021–2030 років. Додаткова перевага здійснення внутрішньо ігрових покупок, надана безкоштовними іграми, є причиною зростання. У безкоштовній грі базова версія гри доступна абсолютно безкоштовно; незважаючи на це, користувач повинен платити за додаткові функції або підвищення рівня.

Аспект компонування безкоштовних ігор є важливим, оскільки клієнт не повинен думати, що рівень занадто складний для просування або занадто легкий для проходження. Реклама, покупки в програмі та покупки віртуальних предметів становлять значну частину доходу творців безкоштовних ігор.

					ДП.КН 23.507.09.000 ПЗ	Арк.
						55
Змн.	Арк.	№ докум.	Підпис	Дата		



Крім того, доступність численних платіжних механізмів у таких фірм дозволяє гравцям вибирати на свій смак, отже збільшуючи дохід від мобільних ігор.

## 5.2 Розрахунок витрат на проєктування

Зведені розрахунки витрат наведено у таблиці 5.1.

Таблиця 5.1 – Зведені розрахунки матеріальних витрат

№ п/п	Найменування матеріальних ресурсів	Факт. витрачено матеріалів	Ціна за одиницю, грн.	Сума, грн	Транспортні витрати (10% від суми)	Загальна сума, грн
1.	Допоміжна література	2	1200	2400	240	2640
2.	Папір (формат А4)	2	80	160	16	176
3.	Ручка кулькова	2	10	20	2	22
4.	Олівець простий	2	10	20	2	22
5.	Диски CD-R	2	15	30	3	33
6.	USB Флешка 32GB	1	330	330	33	363
7.	Зошит, 96 арк.	1	50	50	5	55
8.	Тонер для принтера	1	90	90	9	99
9.	Канцелярські маркери (червоний, зелений)	2	20	40	4	44
Разом						3454,00

З кожним роком сучасні технології розвиваються, розробка мобільних додатків також не стоїть на місці. Їх реалізація сприяє покращенню комунікабельності, якості спілкування, вирішення різноманітних складних завдань. Хорошим професіоналам це приносить досить високий прибуток.

Заробітна плата учасників проєктування проєкту наведено у таблиці 5.2.

Таблиця 5.2 – Розрахунок заробітної плати проєктувальників

№ п/п	Посада виконавця	Оклад, грн/міс	Відрахування, грн/міс	Кількість		Сума з/п, грн.
				чол.	місяців	
1	Технік-програміст	5200	950,5	2	6	4249,5
2	Тестувальник	4600	910	1	6	3690
Усього зарплати:						175122

Загальний кошторис витрат протягом усього проєктування та по його завершенні наведено у таблиці 5.3.

Таблиця 5.3 – Кошторис витрат

Найменування статей витрат	Сума, грн	Обґрунтування
1. Зарплата проєктувальників.	175122	При розробці мобільного додатку потрібно розробити великий функціонал і відповідно протестувати його
2. Відрахування на соціальні потреби.	38526	22% від суми зарплат
3. Контрагентські роботи і послуги.	480	Прибирання приміщення
4. Витрати на відрядження.	870	Похід усієї команди в кінотеатр
5. Інші прямі витрати.	3454	Придбання канцелярських товарів, паперу, маркерів.
6. Усього прямих витрат.	218452	$175122+38526+480+870+3454=218452$
7. Накладні витрати.	4400	25% - витрати на опалення, електроенергію, купівлю комплектуючих для ремонту комп'ютерів
8. Планові накопичення.	3670	Преміювання працівників за хорошу роботу та якісне виконання поставлених задач (20%).
9. Усього, кошторисна вартість проєкту.	226 522	$218 452+4400+3670= 226 522$
10. Податок на додану вартість.	45353,6	20% від усього кошторису
11. Загалом, договірна ціна розробки З/П.	271 875	$226 522+45353= 271 875$

### 5.3 Аналіз та вибір бізнес-моделі

Згідно усіх минулих досліджень, як ринку мобільних ігор так і предметної області додатка, що фактично повністю розроблено точно можна сказати цінність усього процесу для набуття необхідного надалі досвіду. Однак для правильного наповнення техніко-економічного розділу додаток буде розглянуто з комерційної сторони, як проєкт, що може нести прибуток.

Буде описано два можливих сценарії, які можуть існувати разом або окремо. Перший сценарій це бізнес модель Freemium, а інший монетизація додатка через вбудовану рекламу.

