

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

**КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА**

**на тему: «Ігровий додаток змагального характеру»**

За спеціальністю 122 «Комп'ютерні науки»  
освітньо-професійна програма «Інформаційні технології проектування»

**Виконавець роботи:** студент групи ІТ-81 Дачко Артем Віталійович

**Кваліфікаційна робота бакалавра  
захищена на засіданні ЕК  
з оцінкою**

\_\_\_\_\_ «\_\_» \_\_\_\_\_ 2022 р.

Науковий керівник

\_\_\_\_\_  
(підпис)

к.т.н., доц. Федотова Н.А.  
(науковий ступінь, вчене звання прізвище та ініціали)

Засвідчую що у цій дипломній роботі немає  
запозичень праць інших авторів  
без відповідних посилань

Студент \_\_\_\_\_  
(Підпис)

Сумський державний університет  
Факультет електроніки та інформаційних технологій  
Кафедра інформаційних технологій  
Спеціальність 122 «Комп'ютерні науки»  
Освітньо-професійна програма «Інформаційні технології проектування»

**ЗАТВЕРДЖУЮ**

Зав. кафедри ІТ

\_\_\_\_\_ В. В. Шендрик  
«\_\_» \_\_\_\_\_ 2021 р.

**З А В Д А Н Н Я**  
**НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ**

*Дачко Артем Віталійович*

**1 Тема роботи** Ігровий додаток змагального характеру

**керівник роботи** Федотова Наталія Анатоліївна, к.т.н., доцент \_\_\_\_\_,

затверджені наказом по університету від «27» квітня 2022 р. \_\_\_\_\_ №0301-VI

**2 Строк подання студентом роботи** «10» червня 2022 р.

**3 Вхідні дані до роботи** загальний опис бажаного ігрового процесу, список ігор аналогів та конкурентів

**4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)** аналіз предметної області, моделювання та проектування ігрового додатку, розробка додатку

**5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)**

---

**6. Консультанти розділів роботи:**

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
<i>Моделювання та проектування ігровго додатку</i>	<i>Федотова Н.А.</i>		

**7.Дата видачі завдання** 06.10.2021

---

**КАЛЕНДАРНИЙ ПЛАН**

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Оформлення планування робіт	18.04.2022- 20.04.2022	
2	Оформлення технічного завдання	21.04.2022- 23.04.2022	
3	Проведення аналізу предметної області	25.04.2022- 01.05.2022	
4	Проведення проектування ігрового додатку	02.05.2022- 08.05.2022	
5	Розробка ігрового додатку	09.05.2022- 30.05.2022	
6	Тестування ігрового додатку	31.05.2022- 04.06.2022	
7	Оформлення пояснювальної записки	05.06.2022- 10.06.2022	

**Студент**

\_\_\_\_\_

(підпис)

Дачко А.В.

**Керівник роботи**

\_\_\_\_\_

(підпис)

к.т.н., доц. Федотова Н.А.

## РЕФЕРАТ

Тема кваліфікаційної роботи бакалавра «Ігровий додаток змагального характеру»

Пояснювальна записка складається зі вступу, 3 розділів, висновків, списку використаних джерел із 19 найменувань, додатків. Загальний обсяг роботи – 52 сторінка, у тому числі 29 сторінок основного тексту, 2 сторінки списку використаних джерел, 21 сторінка додатків.

Кваліфікаційну роботу бакалавра присвячено розробці гри змагального характеру. В роботі проведено аналіз популярних ігор, формулювання правил гри, моделювання та розробка ігрового додатку.

Результатом проведеної роботи є ігровий додаток «IGRA» та висновки щодо його подальшої долі.

Практичне значення роботи полягає у можливості протестувати на практиці розроблені правила гри, зробити висновки щодо важливості розробки повномасштабної гри на основі представлених правил, або потреби їх імплементації в уже існуючі ігри.

Ключові слова: ІГРОВИЙ ДОДАТОК, ГАР, ПРАВИЛА ГРИ, ІГРОВИЙ РУШІЙ, UNREAL ENGINE 4, ПЕРСОНАЖІ, МАТЧ, РАУНД.

