

Хмельницький національний університет  
Факультет інформаційних технологій  
Кафедра інженерії програмного забезпечення

## КВАЛІФІКАЦІЙНА РОБОТА

Кросплатформна гра для мобільних платформ у жанрі «Аркада» на Unity  
Назва теми

Рівень вищої освіти Перший (бакалаврський)  
Галузь знань 12 «Інформаційні технології»  
Спеціальність 121 «Інженерія програмного забезпечення»  
Освітня програма Освітньо-професійна програма «Інженерія програмного  
забезпечення»

Шифр КвРПЗ 190127.19.03.ПЗ

Виконав студент IV курсу група ПЗ-19-1  М. О. Василенко  
Підпис Ініціали, прізвище

Керівник канд. пед. наук, доцент  Н. І. Праворська  
Науковий ступінь, звання Ініціали, прізвище

Нормоконтролер канд. техн. наук, доцент  О. М. Яшина  
Підпис Ініціали, прізвище

**До захисту допускаю:**  
Завідувач кафедри інженерії програмного забезпечення  Л. П. Бедратюк  
Підпис Ініціали, прізвище

6 червня 2023 р.

Хмельницький 2023

# ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій  
Кафедра Інженерії програмного забезпечення  
Рівень вищої освіти Перший (бакалаврський)  
Галузь знань 12 «Інформаційні технології»  
Спеціальність 121 «Інженерія програмного забезпечення»  
Освітня програма Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ  
Завідувач кафедри 173  
Л. П. Бедратюк  
05 02 2023 р.

## ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Василенко Максим Олексійович

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Кроссплатформна гра для мобільних платформ у жанрі «Аркада» на Unity

Керівник кваліфікаційної роботи канд. пед. наук, доцент Праворська Н. І.

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 01.03.2023 р. № 5

2. Строк подання студентом проекту (роботи) на кафедру 01.06.2023 р.

3. Вихідні дані до проекту (роботи) Матеріали переддипломної практики

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

1.1 Аналіз предметної області 1.2 Аналіз наявного програмно-технічного забезпечення 1.3 Постановка задач та вимог до програмного продукту 2 Проектування програмного забезпечення 2.1 Проектування варіантів використання 2.2 Проектування інтерфейсу користувача 2.3 Реалізація модулів системи 3 Програмна реалізація 3.1 Підключення сервісів до проекту 3.2 Реалізація бази даних 3.3 Реалізація ігрового процесу та його балансування 3.4 Забезпечення безпеки бази даних 3.5 Тестування 3.6 Інструкція користувача 3.7 Вимоги до апаратно-програмних засобів

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

Діаграма варіантів використання, схема бази даних, діаграма життєвого циклу програми

## 6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	О. М. Яшина, доцент, к.т.н.	8.05.23	6.06.23
Антиплагіат	І.В. Гурман, доцент, к.т.н.	5.06.2023	5.06.2023

7. Дата видачі завдання « 02 » січня 2023 р.

## КАЛЕНДАРНИЙ ПЛАН

Назва етапів (розділів) дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1 Ознайомлення з тематикою дипломного проектування, визначення та узгодження індивідуальних тем кваліфікаційних робіт (КвР)	01.12 – 30.12.2023	
2 Збір матеріалу за темою КвР; дослідження предметної області, в якій планується використання програмного забезпечення (ПЗ), визначення задач та вимог, розробка технічного завдання	02.01 – 31.01.2023	
3 Проектування програмного забезпечення	01.02 – 28.02.2023	
4 Програмна реалізація з використанням відповідних засобів розробки	01.03 – 10.04.2023	
5 Тестування програмного забезпечення	11.04 – 30.04.2023	
6 Написання вступу, загальних висновків, оформлення переліку джерел посилання та додатків. Оформлення пояснювальної записки КвР згідно вимог	01.05 – 25.05.2023	
7 Попередній захист КвР	Травень 2023 (згідно графіка)	
8 Перевірка КвР на плагіат, нормоконтроль, отримання відгуків, рецензій та інших супровідних документів. Брошування (зшиття) пояснювальної записки.	26.05 – 30.05.2023	
9 Здача КвР на кафедрі; підготовка КвР для розміщення у репозитарії ХНУ; підготовка до захисту та захист КвР	з 01.06.2023	

Студент

  
 Підпис

Василенко М.О.

Ініціали, прізвище

Керівник проекту

  
 Підпис

Праворська Н. І.

Ініціали, прізвище

## АНОТАЦІЯ

Тема кваліфікаційної роботи «Кроссплатформна гра для мобільних платформ у жанрі «Аркада» на Unity».

Автор проекту: Василенко Максим Олексійович

Керівник проекту: Праворська Наталія Іванівна

Пояснювальна записка: 56 с., 27 рис., 3 табл., 4 дод., 43 джерел.

Графічна частина: 3 креслення.

Об'єктом дослідження є сфера розповсюдження та експлуатації ігрових додатків, що займається реалізацією різних соціальних можливостей через мережу Інтернет

Метою роботи є створення гри в жанрі Аркада в якій за допомогою використання сервісів гравці змагатимуться між собою за найкращий результат.

У кваліфікаційній роботі проведено аналіз предметної області та її інформаційного забезпечення, визначені вимоги до ігрового додатку, розроблена загальна архітектура додатку, спроектована структура бази даних для нього та сучасний інтерфейс.

Для розробки програмної системи використано мову програмування C#, та програмне забезпечення Unity з використанням сервісів Admob, Firebase, Google Play Services.

У результаті проектування здійснена програмна реалізація додатку та його публікація в магазин Google Play Store а також інтегрування бази даних за допомогою якої відбувається управління грою.

30.05.2023

Дата



Підпис

## ВІДОМІСТЬ ДОКУМЕНТІВ

№ рядка	Формат	Позначення документа	Найменування документа	К-сть аркушів	№ екз.	Примітка
			<u>Текстові документи</u>			
1	A4	КвРІПЗ.190129.01.03.ПЗ	Пояснювальна записка	56		
2	A4		Завдання на проект	1		
			проект			
3	A4		Анотація	1		
			<u>Графічні документи</u>			
4	A3	КвРІПЗ.190129.01.03.E8	Діаграма варіантів	1		
			використання			
5	A3	КвРІПЗ.190129.01.03.E8	Схема бази даних	1		
6	A3	КвРІПЗ.190129.01.03.E8	Діаграма життєвого	1		
			циклу програми			

					КвРІПЗ.190127.01.03.ВД			
Змн.	Арк.	№ докум.	Підпис	Дат	Кросплатформна гра для мобільних платформ у жанрі «Аркада»	Літ.	Арк.	Аркуше
Виконав		Василенко М.О.		6.06			1	1
Керівник		Праворська Н.І.		6.06				
Н. Контр.		Яшина О.М.		6.06				
Зав. Каф.		Бедратюк		6.06	Відомість документів	ХНУ, ІПЗ-19-1		

## ЗМІСТ

Вступ.....	8
1 Характеристика предметної області і постановка задачі .....	10
1.1 Аналіз предметної області .....	10
1.2 Аналіз наявного програмно-технічного забезпечення.....	11
1.3 Постановка задач та вимог до програмного продукту .....	15
1.4 Висновки.....	15
2 Проектування програмного забезпечення .....	20
2.1 Проектування варіантів використання.....	20
2.2 Проектування інтерфейсу користувача.....	22
2.3 Реалізація модулів системи.....	26
2.4 Висновки.....	39
3 Програмна реалізація .....	40
3.1 Підключення сервісів до проекту.....	40
3.2 Реалізація бази даних .....	50
3.3 Реалізація ігрового процесу та його балансування .....	53
3.4 Забезпечення безпеки даних.....	56
3.5 Тестування .....	58
3.6 Інструкція користувача .....	60
3.7 Вимоги до апаратно-програмних засобів.....	61
3.8 Висновки.....	62
Висновки.....	63
Перелік джерел посилання .....	64
Додаток А Технічне завдання.....	68
Додаток Б Програмний код основних модулів.....	71
Додаток В Презентаційні матеріали.....	73
Додаток Г Графічна частина.....	79

КвРПЗ.190129.01.03.ПЗ				
Змн.	Арк.	№ докум.	Підпис	Дата
Виконав		Василенко М.О	<i>МВ</i>	6.06
Керівник		Праворська Н.І.	<i>Н.І. Праворська</i>	6.06
Н. контр.		Яшина О.М.	<i>О.М. Яшина</i>	6.06
Зав. каф.		Бедраюк Л. П.	<i>Л.П. Бедраюк</i>	6.06
Кросплатформна гра для мобільних платформ у жанрі «Аркада»				
		Літ.	Арк.	Аркушів
			4	56
ХНУ, ІПЗ-19-1				

## Вступ

Ігри вже досить довгий час тримаються у фокусі актуальності, завдяки великому розвитку Game Development сфери, яка заохочує неймовірну кількість нових інвесторів та спеціалістів, що сподіваються втілити свої дитячі мрії, або надреалістичні фантазії. Відеоігри мають велике культурне значення, і є досить факторів чому вони і надалі будуть розвиватись та йти в ногу з реаліями сьогодення.

Перша головна причина актуальності – технічний прогрес тісно пов'язаний з цією сферою, та кожен новий стрибок дає абсолютно інакший вектор розвитку спеціалістам. Цифрова власність та віртуальна реальність давно популярна тема, але зараз вона стає все більш реальною. Можна згадати проєкт «Sandbox», що в цілому побудований на тому, що гравці заробляють токени, які можна поміняти на справжні гроші, тобто виходить у комп'ютерній грі справжня економіка.

Ще важлива тема – штучний інтелект. Не так давно Chat GPT змінив наше життя, тому можна бути впевненими в тому, що цей феномен знайде собі роль в ігровій індустрії. Вже зараз спеціалісти працюють над тим, щоб персонажі РПГ проєктів підтримували діалог з гравцем, як справжні люди. В кінці кінців це все і справді може перерости в щось неймовірне, тому нейронні мережі це досить сильний аргумент.

Якщо ж говорити про конкретну реальність – кожен творчий проєкт розробники намагаються випускати в багатьох медіа формах, та ігри є великою часткою у сфері популяризації контенту. Споживачі люблять не тільки дивитися на героїв, слухати про них, читати, з'явилась потреба стати ними, відчутти їх на своїй шкурі.

Важливим феноменом у геймдизайні є баланс часу та якості з головною ідеєю. Досить велика кількість хороших проєктів, з справді реалістичною графікою, геніально прописаним сюжетом не знайшли свого місця серед

					КВРІПЗ.190129.19.03.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		5

найкращих представників. Актуальність ідеї – перша та головна задача. Для пошуку тем та тегів для гри, можна звернутися до ЗМІ.

Об'єктом дослідження є структура організації ігрових додатків, які використовують для популяризації контенту, також для рекламної монетизації. Ще одним важливим аспектом у дослідженні є сервіси, які використовуються для обслуговування ігрових додатків. Firebase популярний сервіс, який виконує багато важливих функцій у проектуванні: авторизація, підключення бази даних, обслуговування DeepLink, аналітика, і тд. Admob сервіс, який дозволяє створювати та налаштовувати рекламні блоки в іграх, для заробітку на них. Також є можливість з'єднати та синхронізувати Admob Console, Firebase Console, Google Play Console, Facebook Console.

Предметом дослідження даної кваліфікаційної роботи є процес розробки гри у жанрі аркада на рушієві Unity, його монетизації та аналіз конверсій на основі даних магазину Google Play та сервісів Firebase.

Головною метою при виконанні даної кваліфікаційної роботи є розробка ідеологічно актуальної гри для українців з монетизацією та аналітикою.

Для досягнення мети проекту необхідно вирішити наступні завдання:

- виконати аналіз схожих проектів у конкурентів;
- визначити актуальну ідеологію для стилізації та оформлення гри;
- знайти оптимальний шлях для реалізації жанру аркада;
- визначити вимоги для розроблюваного застосунку;
- розробити діаграми для майбутньої побудови застосунку;
- розробити балансований геймплей;
- підключити сервіси Firebase, Admob;
- створити механізм авторизації;
- створити систему он-лайн рекордів;
- опублікувати додаток у магазині.

					КВРІПЗ.190129.19.03.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		6

# 1 Характеристика предметної області і постановка задачі

## 1.1 Аналіз предметної області

Зараз, завдяки доступній технології, майже кожен може створити свою власну гру. Нинішні ігрові двигуни мають широкий спектр інструментів для розробки, деякі з яких не вимагають знання мов програмування. Хоча процес створення ігор залишається складним і тривалим, наявність доступних ігрових двигунів значно полегшує його. Це має значний вплив на галузь ігор і відеоігор. Термін "ігровий двигун" з'явився в середині 1990-х років у контексті комп'ютерних ігор жанру шутер від першої особи, схожих на популярну на той час Doom. Під час створення перших ігор розробники виявили, що у проектах багато спільного, зокрема, робота з графікою та розрахунок фізики, тому функціонал об'єднували у спеціальні бібліотеки.

Згодом ці бібліотеки переросли у справжні програмні модулі, які зібралися разом і утворили ігровий двигун. До середини першого десятиліття 21 століття, ігрові двигуни були дорогими для компаній, тому на ринку не було багато людей, які грали в ігри. Однак згодом з'явилася багато програм для розробки за доступними цінами, які могли дозволити собі невеликі інді-команди розробників. Деякі компанії, такі як EPIC GAMES, також знизили вартість своїх ігрових двигунів. Нові конкуренти, такі як UNITY 3D, TORQUE 3D та інші, з'явилися на ринковому полі. У результаті утворилася промисловість зі створення ігрових двигунів як окремих інструментів розробки.

Застосування ігрових двигунів значно спрощує процес розробки ігор і зробило його доступнішим для більшої кількості людей. Зараз, завдяки розвитку ігрових двигунів, навіть самотній розробник може створити гру, яка буде містити високоякісну графіку, реалістичну фізику, аудіо-ефекти та багато іншого.

Однією з ключових переваг використання ігрових двигунів є те, що вони дозволяють розробникам сконцентруватися на творчому процесі, не втрачаючи час на написання коду для вирішення рутинних задач. Крім того, ігрові двигуни

					КВРІПЗ.190129.19.03.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		7

забезпечують стандартизацію ігрових систем, що дозволяє розробникам створювати ігри швидше та ефективніше. З появою доступних цін на ігрові двигуни, на ринку з'явилася велика кількість інді-студій, що розробляють свої ігри. Це сприяло диверсифікації ігрового ринку та зростанню кількості незалежних ігор. На рисунку 1.1 зображено інтерфейс сучасної програми для розробки ігрових додатків.

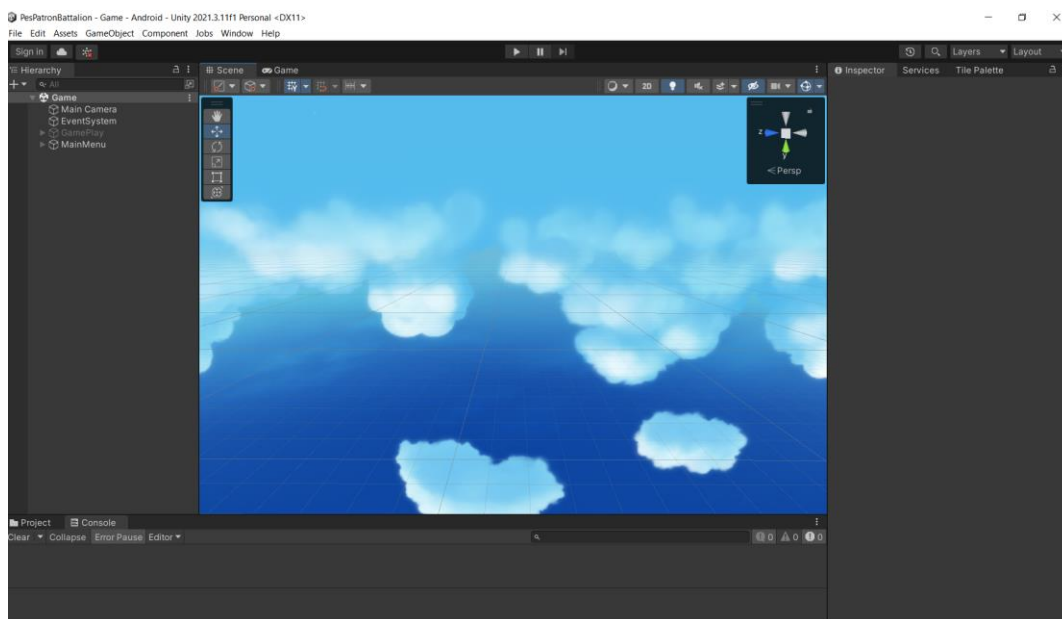


Рисунок 1.1 – Інтерфейс Unity

Такий інтерфейс дозволяє GameDev розробникам мати досить цікавий процес роботи. Такі розробники можуть займатись абсолютно будь-якою роботою пов'язаною з проектом: від створення анімацій до програмування систем та рендерингу середовища.

Підсумовуючи це все: розробка мобільної гри на будь якому ігровому движку, може видатись справді цікавим прибутковим процесом. Адже завдяки сюжетним складовим ігор люди культурно розвиваються, а завдяки цікавим ігровим механікам – отримують унікальний життєвий досвід та розвивають свій мозок. Тому розробка аркади на движку Unity є достойне поле для дослідження та розробки проектування кваліфікаційної роботи.

					КВРІПЗ.190129.19.03.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		8

## 1.2 Аналіз наявного програмно-технічного забезпечення

А зараз час для аналізу конкурентів та ПЗ, на якому вони працюють. Якщо зайти на сторінку топових аркадних ігор, ми одразу ж зрозуміємо, що є безліч піджанрів у них: платформери, стратегії, ранери, клікери, хоррори. Кожен піджанр має велике значення та свою аудиторію, тому слід його обирати уважно. Головна суть аркади в тому, що вона суперечить реалізму, та виконує суто динамічну роль. Для кращого розуміння сучасної ситуації на ринку переходимо на головну сторінку Google Play топу аркад, який зображено на рисунку 1.2.

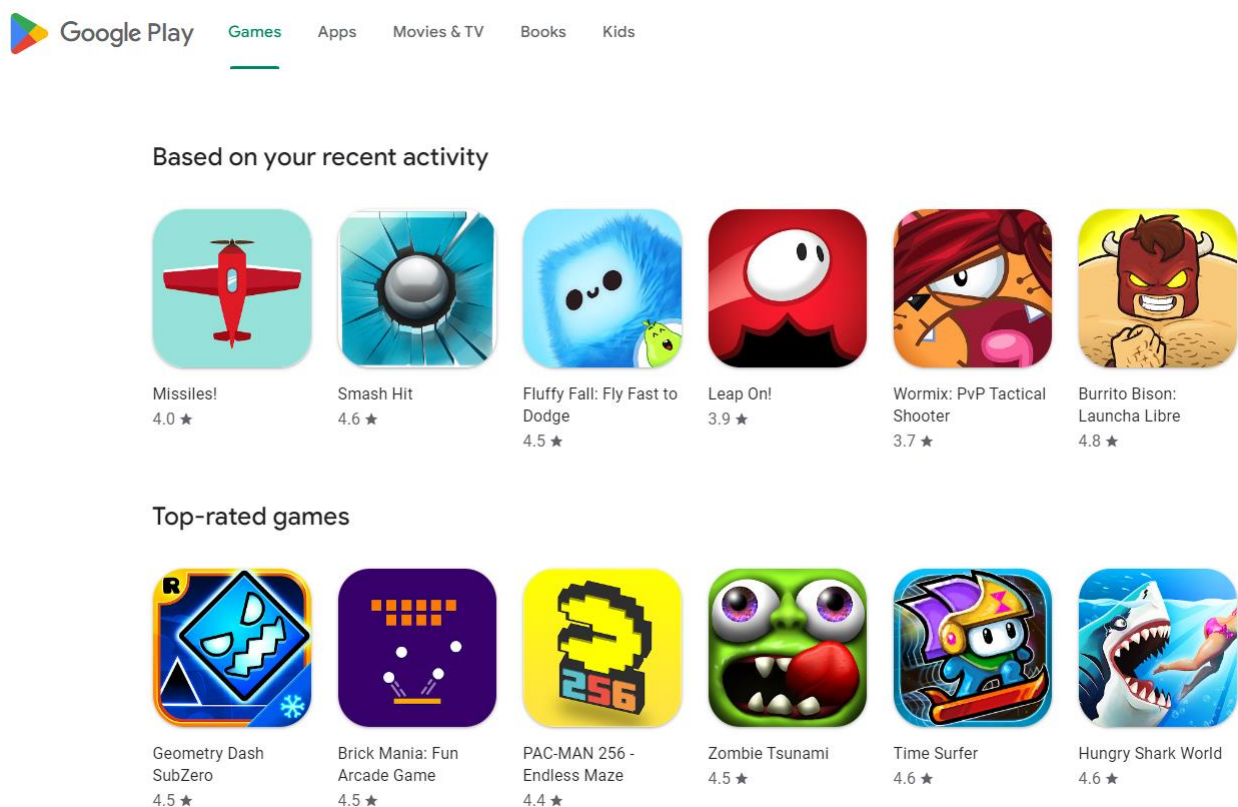


Рисунок 1.2 – Магазин Google Play

Серед популярних ігор зазначимо проекти, які з оптимально малими зусиллями зайняли високі місця в топі. Було виділено проекти: Relax Tower Defense, Make More, Cat Snack Bar. Як бачимо, ці ігри не відрізняються оригінальністю ідеї, але вони взяли, якийсь один тезис та розвили його досить

					КВРІПЗ.190129.19.03.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		9

добре, діючи за правилом «Підсилюємо сильне», і таким чином займають своє місце в рейтингу і не пропускають конкурентів.

Серед сучасних движків виділяються два конкурента Unity та Unreal Engine. У кожного з них є свої переваги та недоліки

Unity3d є сучасним і потужним крос-платформним двигуном для створення ігор та програм, який розробляється компанією Unity Technologies. Однією з головних переваг цього двигуна є вбудований ігровий пристрій, який дозволяє розробникам тестувати свої гри безпосередньо в редакторі. Крім того, Unity підтримує імпорт різних форматів моделей, що дає змогу конструювати їх у зручніших додатках та використовувати в розробці продукту.

Для написання коду Unity підтримує найбільш популярні мови програмування, такі як C# і JavaScript, а іноді використовується й Java в спеціальних випадках. Unity також є крос-платформним двигуном, що дозволяє створювати програми не тільки для комп'ютерів, але й для мобільних пристроїв, ігрових консолей та інших аксесуарів. Крім цього, Unity має велику спільноту користувачів, яка активно допомагає один одному вирішувати проблеми та ділитися досвідом. В додаток до цього, Unity надає користувачам широкі можливості налаштування та оптимізації гри, що дозволяє зменшити навантаження на систему та збільшити продуктивність.

Ще однією важливою характеристикою Unity є його зручний інтерфейс. Редактор Unity дозволяє візуально створювати об'єкти, налаштовувати їх властивості, а також додавати компоненти, які забезпечують функціонал об'єкту. Це дозволяє зосередитися на розробці гри, а не на вирішенні технічних проблем.

Unity також підтримує віртуальну реальність (VR) та доповнену реальність (AR), що дає розробникам можливість створювати ігри та програми, що використовують ці технології. Це зробило Unity одним з найпопулярніших движунів для Unity3d є потужною та популярною платформою, яка дозволяє створювати програми та експортувати їх на різні пристрої, включаючи мобільні телефони, персональні комп'ютери та приставки Nintendo. Для того, щоб

					КВРІПЗ.190129.19.03.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		10

розпочати розробку в Unity3d, необхідно мати деякий досвід у програмуванні на мовах, таких як C# та JavaScript. Однак, завдяки добре розробленому інтуїтивному інтерфейсу, навіть початківці можуть впоратися з використанням цього двигуна.

Однією з головних переваг Unity3d є крос платформа підтримка, яка охоплює не тільки різноманітні пристрої, але і всі три провідні оперативні системи. Це дозволяє розробникам збільшувати базу користувачів та залучати інвесторів.