Важко змусити когось віддати свої гроші за продукт чи послугу, які вони ніколи раніше не пробували. Ось чому безкоштовні пробні версії є популярним способом познайомити когось зі своїми послугами або ігровим додатком. Існує кращий спосіб залучати нових клієнтів, хоча це навіть привабливіше для них – бізнес-модель freemium.

«Freemium» – це термін, який був створений на початку 2000-х років для опису певного способу ведення бізнесу. Це слово є сумішшю «безкоштовно» та «преміум», що вказує на два рівні, які складають цю бізнес-модель.

У більшості успішних бізнес-моделей freemium існує чітка різниця між двома пропозиціями. З безкоштовним планом користувач має доступ до основних функцій інструменту чи служби, а також, можливо, до однієї або двох додаткових функцій, які заохотять його досліджувати далі.

Freemium бізнес-модель особливо популярна у світі. У більшості компаній у цьому просторі конкуренція є жорсткою – це означає, що розробник хоче залучити якомога більше потенційних клієнтів. Конкурентоспроможна цінова стратегія допомагає, але безкоштовний план означає, що людям не доведеться перестрибувати, щоб спробувати саме наш продукт.

Безкоштовна бізнес-модель пропонує багато свободи, гнучкості та часу. Люди можуть ознайомитися з продуктом і перевірити, чи працює він для них безкоштовно та без обмежень у часі. З іншого боку, безкоштовна пробна версія часто надає користувачам повний пакет продуктів або послуг безкоштовно, але вона обмежена в часі.

Безкоштовні пробні версії можуть стати перешкодою для користувачів, які не мають моделі freemium. Часто потрібно буде передати компанії дані своєї картки, щоб активувати безкоштовну пробну версію продукту чи послуги, при цьому з вашої картки буде стягнуто плату за преміум-версію, якщо ви вчасно не скасуєте підписку.

					ДП.КН 23.507.09.000 ПЗ	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		

Додавання безкоштовного продукту до бізнес-моделі може принести багато переваг. Ось деякі з основних позитивних моментів переходу на безкоштовну бізнес-модель або запровадження безкоштовного рівня до наявної лінії.

Простіше бета-тестування. Завдяки безкоштовній бізнес-моделі легше створити потужну базу користувачів, ніж зосереджуватися лише на платних планах. З більшою кількістю користувачів легше знайти людей, які готові зареєструватися як бета-тестери нових функцій або вдосконалень.

Можна шукати клієнтів, які зацікавлені стати тестувальниками, і пропонувати заохочення або винагороду як подяку за витрачений час, як-от безкоштовне оновлення платного плану на кілька місяців або ранній доступ до нових функцій.

Користувачам буде корисно відчувати те, що вони можуть придбати пізніше, а у розробника є захоплена група людей, готових випробувати та оцінити нові преміум-функції. Вигляд моделі freemium зображено на рисунку 5.1.

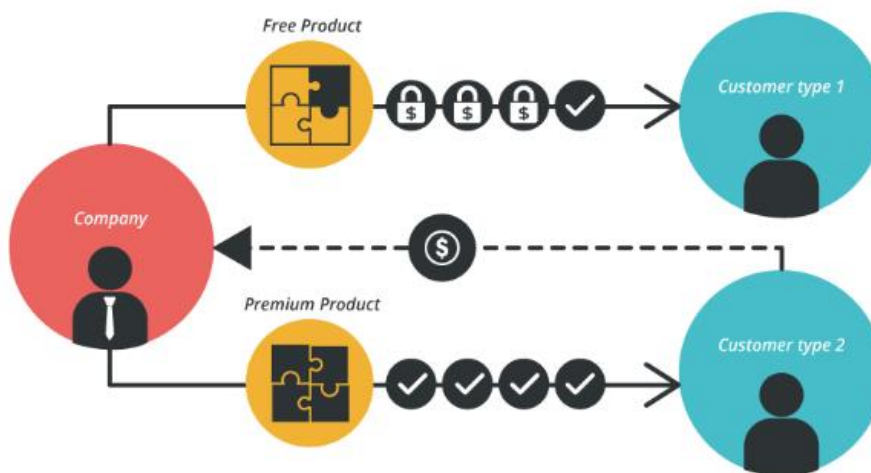


Рисунок 5.1 – Діаграма моделі freemium

Кількість активних користувачів. Завдяки безкоштовній моделі немає потреби користувачам додавати нові щомісячні витрати до свого списку. Їм не

потрібно платити щомісяця за користування послугою або скасовувати, якщо вони не можуть знайти кошти або не отримують схвалення від керівництва. Натомість клієнтська база може зробити продукт частиною їх звичайної рутини. Це має відобразитися на здоровій кількості активних користувачів.