## ЗМІСТ

Вступ.....	3
1 Аналіз предметної області.....	5
1.1 Огляд останніх досліджень і публікацій.....	5
1.2 Аналіз програмних продуктів-аналогів .....	5
1.3 Постановка задачі.....	12
2 Моделювання та проектування ігрового додатку.....	15
2.1 Моделювання.....	15
2.2 Вибір та обґрунтування засобів реалізації .....	20
3 Розробка додатка .....	23
3.1 Архітектура програмного додатку .....	23
3.2 Програмна реалізація .....	25
3.3 Використання програмного додатку .....	27
Висновки .....	29
Список використаних джерел .....	30
Додаток А.....	32
Додаток Б .....	43

## ВСТУП

За останні пару років відеоігри стали невід’ємною частиною життя багатьох людей. Така тенденція була викликана заходами для боротьби з епідемією COVID-19, та різними його штамами. З початку епідемії людство було обмежене у засобах розваги та відпочинку. Ігри ж в свою чергу були абсолютно актуальною розвагою, що була доступною багатьом. Тож очевидно що вони зайняли вільну і досить комфортну нішу.

Виходячи з подібних міркувань велика кількість експертів та засобів масової інформації [4, 10] прогнозує подальше збільшення запитів на нові ігри, та насичення ринку ігрової індустрії фінансами. Взнявши за приклад статистику однієї з найпопулярніших платформ «Steam»[13], становиться очевидним що найбільш популярні проекти – це багатокористувацькі ігри змагального характеру. Зважаючи на наведені факти, будь-яка нова гра має шанси на успіх, особливо якщо це гра змагального характеру з нескінченним та цікавим ігровим процесом.

Маючи певні загальні вимоги до гри було вигадано: концепцію, основну ігрову механіку та загальний набір правил майбутньої гри. Зважаючи на те, що ігор величезна кількість, головна вимога до нової – це певна унікальність та цікавий ігровий процес. Зробивши опис ігрового процесу групі знайомих геймерів отримано позитивні відгуки про задум. В онлайн магазинах ігор не було виявлено абсолютно точної реалізації вигаданої концепції.

Мета кваліфікаційної роботи – створити ігровий додаток, що буде реалізовувати в собі основні ігрові механіки, та правила гри. Перевірити вигадані правила на життєздатність та зробити висновки що до доцільності повномасштабної розробки гри.

Основні задачі:

- описати правила гри та ігровий процес

- проаналізувати аналоги, виокремити основні механіки, сформулювати вимоги до ігрового додатку
- реалізувати ігровий додаток на основі ігрового рушія
- перевірити ігровий процес серед гравців

## 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

### 1.1 Огляд останніх досліджень і публікацій

Ігри давно вже зайняли місце серед більш класичних розваг. Провести вечір за проходженням улюбленої сюжетної гри, чи намагаючись перемогти в команді з друзями в якій-небудь королівській битві, давно вже стало нормою на рівні з походом в кінотеатр чи кафе. Але так як і один фільм неможливо дивитися нескінченну кількість разів, та і ігри набридають. Отже світ відеоігор потребує все нових і нових ігор, з новими механіками та сюжетами.

За статистикою представленою постачальником ігор «Steam» велика кількість гравців постійно грають в яку-небудь онлайн гру. Більшість представлених ігор мають змагальний характер. Такі ігри як: «Counter-Strike: Global Offensive», «Dota 2», «PUBG: BATTLEGROUNDS», «Apex Legends», «Team Fortress 2» - все це ігри, кожна з яких має декілька сотень тисяч одночасно активних гравців. Більшість ігор наведених в статистиці має змагальний характер.

### 1.2 Аналіз програмних продуктів-аналогів

#### **Team Fortress 2**

Гра розрахована на 2 команди, які змагаються у різних режимах. Кожен гравець вправі обрати одного з 9 наявних у грі героїв для участі у матчі. Різноманітності гри додають різні арени та режими. Також кожному персонажу можна обрати власний набір спорядження що робить простий для освоєння ігровий процес більш цікавим та веселим. Кожна команда може містити будь-яку кількість гравців але рівну ( $\pm 1$ )



кількості гравців у команді противника. Персонажі у гравців можуть повторюватися [14]. Процес гри представлений на рисунку 1.1.