Нарешті, варто зазначити, що Unity3d доступний у безкоштовній версії, яка містить велику кількість функціоналу. Це дає можливість новачкам випробувати цей framework та розробляти свої ігри без великих витрат. Хоча, варто відзначити, що платні версії мають більше можливостей та функцій, які можуть допомогти розробникам підвищити якість своїх продуктів. На рисунку 1.2 зображено види ліцензій Unity

	<b>Personal</b> Free Start creating with the free version of Unity <a href="#">Get started</a> Are you a student? <a href="#">Get the free Student plan</a>	<b>Plus</b> €369 /yr per seat More functionality and resources to power your projects <a href="#">Choose plan</a>	<b>Pro</b> €1,877 /yr per seat Complete solution for professionals to create and operate <a href="#">Choose plan</a>	<b>Enterprise</b> Success at scale for large organizations with ambitious goals <a href="#">Contact us</a> For large teams
<b>Create</b>				
① Unity real-time development platform	✓	✓	✓	✓
① Unity Visual Scripting	✓	✓	✓	✓
① Unity Version Control (3 users and 5GB storage)	✓	✓	✓	✓
① Splash screen customization	—	✓	✓	✓

Рисунок 1.2 – Види ліцензій Unity

У загальному, існує три версії ліцензій для рушія, розробленого компанією. Безкоштовна версія ліцензії називається Personal, але її можна використовувати тільки для особистого використання. Pro та Plus є платними версіями ліцензії.

Застосування індивідуальної роздільної здатності дозволено тільки фірмам, щорічний обіг яких не перевищує 100 тисяч доларів. Якщо ж фірма має обіг від 100 тисяч до 200 тисяч доларів, то рекомендовано використовувати версію Plus, а в інших випадках можна використовувати лише версію Pro. Усі версії ліцензій надають доступ до офіційного торгового центру, який містить у собі як моделі, так і зображення, а також цілі готові програмні модулі, які можуть значно розширити функціонал гри. Також, за допомогою торгового центру можна продавати свої збірки контенту.

Проте, подібні фішки може запропонувати й Unreal Engine. Тому було б досить доречно порівняти ці два движки за найважливішими параметрами. Порівняння движків зображено в таблиці 1.1

Таблиця 1.1 Порівняння Unity та Unreal

Специфікація	Unity	Unreal Engine
Мови програмування	Unity використовує мови C# та JavaScript	Unreal використовує мову C++
Швидкодія	Unity працює з більш високорівневим кодом, тому оптимізація має менший рівень	C++ є більш низькорівневим, тому і можливості оптимізації вищі
Платформи	Unity має дещо більше можливостей для розробки мобільних ігор, тоді як Unreal більш підходить для розробки ігор для великих консолей та ПК.	Unreal підтримує більшість популярних платформ, таких як PC, Mac, iOS, Android, Xbox, PlayStation, і т.д.
Мультиплеєр	в Unity для його створення доведеться підключати сторонні програми	Вбудований мультиплеєр
Підтримка розробників	Часто виходять нові версії, це має свої переваги та недоліки.	Навчальні матеріали по движку.

UE4 (Unreal Engine 4) ідеально підходить для швидкого створення якісного прототипу та розробки великих ігор. Він має відкритий код, але для ефективної роботи потрібні знання мови програмування C++.

Unity, з іншого боку, має трохи нижчі системні вимоги, а сам движок та проекти, створені на ньому, займають менше місця на диску. Він є найкращим вибором для створення невеликих 2D та 2.5D ігор, особливо для продуктів з сенсорним інтерфейсом.

Вибір між UE4 і Unity залежить від проекту, над яким працює студія. Якщо це мобільна гра у 2D або 2.5D, Unity буде очевидним вибором через його простоту використання. Unreal Engine не був створений спеціально для 2D-ігор, тому реалізація проекту може додати зайвої складності. Однак, якщо планується розробка симуляторного жанру, Unreal Engine буде кращим варіантом. Згідно з проектом, Unity є кращим вибором. На рисунку 1.3 показаний інтерфейс Unreal Engine, що демонструє, що сучасні движки мають подібний компонентний підхід, який візуально схожий.

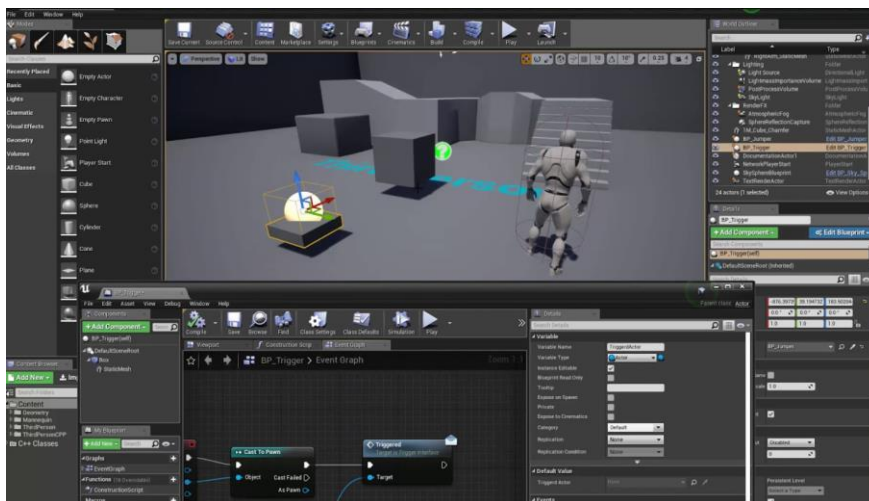


Рисунок 1.3 – Інтерфейс Unreal Engine

У цьому розділі було розглянуто два популярних сучасних движка для розробки мобільних ігор, та вирішено, що Unity підходить для виконання завдання краще, ніж Unreal Engine.

					КВРІПЗ.190129.19.03.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		13

### 1.3 Постановка задач та вимог до програмного продукту

Визначення вимог до програмного забезпечення та розробка технічного завдання Створення специфікації – важливий етап у розробці будь-якого проекту, в тому числі і проектів Unity. Специфікація – це документ, який визначає всі вимоги та умови, які повинні бути виконані для успішної реалізації проекту. Він містить усі необхідні вказівки щодо функціональності проекту, архітектури, дизайну, специфікацій і технічних вимог.

Перша і найважливіша функція технічного завдання полягає в тому, щоб встановити основу для проекту. Прагнення до здатності розуміти очікування замовника і можливості команди до їх реалізації. Визначити обсяг і складність проекту, що дозволяє правильно спланувати роботу команди і визначити терміни реалізації проекту, а також передбачити різні надзвичайні ситуації. Забезпечення кращого спілкування між розробниками, тестувальниками та клієнтами також може допомогти уникнути непорозумінь і помилок у процесі. Це важливо, оскільки фази розробки унікальні та часто мають багато деталей.

Загалом, вибір плагіна та технології двигуна є важливим кроком у створенні проекту на Unity. Правильний вибір дозволить розробляти проекти швидше, ефективніше та якісніше. Однак вибір не повинен бути занадто складним або дорогим.

За для закладання функціональних вимог, опишемо геймплей гри. Гравці будуть приймати замовлення від клієнтів і обслуговувати їх за допомогою таксі. Крім того, гравець має можливість керувати автомобілями таксопарку та покращувати їх технічні характеристики, будувати різноманітні споруди та наймати нових водіїв, усе це є невід'ємною частиною розвитку гри.

Опис інтерфейсу: Головне меню повинно містити такі пункти: Авторизація, Геймплей, Налаштування, Довідка, Вихід. У грі повинні бути реалізовані наступні елементи інтерфейсу: форма для реєстрації, геймплей гри, де очевидно

					КВРІПЗ.190129.19.03.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		14

показано, що гравцю потрібно робити, а також необхідна панель з результатами гравців. В якості піджанру аркади було обрано – Tower Defense.

Жанр Tower Defense – це вид комп'ютерних ігор, у яких гравцеві потрібно захищати свою територію від наступаючих хвиль ворогів, будуючи та розміщуючи різноманітні захисні споруди – вежі. Основна мета гравця в жанрі Tower Defense – не дозволити ворогам досягти кінцевої точки або бази, яку гравець захищає. Функціональні вимоги цього жанру включають наявність:

- карти: графічне поле поділене на сітку або траси, по яких рухаються вороги. Деякі карти можуть мати перешкоди, ландшафтні особливості або шляхи з різною довжиною, що впливають на стратегію розташування захисних веж;

- захисних веж: гравець може будувати різні типи веж для боротьби з ворожими силами. Кожна вежа має свої унікальні властивості. Гравець може вдосконалювати вежі, щоб покращити їх ефективність;

- ресурсів: гравець може мати обмежені ресурси, які використовуються для будівництва та покращення веж. Ресурси можна отримати під час гри або за певні досягнення;

- хвиль ворогів: вороги атакують територію гравця у визначені хвили. Кожна хвиля може містити різні типи ворогів з різними здібностями та силою;

- системи прогресу: гра може мати систему рівнів або кампаній, де гравець поступово зустрічає все більш складніші виклики та отримує можливість отримати нові вежі або покращення;

Наступний крок – визначення стилістики та теми гри. Так як, цей жанр тісно сплітається з воєнною тематикою, а зараз в нашій країні війна, є досить актуальним пов'язати ці тези. Проте, важливо щоб гра виглядала дружелюбною за для її легкої популяризації, тому головним героєм гри буде Пес Патрон – відважний персонаж, який у мальтяжному стилі захищає свою країну та землю. Це досить популярний герой у сучасній Україні завдяки пісням та медіа мемам.

У розділі було описано послідовність задач, які будуть поступово вирішуватися в процесі створення оформлення застосунку та стилізації.

					КВРІПЗ.190129.19.03.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		15

## 1.4 Висновки. Постановка задачі

До функціональних вимог відносимо наявність динамічного ігрового процесу, складність якого росте з рівнями. Слід додати можливість авторизації, збереження результатів гравця в он-лайн таблиці, монетизація гри за допомогою сервісу Admob, та можливість впливати на балансування значеннями через консоль сервісу Firebase. Також важлива наявність підтримки всіх екранів сучасних смартфонів. На основі зроблених висновків можна зробити перший прототип програми та технічне завдання. Технічне завдання було показано в Додатку А. Перший прототип зображено на рисунку 1.4.

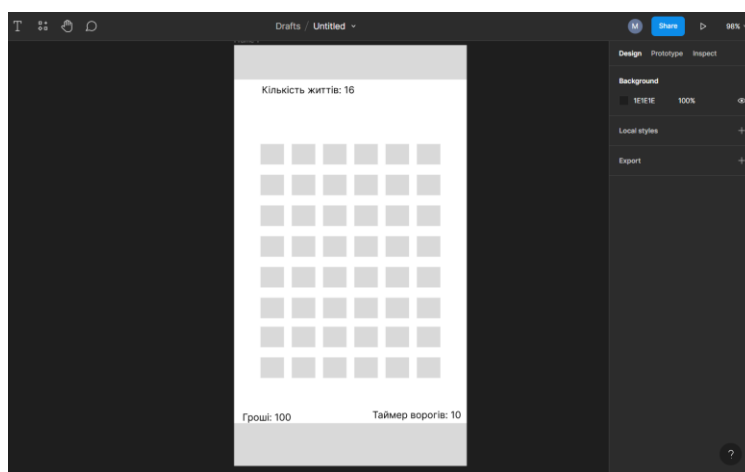


Рисунок 1.4 – Перший прототип гри створений в Figma

До нефункціональних вимог ігрового додатку можна віднести той факт, що оформлення потребує яскравих кольорів, та графіка, які повинні виглядати приємно, та безпечно. Окрім того вітається створення в додатку патріотичної символіки, та українських культурних тезисів пов'язаних з перемогою, геройствами. На місце веж, як це буває у звичайному Tower Defense на їх місце можна взяти собак різних порід, які кидатимуть м'ячами у ворогів. Це все додаватиме грі приємної атмосфери. Як підсумок, визначення вимог один з найважливіших, адже від цього залежить працездатність та практичне застосування всієї роботи.

					КВРІПЗ.190129.19.03.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		16

## 2 Проектування програмного забезпечення

### 2.1 Проектування варіантів використання

Проектування варіантів використання, також відоме як проектування сценаріїв використання, є процесом визначення та опису функціональності системи на основі взаємодії між акторами та самою системою. Варіанти використання, або сценарії, описують послідовність подій, які відбуваються під час взаємодії користувачів з системою, з метою досягнення певних цілей або виконання певних функцій.

Розроблений застосунок у готовому виді – це гра, розроблена на Unity, яка доступна для скачування на Google Play. Гравці можуть знайти гру в магазині і встановити її на свої пристрої. Після завершення гри, користувачі мають можливість залишити свій відгук про гру та виставити оцінку, що допомагає визначити її популярність.

У грі є монетизація через Admob, де гравці можуть отримувати вигоду, переглядаючи рекламні банери або відеоролики. Це дозволяє гравцям отримувати монети або бонуси, які можуть використовувати в грі для поліпшення свого досвіду.

Також у грі є посилання на соціальні мережі фансервісу, такі як Telegram або TikTok, Instagram. Це дає можливість гравцям з'єднатися з фанатами гри та ділитися своїми досягненнями, думками та фотографіями.

Гра також має геймплей, де гравці можуть змагатися один з одним. Це може бути режим змагань, де гравці можуть вибирати своїх опонентів або брати участь у випадкових поєдинках з іншими гравцями. Це додає елемент змагання та соціальної взаємодії до гри.

Окрім цього, у грі є можливість обміну контактами між користувачами. Гравці можуть залишати свої контакти, такі як електронна пошта або ім'я користувача, і інші гравці можуть знайти ці контакти та встановити зв'язок з ними поза межами гри.

					КВРІПЗ.190129.19.03.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		17

У грі також присутня звичайна реклама, яка може бути оновлена через базу даних Firebase. Це дає мені, як розробнику, можливість актуалізувати текст та змінити рекламні ресурси без необхідності оновлювати саму гру. Завдяки цьому, буде змога ефективно просувати нові продукти, послуги або події, розміщуючи оновлену рекламу прямо в грі. Це дозволяє зберігати гнучкість та контроль над рекламними матеріалами, що сприяє покращенню взаємодії з гравцями та максимізує прибуток від рекламних показів. Діаграма використання зображена на рисунку 2.1.

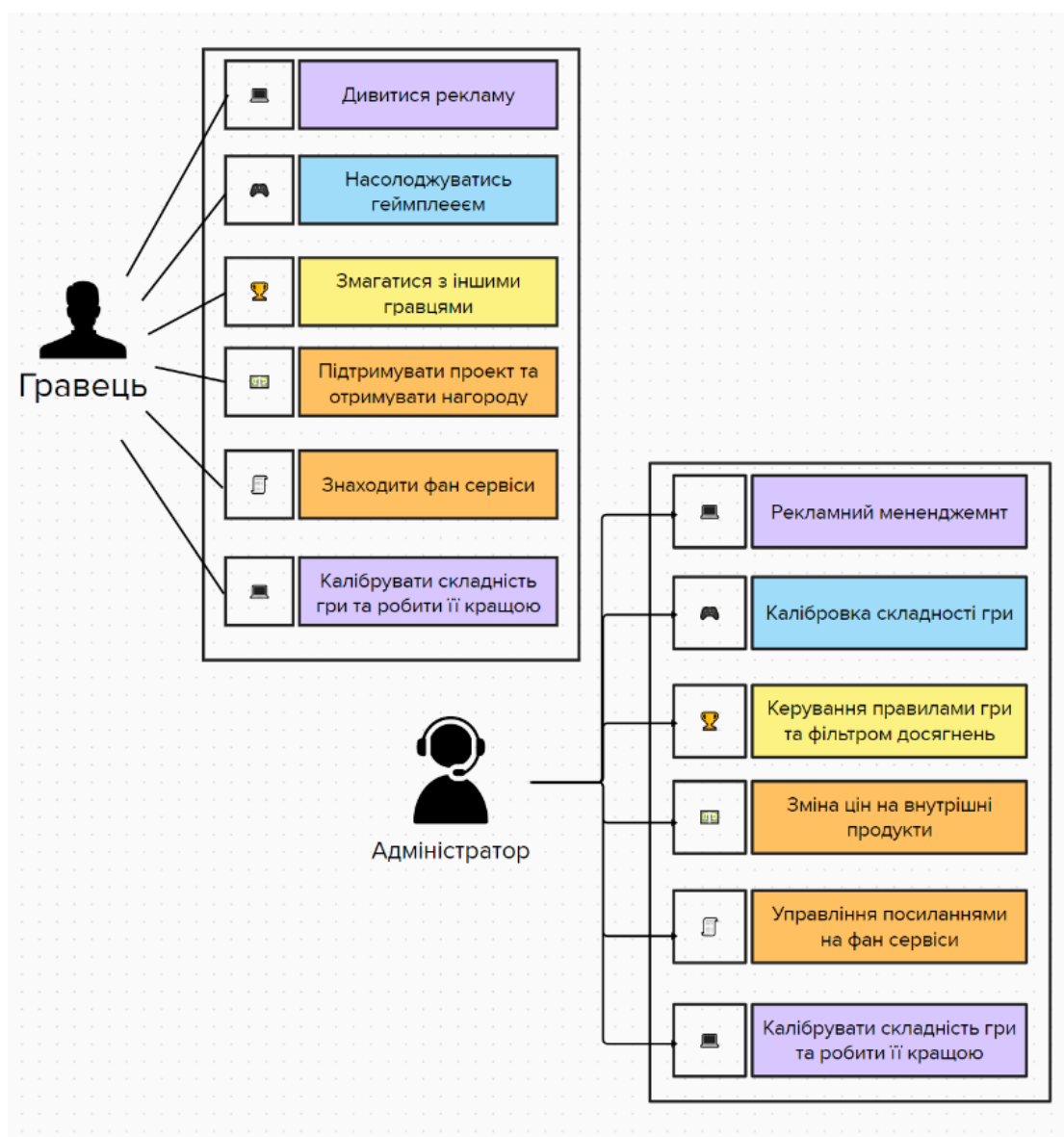


Рисунок 2.1 – Діаграма варіантів використання

## 2.2 Проектування інтерфейсу користувача

При проектуванні інтерфейсу користувача в Unity з використанням Canvas та інших елементів, розглянемо кожне вікно по черзі. Авторизація та реєстрація надає можливість користувачам увійти до гри або створити новий обліковий запис. Тут зазвичай розташовані поля для введення логіна та пароля, кнопки "Увійти" та "Зареєструватися". Метою цього вікна є забезпечення зручного доступу до гри для зареєстрованих користувачів та створення нових облікових записів. Вікна авторизації зображені на рисунку 2.2.

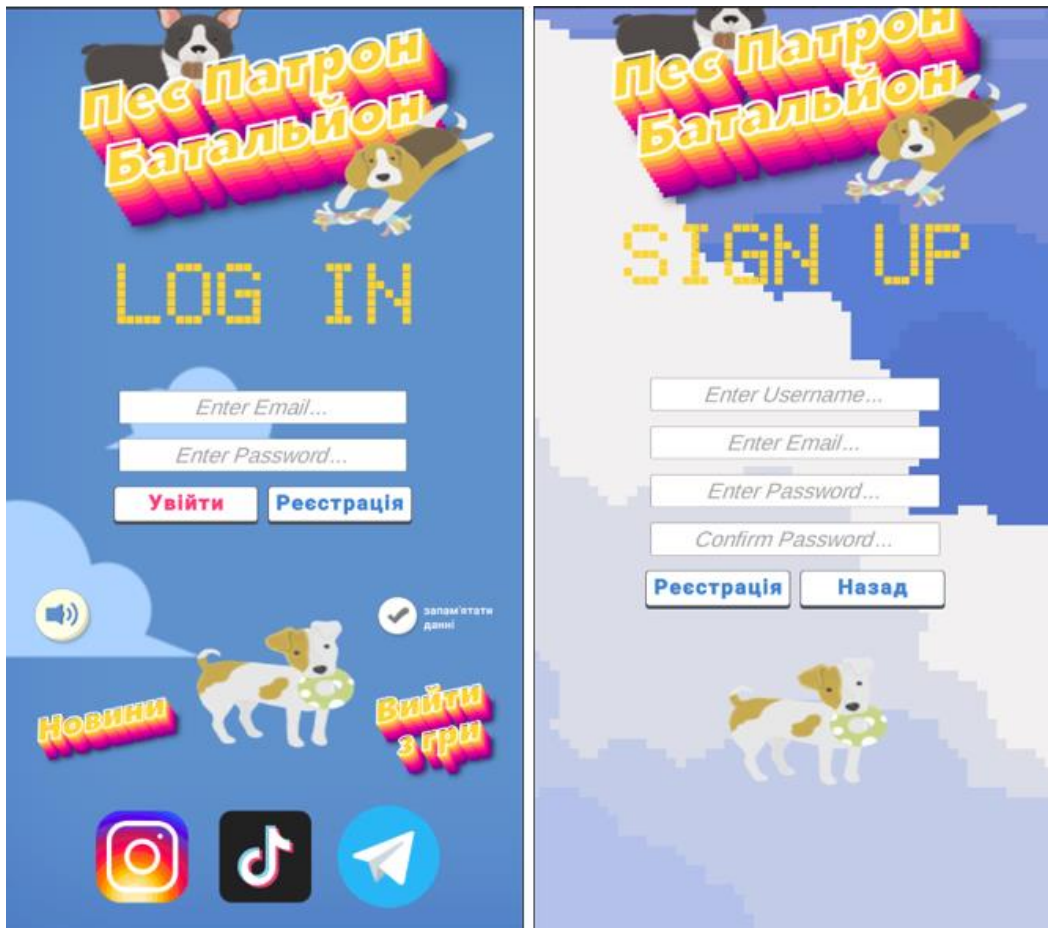


Рисунок 2.2 – Вікна авторизації

Геймплейне меню: це вікно містить основні функції гри, такі як початок нової гри, завантаження попередніх збережених сеансів, налаштування гри, вихід

					КВРІПЗ.190129.19.03.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		19

з гри тощо. Тут розташовані кнопки, які викликають відповідні дії. Головна мета цього вікна – надати користувачеві зручний спосіб управління функціями гри, є два його режими – режим готовності до гри та сама гра. Обидва вікна зображено на рисунку 2.3.

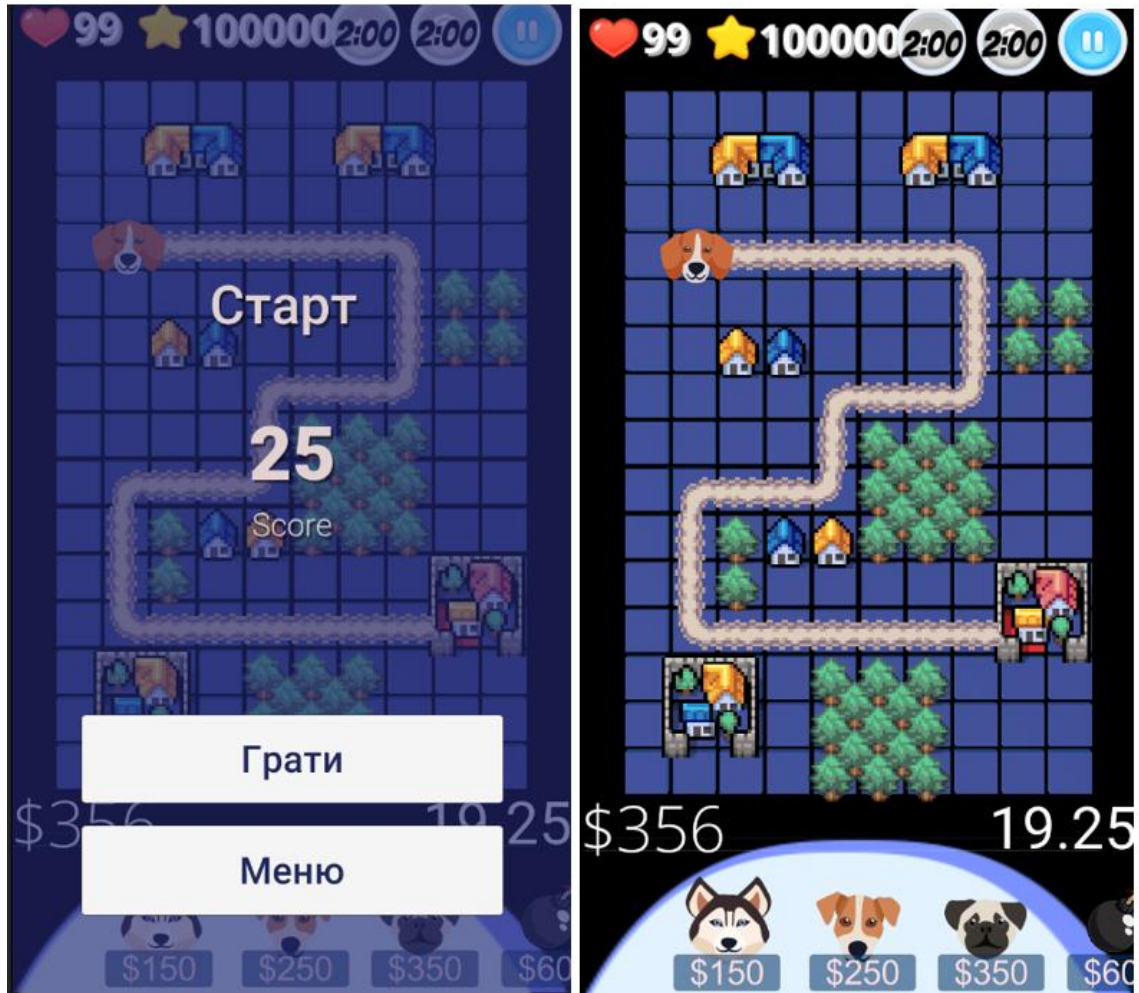


Рисунок 2.3 – Вікна геймплею та режим паузи

У вікні профілю гравця користувач має можливість зануритись у свій особистий простір і керувати своїм профілем в грі. Це вікно містить широкий спектр інформації та налаштувань, що дозволяє гравцю налаштувати свій профіль відповідно до власних уподобань. Основною складовою частиною вікна профілю є інформаційний блок, де відображається основна інформація про гравця. Це може включати його ім'я, рівень досягнень, статистику проходження різних режимів гри, кількість набраних очок тощо. Крім того, можуть бути

					КВРІПЗ.190129.19.03.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		20

відображені відзнаки або нагороди, які гравець отримав за досягнення певних меток у грі.