Щоб визначити, скільки активних користувачів, спочатку потрібно визначити, що для означає «активний». Для більшості підприємств це просто записується, коли хтось отримує доступ до інструменту чи продукту — або робить це протягом певного періоду часу.

Перехід з безкоштовних планів на платні. Найпростіший спосіб визначити, чи успішна безкоштовна версія чи ні, — це подивитись, скільки людей перейшли на преміум-рівень. Це може бути проста статистика для запису та моніторингу, але вона потужна, оскільки це пряма та визначена дія в результаті наявності безкоштовного рівня.

Відстежувати конверсії легко — просто потрібно переглянути дані за попередній місяць, щоб дізнатися, скільки користувачів оновили свій план за цей період.

Порівняти результати з місяцями, коли були доступні лише платні плани, і подивіться, як це вплинуло на дохід. Якщо модель freemium працює, то коефіцієнт конверсії має бути здоровим, і можна буде побачити позитивне відображення у цифрах доходу.

Яскравий приклад успішної freemium моделі це такі застосунки:

– WordPress спрощує людям виносити свій внесок в Інтернет завдяки безкоштовному плану, який дозволяє будь-кому створити простий веб-сайт за мить. Однак, щоб розблокувати кращі функції та покращити свій досвід, їм потрібно буде підписатися на платний план.

Розуміючи, наскільки розчаровує реклама, користувачі можуть видалити рекламу WordPress.com, оформивши план преміум-рівня. Мало того, вони отримують безкоштовний домен на рік — обов'язковий атрибут для будь-якого серйозного професіонала. Крім першого платного рівня, плани масштабуються

					ДП.КН 23.507.09.000 ПЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

залежно від функцій і потреб. Цікавою особливістю є те, що вони виділили кнопку реєстрації для плану Premium іншим кольором.

– Slack є одним із найпопулярніших інструментів обміну повідомленнями для бізнесу, і його безкоштовний план є однією з причин цього. Можна налаштувати робочий простір на Slack і запропонувати всім спілкуватися разом за мить, не отримуючи схвалення для нового платного інструменту та не проходячи через тривалий процес.

Цей популярний інструмент обміну повідомленнями має безкоштовний рівень і три преміум-рівня вище. Безкоштовний тарифний план має всі основні функції, але кількість попередніх повідомлень, до яких ви можете отримати доступ, обмежена. Це розумний спосіб заохотити людей до оновлення — коли вони досягнуть ліміту в 10 000 повідомлень, вони не захочуть раптом пропустити попередні розмови. Тому вони, ймовірно, перейдуть на платний план.

					ДП.КН 23.507.09.000 ПЗ	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дата		

## ВИСНОВКИ

В результаті дипломного проектування та розробки згідно визначених вимог було успішно досягнуто генеральної цілі – створення повноцінної готової до використання мобільної гри для операційної системи Android. Звичайно повз генеральну ціль, яка не може існувати без виконання менших цілей, було виконано багато складних теоретичних та практичних задач, які були попередньо визначені.

Визначення цих самих вимог також вважається однією з задач дипломного проектування. Отже спочатку було проведено аналіз подібних мобільних ігор у вільному доступі та на його основі проводилась подальша робота над проектом.

Встановлено функціональні вимоги та відображено майбутню структуру додатка за допомогою діаграм UML, а саме:

- Діаграма варіантів використання, що дозволила встановити потрібних акторів системи та визначити базовий функціонал, який потрібно розробити.
- Діаграма класів, що встановила майбутній розподіл класів системи та їх взаємозв'язки, які будуть наповнені програмним кодом.
- Діаграма послідовності, яка візуалізує майбутню функціональну роботу додатка.

Після проектування та вибору архітектури з методологією розробки виконано написання коду та усіх потрібних функцій всередині визначених класів.

Розроблений ігровий додаток було протестовано на емуляторі всередині Android Studio в процесі написання коду. Після повної розробки готовий продукт також було повторно протестовано на реальному мобільному пристрої, що дозволило повністю виключити будь-які несправності.