Рисунок 1.1 – Team Fortress 2

## Dota 2

Гра містить в собі лише одну арену, на якій відбуваються мільйони матчів, деякі з призовий фондом в мільйони доларів. Цілі гри кожного разу однакові, знищити головну будівлю команди суперника, але кожен матч унікальний. Це зумовлено великою різноманітністю персонажів, більше 120, з яких в одному матчі бере участь 10, по 5 в кожній команді [5]. Кожен гравець грає за унікального персонажа в матчі. Велика варіативність предметів, які в процесі гри можуть збирати персонажі для покращення своїх характеристик та можливостей, урізноманітнюють різні матчі, навіть, якщо в них однакові набори персонажів в кожній команді (рис. 1.2).



Рисунок 1.2 – Dota 2

## Paladins

Дві команди по 5 гравців змагаються в різних режимах. Кожен гравець перед початком матчу обирає персонажа яким буде грати увесь матч. Персонажі в матчі не повторюються. Кожен персонаж може бути покращений вдало підібраним набором «модів» перед початком матчу, та системою прокачки підчас раундів [8]. Кожен матч триває до двох перемог в раунді однієї з команд. Різноманітності грі додають великий вибір героїв перед початком матчу, вибір набору модів, випадкова карта та режим гри (рис. 1.3).



Рисунок 1.3 – Paladins

## PUBG: BATTLEGROUNDS

Гра представляє собою велику королівську битву. На одній великій карті методом десанту з літака з'являється велика кількість гравців де кожен сам за себе, до 100 осіб керують однаковими персонажами, головна задача яких вціліти та знищити всіх суперників. Всю зброю та спорядження гравці знаходять, випадковим чином розкидані по карті. Останній вцілілий вважається переможцем. З плином часу активна зона арени зменшується, і всі хто не знаходяться в ній отримують періодичний ушкодження. Різноманітності гри додає випадково розкидана зброя та спорядження, випадковий напрям польоту десантного літака над ареною, та випадковий центр активної зони до якої вона звужується [9].

Найбільш активний та цікавий ігровий процес відбувається на початку гри коли в одній локації «багатій на зброю та спорядження зустрічаються гравці», та в кінці коли активна зона «зводить» до купи вцілілих гравців з різним рівнем екіпірування (рис. 1.4).



Рисунок 1.4 – PUBG: BATTLEGROUNDS

### Counter-Strike: Global Offensive

Арена на 2 команди по 5 гравців. Простий ігровий процес: знищити суперника або зробити мінування/не допустити Мінування однієї з двох точок [2]. Різноманітність ігрового процесу забезпечує обрана зброя гравцями, та різні карти /арени на яких проходять матчі. Ігровий процес зображено на рисунку 1.5.



Рисунок 1.5 – Counter-Strike: Global Offensive

### Режим Deathmatch

В багатьох іграх представлений даний режим, у кожній зі своїми особливостями. Основна ігрова механіка полягає в нищенні всіх противників що знаходяться на карті. Якщо вас знищили ваш герой відродиться через певний час в спеціальній точці. Перемагає той хто знищить найбільшу кількість противників за відведений час. З одного боку режим простий та архаїчний, а з іншого він лежить в основі багатьох сучасних ігор змагального характеру [1, 12].

Надихаючись представленими іграми, та беручи від них найцікавіші ігрові механіки, за суб'єктивною оцінкою, було вигадано та продумано основний ігровий процес розроблюваного ігрового додатку. Головний недолік всіх представлених ігор – нарікання гравців на ігровий баланс, а точніше його недостатність. В одних іграх деякі персонажі сильніші за інших що ще до початку матчу надає перевагу, в інших карти не збалансовані (одна сторона перемагає частіше за іншу). Розроблений ігровий процес має намагатися уникати головних недоліків усіх ігор (табл. 1.1).

Таблиця 1.1 – Порівняльна характеристика ігор аналогів

Критерії порівняння	Ігри					
	Team Fortress 2	Dota 2	Paladins	PUBG: BATTLEGROUNDS	Counter-Strike: Global Offensive	IGRA
Невелика арена та кількість гравців на ній	+	+	+	-	+	+
Різноманітність персонажів	+	+	+	-	-	+
Ігровий баланс	±	±	±	±	±	+?
Активний ігровий процес	+	+	+	±	+	+
Різноманітність ігрового процесу	+	+	+	+	±	+
Залежність результату від власної майстерності	-	-	-	+	+	+

+? – ігровий баланс представлений як твірна частина ігрового процесу

### 1.3 Постановка задачі

Метою проекту є створення ігрового додатку, що буде містити в собі основу ігрового процесу, правила проведення раундів та матчів, а також основні ігрові механіки. Розроблений програмний продукт має надати можливість демонстрації та випробовування ігрового процесу групі геймерів.