У вікні профілю також може бути доступне меню аватара. Користувач має можливість вибрати картинку профілю, яка буде відображати його у грі. Це може бути набір заздалегідь встановлених зображень, які користувач може вибрати безкоштовно. Альтернативно, можуть бути доступні преміальні аватари, які можна отримати за допомогою донатів або спеціальних досягнень у грі. Гравець може переглядати список доступних аватарів, попередньо переглядаючи їх зображення та вибираючи той, який найбільше йому подобається або відповідає його стилю. Вікно профілю та меню вибору аватару зображено на рисунку 2.4.

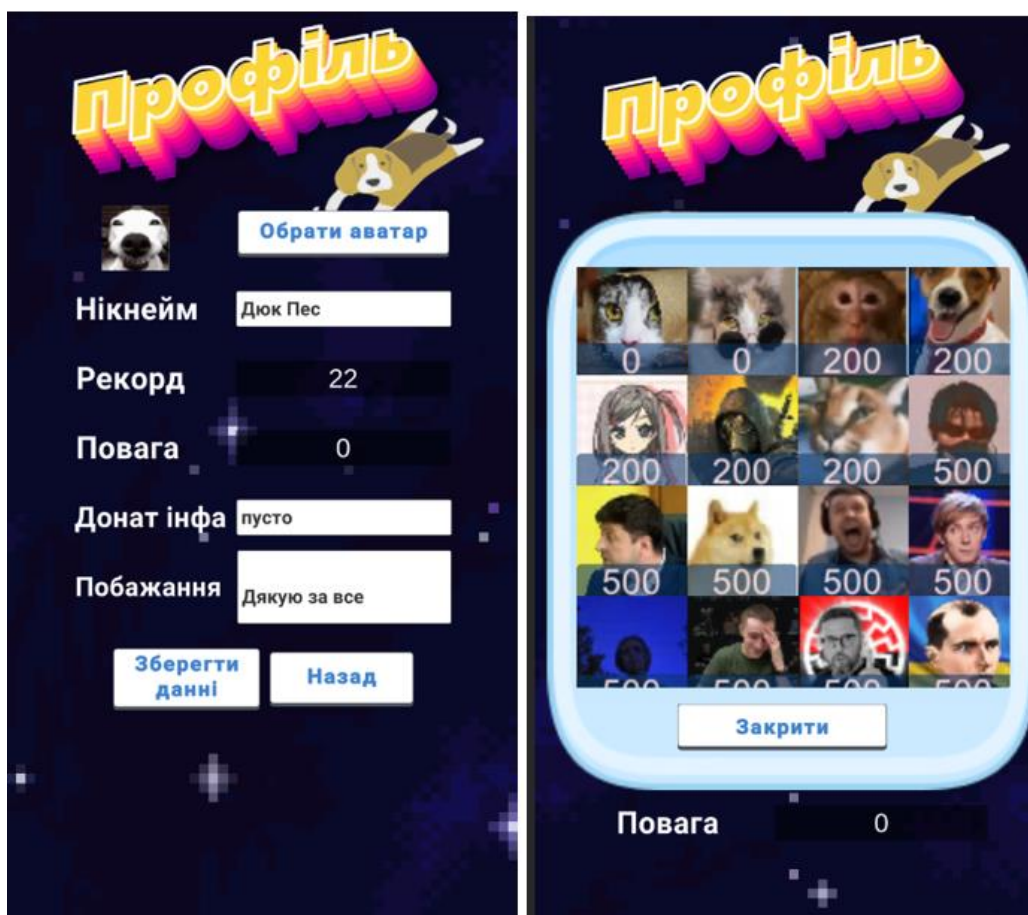


Рисунок 2.4 – Вікно налаштування профілю та аватару

У вікнах "Таблиця рекордів" та "Таблиця донатерів" гравці мають можливість ознайомитися з найкращими результатами та досягненнями інших

					КВРІПЗ.190129.19.03.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		21

учасників гри. Ці вікна відображаються у вигляді списку гравців разом з їхніми досягненнями, які можуть бути відсортовані за певними критеріями, такими як найвищий рівень, найбільша кількість очок або найбільше пожертвування. Відображення гравців у таблиці рекордів та таблиці донатерів може бути організовано у вигляді сторінок, що дозволяють користувачам перегортати список та переглядати більше результатів. Це зручно, коли в таблиці міститься велика кількість гравців і гравці можуть переглянути більше інформації, прокручуючи сторінки.

Однією зі специфічних можливостей, яку надає он-лайн база даних Firebase, є можливість змінювати кількість відображених гравців у таблиці. Це дозволяє розробникам налаштовувати розмір таблиці в залежності від потреб аудиторії гри. Наприклад, якщо гра набуває популярності та має велику кількість гравців, розробники можуть збільшити кількість відображених гравців у таблиці, щоб забезпечити відображення більшого числа досягнень. З іншого боку, якщо гра має меншу аудиторію, розробники можуть скоротити кількість відображених гравців, щоб відображати лише найкращі результати. Вікнах "Таблиця рекордів" та "Таблиця донатерів" зображені на рисунку 2.5

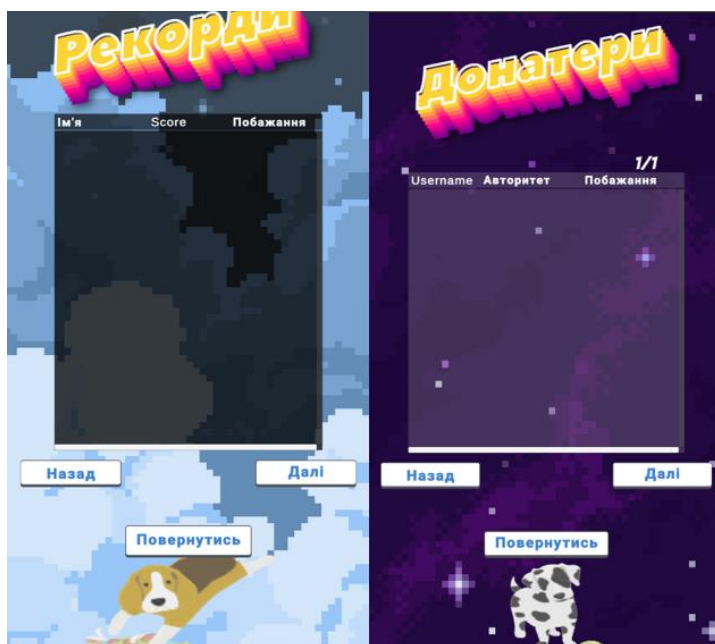


Рисунок 2.5 – Вікна досягнень гравців

					КВРІПЗ.190129.19.03.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		22

## 2.3 Реалізація модулів системи

Однією з ключових переваг роботи з Unity є те, що вам не потрібно винаходити колесо. Так як у Unity є вбудовані засоби та функції, що допомагають розробникам швидко створювати графічні ігри. Однак, для того, щоб успішно працювати в Unity, потрібно дотримуватись кількох принципів. Перш за все, потрібно правильно розподіляти код для того, щоб у майбутньому це не призвело до великих труднощів. Це може бути досягнуто шляхом створення логічних модулів та класів, які мають конкретні відповідності та функції. Наприклад, якщо розробник створює ігрового персонажа, варто розділити його логіку та візуальний вигляд на різні класи. Таким чином, коли розробник згадає в майбутньому про цей персонаж, то можна знайти всі його функції та властивості у відповідному класі, а не розкидані по всьому коду. Також, варто дотримуватись інших правил програмування, таких як неймінг змінних та функцій, коментування коду та використання версійного контролю. Це дозволить розробнику легко знаходити та зрозуміло змінювати код навіть після тривалої перерви у роботі над проектом. Велика сила Unity у використанні уже створених матеріалів, та ідей, а також паттернів програмування, що стануть варіантами готових рішень проблем, які часто виникають у розробці таких проектів.

Шаблони проектування – це загальні рішення типових проблем, які зустрічаються в інженерії програмного забезпечення. Це ще не готові рішення, які можна скопіювати та вставити у свій код, але можна розглядати шаблони проектування, як додаткові інструменти у своєму наборі інструментів. Варто інтегрувати їх, щоб покращити читабельність коду та зробити кодову базу чистішою.

Щоб зрозуміти, який паттерн слід використати, варто розглянути популярні випадки використання різних паттернів під різні ситуації в геймдеві. Кожен шаблон має свої позитивні та негативні моменти, тому досить важлива задача – обрати найбільш вигідний з них. Компонентний підхід є однією з основних

					КВРІПЗ.190129.19.03.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		23

концепцій у Unity і використовується для створення об'єктів та взаємодії з ними. У Unity компонент є окремим модулем функціональності, який можна додати до програмного об'єкту. Кожен компонент представляє певну функціональність, таку як рух, зіткнення, здоров'я, зброя тощо. Розробник може додавати нові компоненти до своїх об'єктів для розширення їх функціональності. Компоненти можуть бути додані до ігрового об'єктів з допомогою інтерфейсу Unity Editor або програмно з використанням коду Unity. Кожен компонент має свої властивості та методи, які можна настроїти для різних потреб життєвого циклу програми. Крім того, компоненти можуть взаємодіяти між собою, що дозволяє створювати складні системи та ігрові механіки.

Наприклад, компонент руху може взаємодіяти з компонентом зіткнення, щоб перевірити, чи стикається об'єкт з іншим об'єктом, та виконувати певні дії у випадку зіткнення. Загалом, компонентний підхід дозволяє створювати гнучкі та легкі для розширення системи геймплею, що робить Unity дуже популярним серед розробників. Компонентний підхід у Unity можна вважати патерном проектування, оскільки він відповідає шаблону "Компонент", який входить до категорії структурних паттернів проектування. На рисунку 2.1 на прикладі гри аркади в піджанрі Tower Defense показано структуру проектування об'єктів у грі за патерном «Компонент».

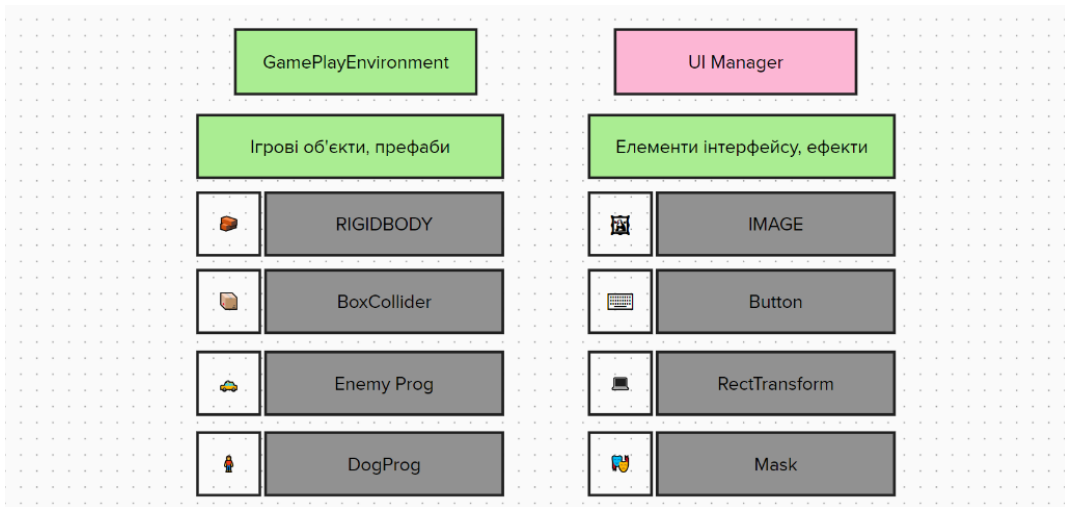


Рисунок 2.1 – Патерн проектування компонент

Патерн Pool System – це патерн проектування, який використовується в багатьох ігрових двигунах, включаючи Unity. Він дозволяє ефективно керувати кількістю об'єктів в грі, знижуючи ризик перевантаження пам'яті і процесора.

В Unity ідея патерну Pool System використовується для керування екземплярами об'єктів, таких як частинки, вороги, предмети, ефекти та інше. Замість того, щоб створювати новий екземпляр кожного об'єкту при його використанні, пул об'єктів зберігає заздалегідь створені екземпляри в пам'яті і повертає їх для використання при потребі. Pool System включає в себе наступні елементи:

- об'єкт-пул – це список об'єктів, які створюються заздалегідь і зберігаються в пам'яті. Об'єкти можуть бути використані у грі, а потім повернуті до пулу для наступного використання;

- менеджер пулу – це об'єкт, який керує створенням і поверненням об'єктів з пулу. Він також відповідає за відслідковування кількості доступних екземплярів у пулі;

- клієнт – це об'єкт, який використовує екземпляр з пулу. Клієнт повинен запитати менеджера пулу про доступні екземпляри, використовувати їх і повернути їх до пулу, коли вони стануть зайвими.

Конкретно у даному проекті, головна ідея використання цього патерну полягає в тому, для «Турелей» потрібен механізм створення куль, а в Unity, що створення, що видалення досить громіздкий процес, якщо говорити про велику кількість турелей. Тому з самого початку треба створити максимальну кількість можливих куль, а згодом просто вимкнути непотрібні. Проте, це не єдиний спосіб для використання цього потужного програмного застосунку. Існує ще один хороший спосіб використати цю концепцію – він полягає у системі звукового супроводу для гри. Звукова система в Unity побудована таким чином, що є якась кількість AudioSource - компонентів, які можуть вмикати звуковий уривок з різними налаштуваннями під нього. Одночасно він може програвати лише один звуковий кліп, тому, звісно, доведеться використовувати декілька таких модулів.

					КВРІПЗ.190129.19.03.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		25

Кожен із створених збільшує навантаження, та потребує достатню кількість ресурсів. Тому є досить непоганим у нашому випадку спосіб нового використання існуючих об'єктів. За допомогою перевірки шукаємо компонент, який зараз не програє звук, та замінюємо у ньому кліп на потрібний. Таким же способом можна управлятися зі спецефектами, тому що вони, маючи Target Graphic мають свій час анімації та працюють абсолютно таким же чином. Іноді пулові об'єкти можуть відрізнятися між собою або мати індивідуальну організацію, для поштучної обробки пулова система теж дуже добре підходить, адже це ще один момент в циклі програми де можна зробити правки. Таким чином цей патерн є більше концепцією, а не практичним застосуванням. Демонстрація його роботи зображена на рисунку 2.2.

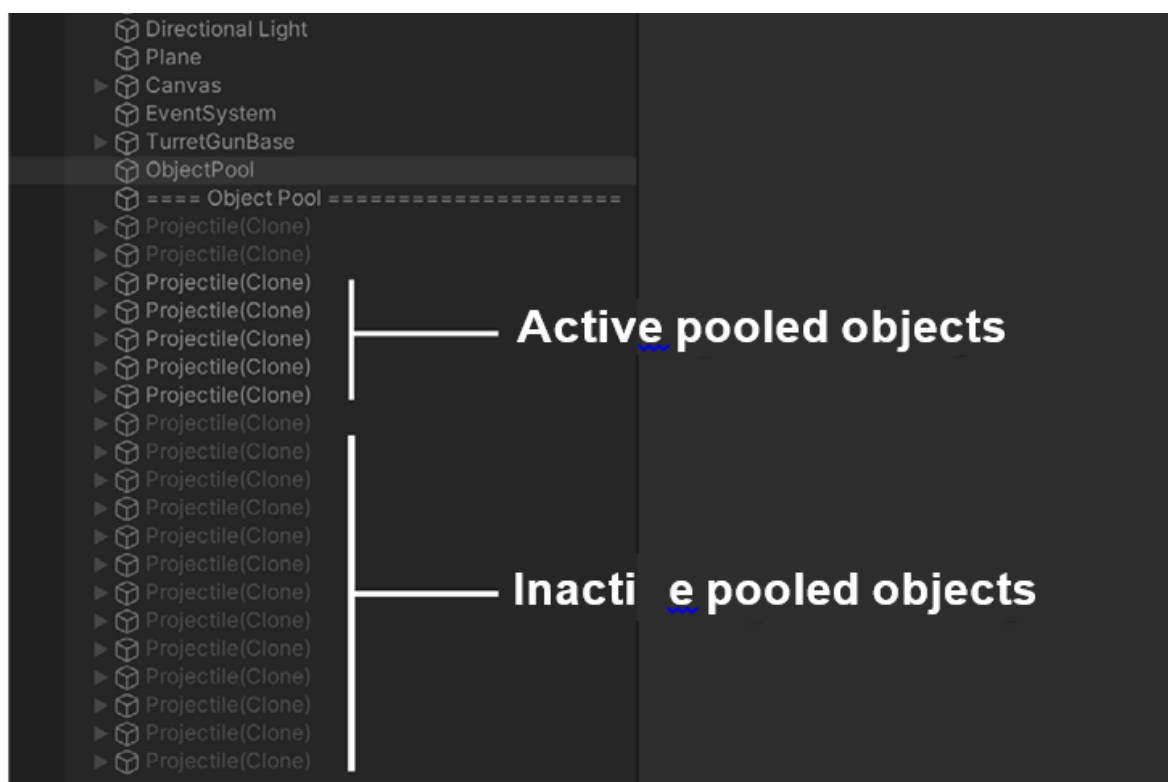


Рисунок 2.1 – Патерн проектування PoolSystem

На рисунку добре видно, що список об'єктів у пуловому клієнті досягає близько 20, та за допомогою скрипту, в якому через цикл вмикаються та вимикаються об'єкти він має змогу балансувати навантаженням на систему.

					КВРІПЗ.190129.19.03.ПЗ	Арк.
						26
Зм.	Арк.	№ докум.	Підпис	Дата		

Фабричний метод – це один з патернів проектування в програмуванні, що дозволяє створювати об'єкти певного класу, не вказуючи конкретних типів об'єктів, які будуть створені. Замість цього, створення об'єктів делегується підкласам, які визначають, який тип об'єкту повинен бути створений. Загалом, фабричний метод дозволяє розділити процес створення об'єкту від його використання. Це дозволяє змінювати типи створюваних об'єктів без зміни коду, що використовує ці об'єкти. Також це дозволяє скористатись поліморфізмом для роботи з об'єктами різних типів через спільний інтерфейс.

Конкретно у випадку створення Tower Defense гри велика користь цього патерну є в тому, що ігрові об'єкти виносяться в абстракцію, яку зручно налаштовувати та розширювати. На рисунку 2.3 показано меню налаштування турелей з інтерфейсу Unity, який можливий завдяки фабричному методу.

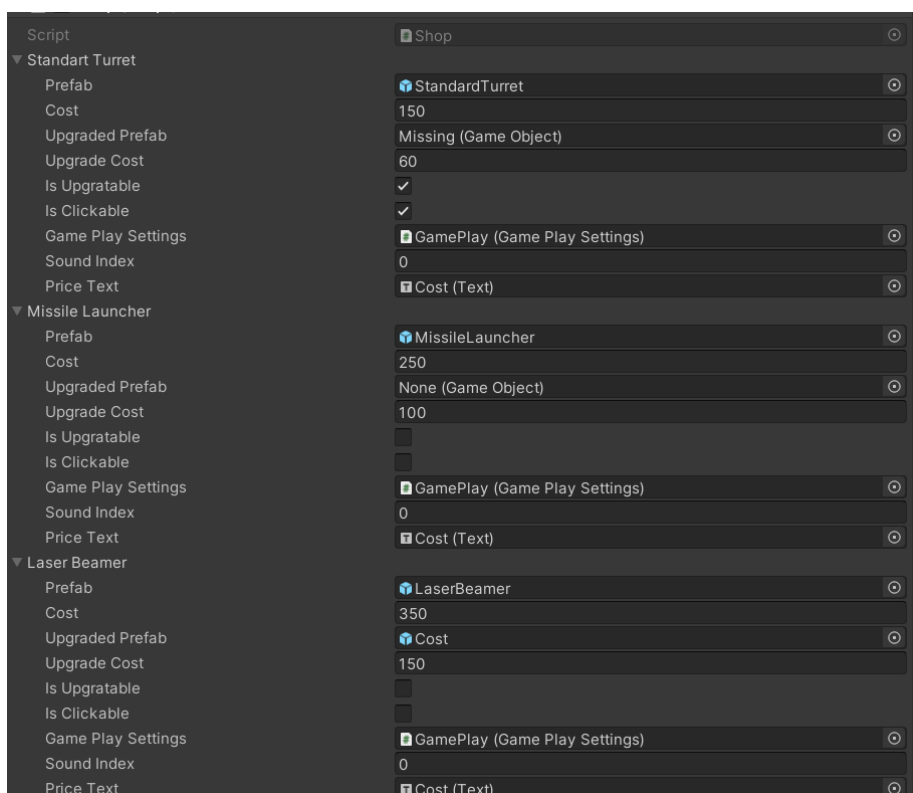


Рисунок 2.3 – Меню налаштування турелей

Інша важлива функція патерну «Фабрика» створення об'єктів від їх подальшого використання, що сприяє збільшенню гнучкості програми та

полегшенню її розширення. За допомогою Діаграми класів зображеної на рисунку 2.4 можна побачити, що клас фабрика є найголовнішим у цьому ігровому додатку, адже завдяки ньому будуть створюватися вороги на шляхах, а також будуватиметься рівень.