Результатом усіх цих віх цілого дипломного проекту і стало досягнення генеральної цілі – створення ігрового мобільного застосунку «CardMatch» з використанням методологію розробки RUP.

					ДП.КН 23.507.09.000 ПЗ	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дата		

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Майкл Бардетт. Android App Development for Dummies. М.: USA, 2012.
2. Білл Філіпс, Кріс Стюарт, Брайан Харді. Android Programming: The Big Nerd Ranch Guide. М.: Big Nerd Ranch, 2020.
3. Рето Майер. Android 4. Програмування застосунків для планшетних комп'ютерів і смартфонів. М.: ЄКСМО, 2013. 816 с.
4. Ендрю Ханта і Девіда Томаса. The Pragmatic Programmer. М.: Addison-Wesley Professional, 1999.
5. Android. *Вікіпедія* - вільна енциклопедія: веб-сайт. URL: <https://ua.wikipedia.org/wiki/Android> (дата звернення: 24.12.2022).
6. Android\_Studio. *Вікіпедія* - вільна енциклопедія: веб-сайт. URL: [https://ua.wikipedia.org/wiki/Android\\_Studio](https://ua.wikipedia.org/wiki/Android_Studio) (дата звернення: 24.12.2022).
7. Java. *Вікіпедія* - вільна енциклопедія: веб-сайт. URL: <https://uk.wikipedia.org/wiki/Java> (дата звернення: 24.12.2022).
8. Документація Git. *Git*: веб-сайт. URL: <https://git-scm.com> (дата звернення: 24.12.2022).
9. Документація Realm. *Realm*: веб-сайт. URL: <https://realm.io> (дата звернення: 24.12.2022).
10. Документація. Java *Java*: веб-сайт. URL: <https://google.github.io/java> (дата звернення: 24.12.2022).

					ДП.КН 23.507.09.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		63



# ДОДАТКИ

## Додаток А

### Програмний код

#### Лістинг А1 – Код класу EventBus

```
public class EventBus {

    private Handler mHandler;
    private static EventBus mInstance = null;
    private final Map<String, List<EventObserver>> events =
Collections.synchronizedMap(new HashMap<String, List<EventObserver>>());
    private Object obj = new Object();

    private EventBus() {
        mHandler = new Handler();
    }

    public static EventBus getInstance() {
        if (mInstance == null) {
            mInstance = new EventBus();
        }
        return mInstance;
    }

    synchronized public void listen(String eventType, EventObserver
eventObserver) {
        List<EventObserver> observers = events.get(eventType);
        if (observers == null) {
            observers = Collections.synchronizedList(new
ArrayList<EventObserver>());
        }
        observers.add(eventObserver);
        events.put(eventType, observers);
    }

    synchronized public void unlisten(String eventType, EventObserver
eventObserver) {
        List<EventObserver> observers = events.get(eventType);
        if (observers != null) {
            observers.remove(eventObserver);
        }
    }

    public void notify(Event event) {
        synchronized (obj) {
            List<EventObserver> observers = events.get(event.getType());
            if (observers != null) {
                for (EventObserver observer : observers) {
                    AbstractEvent abstractEvent = (AbstractEvent) event;
                    abstractEvent.fire(observer);
                }
            }
        }
    }
}
```

					ДП.КН 23.507.09.000 ПЗ	Арк.
						64
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        public void notify(final Event event, long delay) {
            mHandler.postDelayed(new Runnable() {
                @Override
                public void run() {
                    EventBus.this.notify(event);
                }
            }, delay);
        }
    }
}

```

## Лістинг А2 – Код класу EventObserverAdapter

```

public class EventObserverAdapter implements EventObserver {

    public void onEvent(FlipCardEvent event) {
        throw new UnsupportedOperationException();
    }

    @Override
    public void onEvent(DifficultySelectedEvent event) {
        throw new UnsupportedOperationException();
    }

    @Override
    public void onEvent(HidePairCardsEvent event) {
        throw new UnsupportedOperationException();
    }

    @Override
    public void onEvent(FlipDownCardsEvent event) {
        throw new UnsupportedOperationException();
    }

    @Override
    public void onEvent(StartEvent event) {
        throw new UnsupportedOperationException();
    }

    @Override
    public void onEvent(ThemeSelectedEvent event) {
        throw new UnsupportedOperationException();
    }