Для досягнення поставленої мети треба вирішити наступні завдання:

- розробити технічне завдання додатку;
- провести планування робіт та ризиків;
- створити інтерфейс головного меню, створення та підключення до серверу;
- розробити арену для гри;
- розробити режим роботи у форматі локального сервера;
- розробити режим підключення до сервера;
- розробити механізм підключення гравців та зміни їх героїв;
- створити героїв з різними параметрами;
- впровадити правила гри на сервер;
- провести тестування додатку;
- оформити супровідну документацію.

#### **Правила матча**

Для кожного гравця матч починається з вибору персонажа для першого раунду. Коли кожен з присутніх гравців обрав собі персонажа відбувається автоматичний розподіл персонажів за точками відродження. У кожному наступному раунді гравці

будуть отримувати під своє керування нового персонажа суперника, але точка Відродження для цього персонажа змінюватися не буде протягом всього матчу.

За результатами кожного раунду гравцям будуть нараховані бали, в залежності від його досягнень. Нараховані бали за раунд будуть додаватися до поточної кількості балів за матч. Після завершення останнього раунду кожен гравець буде мати певну суму балів за всі раунди. Переможцем всього матчу буде гравець з найбільшою кількістю балів.

### **Правила одного раунду**

Раунд починається з відліку часу, після якого гравець отримує контроль над своїм персонажем. Персонаж може переміщуватися по арені, атакувати інших персонажів або намагатися захопити центральну зону. Кожен гравець може робити все що завгодно, але найбільшу кількість балів за раунд отримує або той хто захопить центральну зону, або той чий персонаж залишиться останнім на арені.

Головним об'єктом оцінювання та порівняння майстерності гравців у кожному раунді, а потім і у всьому матчі є бали. Кожен гравець отримує свої бали по завершенню його участі у раунді. Для цього створена та покладена в основу ігрового процесу система розподілу балів.

Кожен гравець починає раунд з кількістю балів рівній **0**. Якщо одиниці здоров'я підконтрольного гравцю персонажа опускаються до **0 (мінімального значення)**, то гравець покидає арену та раунд для нього завершується, і він отримує бали рівні кількості гравців що вибули з раунду на даний момент (*перший гравець що вибув отримує 0, останній хто залишився n-1 балів, де n – кількість гравців що брали участі у раунді*).

Існує ще один спосіб завершити раунд, а саме захопити центральну зону. Для захоплення зони потрібно перебувати в цій зоні неперервно протягом певного проміжку часу, наприклад 30 секунд. Після захвату центральної зони одним з гравців раунд завершується для всіх хто в ньому ще залишився, і відбувається розподіл балів. Гравець, що захопив зону, отримує кількість балів рівну кількості гравців у цьому раунді (**n**), всі інші гравці отримують бали за схемою «n-1» (*якщо захват*



*відбувається до того як будь-хто з гравців вибуває з раунду, то розподіл балів наступний: гравець що захопив зону отримує n балів, всі решта отримують 0 балів).*

Після того як гравець вибув з раунду його персонаж зникає, як і контроль над ним. Дочекавшись завершення раунду для всіх гравців, всі переходять до наступного раунду з новим персонажем під контролем.

### **Концепція персонажів які беруть участь у матчі**

Справедливий баланс, оснований на правилах матчу, дає можливість створювати персонажів не зважаючи на потребу робити їх рівномірно сильними. Тобто для першої ітерації гри достатньо буде персонажів які будуть різнитися одним домінуючим параметром (одиниці здоров'я, разовий ушкодження, швидкість переміщення, темп атаки(скорострільність), розмір персонажа та ін.).

## 2 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ ІГРОВОГО ДОДАТКУ

### 2.1 Моделювання

Для швидкого розуміння роботи ігрового додатку та процесу його використання змодельовано типові ситуації. Результати подібного моделювання представимо у нотації IDEF0 (рис. 2.1). Головна задача покладена на додаток це демонстрація та можливість випробувати ідеї та правила гри.