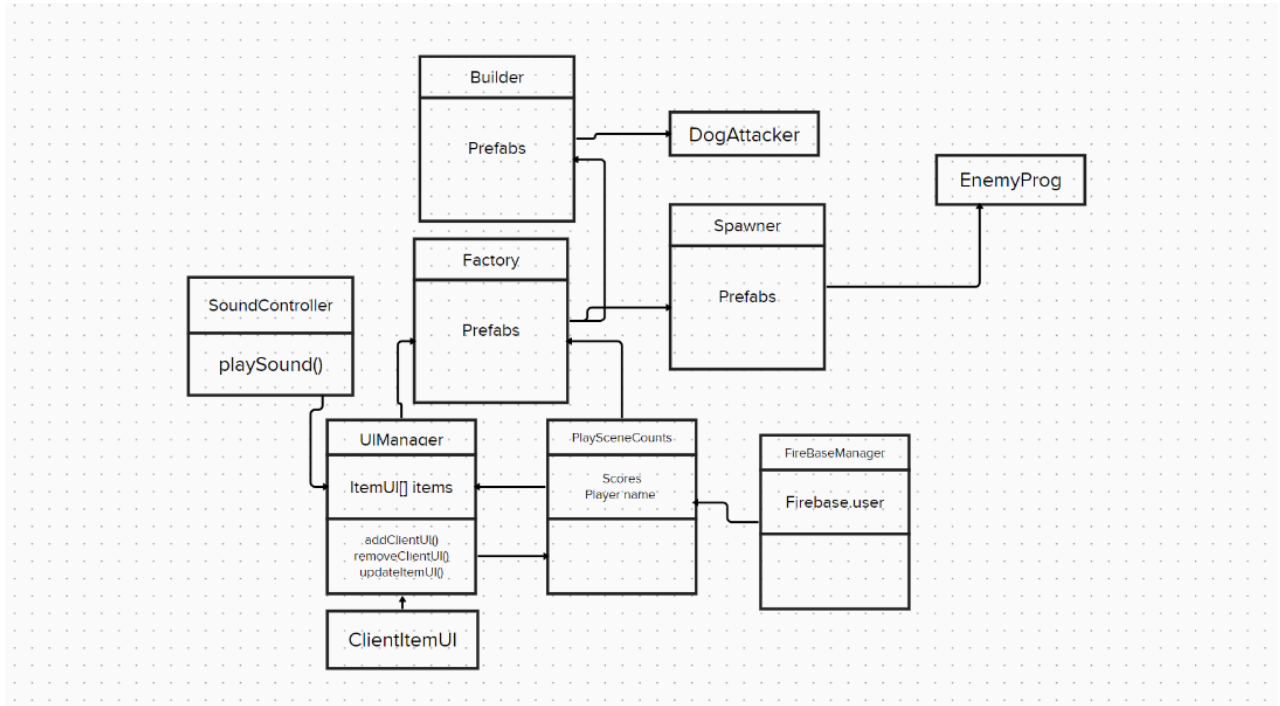


Рисунок 2.4 – Діаграма класів

Як видно з діаграми класів, що виникли деякі залежності класу фабрики від префабів, проте завдяки організації фабрики у патерні, маємо досить легкий спосіб по-перше керувати подіями у грі, а по друге досить легкий у плані написання спосіб для створення алгоритмів та життєвого шляху гри. Фабрика, також підлягає під шаблон проектування Singleton, що робить її статичною. Singleton – це патерн проектування в об'єктно-орієнтованому програмуванні, який використовується для обмеження кількості екземплярів класу до одного та забезпечення глобального доступу до цього єдиного екземпляра. Іноді, це важливо для створенні різних управляючих об'єктів, які знаходяться на сцені. Адже, за логікою програми контролер повинен бути один. Singleton забезпечую єдиний екземпляр класу в межах програми. Цей екземпляр доступний з будь-якої

частини програми, що дозволяє легко отримувати доступ до спільних ресурсів. Концепція забезпечує контроль над процесом створення єдиного екземпляра, забороняючи створення додаткових екземплярів класу. Переваги використання Singleton:

- використання патерну дозволяє економити системні ресурси, так як створення лише одного екземпляра класу зменшує навантаження на пам'ять та процесор;

- ідея забезпечує зручний глобальний доступ до ресурсів, що спільно використовуються різними частинами програми;

- розробка дозволяє контролювати доступ до спільних ресурсів та уникнути колізій при одночасному доступі кількох об'єктів до спільних ресурсів.

Недоліки використання Singleton :

- оскільки патерн має глобальний доступ, він може ускладнювати процес тестування, оскільки не завжди можна легко замінити єдиний екземпляр на об'єкт або підробити його поведінку під час тестування;

- затримки при ініціалізації: іноді ініціалізація залежностей може бути витратною за часом, оскільки він може бути створений лише при першому виклику, а не при старті програми. Це може спричинити затримки в роботі програми, якщо є потреба в швидкому доступі до ресурсів патерну.

При розробці було використано цей патерн для різних систем управління головними компонентами: інтерфейсом, фізичними об'єктами на сцені, звуковими кліпами та спецефектами.

Основна ідея застосування у грі полягає в тому, що при стартовому виклику базових методів «Awake» чи «Start» скрипт перевіряє статичні змінні на налічування такого об'єкту в сцені. У першому випадку він спостерігає, що змінна є пустою та залишає посилання на себе, в іншому випадку, якщо місце вже зайняте, він просто видаляється або деактивується. Іноді в Unity іграх доводиться мати справу з багатьма сценами, які пов'язані деякими даними між собою.

					КВРІПЗ.190129.19.03.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		29

Об'єкти, які мають діло з таким патерном зазвичай виступають в кодї статичними структурами.

Щоб показати важливість застосування цього патерну конкретно на проєкті аркадної гри, було створено схему, яка зображена на рисунку 2.5, що показує головні класи програми, які не допускають наявності дублікатів. Як видно за рисунком, ця концепція стосується абсолютно всіх важливих управляючих частин ігрового додатку, саме тому не зважаючи на свою простоту ідея цього патерну розглядається в багатьох книгах та навчальних матеріалах по Unity.

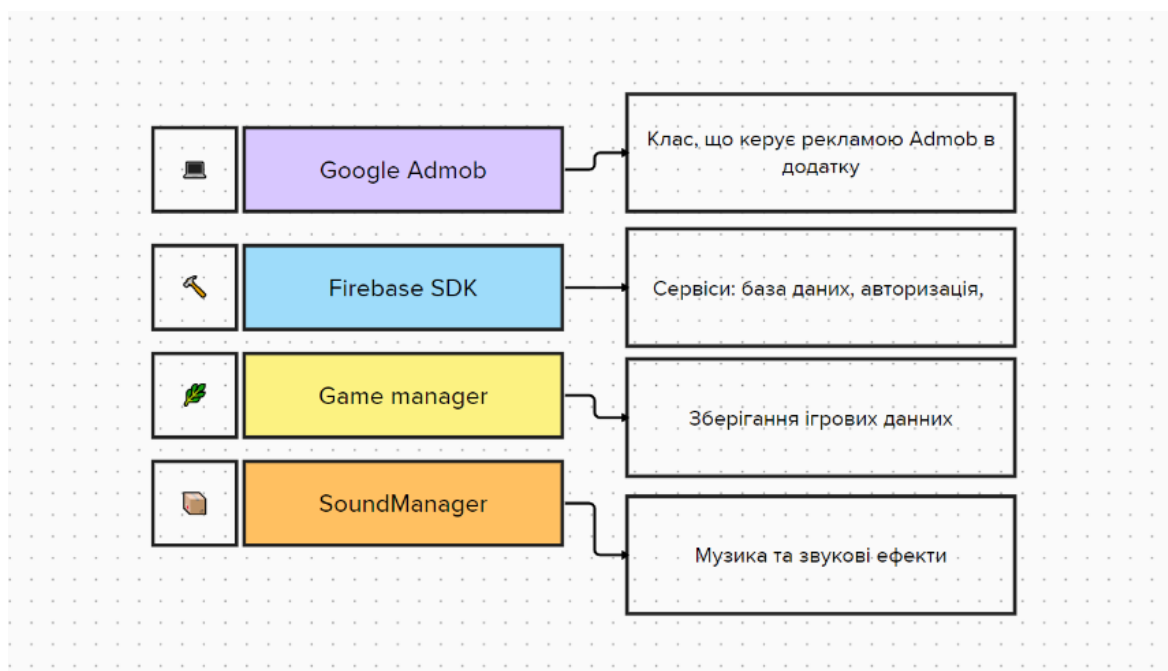


Рисунок 2.5 – Класи, що підлягають під концепцію «Singleton»

Наступний патерн, який дозволяє використати Unity є State, який відноситься до патернів поведінки і використовується для реалізації різних станів об'єкта і управління їх змінами. Він дозволяє об'єкту змінювати свою поведінку в залежності від його поточного стану без зміни його класу. Це корисно, адже перетинається з деякими жорсткими принципами SOLID.

Завдяки цьому шаблону проектування є змога розширяти класи не змінюючи їх реалізацію та створювати модулі програми так, що їх буде легко повторно використовувати.

## Основні елементи патерну State:

– контекст : це клас, який містить об'єкт, поведінка якого змінюється в залежності від його поточного стану. Він взаємодіє з об'єктами станів, викликаючи їх методи відповідно до поточного стану. Контекст може мати внутрішній стан або може посилатись на зовнішній об'єкт стану;

– стан : це інтерфейс або базовий клас, який визначає спільний інтерфейс для всіх можливих станів, в які може переходити контекст. Він містить декілька методів, які визначають поведінку, яка залежить від стану. В програмуванні це може бути інтерфейс або абстрактний клас, який визначає методи, які мають бути реалізовані в кожному конкретному стані;

– конкретні стани - це класи, які реалізують інтерфейс або успадковують базовий клас стану і реалізують конкретну поведінку для кожного стану. Кожен конкретний стан відповідає за свою власну поведінку в залежності від поточного стану об'єкта. Вони можуть містити внутрішні дані, методи, які виконують конкретні дії, і методи, які визначають перехід;

– клієнт : це клас або об'єкт, який використовує контекст і взаємодіє з ним. Клієнт може викликати методи контексту для зміни стану об'єкта або отримання поточного стану об'єкта. Клієнт також може взаємодіяти з об'єктами конкретних станів через інтерфейс або базовий клас стану.

Концепція State відноситься до патернів поведінки і використовується для реалізації різних станів об'єкта і управління їх змінами. Він дозволяє об'єкту змінювати свою поведінку в залежності від його поточного стану без зміни його класу. Це корисно, адже перетинається з деякими жорсткими принципами SOLID. Завдяки цьому шаблону проектування ми можемо розширяти класи не змінюючи їх реалізацію. Застосування патерну в програмуванні може бути різноманітним.

Наприклад, він може бути використаний для управління станами об'єкта, такими як стан гри, стан замовлення або стан користувача, дозволяючи визначити різні стани об'єкта та переходи між ними. Він також може бути використаний для

					КВРІПЗ.190129.19.03.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		31

реалізації різної поведінки об'єкта в залежності від його поточного стану, так як поведінка може змінюватись динамічно. Крім того, State може бути використаний для заміни вкладених умовних операторів або великої кількості операторів «if-else» в коді, роблячи код більш структурованим та зрозумілим. Він дозволяє розподілити різні варіанти поведінки об'єкта на окремі класи станів, що спрощує підтримку та розширення коду.

Таким чином, можна використовувати цей патерн також до станів системи управління компонентами, всюди де є умовні переходи реалізації. У влаштованому Animator для управління анімаціями також використовується цей патерн. А змінювати його властивості можна через елементи інтерфейсу. Це дозволяє дуже добре передбачити всі можливі стани об'єкта та коректного його відображення. З допомогою цього патерну, можна виявити всі зміни вікон гри, та побудувати на основі цього діаграму станів. Її зображено на рисунку 2.6

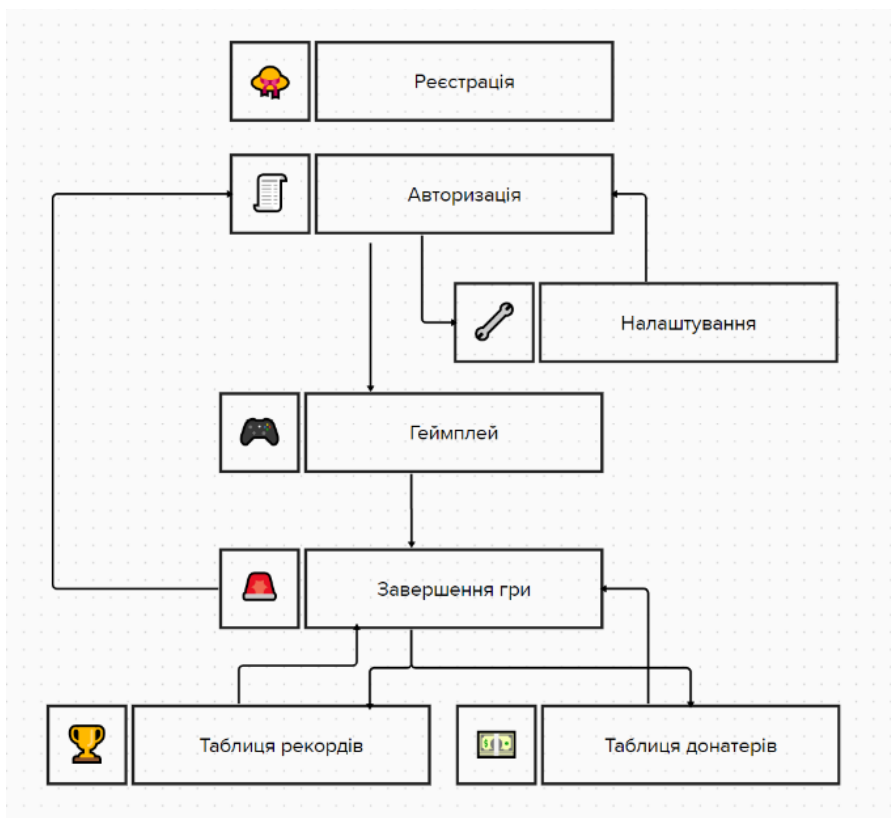


Рисунок 2.6 – Діаграма станів гри

Патерн Observer (спостерігач) – це патерн проектування, який використовується в об'єктно-орієнтованому програмуванні для реалізації взаємодії між об'єктами, де зміна стану одного об'єкта призводить до автоматичного сповіщення і оновлення інших об'єктів, які на нього підписані. В ході розробки гри може відбутися багато різних подій, таких як знищення ворогів чи отримання бонусів. Тому знадобиться механізм, який дозволить об'єктам сповіщати інші об'єкти, не маючи прямих посилань на них, що допоможе уникнути непотрібних залежностей.

Шаблон спостерігача є одним з розповсюджених рішень для таких ситуацій. Він дозволяє об'єктам спілкуватися, залишаючись слабо зв'язаними за допомогою залежності "один до багатьох". Коли один об'єкт змінює свій стан, всі спостерігачі, які на нього підписані, автоматично отримують повідомлення. Це подібно до радіовежі, яка передає різні сигнали багатьом слухачам. Схема спостереження працює на зразок радіовежі, де суб'єкт транслює сигнали спостерігачам. Об'єкт, який виконує трансляцію, називається суб'єктом, а інші об'єкти, які слухають, – спостерігачами.

Цей шаблон слабо зв'язує суб'єкт, оскільки він не знає про наявність спостерігачів і не має відносин з тим, що вони роблять після отримання сигналу. У той же час, спостерігачі мають залежність від суб'єкта, але не знають про наявність один одного. Події є таким поширеним шаблоном, що вони вже вбудовані в мову C#. Проте можна створити і власні події, але це, як правило, не потрібно, оскільки мова C# вже надає таку можливість.

Конкретно у випадку створення мобільної аркадної гри цей патерн є корисний для створення різних об'єктів, що пов'язані з «View» модулем програми. Адже кожен раз, коли у важливих параметрах є зміни, активується подія, які делегують методи зміни даних у UI. Це є гарним рішенням з точки зору, що не потрібно ставити такі завдання у метод «Update», адже він виконується кожен кадр гри, і це є складний процес. Часто для реалізації патернів у ігрових застосунках використовуються інтерфейси, виносячи все важливе в абстракцію.

					КВРІПЗ.190129.19.03.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		33

Інтерфейси – це типи, які дозволяють визначати методи та властивості, які будуть реалізовані різними класами. Вони дозволяють програмістам визначати спільні засоби взаємодії між класами та використовувати поліморфізм для роботи з різними об'єктами, які реалізують ці інтерфейси.

Наприклад, в іграх часто є багато об'єктів, які повинні взаємодіяти з гравцем, такі як вороги, предмети, пастки тощо. Інтерфейс може бути використаний для визначення спільних методів та властивостей, які повинні бути реалізовані цими об'єктами, наприклад метод для атаки гравця або метод для зменшення здоров'я гравця. Один з прикладів використання інтерфейсів в програмуванні ігор на Unity – це інтерфейс «IHealth» для управління здоров'ям персонажа. Цей інтерфейс містить методи та властивості для управління здоров'ям персонажа, такі як «TakeDamage», що дозволяє приймати пошкодження, «Heal», що відновляє здоров'я та «Die», який знищує об'єкт. Різні об'єкти в грі, такі як гравець та вороги, можуть реалізувати цей інтерфейс та взаємодіяти з ним.

Інтерфейси також можуть бути використані для створення більш гнучких систем зберігання та передачі даних між різними об'єктами в грі. Наприклад, в грі можуть бути різні об'єкти, які мають спільні властивості, такі як розмір, колір тощо. Використання інтерфейсів дозволяє програмістам визначити ці спільні властивості та методи у окремому інтерфейсі, який можна використовувати для реалізації цих об'єктів. Інтерфейси також дозволяють забезпечувати більш гнучку архітектуру програми, що дозволяє змінювати реалізацію класів, не змінюючи способу, яким класи взаємодіють з іншими об'єктами в системі. Це забезпечує зменшення залежності між класами та полегшує підтримку коду в майбутньому.

Інтерфейси також можуть допомогти уникнути повторюваного коду. В грі може бути багато об'єктів, які мають спільні функції, наприклад, об'єкти, які можна зібрати та використати.

Використання інтерфейсу дозволяє реалізувати ці функції у одному місці та використовувати їх для всіх об'єктів, які реалізують цей інтерфейс. Крім того,

					КВРІПЗ.190129.19.03.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		34

інтерфейси допомагають відокремлювати інтерфейс користувача від логіки гри. Це дозволяє змінювати інтерфейс користувача без зміни логіки гри та спрощує розробку крос платформних ігор.

Інтерфейси є потужним інструментом у програмуванні ігор на Unity, який дозволяє розробникам створювати більш гнучкі та модульні системи, зменшувати залежності між класами. У нашому випадку один із непоганих способів використання інтерфейсів є створення модулів, які легко замінюють один одного. На рисунку 2.6 показано один з прикладів як можна використовувати інтерфейси в іграх.

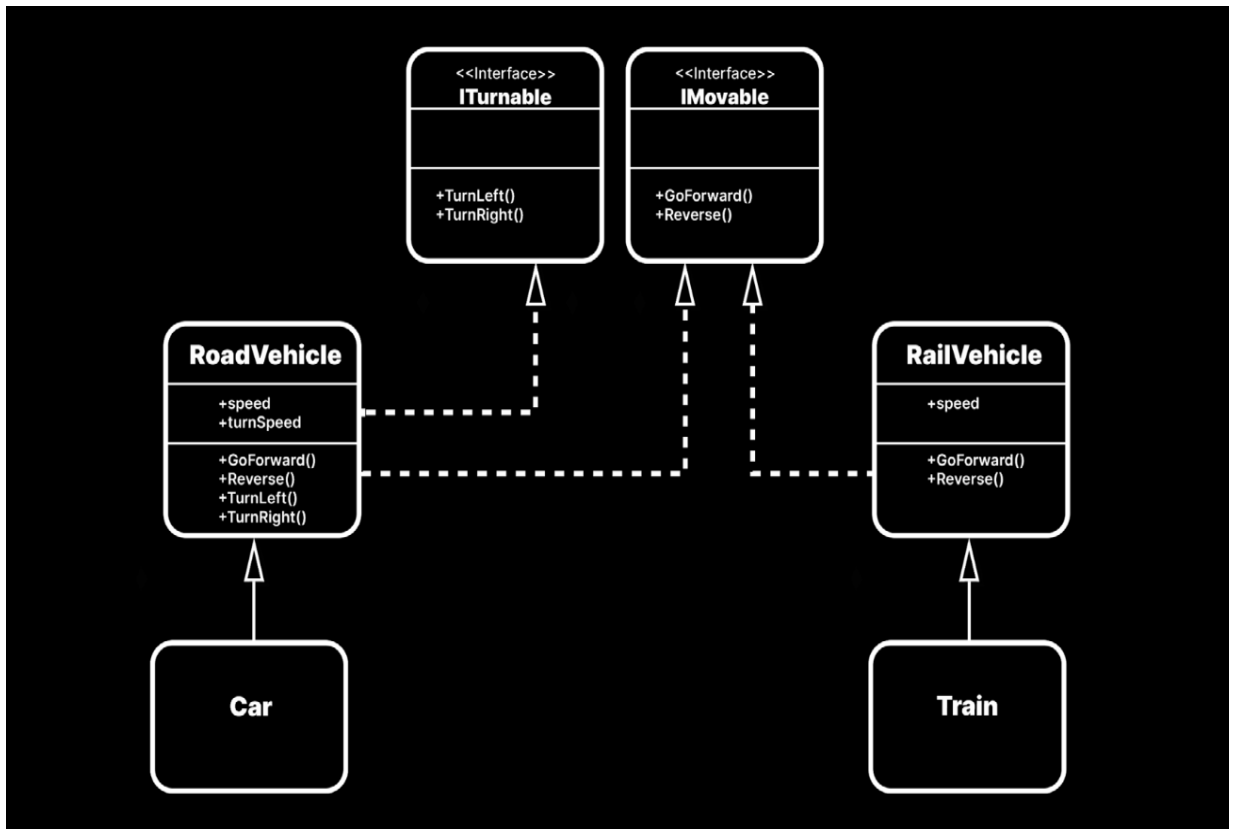


Рисунок 2.6 – Використання інтерфейсів в іграх

Тобто кожна система вона так чи інакше повинна була б бути пов’язана з іншими компонентами та залежати від них, але ми можемо зробити цю залежність від абстракції та утворити нашій грі новий вектор розвитку. Або також це досить таки непоганий спосіб для того, щоб тестувати наші модулі.

## 2.4 Висновки

Проектування додатку і його вікон є надзвичайно важливим етапом у розробці, оскільки воно визначає функціональність, ефективність та користувацький досвід. В процесі розробки нашого додатку було враховано ці аспекти, що сприятиме досягненню успіху проекту. На початку була створена діаграма варіантів використання, яка відображає різні можливості та функції, доступні для користувачів. Це дозволило при розробці чітко уявити, які опції надаватимуться гравцям і як вони будуть взаємодіяти з додатком. Крім того, діаграма класів була спроектована для моделювання структури додатку, його компонентів та залежностей між ними.

У процесі проектування було використано певні ідеї програмування, такі як factory, state, singleton та pool system. Factory дозволив нам створювати об'єкти з різними параметрами з одного місця, спрощуючи процес створення об'єктів. State був застосований для забезпечення різних станів геймплею і взаємодії з ними. Singleton використовувався для створення єдиного екземпляру деяких класів, наприклад, для керування головним меню гри. Pool system системи був використаний для ефективного управління об'єктами у грі, зокрема для управління персонажами або снарядами.

Інтерфейс додатку був пристосований до геймплею в жанрі Tower Defense. Це означає, що інтерфейс надає зручний спосіб керування та взаємодії з грою, дозволяючи гравцям легко розмішувати захисні вежі, управляти персонажами або виконувати інші необхідні дії.

Узагальнюючи, проектування додатку та його вікон, включаючи діаграми варіантів використання та класів, а також використання патернів програмування та передбачення сучасних технологій, допомагають уточнити вимоги до проекту та відкривають широкі можливості для розвитку та розширення додатку у майбутньому. У випадках, коли до роботи над великою системою долучаються нові робітники, схеми грають ключову роль в ознайомленні з системою.

					КВРІПЗ.190129.19.03.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		36

### 3 Програмна реалізація

#### 3.1 Підключення сервісів до проекту

Unity є потужним інструментом для розробки ігор та програм, проте, іноді вона сама не може задовольнити всі функціональні вимоги проекту. В таких випадках варто розглянути підключення додаткових сервісів, таких як Firebase, AdMob та Google Play, щоб розширити можливості вашої програми.

Firebase – це платформа розробки мобільних та веб-додатків, яка надає різноманітні сервіси, які можна інтегрувати в Unity. Один з головних розділів Firebase – це сервіси для зберігання та обміну даними в реальному часі. Firebase Realtime Database дозволяє зберігати та синхронізувати дані між різними пристроями, що дозволяє створювати багатокористувацькі ігри та програми зі спільними даними. Firebase також надає можливості аутентифікації користувачів, зберігання файлів, надсилання повідомлень та багато інших функцій, які можуть бути корисними для вашого проекту.

AdMob – це сервіс реклами для мобільних додатків, який дозволяє розробникам включити рекламу в ігри та програми. За допомогою AdMob є змога показувати банерну рекламу, відеорекламу та рекламу на виході з програми. Це може бути корисним засобом монетизації вашого проекту та заробітку на ньому.