    @Override
    public void onEvent(GameWonEvent event) {
        throw new UnsupportedOperationException();
    }

    @Override
    public void onEvent(BackGameEvent event) {
        throw new UnsupportedOperationException();
    }

    @Override
    public void onEvent(NextGameEvent event) {
        throw new UnsupportedOperationException();
    }

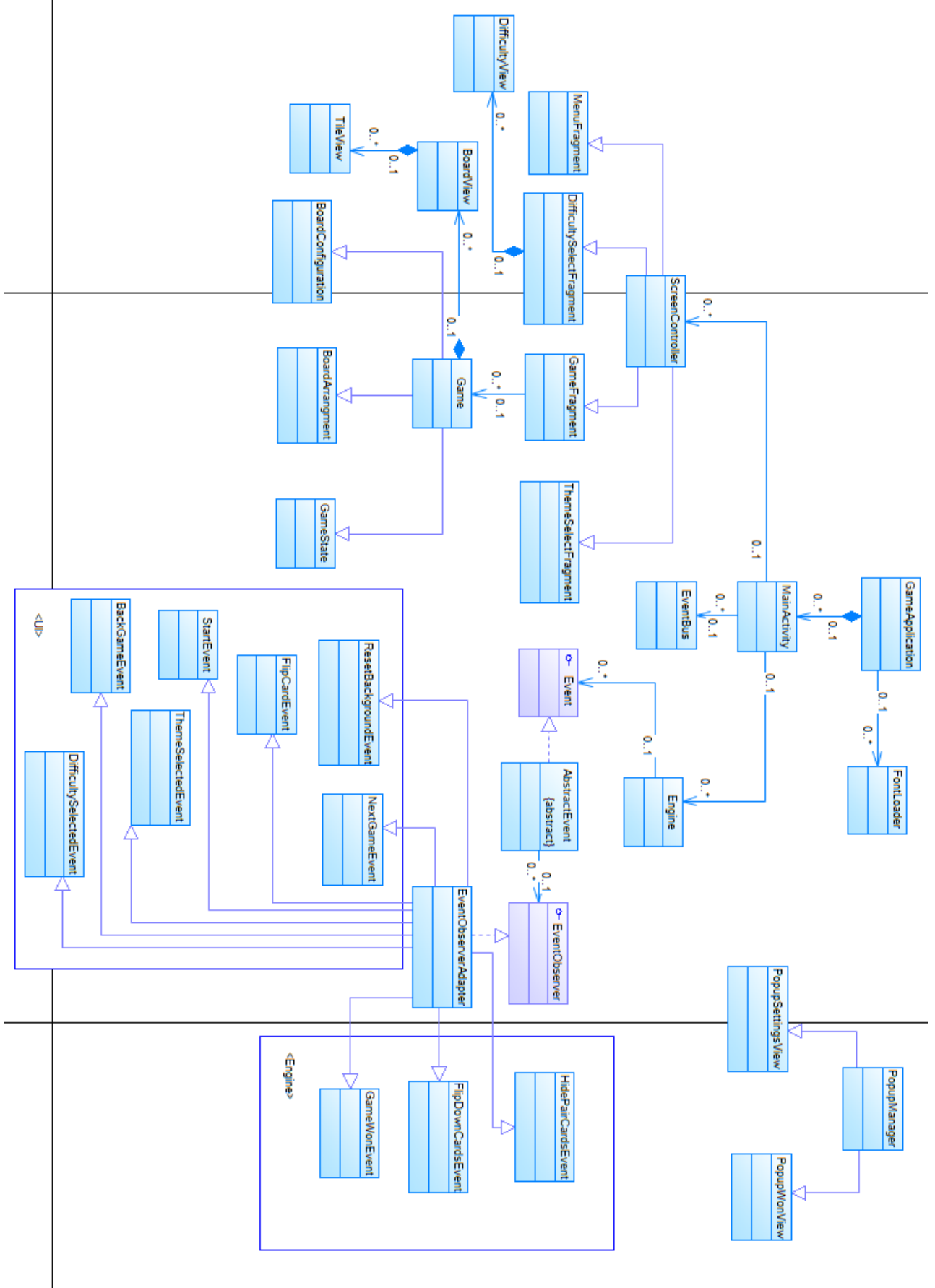
    @Override
    public void onEvent(ResetBackgroundEvent event) {
        throw new UnsupportedOperationException();
    }
}

```

					ДП.КН 23.507.09.000 ПЗ	Арк.
						65
Змн.	Арк.	№ докум.	Підпис	Дата		

# Додаток Б

## Діаграма класів



## РЕЦЕНЗІЯ

на дипломний проєкт

студента відділення комп'ютерних технологій

Галицького фахового коледжу імені В'ячеслава Чорновола

студента IV курсу групи КН-41

Зьомка Володимира Анрійовича

Спеціальність 122 „Комп'ютерні науки”

Обсяг дипломного проєкту: 66 стор.

Кількість аркушів креслень 0 арк.

Кількість сторінок записки: 69 стор.

Тема: «Ігровий мобільний застосунок "CardMatch" з використанням методологію розробки RUP»

Актуальність теми полягає у широкому використанні мобільних застосунків та виокремлення переваг операційної системи Android не лише для кожного користувача окремо, а як величезної індустрії в сфері розвитку сучасних технологій.

Практична або теоретична цінність опрацьованих питань: використано середовище розробки Android Studio, весь графічний інтерфейс для користувача спроектований та побудований згідно кращих практик UI/UX дизайну. Для реалізації програмного коду функціональності використано мову програмування Java. Проведено аналіз подібних мобільних ігор у вільному доступі та на його основі проводилась подальша робота над проєктом. Встановлено функціональні вимоги та відображено майбутню структуру додатка за допомогою діаграм UML. Після проєктування та вибору архітектури з методологією розробки виконано написання коду та усіх потрібних функцій всередині визначених класів.

Недоліки роботи: в роботі варто відобразити скріни програмного засобу на етапі розробки.

Загальний висновок: робота відповідає вимогам до дипломних проєктів освітньо кваліфікаційного рівня «молодший спеціаліст» зі спеціальності 122 «Комп'ютерні науки» та заслуговує оцінку «задовільно»

Рецензент

 Н.Г.Гавришків

« 23 » 06 2023 р.

**ВІДГУК**  
на дипломний проєкт  
студента відділення комп'ютерних технологій  
Галицького фахового коледжу імені В'ячеслава Чорновола

студента IV курсу групи КН-41  
Зьомка Володимира Андрійовича \_\_\_\_\_  
(прізвище та ініціали)

Спеціальність 122 «Комп'ютерні науки»

Керівник ДП \_\_Кузик Василь Миколайович \_\_\_\_\_

Тема: "Ігровий мобільний застосунок "CardMatch" з використанням методології розробки RUP"

\_\_\_\_\_

1. Загальна характеристика студента під час написання ДП: \_\_студент проявив себе як фахівець, який володіє засобами проєктування та створення андроїд-додатків

\_\_\_\_\_

2. Практична або теоретична цінність опрацьованих питань: \_\_\_\_\_  
полягає в програмній реалізації інформаційної системи – андроїд додатку ігрової програми. Розроблена система підвищує якісний рівень ігрових застосунків для дітей відповідного вікового періоду. Розвиває творче мислення, увагу. \_\_\_\_\_

\_\_\_\_\_

3. Недоліки роботи: \_\_\_\_\_ до недоліків можна віднести слабе посилання на джерела використаної літератури \_\_\_\_\_

\_\_\_\_\_

4. Загальний висновок \_\_дипломний проєкт виконаний згідно вимог і заслуговує на позитивну оцінку \_\_\_\_\_

\_\_\_\_\_

Керівник дипломного проєкту

  
\_\_\_\_\_

Кузик Василь Миколайович, викладач ЦК Інформатики та комп'ютерних дисциплін

23 червня 2023р.

\_\_\_\_\_

Ім'я користувача:  
Василь Кузик

Дата перевірки:  
14.06.2023 13:27:57 EEST

Дата звіту:  
14.06.2023 13:31:47 EEST

ID перевірки:  
1015600848

Тип перевірки:  
Doc vs Internet + Library

ID користувача:  
100012366

Назва документа: Зьомко

Кількість сторінок: 57 Кількість слів: 10945 Кількість символів: 83419 Розмір файлу: 2.42 MB ID файлу: 1015249180

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

4.9%  
Схожість

Найбільша схожість: 2.59% з джерелом з Бібліотеки (ID файлу: 1011429600)

2.8% Джерела з Інтернету

174

Сторінка 59

3.32% Джерела з Бібліотеки

39

Сторінка 60

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%  
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Підозріле форматування

9  
сторінок