Google Play – це платформа розповсюдження мобільних додатків для пристроїв з операційною системою Android. Інтеграція з Google Play дозволяє розробнику публікувати свої ігри та програми в Google Play Store, де користувачі зможуть знайти та завантажити ваші додатки. Google Play також надає можливості керування рейтингами та відгуками, аналітики використання додатків, підписки та інші функції.

Застосування Firebase в проекті Unity дозволяє значно розширити можливості гри чи програми. Наприклад, можна використовувати Firebase для реалізації мультиплеєра, де гравці зможуть взаємодіяти один з одним у реальному часі. Також є можливість використовувати Firebase для зберігання даних гравців,

					КВРІПЗ.190129.19.03.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		37

досягнень, статистики гри та інших даних, які потрібні для коректної роботи вашої програми. Крім того, Firebase надає зручні інструменти для аналізу та монетизації вашого проекту.

Узагалі, підключення додаткових сервісів до Unity дозволяє розробникам скористатися готовими рішеннями, зекономити час і зусилля на розробку додаткових функцій, а також розширити можливості вашого проекту. Вибір конкретних сервісів залежить від потреб гри чи програми і того, які функції необхідно реалізувати.

Для успішного використання Firebase у проекті потрібно мати обліковий запис Google для доступу до консолі Firebase та налаштування проекту. Процес налаштування Firebase показано на рисунку 3.1. В першу чергу потрібно вказати назву пакетів, а також скачати конфіг файл, який згодом потрібно перемістити у папку проекту Unity.

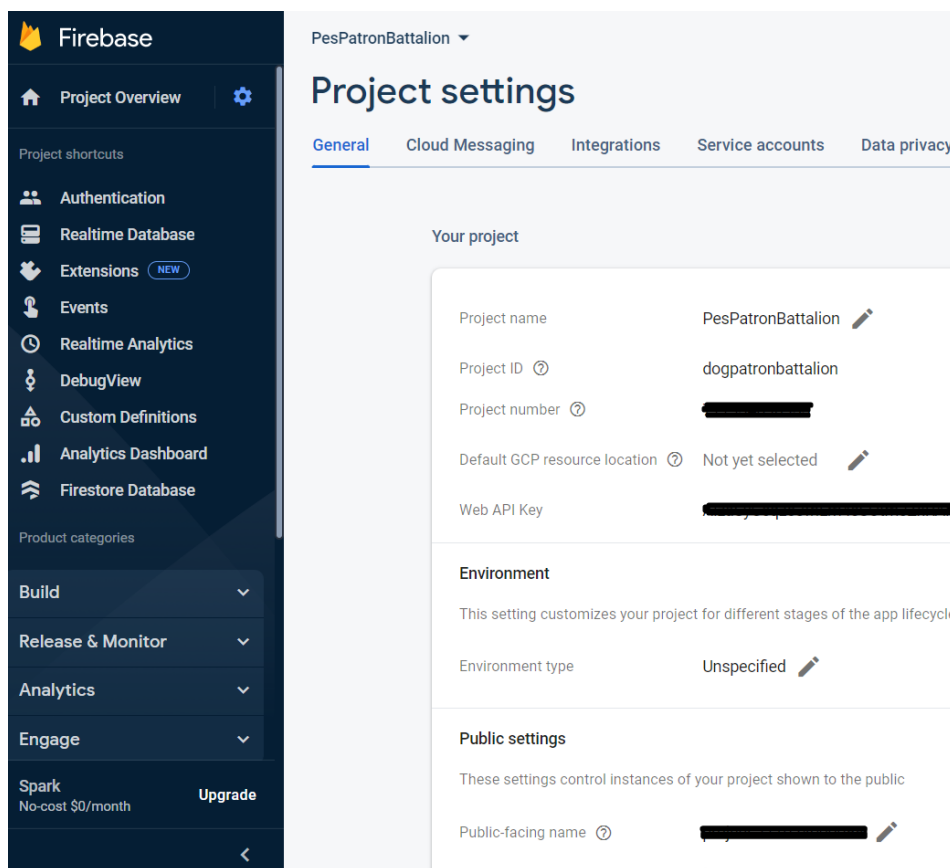


Рисунок 3.1 – Налаштування консолі Firebase

					КВРІПЗ.190129.19.03.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		38

Наступним кроком, потрібно визначити, які саме функції з Firebase потребує програмний застосунок, щоб не ускладнювати його без причини. Серед тих, які необхідні аркадній грі:

- Firebase Dependency Manager;
- Firebase App;
- Firebase Analytics;
- Firebase Database;
- Firebase Auth.

В подальшому, їх потрібно встановити у проект у правильному порядку, щоб не виникало помилок, адже кожен з цих плагінів має залежності з іншими для коректної роботи. Коректний порядок їх встановлення зображено на рисунку 3.2. Після цього залишається лише додати config файл та синхронізувати всі плагіни в коді. Код синхронізації Firebase SDK подано в Додатку А.

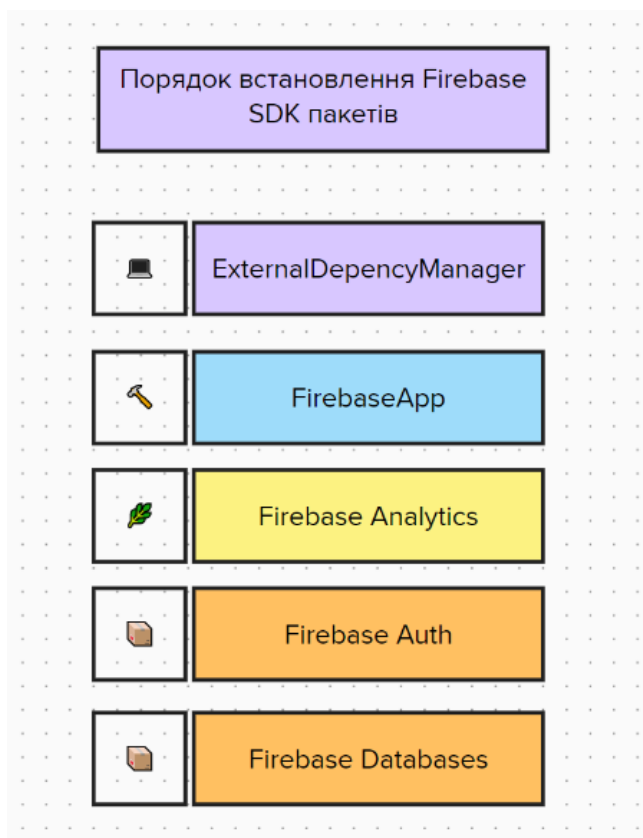


Рисунок 3.2 – Порядок встановлення пакетів Firebase

Після налаштування Firebase саме час додати у проект Google Admob, адже файли проекту містять залежності, які зараз задовольняють цей плагін. Для використання Google Admob потрібно теж створити свій обліковий запис на сайті. Процес створення облікового запису та додавання гри зображено на рисунку 3.3. Як видно з рисунку, гру створено, проте вона потребує перевірки від працівників Google, також для коректного відображення реклами є необхідність в публікації додатку в Google Play Store.

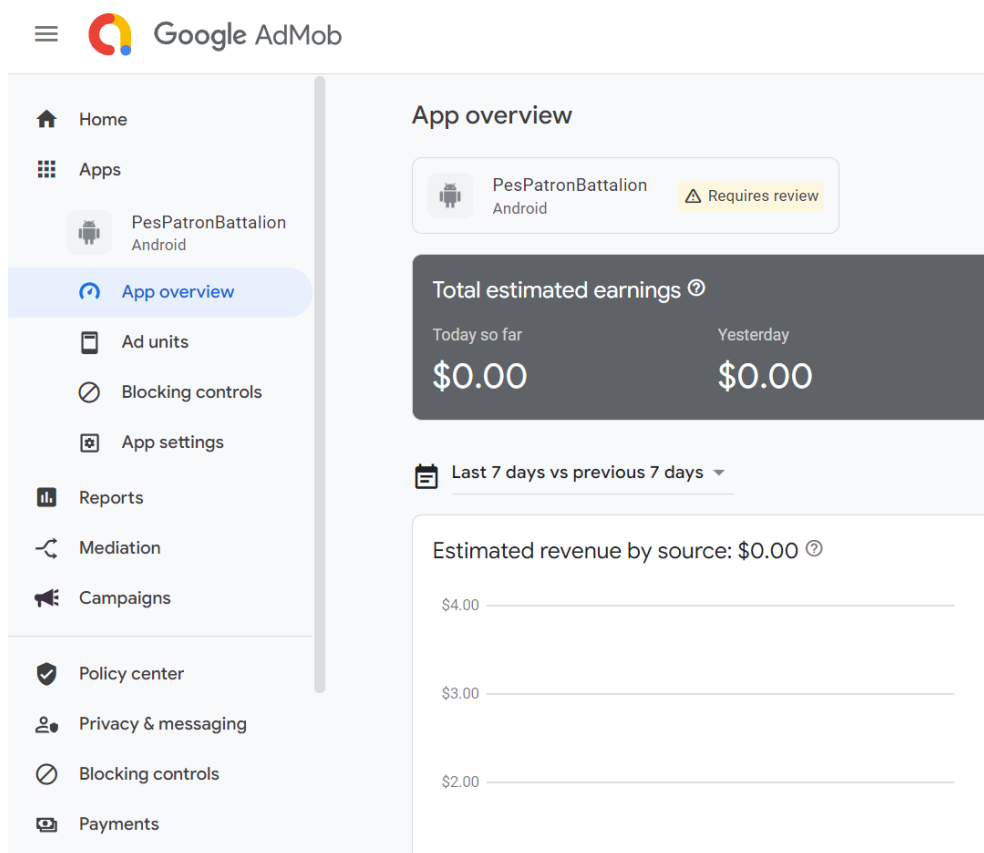


Рисунок 3.3 – Налаштування консолі Google Admob

Рекламний блок може бути різних типів, таких як банери, виникаючі вікна, відеореклами тощо, і має свої властивості та налаштування, залежно від типу реклами, який ви обираєте.

У рекламному блоку визначається розміщення реклами в ігровому додатку або на сервісі чи веб-сайті. Це може бути статичне місце на екрані, де відображається банер або виникаюче вікно, або може бути інтегровано у вміст

					КВРІПЗ.190129.19.03.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		40

або графічний інтерфейс користувача. AdMob пропонує кілька типів рекламних блоків, які можна використовувати для відображення реклами в мобільному додатку користувача:

– банери – це один з найпоширеніших типів рекламних блоків. Вони відображаються зазвичай у верхній або нижній частині екрану і мають фіксований розмір. Банери можуть містити текстову, графічну або об’ємну рекламу;

– виникаючі вікна – це повноекранна реклама, яка відображається в певний момент, наприклад, перед переходом до нового рівня в грі або під час завантаження нової сторінки на веб-сайті. Вони надають більшу візуальну привабливість та можуть містити різноманітний контент, такий як зображення, відео або інтерактивні компоненти.;

– реклама з нагородами – це тип реклами, який пропонує користувачам отримати якусь вигоду, наприклад, додаткові гроші в грі або пропуск рівня, у обмін на перегляд відеореклами або взаємодію з нею. Цей тип реклами зазвичай залучає більше уваги користувачів і може бути ефективним способом залучення до рекламодавців;

– відеореклама – це тип реклами, який відтворює відеоролики рекламного характеру. Вона може бути вставлена в місця, де очікується відеоконтент, наприклад, під час перегляду відео на веб-сайті або в мобільній грі. Відеореклама може бути короткою або довшою, залежно від налаштувань рекламодавців.

Управління рекламними блоками з консолі (панелі керування) має численні переваги, зокрема щодо контролю за вмістом та поведінкою реклами у вашому додатку або на веб-сайті. Один із основних факторів, які роблять цей підхід цінним, – це забезпечення дотримання правил магазину або платформи, на якій ви публікуєте свій додаток.

Платформи, такі як Google Play або App Store, можуть мати обмеження щодо типів рекламних матеріалів, які можуть бути використані в додатках.

					КВРІПЗ.190129.19.03.ПЗ	Арк.
						41
Зм.	Арк.	№ докум.	Підпис	Дата		

Крім того, управління рекламними блоками з консолі також полегшує медіацію рекламних мереж. Медіація рекламних блоків дозволяє розробнику працювати з кількома рекламними мережами одночасно і автоматично вибрати найбільш вигідну рекламу для розроблюваного додатка або веб-сайту. Завдяки управлінню з консолі, можна легко налаштувати та керувати медіацією рекламних блоків, вибираючи потрібні рекламні мережі та пріоритети.

Конкретно у проєкті використовуватиметься найпопулярніший тип реклами «Винагородження за показ», тобто під час гри, якщо реклама має замовлення та встигла завантажитись, гравець може отримати бонус за показ. Налаштування рекламного блоку зображено на рисунку 3.4.

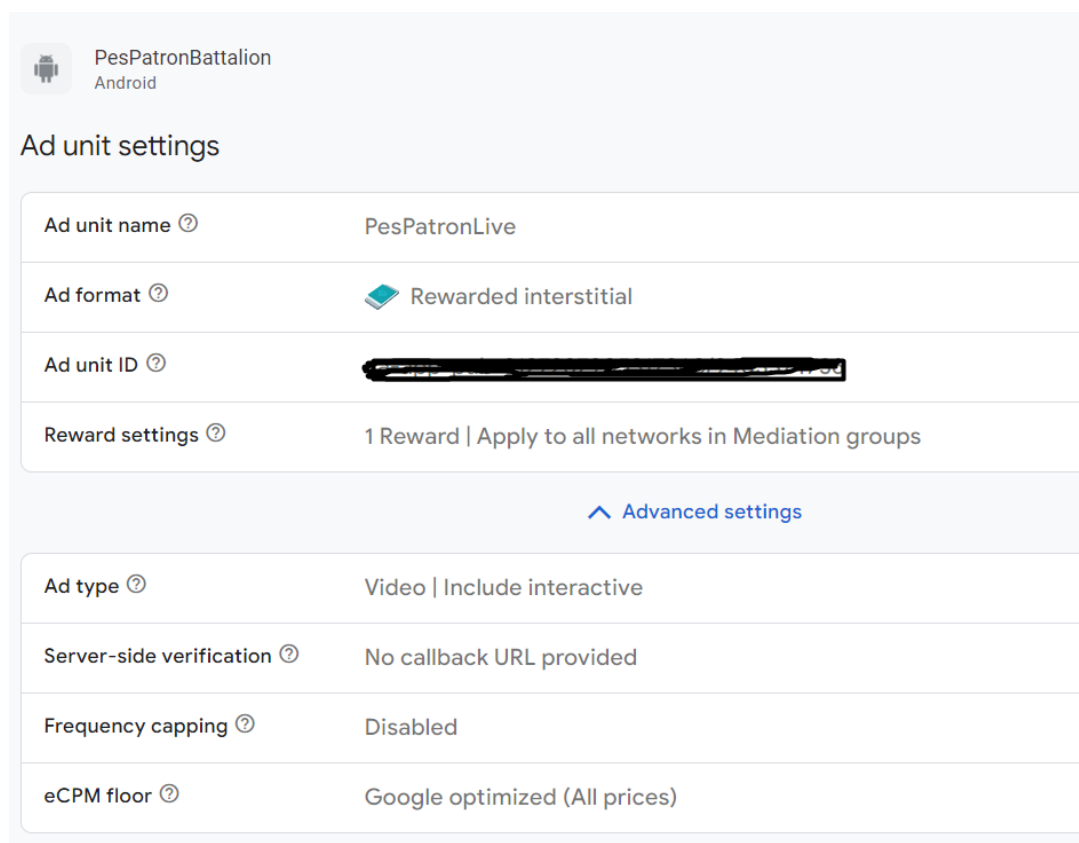


Рисунок 3.3 – Налаштування Rewarded Ad в консолі Admob

Важливо, що за правилами сервісу, для кожної функції потрібно використовувати окремий рекламний блок, навіть якщо тип у них однаковий.

					КВРІПЗ.190129.19.03.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		42

Необхідність в цьому кроці – коректна зв'язність всього функціоналу та аналітики.

Публікація додатку в Google Play Store – це важливий крок для розповсюдження додатку серед користувачів Android-пристроїв. Однак, цей процес має свої тонкощі та вимоги, які потрібно враховувати для успішної публікації.

Перш за все, перед публікацією додатку в Google Play Store, потрібно зареєструвати обліковий запис розробника в Google Play Console. Це веб-платформа, яка дозволяє керувати додатками, включаючи налаштування, публікацію та монетизацію.

Після реєстрації облікового запису розробника, можна створити новий список додатків, де вказати назву, опис, вікові обмеження та інші важливі деталі про додаток. Також можна вибрати категорію, в якій буде розміщений додаток.

Одним з важливих кроків є налаштування реклами в Google Play Console. Тут можна підключити обліковий запис AdMob і налаштувати рекламні блоки, які будуть використовуватися в додатку. Можна вибрати різні типи рекламних блоків, які найкраще відповідають потребам та вимогам.

Крім того, Google Play Console надає можливість використовувати інші сервіси Google для поліпшення додатку. Наприклад, можна інтегрувати Firebase, щоб отримати доступ до різноманітних функцій, таких як аналітика, аутентифікація користувачів, зберігання даних тощо. Також можна використовувати Google Play Games Services для ігрових можливостей, таких як таблиці досягнень та багато іншого.

За допомогою Google Play Console можна керувати всіма цими сервісами з одного місця. Можна переглядати статистику, отримувати звіти про доходи від реклами, проводити А/В-тести, налаштовувати випуск оновлень додатку та багато іншого.

Важливо враховувати, що кожен сервіс може мати свої власні вимоги та інструкції, які потрібно дотримуватися для правильної настройки та

					КВРІПЗ.190129.19.03.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		43

функціонування. Ці всі параметри загалом повинні допомагати просувати цифровий продукт у видачі рекомендацій в магазині.

Під час публікації додатку в Google Play Store, Google може запитувати різні питання для отримання детальної інформації про додаток. Список питань, які необхідно заповнити та матеріалів що потрібно завантажити:

- короткий та довгий опис гри;
- вибір її основного жанру, в моєму випадку: аркада;
- вікові обмеження та цільова аудиторія гри;
- 6 скріншотів, серед яких зображення для планшетів телефонів, та персонального комп’ютера;
- іконка додатку у розмірі 512x512;

Заповнена анкета з графічними матеріалами для гри в Google Play Console зображена на рисунку 3.4.

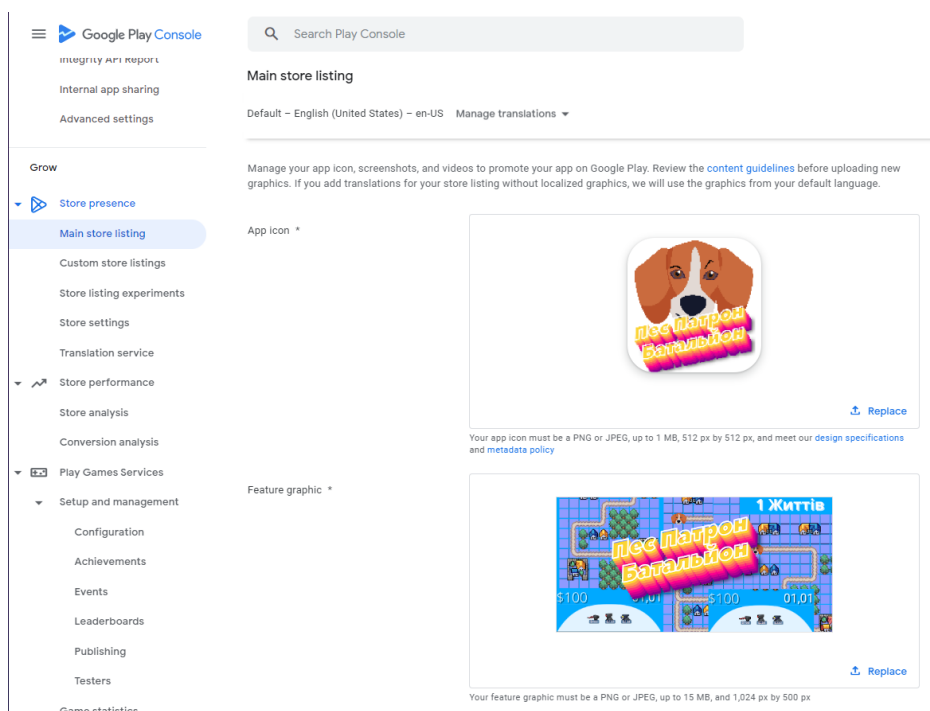


Рисунок 3.4 – Заповнена анкета гри в консолі Google Play

Основна мова додатку – українська, тому потрібно написати його повну назву, розгорнутий та короткий опис продукту. Назва гри повинна бути

					КВРІПЗ.190129.19.03.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		44

привабливою, та відображати її сутність. Вона може бути короткою, але ефективною, захоплюючи увагу потенційних гравців. Важливо також переконатися, що назва не порушує правила магазину та не суперечить авторським правам.

Опис гри має зацікавити гравців і збудити їх бажання спробувати її. Використовуйте привабливі заголовки та підзаголовки, які чітко передають переваги та особливості гри. Важливо описати унікальні функції та аспекти, щоб привернути увагу гравців.

Прикріплення привабливих зображень гри є важливим елементом маркетингової стратегії. Вони допомагають користувачам отримати візуальне уявлення про гру і переконати їх у її якості та цікавості. Варто використати високоякісні зображення з різних ракурсів, щоб продемонструвати різні аспекти ігрового процесу.

Заповнена анкета з текстом сторінки для гри в Google Play Console зображена на рисунку 3.5.

### Listing assets

Check the [Metadata policy](#) and [Help Center guidance](#) to avoid common issues with your store listing. Review all [program policies](#) before submitting your app.

If you're eligible to [provide advance notice](#) to the app review team, contact us before publishing your store listing.

App name *	<input type="text" value="Пес Патрон -Батальйон"/>	This is how your app will appear on Google Play <span style="float: right;">21 / 30</span>
Short description *	<input type="text" value="Пес-Патрон Батальйон - захоплююча аркадна гра у жанрі tower defense."/>	A short description for your app. Users can expand to view your full description. <span style="float: right;">68 / 80</span>
Full description *	<div style="border: 1px solid #ccc; padding: 5px;"><p>Пес-патрон: <u>Захисники Дому</u>" - захоплююча аркадна гра у жанрі tower defense, де гравцеві доведеться взяти участь у захисті свого дому від нашестя ворогів за допомогою вірного і відданого пса-Патрона. Вам належатиме встати на захист вашого дому і відбити хвилю нападників, використовуючи унікальні вміння вашого Пса-патрона.</p></div>	<span style="float: right;">322 / 4000</span>

Рисунок 3.4 – Налаштування текстових матеріалів в магазині

					КВРІПЗ.190129.19.03.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		45



### 3.2 Реалізація бази даних

Робота з базами даних є важливим аспектом розробки ігор, оскільки вона дозволяє зберігати, організувати та отримувати доступ до великих обсягів даних, необхідних для функціонування гри. Бази даних в іграх використовуються для збереження інформації про гравців, їх досягненнях, станах гри, налаштуваннях, а також для забезпечення механізмів збереження прогресу та спільної гри в режимі он-лайн.

Застосування баз даних в ігровій розробці має наступні переваги. По-перше, вони дозволяють зберігати різноманітні дані, такі як профілі гравців, історія гри, статистика, налаштування тощо. Це дозволяє гравцям повертатися до гри з різних пристроїв і зберігати свій прогрес.

Крім того, бази даних використовуються для підтримки он-лайн-ігор та мультиплеєра. Вони дозволяють гравцям взаємодіяти між собою, обмінюватися даними та спільно грати. Бази даних забезпечують безперебійний обмін інформацією між гравцями, а також збереження результатів і прогресу гри.

Firestore Realtime Database – це хмарна база даних, розроблена компанією Google, яка пропонує реальний час синхронізації даних для додатків. Цей сервіс має кілька значних переваг, що роблять його популярним серед розробників ігор та інших додатків. Однією з головних переваг Firestore Realtime Database є його спрощена модель даних. Дані зберігаються у форматі дерева JSON, що дозволяє легко організувати і структурувати дані. Це робить його зручним для зберігання складних структур даних, таких як конфігураційні налаштування, ігровий стан, рейтинги гравців тощо.

Ще однією перевагою Firestore Realtime Database є його здатність до миттєвої синхронізації даних в режимі реального часу. Це означає, що зміни, внесені в базу даних одним користувачем, автоматично відображаються на пристроях інших користувачів, підключених до додатку. Це дозволяє створювати

					КВРІПЗ.190129.19.03.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		47

ігри зі спільною грою, де гравці можуть бачити та взаємодіяти один з одним в реальному часі.

Firestore Realtime Database також відмінно масштабується і забезпечує високу доступність. Сервіс автоматично реплікує дані на кількох серверах, що гарантує, що дані завжди будуть доступні із найменшими можливими затримками. Крім того, Firestore Realtime Database має можливості управління правами доступу, що дозволяє обмежувати доступ до даних та забезпечувати безпеку і конфіденційність, ці правила можна редагувати в консолі. Процес редагування бази даних в консолі Firestore зображено на рисунку 3.5. На сервісі це відбувається у вигляді файлу в який можна вносити зміни редагуючи ключі та створення їм нащадків, обираючи потрібні для зберігання типи даних.

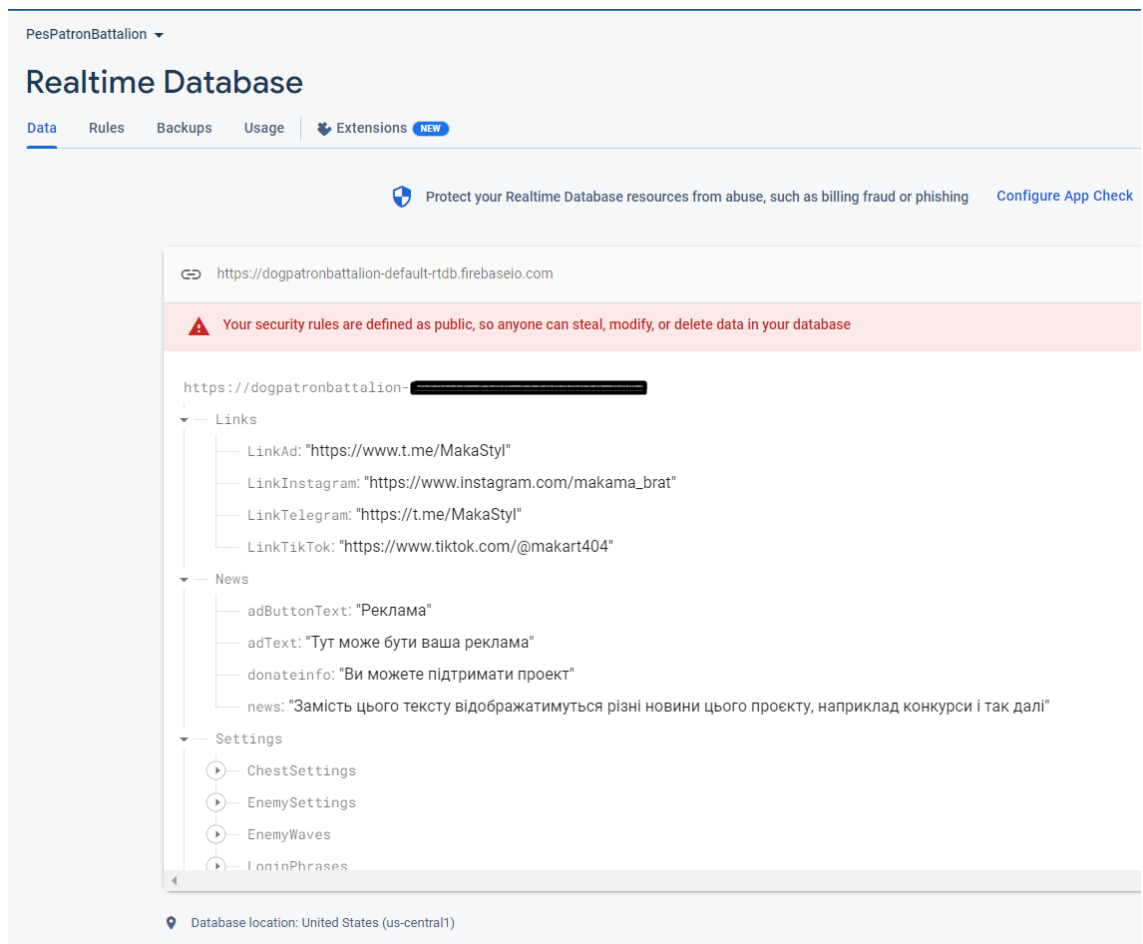


Рисунок 3.5 – Редагування бази даних з консолі Firebase

					КВРІПЗ.190129.19.03.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		48

Однією з переваг Firebase Realtime Database є можливість редагування бази даних графічним способом. Інтуїтивний інтерфейс Firebase Console дозволяє зручно створювати, модифікувати та видаляти дані без необхідності написання складних запитів. Це особливо корисно для розробників, які не мають глибоких знань SQL або інших мов запитів.

Проте, навіть при зручному графічному інтерфейсі роботи з Firebase Realtime Database, рекомендується враховувати особливості проекту і створити схему бази даних, щоб визначити структуру таблиць і зв'язки між ними. У цьому проекті загалом в базі даних має бути в наявності 4 таблиці. Перша таблиця містить список користувачів з їх властивостями. Наступна таблиця містить данні кнопок соціальних мереж. Третя відповідає за тексти, які відображатимуться в новинах проекту. І остання відповідає за налаштування балансу гри. Побудована Діаграма схема даних бази даних зображена на рисунку 3.6.

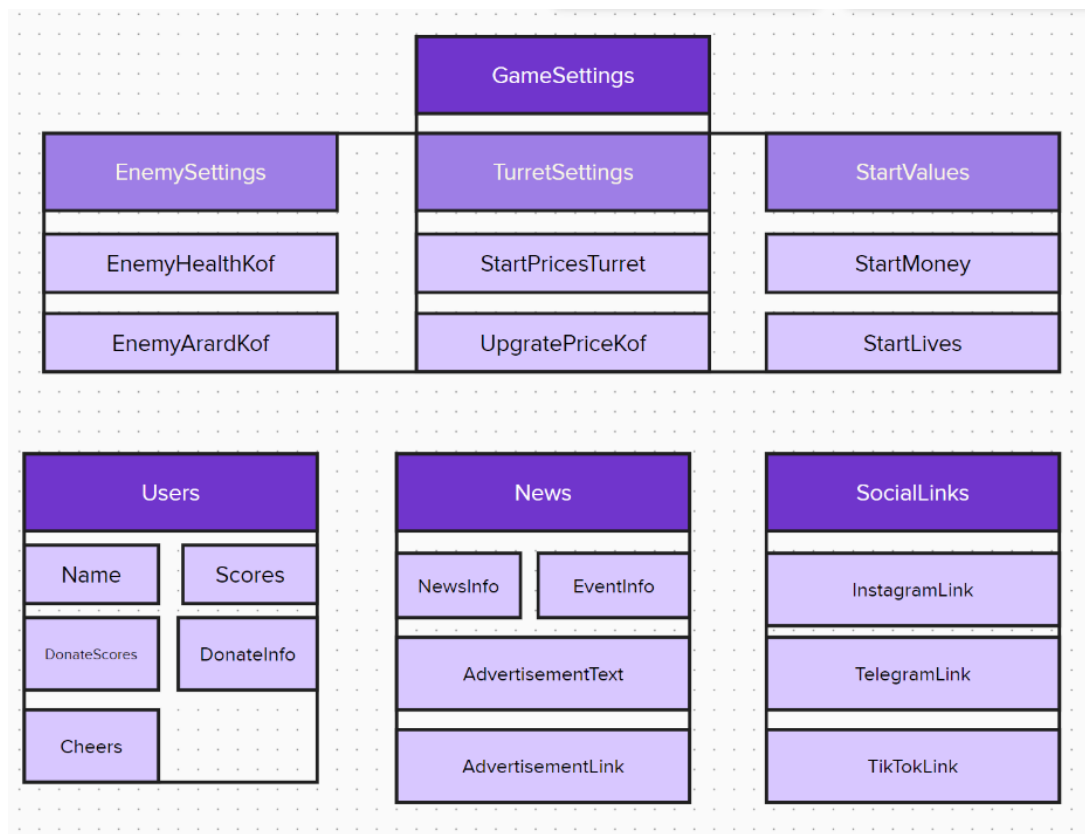


Рисунок 3.6 – Схема даних діаграм

### 3.3 Реалізація ігрового процесу та його балансування

У грі жанру Tower Defense основою ігрового процесу є будівництво турелей гравцем, які призначені для знищення ворожих сил. Гравець має можливість поліпшувати турелі за певну ціну. При знищенні ворогів гравець отримує винагороду. З кожною наступною хвилиною ворогів їх кількість зростає, а також збільшуються їх параметри, такі як життя та ціна за знищення. Таким чином, гра поступово стає складнішою і вимагає від гравця аналітичного мислення та стратегічних рішень.

Окрім повного програмного циклу, гравець також повинен урахувати, куди розміщувати турелі. Важливими факторами вибору турелі є ціна та її ефективність. Гравець повинен збалансувати свої фінансові можливості з потребою в потужних турелях, щоб впоратися зі зростаючою складністю гри.

Крім того, гравець має можливість використовувати бонуси. Кожні 60 секунд гравець може отримати 2 бонуси – один за перегляд реклами, інший надається безкоштовно. Ці бонуси впливають на ігровий процес, надаючи гравцеві додаткові переваги або ресурси для подальшої стратегії.

Враховуючи всі ці параметри, важливо розробити формули, які враховуватимуть кожен з них.

Наприклад, формула для розрахунку ефективності турелі може враховувати вартість покращень та їх вплив на ворожі сили. Формули для зростання складності гри можуть базуватися на відсотку збільшення кількості ворогів та їх параметрів з кожною новою хвилиною. Дослідження та розвиток таких формул дозволяють досягти балансу в грі та створити випробування, яке буде цікавим та захоплюючим для гравців.

Один з ключових аспектів гри в жанрі Tower Defense полягає у зберіганні параметрів в он-лайн базі даних, такій як Firebase Database. Система забезпечує зручний та надійний механізм для зберігання даних великої кількості гравців у реальному часі.

					КВРІПЗ.190129.19.03.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		50



В збалансованому ігровому процесі важливо встановити середню кількість часу, яку гравець витратить на одну спробу. Ця загальна тривалість повинна бути в межах приблизно 5 хвилин. Водночас, останні 2 хвилини гри повинні бути найбільш інтенсивними та вимагати від гравця більше уваги та навичок.

Для більш детального аналізу параметрів та їх впливу на хід гри, корисно використовувати графічне зображення. На рисунку 3.7 вказано, що місця для турелей обмежені, покращення турелей, та їх шкода ворогам теж обмежені, все веде до того, щоб гравець все ж таки програв. Проте це не важливо, це гра на рекорд, на результат, і кожна спроба це перемога!



Рисунок 3.7 – Інтерфейс Unreal Engine

### 3.4 Забезпечення безпеки даних

Firebase дозволяє прописувати правила доступу до бази даних за допомогою Firebase Realtime Database Rules. Ці правила можна визначати за допомогою спеціальної скриптової мови, яка є унікальною для Firebase і називається Firebase Security Rules Language (або просто Security Rules Language).

Firebase Security Rules Language є спеціальною мовою, розробленою командою Firebase, щоб забезпечити безпеку та контроль доступу до бази даних. Ця мова використовується для визначення правил доступу до окремих шляхів у базі даних, включаючи можливості читання, запису, оновлення та видалення даних. Головна мета використання правил Firebase Realtime Database полягає в забезпеченні безпеки та захисту даних, а також контролі доступу до бази даних. Це дозволяє розробникам точно налаштувати, які користувачі або групи користувачів мають доступ до певних даних, які дії вони можуть виконувати (наприклад, читати, записувати, оновлювати) і які обмеження повинні застосовуватися. Використання правил Firebase Realtime Database дозволяє забезпечити конфіденційність даних, уникнути несанкціонованого доступу, зменшити ризик втрати даних та забезпечити відповідність правилам безпеки даних. Крім того, це дає можливість розробникам керувати доступом до бази даних залежно від потреб своєї програми чи додатка.

У базі даних, яка розроблена для проекту, застосовані такі правила, які дозволяють користувачу маніпулювати своїми особистими даними, такими як опис облікового запису, ім'я, та кількість балів. Він може змінювати ці дані відповідно до своїх потреб та вподобань.

Однак, є деякі обмеження щодо зміни донатних балів користувачем. Донатні бали є спеціальна оцінка, яка може бути змінена тільки адміністратором у профілі користувача. Це зроблено для забезпечення правильності та контролю за процесом додавання цих балів. Тому, змінна цих параметрів доступна тільки

					КВРІПЗ.190129.19.03.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		53

для зчитування користувачем, щоб він міг переглядати свій поточний бал, але не мав можливості самостійно їх змінювати.

Крім того, існують різні настройки ігрової системи, які також доступні тільки для зчитування користувачем. Ці настройки визначають параметри геймплею і корегуються адміністратором з панелі управління. Це зроблено для забезпечення рівноваги та уникнення можливості недобросовісного змінювання цих настройок з боку користувача.

На жаль, навіть при використанні бази даних Firebase існує можливість шахрайства та неправильного змінення гравцями гральних параметрів під час виконання програми. Це може статися за допомогою спеціального програмного забезпечення, яке шукає схожі за значенням ігрові параметри і змінює їх під час процесу гри. Такі шахрайські програми можуть використовувати пошук значень, аналізувати відображення ігрових параметрів в диспетчері задач або змінювати значення безпосередньо в пам'яті процесу гри. Це дає можливість гравцям змінювати кількість балів, грошей або життів, що впливає на їх результати та прогрес у грі. Однак, з метою захисту від такого шахрайства, варто розробити механізми, що контролюють та захищають гральні дані.

Конкретно у розробленій грі перед записом кількості балів у базу даних здійснюється перевірку, щоб впевнитися, що гравець може досягти такої кількості, враховуючи час, витрачений на гру.

Наприклад, якщо ви знаєте, що новий ворог появляється кожні 30 секунд, а гравець отримує 10 балів за знищення кожного ворога, можна обчислити максимальну оцінку партії, яку можна набрати за певний час гри, враховуючи навіть деякі випадкові ігрові події, кожна хвиля все одно має свій максимум

Аналогічні обмеження можна застосовувати і до інших ігрових параметрів. Наприклад, є обмеження на кількість життів що має гравець, та час бонусу який його дає. Перед записом цієї кількості до бази даних можна перевірити, чи не перевищує гравець максимально допустимий обсяг ресурсів грошей, здоров'я, турелі тощо.

					КВРІПЗ.190129.19.03.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		54

### 3.5 Тестування

У тестуванні в Unity існує кілька видів тестування, які можуть бути використані для забезпечення якості програмного продукту.

Один з основних видів тестування – це модульне тестування. Воно спрямоване на перевірку окремих компонентів або модулів коду, що дозволяє виявляти та виправляти помилки на ранніх етапах розробки. Модульне тестування зазвичай проводиться незалежно від інших компонентів програми, що дозволяє ізолювати проблему та зосередитися на конкретній функціональності. Це забезпечує більшу стабільність та надійність коду.

Ще один вид тестування – це інтеграційне тестування. Воно використовується для перевірки взаємодії між різними компонентами системи. Інтеграційне тестування дозволяє виявити можливі проблеми, які виникають при злитті різних компонентів та їх спільній роботі. Цей вид тестування допомагає забезпечити, що різні частини програмного продукту працюють разом правильно та без проблем.

Також, важливим видом тестування є тестування на реальних пристроях або фізичних пристроях. Це включає тестування програмного продукту на різних пристроях, таких як мобільні телефони, планшети або віртуальні реальності, для перевірки сумісності та виявлення проблем, які можуть виникнути на конкретних пристроях. Тестування на реальних пристроях допомагає забезпечити, що програмний продукт працює на різних платформах із заданою якістю та функціональністю.

Однією з переваг тестування в Unity є забезпечення якості програмного продукту. Тестування дозволяє виявляти та виправляти помилки ще до випуску програми, що робить її більш стабільною та надійною. Воно також допомагає забезпечити правильну роботу різних компонентів програми та їх взаємодію. Крім того, тестування на реальних пристроях дозволяє перевірити сумісність та

					КВРІПЗ.190129.19.03.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		55

взаємодію програми з різними пристроями, що забезпечує задоволення користувачів та підвищує репутацію програмного продукту.

Отже, застосування різних видів тестування в Unity допомагає забезпечити якість, стабільність та надійність програмного продукту, а також забезпечує задоволення користувачів та підвищує його конкурентоспроможність. У грі про пса Патрона були використані різні види тестування для забезпечення якості програмного продукту, і результати були дуже задовільними.

Наприклад, для перевірки окремих компонентів та модулів коду було проведено модульне тестування. Це дозволило виявити й виправити помилки на ранніх етапах розробки. Модульне тестування допомогло забезпечити стабільність та надійність коду, що має велике значення для успішної роботи гри.

Також було проведено інтеграційне тестування для перевірки взаємодії різних компонентів гри. Це дозволило виявити та усунути можливі проблеми, які можуть виникнути при спільній роботі компонентів. Інтеграційне тестування гарантує, що різні частини гри працюють разом правильно та без проблем.

Крім того, система була протестована на реальних пристроях, таких як мобільні телефони та планшети, щоб перевірити сумісність та виявити можливі проблеми на конкретних пристроях. Тестування на реальних пристроях підтвердило, що гра працює на різних платформах з високою якістю та функціональністю.

Застосування цих різних видів тестування у грі Tower Defense дозволило забезпечити якість, стабільність та надійність гри. Всі тести пройшли успішно, що говорить про високий рівень якості програмного продукту та гарантує задоволення користувачів, які зможуть насолоджуватись грою без технічних проблем.

У додаток до різних видів тестування, в грі також була використана консоль Firebase для відстеження активності користувачів. Ця інструментальна платформа дозволила нам отримати детальну статистику про правильну роботу даних у грі.

					КВРІПЗ.190129.19.03.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		56

### 3.6 Інструкція користувача

Для успішної експлуатації гри потрібно зробити наступні кроки:

- відкрити магазин Google Play, знайти гру "Юніті Аркада: Tower Defense" і натиснути кнопку "Встановити";
- знайти іконку гри на своєму пристрої та натиснути на неї, щоб відкрити гру;
- після запуску гри користувач потрапить в головне меню. Тут можна увійти у свій обліковий запис, або створити новий, також можна виключити звук у грі, або перейти у соціальні мережі. Після входу можна почати гру;
- ваше завдання в грі – захистити свою територію від ворогів, розташовуючи турелі для атаки на їхній шлях;
- розмістіть турелі, тапнувши на відповідне місце на екрані. Вибрати тип башти та її місце розташування з розумом, щоб ефективно стримувати ворогів;
- щоб поставити турель потрібно один раз тапнути на обраний тип турелі в нижній панелі магазину, варто впевнитись, що на балансі вистачає ресурсів для її побудування;
- окрім того, щоб покращити турель, потрібно клікнути її один раз, після кліку з'явиться панель на якій є кнопки «Покращити» та «Продати». Обираємо потрібну нам;
- гра автоматично зберігає найкращі досягнення;
- є змога переглянути он-лайн таблиці рекордів до яких можна перейти через головне меню;
- у цьому ж меню можна обрати налаштування профілю де є можливість змінити ім'я гравця та його аватар.

					КВРІПЗ.190129.19.03.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		57

### 3.7 Вимоги до апаратно-програмних засобів

Для коректного успішного використання ігрового додатку, пристрій повинен відповідати вимогам:

- операційна система: Android або iOS;
- мінімальна версія операційної системи: Android 5.0 або iOS 10.0;
- процесор: мінімум двоядерний процесор;
- оперативна пам'ять: мінімум 2 ГБ RAM;
- вільне місце на пристрої: залежно від розміру гри, рекомендується мати не менше 500 МБ вільного місця;
- батарея: рекомендується наявність зарядженої батареї для тривалого ігрового процесу;
- Інтернет-з'єднання: для підключення до мережі для змагань з іншими гравцями та синхронізації рекордів.

На рисунку 3.8 зображено приклад пристрою що є цільовим та задовольняє всі вимоги.

#### ХАРАКТЕРИСТИКИ:

Тип	Смартфони
Бренд	Redmi
Серія	Xiaomi Redmi Note 11 4G
Тип екрану	AMOLED
Діагональ екрану	6.5"
Роздільна здатність екрану	2400x1080
Процесор	MediaTek Helio G88
Кількість ядер	8
Графічний чіп	Mali-G52 MC2
Об'єм оперативної пам'яті	6ГБ
Об'єм накопичувача	128ГБ
Основна камера	50 + 8 + 2 Мп
Об'єктив основної камери	Ширококутний (стандартний) / Ультраширококутний / Макро
Запис відео	1080p FullHD (30fps)
Фронтальна камера	8 Мп
Бездротовий зв'язок	Wi-Fi / Bluetooth 5.1

Рисунок 3.8 – Характеристики цільового пристрою

					КВРІПЗ.190129.19.03.ПЗ	Арк.
						58
Зм.	Арк.	№ докум.	Підпис	Дата		

### 3.8 Висновки

У розділі було описано послідовність задач, які були вирішені в процесі розробки застосунку Tower Defense. Було успішно виконано підключення до проекту сервісів Unity, таких як Firebase, AdMob та Google Play services. Це дозволило отримати доступ до різноманітних функцій та можливостей цих сервісів, які були успішно використані в застосунку. Далі, було проведено аналіз різних способів захисту бази даних для забезпечення безпеки і цілісності інформації. Результати аналізу допомогли обрати найефективніші заходи безпеки для нашого застосунку, забезпечуючи надійний захист даних користувачів.

Окрім того, було проведено тестування роботи цих сервісів, включаючи Firebase, AdMob та Google Play services, щоб переконатися в їхній правильній роботі та взаємодії з іншими компонентами застосунку. В ході тестування було виявлено, що ці сервіси працюють бездоганно і відповідають нашим вимогам щодо функціональності та продуктивності.

Додатково, важливо зазначити, що підключення до сервісів Unity, таких як Firebase, AdMob та Google Play services, відіграє важливу роль у створенні монетизаційної моделі проекту Tower Defense.

Після завершення розробки та перед запуском, було проведено перевірку та затвердження проекту адміністрацією платформи. Цей процес мав на меті переконатися, що проект відповідає всім вимогам та стандартам безпеки, етики та якості. Після успішної перевірки проекту адміністрацією, весь функціонал і можливості, включаючи підключені сервіси, були активовані та вступили в роботу.

В результаті виконаних завдань з підключення до сервісів Unity, аналізу безпеки бази даних та протестування роботи сервісів було забезпечено успішну розробку застосунку Tower Defense з використанням надійних технологій та функціональних можливостей.

					КВРІПЗ.190129.19.03.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		59

## ВИСНОВКИ

Під час розробки гри Аркади на Unity у піджанрі Tower Defense для мобільних платформ Android та IOS, було проведено широкий спектр досліджень та виконано різноманітні завдання з метою створення високоякісного та успішного продукту. Основними компонентами досліджень та розробки були:

– вивчення предметної області та актуальних тенденцій: Було проведено аналіз сучасних ігрових трендів, піджанру Tower Defense, а також досліджено актуальність теми гри про пса Патрона;

– аналіз конкурентних рішень: було проведено огляд існуючих ігрових продуктів у жанрі Аркада, зосереджуючись на їх основних перевагах та недоліках. Цей аналіз допоміг визначити ключові елементи та особливості, які дозволяють їм бути успішними;

– встановлення завдань та вимог: були сформульовані конкретні цілі проекту, включаючи дослідження монетизації проекту, розробку системи реєстрації, створення он-лайн таблиці рекордів, інтеграцію системи донатів та нагородження гравців та синхронізацію конверсій кліку гравців по соціальним мережам, а також варіант вирішення проблеми зі зміною геймплею без оновлення додатку;

– розробка функціональності: було реалізовано різноманітні елементи гри, включаючи ігровий процес, систему пса-патрона, різні типи ворогів та турелей, систему управління та багато іншого.

В результаті розробки була створена крос платформна мобільна гра, яка пропонує захопливий ігровий процес у стилістичному оформленні про пса-Патрона.

Проект впроваджений і готовий до монетизації шляхом інтеграції системи донатів, Admob та залученням гравців через он-лайн таблицю рекордів та соціальні мережі.

					КВРІПЗ.190129.19.03.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		60

## ПЕРЕЛІК ДЖЕРЕЛ

1. "The Art of Game Design: A Book of Lenses" by Jesse Schell - – Режим доступу: <https://www.amazon.com/Art-Game-Design-Lenses-Third/dp/1138632058> (дата звернення – 2023-04-02).
2. "Game Programming Patterns" by Robert Nystrom - – Режим доступу: <https://gameprogrammingpatterns.com/> (дата звернення – 18 квітня 2023).
3. "Unity in Action: Multiplatform Game Development in C#" by Joe Hocking - – Режим доступу: <https://www.manning.com/books/unity-in-action-second-edition> (дата звернення – 2023-05-22)
4. "The Art of Game Design: A Book of Lenses" by Jesse Schell USA, 2008; pp 100-110.
5. "Game Engine Architecture" by Jason Gregory - – Режим доступу: <https://www.amazon.com/Game-Engine-Architecture-Jason-Gregory/dp/1568814135> (дата звернення – 2023-05-01)
6. Game Informer – Режим доступу: <https://www.gameinformer.com/> (дата звернення - 2023-05-20)
7. Unity Technologies Blog – Режим доступу: <https://blogs.unity3d.com/> (дата звернення - 2023-05-21)
8. GDC (Game Developers Conference) – Режим доступу: <https://gdconf.com/> (дата звернення - 2023-05-22)
9. Unity Asset Store Forum – Режим доступу: <https://forum.unity.com/forums/asset-store.82/> (дата звернення - 2023-05-23)
10. GameDev.tv - Learn Game Development – Режим доступу: <https://www.gamedev.tv/> (дата звернення - 2023-05-24)
11. Найважливіші архітектурні шаблони, які необхідно знати – Режим доступу: <https://devzone.org.ua/post/nayvazhlivishi-arkhitekturni-shablonyakineobkhidno-znati> (дата звернення - 2023-04-24)

					КВРІПЗ.190129.19.03.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		61

12. Game Jolt - Indie Games for the Love of It – Режим доступу: <https://gamejolt.com/> (дата звернення - 2023-05-26)
13. Unity Answers - QA for Unity Developers – Режим доступу: <https://answers.unity.com/> (дата звернення - 2023-05-27)
14. GameAnalytics - Player Analytics for Games – Режим доступу: <https://gameanalytics.com/> (дата звернення - 2023-05-28)
15. "Game Programming Patterns" by Robert Nystrom USA, 2014; pp 200-220
16. "Game Design Workshop: A Playcentric Approach to Creating Innovative Games" by Tracy Fullerton USA, 2014; pp 250-270
17. "The Game Developer's Dictionary: A Multidisciplinary Lexicon for Professionals and Students" by Dan Roy USA, 2019; pp 180-200.
18. "Introduction to Game Design, Prototyping, and Development: From Concept to Playable Game with Unity and C#" by Jeremy Gibson Bond USA, 2013; pp 250-270.
19. "Game Engine Architecture" by Jason Gregory USA, 2014; pp 400-420.
- Brathwaite, B., & Schreiber, I. (2009). Challenges for Game Designers. Boston, MA: Course Technology.
20. Salen, K., & Zimmerman, E. (2004). Rules of Play: Game Design Fundamentals. Cambridge, MA: The MIT Press.
21. Adams, E., & Rollings, A. (2007). Fundamentals of Game Design. Upper Saddle River, NJ: Prentice Hall.
22. Fullerton, T. (2008). Game Design Workshop: A Playcentric Approach to Creating Innovative Games. Burlington, MA: Morgan Kaufmann.
23. Rabin, S. (Ed.). (2005). AI Game Programming Wisdom. Boston, MA: Course Technology.
24. Moore, M. (2008). Introduction to the Theory of Computation. Upper Saddle River, NJ: Prentice Hall.
25. Bates, B. (2004). Game Design. Thomson Learning.

					КВРІПЗ.190129.19.03.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		62

26. Crawford, C. (2003). Chris Crawford on Game Design. Indianapolis, IN: New Riders.
27. Rollings, A., & Adams, E. (2003). Andrew Rollings and Ernest Adams on Game Design. Indianapolis, IN: New Riders.
28. Koster, R. (2005). A Theory of Fun for Game Design. Scottsdale, AZ: Paraglyph Press.
29. Bateman, C., & Boon, R. (2014). 21st Century Game Design. Wellesley, MA: AK Peters/CRC Press.
30. Schuyttema, P. (2010). Real-Time Shadows. Wellesley, MA: AK Peters/CRC Press.
31. Gamasutra - The Art & Business of Making Games – Режим доступу: <https://www.gamasutra.com/> (дата звернення - 2023-06-01)
32. GameSpot – Режим доступу: <https://www.gamespot.com/> (дата звернення - 2023-06-02)
33. Kotaku – Режим доступу: <https://kotaku.com/> (дата звернення - 2023-06-03)
34. Rock, H. (2003). Tales from the Screen: Endings and Beginnings. Boston, MA: MIT Press.
35. Juul, J. (2010). A Casual Revolution: Reinventing Video Games and Their Players. Cambridge, MA: The MIT Press.
36. Swink, S. (2009). Game Feel: A Game Designer's Guide to Virtual Sensation. Boston, MA: Morgan Kaufmann.
37. El-Nasr, M. S., Drachen, A., & Canossa, A. (2018). Game Analytics: Maximizing the Value of Player Data. New York, NY: Springer.
38. Hunicke, R., LeBlanc, M., & Zubek, R. (2004). MDA: A Formal Approach to Game Design and Game Research. Proceedings of the AAAI Workshop on Challenges in Game AI, 1-5.

					КВРІПЗ.190129.19.03.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		63

39. Squire, K. (2006). From Content to Context: Videogames as Designed Experience. *Educational Researcher*, 35(8), 19-29. Salen, K., & Zimmerman, E. (2003). *The Rules of Play: Game Design Fundamentals*. Cambridge, MA: The MIT Press.
40. Rollings, A., & Morris, D. (2000). *Game Architecture and Design*. Indianapolis, IN: New Riders.
41. Egenfeldt-Nielsen, S., Smith, J. H., & Tosca, S. P. (2016). *Understanding Video Games: The Essential Introduction*. New York, NY: Routledge.
42. Schell, J. (2008). *The Art of Game Design: A Deck of Lenses*. Pittsburgh, PA: ETC Press.
43. Adelson, B., & Lynch, T. (2019). *Unreal Engine 4 Game Development in 24 Hours*. Indianapolis, IN: Sams Publishing.

					КВРІПЗ.190129.19.03.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		64

ДОДАТОК А  
(обов'язковий)

**ТЕХНІЧНЕ ЗАВДАННЯ**

## **Введення**

Робота виконується в рамках проекту «Кросплатформна гра для мобільних платформ у жанрі «Аркада» на Unity»

### **1 Підстава для розробки**

Підставою для розробки є «Завдання на кваліфікаційну роботу», затверджене завідувачем кафедри інженерії програмного забезпечення.

Найменування розробки: Кросплатформна гра для мобільних платформ у жанрі «Аркада» на Unity»

### **2 Призначення розробки**

Функціональні вимоги включають: геймплей, аутентифікацію, систему донатів, синхронізація конверсій. Кожен з цих етапів повинен бути виконаний та протестований практично.

Геймплей:

- Розробка системи Tower Defense, включаючи ворогів, турелі, рівні складності, тощо;
- Реалізація ігрового процесу з можливістю розміщення баштурок для захисту від ворогів;
- Розробка системи керування грою, включаючи тапи, свайпи, тощо;
- Визначення різних типів ворогів з різними характеристиками та здатностями.

#### Антифікація:

- Розробка системи аутентифікації для запобігання шахрайству та читерству в грі;
- Розробка системи он-лайн таблиці рекордів для збереження найкращих досягнень гравців;
- Інтеграція з веб-сервером для зберігання та оновлення даних рекордів;

#### Система донатів винагороджень гравцям:

- Розробка системи донатів для надання винагород гравцям за підтримку проекту, де гравець може обрати аватар за свій донат;

#### Синхронізація конверсій кліків гравців по соціальним мережам:

- Розробка системи синхронізації конверсій кліків гравців зі сторонніми соціальними мережами;

#### Технічні вимоги:

- Платформи: Android та iOS;
- Мова програмування: Unity та C#;
- Веб-сервер: Інтеграція з веб-сервером для збереження та оновлення даних он-лайн таблиці рекордів;

#### Вимоги до завершення проекту:

- Готовий продукт повинен бути впроваджений та готовий до монетизації;

Функціональні вимоги повинні бути належним чином реалізовані та протестовані.

## ДОДАТОК Б (обов'язковий)

### ПРОГРАМНИЙ КОД ОСНОВНИХ МОДУЛІВ

#### Г.1 Програмний код модуля FirebaseManager

```
using Firebase.Analytics;
using Firebase.Database;
using System;
using System.Collections;
using System.Collections.Generic;
using System.Threading.Tasks;
using UnityEngine;
using UnityEngine.UI;

public class FirebaseInit : MonoBehaviour
{
    DatabaseReference reference;
    public FirebaseAuth firebaseAuth;

    //public TextMeshProUGUI textMeshProUGUI;
    //public TextMeshProUGUI textMeshProUGUI2;

    private bool isDependenciesChecked = false;

    private void Start()
    {
        reference = FirebaseDatabase.DefaultInstance.RootReference;

        InitializeFirebase();
    }

    private async void InitializeFirebase()
    {
        await WaitForDependencies();
        Debug.Log("Firebase dependencies initialized");
        Application.logMessageReceived += LogCallback;
        FirebaseAnalytics.LogEvent("custom_progress_event", "percent",
0.4f);

        firebaseAuth.AuthGoogle();
    }
}}
```

## Г.2 Програмный код модуля GameManager

```
public GameObject gameReadyMenu;
public GameObject gamePauseMenu;
public GameObject completeLevelUI;
public GameOver gameOver;
public bool isGamePlay;
public PlayerStats playerStats;
public WaveSpawner waveSpawner;
public BuildManager buildManager;
public PauseMenu pauseMenu;
public ChestTimerProg chestAd;
public ChestTimerProg chest;
public GameplaySettings gamePlaySettings;
public SoundManager soundManager;
public FirebaseManager firebaseManager;
public Shop shop;
private bool downedGame;
private int countCheck;
private void Awake()

// Update is called once per frame
void Update()
{
    if(isGamePlay && PlayerStats.Lives == 0) {
        EndGame();
    }
}

private void OnEnable()
{
    makeDefaultStart();
}

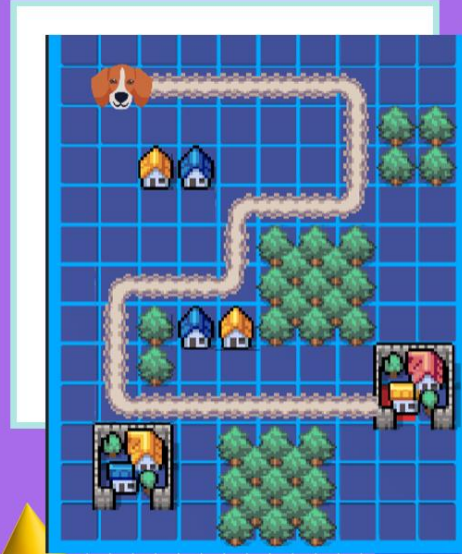
void EndGame () {
    waveSpawner.makeAsDefault();
    buildManager.clearTurrets();
    waveSpawner.clearAllEnemies();
    gameOver.makeGameOver();
    chestAd.makeDefault();
    chestAd.makeDefault();
    waveSpawner.makeAsDefault();
    GameIsOver = true;
    isGamePlay = false;
    gameReadyMenu.SetActive(true);
    soundManager.playLoseSound();
    if (!downedGame)
    { firebaseManager.updateScoreboardBest(PlayerStats.Scores); }
    StopAllCoroutines();
}
```

ДОДАТОК В  
(обов'язковий)

**ПРЕЗЕНТАЦІЙНІ МАТЕРІАЛИ**

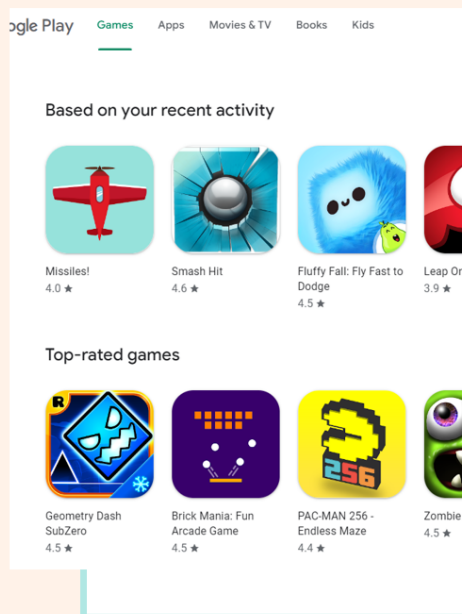
# Unity Game Development

КВАЛІФІКАЦІЙНА РОБОТА на тему  
Кросплатформна гра для мобільних платформ  
у жанрі «Аркада» на рушіві Unity  
Виконав студент групи ПЗ-19-1  
Василенко Максим  
Керівник канд. пед. наук, доцент Н. І.  
Праворська



Слайд 1 – Титулка

## Постановка задачі



Головною метою при виконанні кваліфікаційної роботи є розробка кросплатформної гри з сучасним функціоналом, у якому передбачено авторизацію користувачів, порівняння їх результатів, а також монетизація проекту з імплантацією сервісів Firebase для відслідковування конверсій

Для досягнення цих цілей слід :

- Дослідити сферу гейм дизайну ;
- Визначити актуальну тему для додатку
- Обрати плаґаґіни та графіку для гри
- Розробити гру використовуючи додаткові інструменти
- Підключити Firebase SDK та Admob SDK
- Зробити публікацію гри в Google Play

Слайд 2 – Постановка задачі

## Актуальність



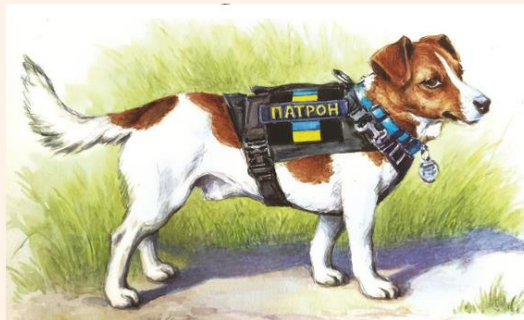
Слайд 3 – Актуальність теми

## Вибір тематики плагінів та піджанру



Tower Defense, TD, або **Захист вежами** – досить відомий піджанр відеоігор. В іграх цього жанру гравець зазвичай має єдине завдання — відбити напад ворогів, які наступають хвиля за хвилиною. Для цього в перших іграх жанру, які й дали йому цю назву, гравець будував різноманітні вежі на заздалегідь визначених ключових ділянках.

У зв'язку з подіями в країні воєнними подіями – тематика патріотизма є досить актуальною, і була обрана для втілення у життя. За сюжетом гри Батальйон пса Патрона захищається від навали ворогів



Слайд 3 – Вибір тематики та піджанру

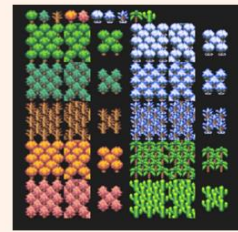
# Використані плагіни

NavMesh – Плагін для пошуку шляхів для штучного інтелекту.

Tile Editor – використання тайлової графіки для декоризації рівнів

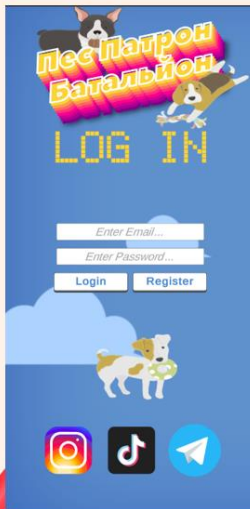
Google Admob – монітезація ігрового додатку

Firebase – Створення авторизації, бази даних, та аналітики і відслідковування конверсій



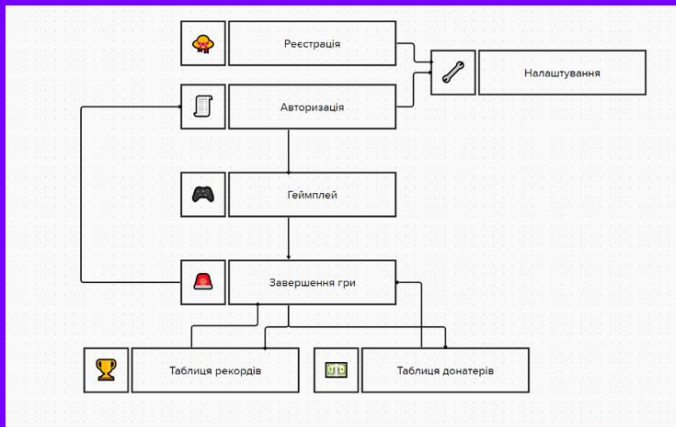
Слайд 4 – Використані додаткові засоби

# Створення інтерфейсу додатку

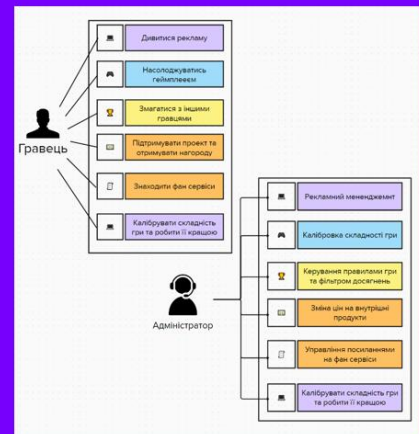


Слайд 5 – Проектування та створення інтерфейсу

# Проектування життєвого циклу та модулів

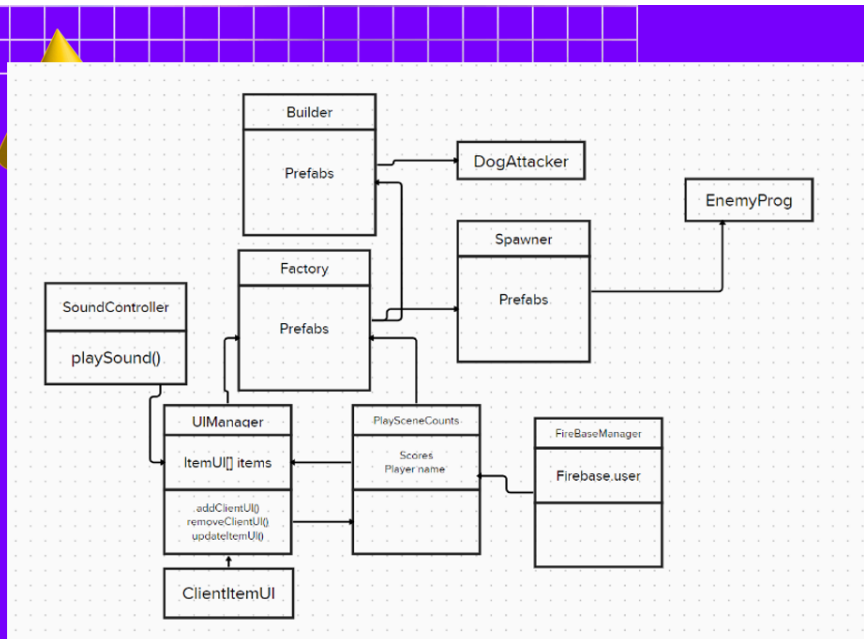


**Життєвий цикл**



**Варіанти використання**

Слайд 6 – Життєвий цикл та діаграма варіантів використання



**Діаграма класів**

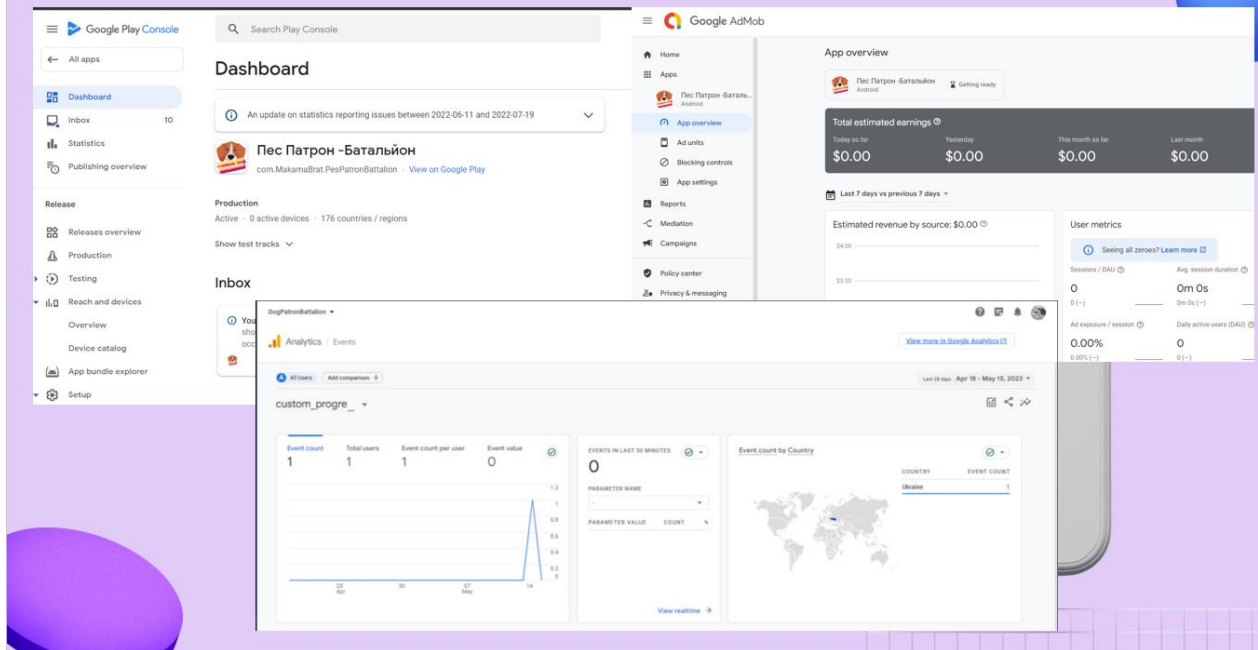
Проектування діаграми Класів передбачає використання 4 паттернів програмування

- Синглетон
- Пулова система
- Фабрика
- Стан

Кожен з них має свої цілі та концепцію для застосування.

Слайд 7 – Діаграма класів

## Публікація, налаштування кабінетів



Слайд 7 – Публікація, налаштування кабінетів

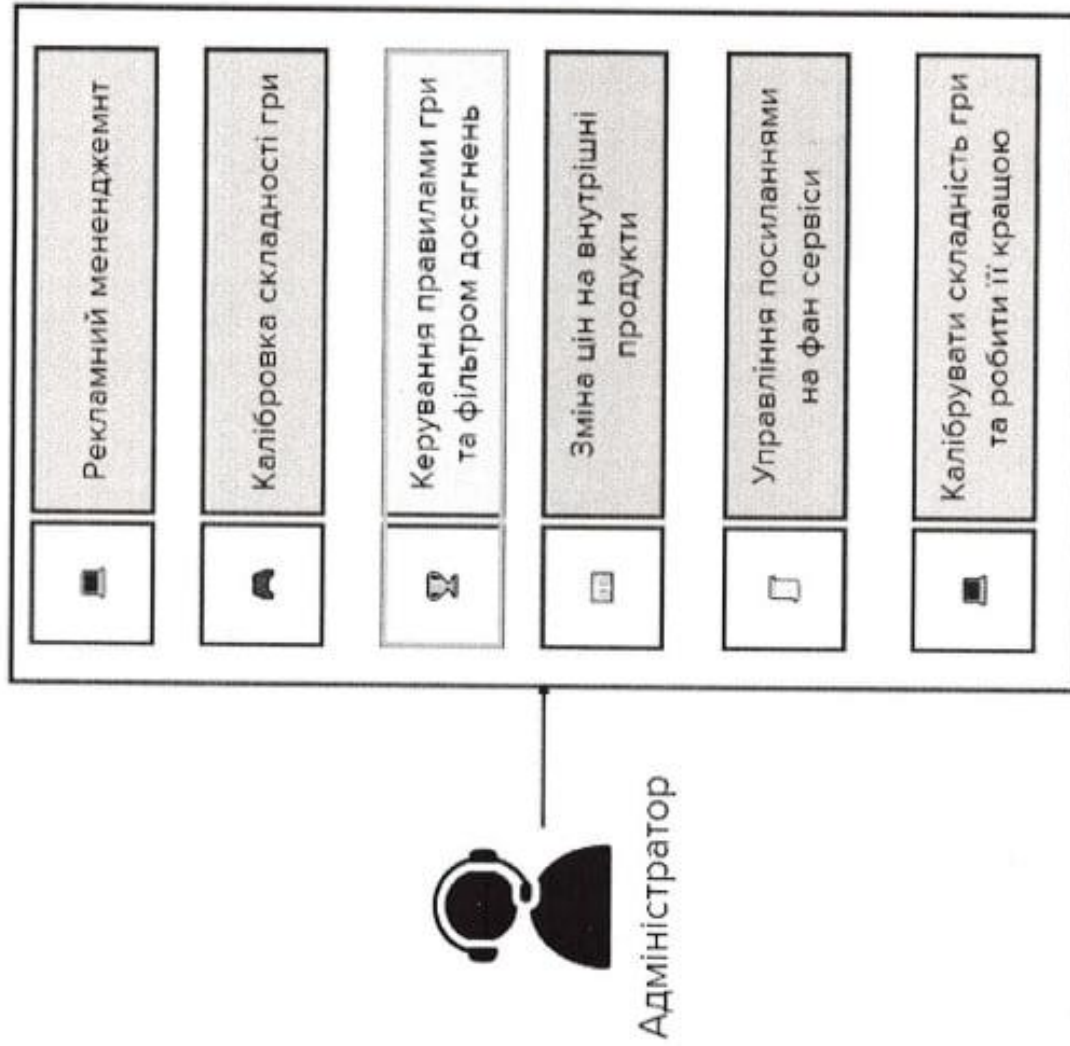
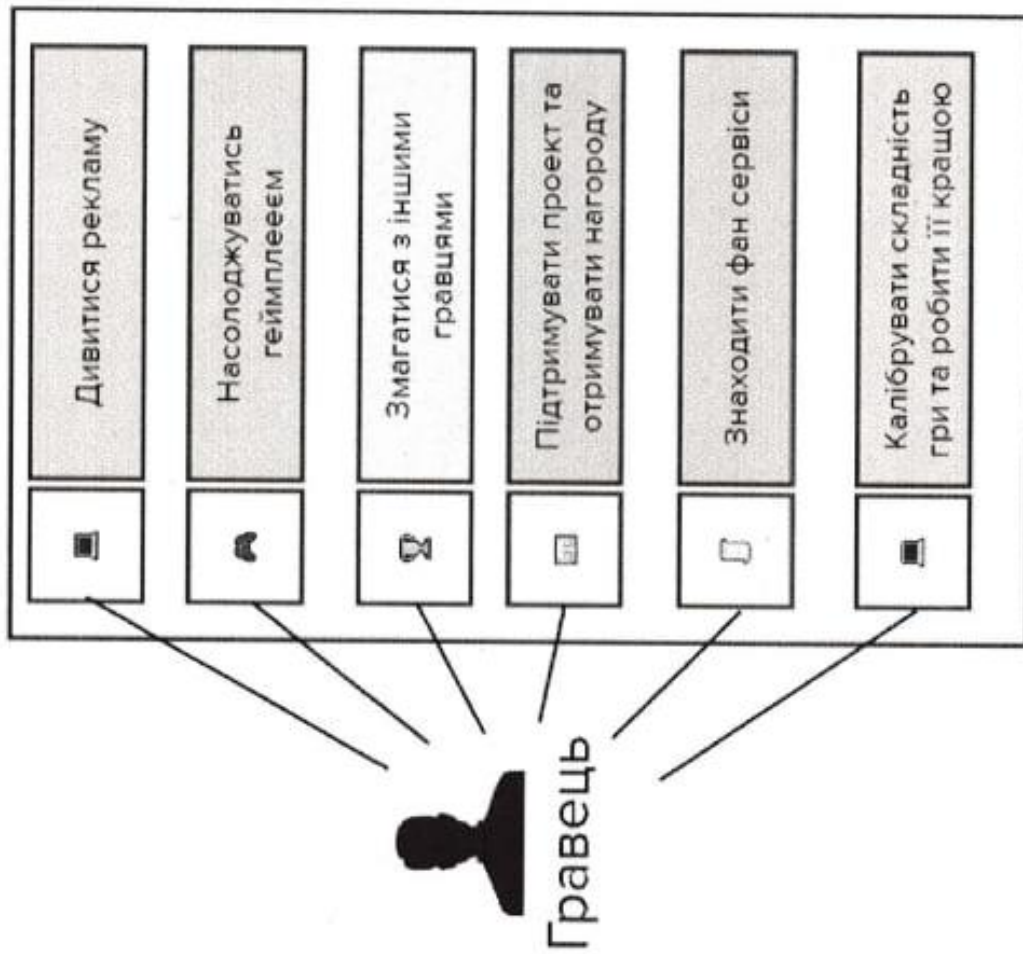
## Висновок

The image shows a screenshot of the Google Play store interface. At the top, there are tabs for 'Games', 'Apps', 'Movies & TV', 'Books', and 'Kids'. Below this, there are two sections: 'Based on your recent activity' and 'Top-rated games'. The 'Based on your recent activity' section shows four game icons with their names and ratings: Missiles! (4.0 ★), Smash Hit (4.6 ★), Fluffy Fall: Fly Fast to Dodge (4.5 ★), and Leap On! (3.9 ★). The 'Top-rated games' section shows four more game icons: Geometry Dash SubZero (4.5 ★), Brick Mania: Fun Arcade Game (4.5 ★), PAC-MAN 256 - Endless Maze (4.4 ★), and Zombie 1 (4.5 ★).

Відеоігри – це велика сфера впливу на цей світ, та теоретичні і практичні застосування кваліфікаційної роботи це цілком показують. Адже з одного боку проект гра про Пса Патрона є культурно актуальний, з іншої сторони він цікавий для молоді, і найголовніше він має два види монетизації, що справді може практично давати якийсь дохід. Окрім того аналітичні можливості дозволяють робити рекламні кампанії з урахуванням конверсій, що дозволить додатку популяризуватися краще, та приносити вищі доходи, а ще це аналітичні данні це хороший інструмент для мірила якості продукту

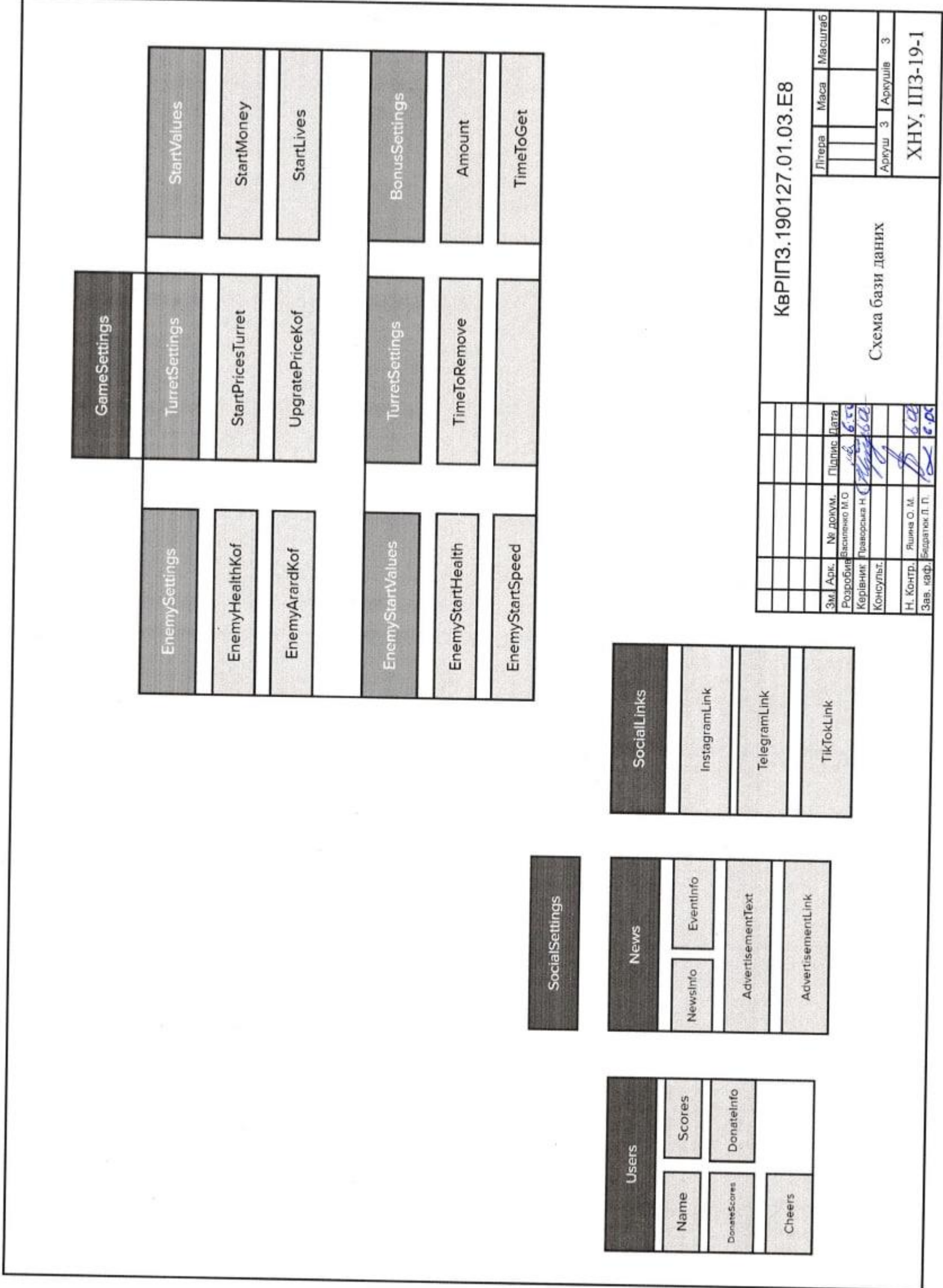
Слайд 8 – Висновок

**ГРАФІЧНА ЧАСТИНА**



КвРІПЗ.190127.01.03.Е8		Літера	Маса	Масштаб
Діаграма варіантів використання		Аркуш 2	Аркуш 3	
Зм. Арк.	№ докум.	Підпис	Дата	
Розробник	Васильченко М.О.		14.06.2016	
Керівник	Правороска Н. І.		14.06.2016	
Консульт.				
Н. Контр.	Яшина О. М.		14.06.2016	
Зав. каф.	Бедрачок Л. П.		14.06.2016	

ХНУ, ІПЗ-19-1



**GameSettings**

EnemySettings	TurretSettings	StartValues
EnemyHealthKof	StartPricesTurret	StartMoney
EnemyArardKof	UpgradePriceKof	StartLives
EnemyStartValues	TurretSettings	BonusSettings
EnemyStartHealth	TimeToRemove	Amount
EnemyStartSpeed		TimeToGet

**SocialSettings**

NewsInfo	EventInfo
AdvertisementText	
AdvertisementLink	

**SocialLinks**

InstagramLink
TelegramLink
TikTokLink

**Users**

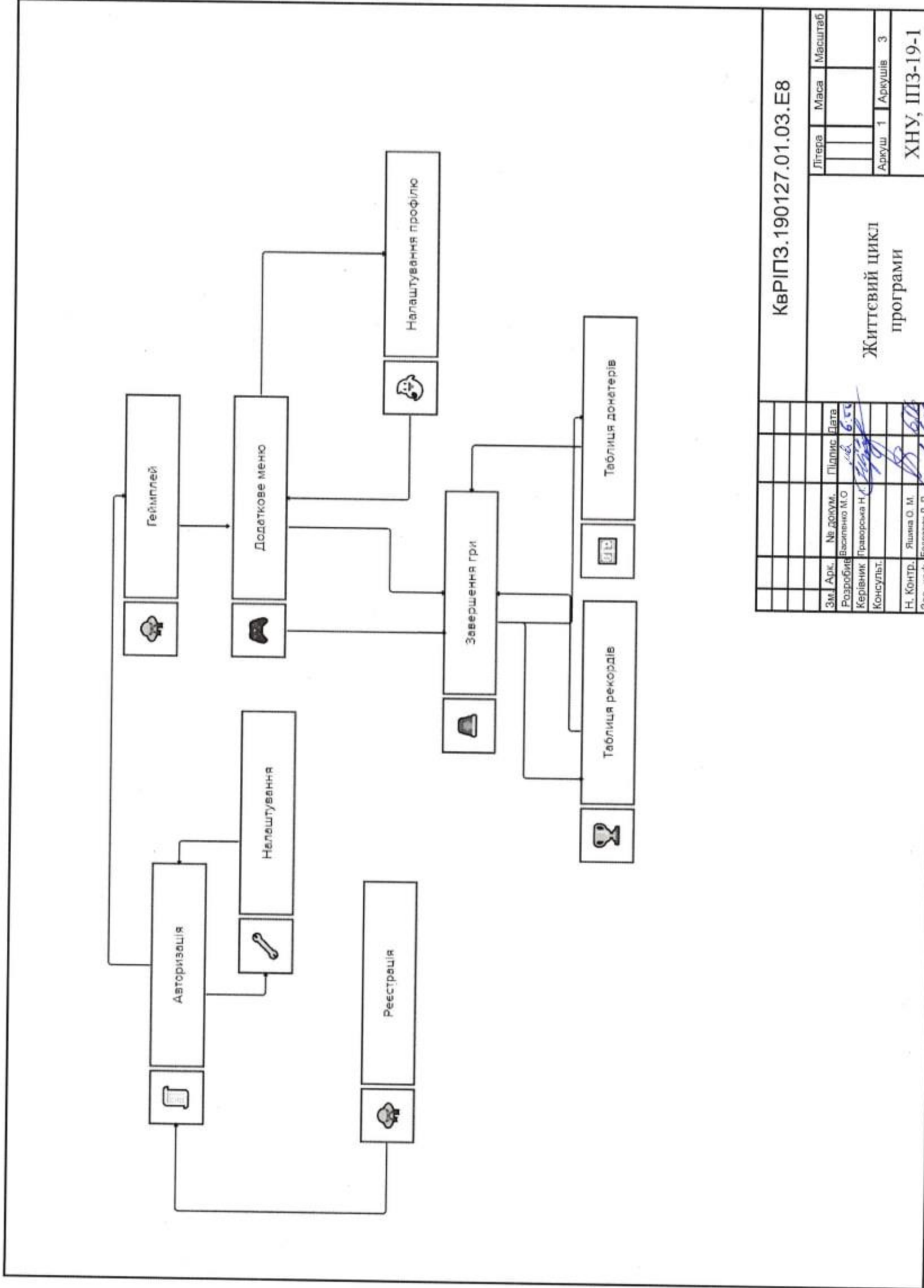
Name	Scores
DonateScores	DonateInfo
Cheers	

Змі. Арк.	№ док.м.	Підпис	Дата	Літера	Маса	Масштаб
Розробник	Васильченко М.О.	<i>[Signature]</i>	16.06.20			
Керівник	Траворська Н.	<i>[Signature]</i>				
Консульт.						
Н. Контр.	Яшина О.М.	<i>[Signature]</i>				
Зав. каф.	Березник Л.П.	<i>[Signature]</i>				

КвРІПЗ.190127.01.03.E8

Схема бази даних

ХНУ, ІПЗ-19-1



КвРІПЗ.190127.01.03.Е8

Літера	Маса	Масштаб
Аркус 1	Аркусів	3

ЖИТТЄВИЙ ЦИКЛ програми

Зм. Арк.	№ докум.	Підпис	Дата
Розробник	Васильченко М.О.	<i>[Signature]</i>	2019.06.26
Керівник	Григоренко Н.С.	<i>[Signature]</i>	
Консульт.			
Н. Конгр.	Яшина О. М.	<i>[Signature]</i>	
Зав. каф.	Безруток Л. П.	<i>[Signature]</i>	

## **СУПРОВІДНІ ДОКУМЕНТИ**

Завідувачу кафедри інженерії програмного  
забезпечення проф. Бедратюку Л. П.

здобувача вищої освіти

Василенка М. О.

Прізвище, ініціали

факультет ІТ, 4 курс, група ІПЗ-19-1

### ЗАЯВА

З правилами чинного Положення «Про дотримання академічної доброчесності в Хмельницькому національному університеті» від 26.09.2020 (зі змінами від 26.11.2020), згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений. Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

05.02.2023

дата



підпис



Ім'я користувача:  
Кафедра ІПЗ

ID перевірки:  
1015426239

Дата перевірки:  
05.06.2023 11:29:29 EEST

Тип перевірки:  
Doc vs Internet + Library

Дата звіту:  
05.06.2023 11:32:18 EEST

ID користувача:  
100005589

Назва документа: **Кваліфікаційна\_робота\_Василенко\_перевірка\_на\_антиплагіат**

Кількість сторінок: 73 Кількість слів: 12671 Кількість символів: 94334 Розмір файлу: 5.28 MB ID файлу: 1015088396

## 5.34% Схожість

Найбільша схожість: 2.16% з джерелом з Бібліотеки (ID файлу: 1015040217)

3.39% Джерела з Інтернету 417 ..... Сторінка 75

2.86% Джерела з Бібліотеки 83 ..... Сторінка 77

## 0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

## 0% Вилучень

Немає вилучених джерел

## Anti-Plagiarism v-15.257

**Максимальне співпадіння з одним документом 1.0%**

**Словники перевірки: en\_US, ru\_RU, ua\_UA. Помилки в документах: 8%**

ID: 114722 Назва: БКР Кроссплатформна гра для мобільних платформ у жанрі «Аркада» на Unity Додано в БД: 2023-06-05 Автора: Василенко М.О. Керівники: Праворська Н.І. к.п.н. доц. Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	79107	687	1407 (2%)	21 (3%)

### Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

освітнього ступеня «Бакалавр»

Дипломник Василенко Максим Олексійович

Тема Кроссплатформна гра для мобільних платформ у жанрі «Аркада» на Unity

Спеціальність 121 – Інженерія програмного забезпечення

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3; кількість сторінок записки 56

1. Короткий зміст пояснювальної записки та прийнятих рішень У кваліфікаційній роботі було проведено дослідження та проаналізовано предметну область створення аркадних ігор. Були визначені функціональні та нефункціональні вимоги до гри, які включали у себе механіку геймплею, графічний дизайн, звукове супроводження та інші аспекти, що впливають на користувацький досвід. В процесі аналізу існуючих аркадних ігор на ринку були розглянуті їх переваги та недоліки. Цей аналіз підкреслив актуальність розробки нового програмного забезпечення, орієнтованого на надання унікального геймплею. Для реалізації були використані інструменти розробки, зокрема фреймворк Unity, який забезпечує потужні можливості для створення графіки, фізики, анімації та управління грою. Використання Unity дозволило ефективно реалізувати всі задачі, пов'язані з розробкою гри. Після розробки програмного забезпечення було проведено тестування гри з метою перевірки її функціональності, стабільності та відповідності вимогам. Результати тестування підтвердили, що розроблене програмне забезпечення працює коректно та готове до експлуатації.

2. Висновок про відповідність роботи поставленому завданню Кваліфікаційна робота виконана відповідно до поставленого завдання та з дотриманням всіх вимог.

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки та передових методів роботи В даній кваліфікаційній роботі було виконано послідовний ряд завдань, які сприяли розробці програмного продукту – аркадної гри, створеної на платформі Unity. У вступному розділі була обґрунтована актуальність теми, визначена загальна мета та конкретні завдання дипломного проектування. У першому розділі було проведено детальний аналіз предметної області, включаючи вивчення сучасних рішень та існуючих аркадних ігор на ринку. В результаті цього аналізу було визначено, що оптимальною архітектурою для розробки гри буде монолітна архітектура в поєднанні з моделлю клієнт-сервер. Це дозволило забезпечити ефективну взаємодію гри з користувачем та серверними ресурсами. У третьому розділі були виконані необхідні підготовчі роботи, включаючи налаштування залежностей та сервісів для розробки гри. Також було реалізовано програмні модулі, які відповідали за різні аспекти гри, такі як графіка, фізика, управління та інші. Описано особливості реалізації цих модулів, які сприяли створенню функціонально повного програмного продукту.

4. Позитивні сторони роботи Тематика кваліфікаційної роботи, пов'язана з розробкою гри на платформі Unity є актуальною і цікавою. На сьогоднішній день ігрова індустрія швидко розвивається, і в Україні є значна потреба у нових ігрових продуктах У процесі розробки було застосовано новітні технології, що дозволило побудувати програмний продукт високої якості. Також були використані актуальні архітектурні рішення, що сприяли ефективній розробці та підтримці гри. Зазначена тематика кваліфікаційної роботи має потенціал для подальшого розвитку і комерціалізації, оскільки ігрова індустрія є перспективною галуззю, яка залучає все більше користувачів та геймерів.

5. Негативні сторони Одним з основних елементів гри є система взаємодії гравців за категоріями, яка дозволяє гравцям бачити один одного в таблицях досягнень. Однак, в рамках роботи було виявлено, що ця система є досить легкою та не забезпечує повний набір інструментів для взаємодії користувачів. Тому, як негативна сторона роботи, було відзначено слабкий соціальний функціонал.

6. Оцінка графічного оформлення та пояснювальної записки Графічне оформлення виконано відповідно до теми кваліфікаційної роботи та подано у вигляді діаграм і рисунків. Пояснювальна записка оформлена згідно вимог чинних стандартів. Графічна частина складає 3 діаграми, зокрема діаграма варіантів використання, діаграма життєвого циклу та схема бази даних

7. Відгук про кваліфікаційну роботу в цілому Кваліфікаційна робота заслуговує позитивної оцінки. Матеріал пояснювальної записки структурований, послідовний, чіткий та простий, що дозволяє чітко зрозуміти викладений матеріал у рамках тематики проектування. Графічний матеріал надає можливість наочно побачити деталі проектування системи.

8. Інші зауваження

9. Оцінка кваліфікаційної роботи Кваліфікаційна робота виконана у повному обсязі, відповідає поставленій задачі та заслуговує на оцінку «добре».

РЕЦЕНЗЕНТ Бобровнікова Кіра Юліївна, кандидат технічних наук, доцент кафедри комп'ютерної інженерії та інформаційних систем (КІС) ХНУ

" 05 "

06

2023 р.

(підпис)

**РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ  
КАФЕДРИ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ  
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатами звіту/звітів подібності щодо роботи, продуктованими програмно-технічним засобом (ами) перевірки текстів на плагіат:

Назва: Кросплатформна гра для мобільних платформ у жанрі «Аркада» на Unity

Автор: Василенко Максим Олексійович

Спеціальність: 121 – Інженерія програмного забезпечення

Освітня програма: Освітньо-професійна програма «Інженерія програмного забезпечення»

Науковий керівник: Праворська Н. І канд. пед. наук, доцент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	<b>відповідає</b>
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої й електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того, як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
5	Інше:	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) у тексті кваліфікаційної роботи системами перевірки на плагіат виявлено схожість з деякими документами в частині загальноживаних обов'язкових словосполучень у стандартних бланках (титулка, відомість документів), у структурі змісту, назвах розділів/підрозділів тощо, у назвах публікацій у переліку джерел посилання;

2) в якості запозичень системою було зафіксовано деякі послідовності вихідного коду і посилання на бібліотеки, які є стандартними мовними конструкціями програмування та не можуть розглядатися як об'єкт авторських прав і, відповідно, їх порушення;

3) усі запозичення є фрагментарними або мають належним чином оформленні посилання;

4) виявлені модифікації тексту не впливають на відсоток схожості.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів ідентичності/схожості, складає 5,34% і адресується до 500 джерел, що, з урахуванням наведених обґрунтувань, відповідає характеру теми і свідчить на користь кваліфікаційної роботи.

Дата 6.06

Завідувач кафедри

Гарант освітньої програми

Керівник кваліфікаційної роботи


Леонід БЕДРАТЮК

Леонід БЕДРАТЮК

Праворська Н. І