У якості вхідних даних представлені дані гравця, що у свою чергу складаються з загальної інформації про гравця та потоку його команд під час матчу. Механізми даного процесу представлені програмним продуктом «IGRA», апаратними механізмами, програмними механізмами та гравцем що в наданому екземплярі додатку задає команди підконтрольному йому герою. Потік вихідної інформації складається з результатів матчу, присвоєння статусу кращого гравця та усвідомлення ігрового процесу. Обмеження ігрового процесу представлені правилами матчу та раунду, а також технічними вимогами для використання ігрового додатку.

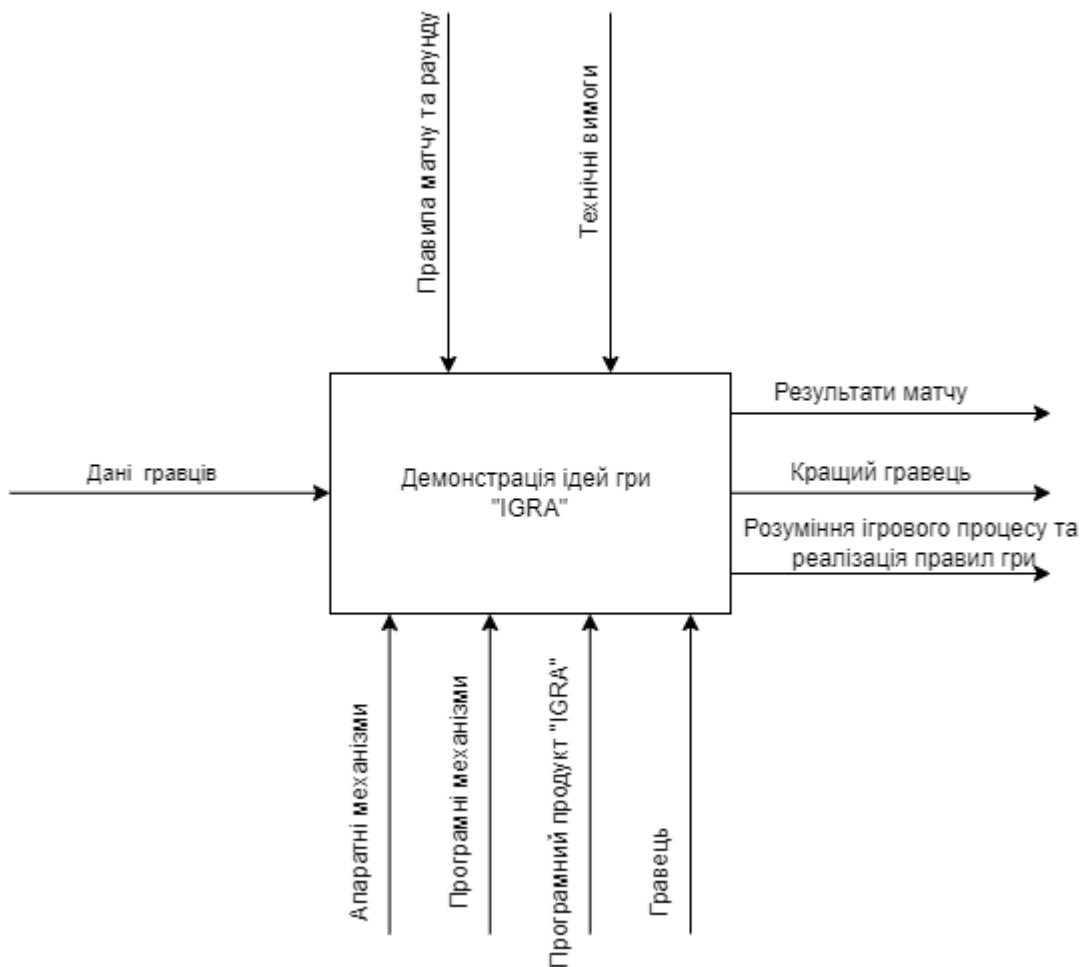


Рисунок 2.1 – Контекстна діаграма демонстрації ігрового процесу «IGRA»

Загальної інформації представленої в діаграмі недостатньо для розуміння процесу використання додатку та представлення його можливостей. Для рішення даної проблеми проведено декомпозицію 1-го рівня (рис. 2.2). На представленій діаграмі процес поділений на три основні частини: створення лобі, матч, розрахунок результатів за матч [6].

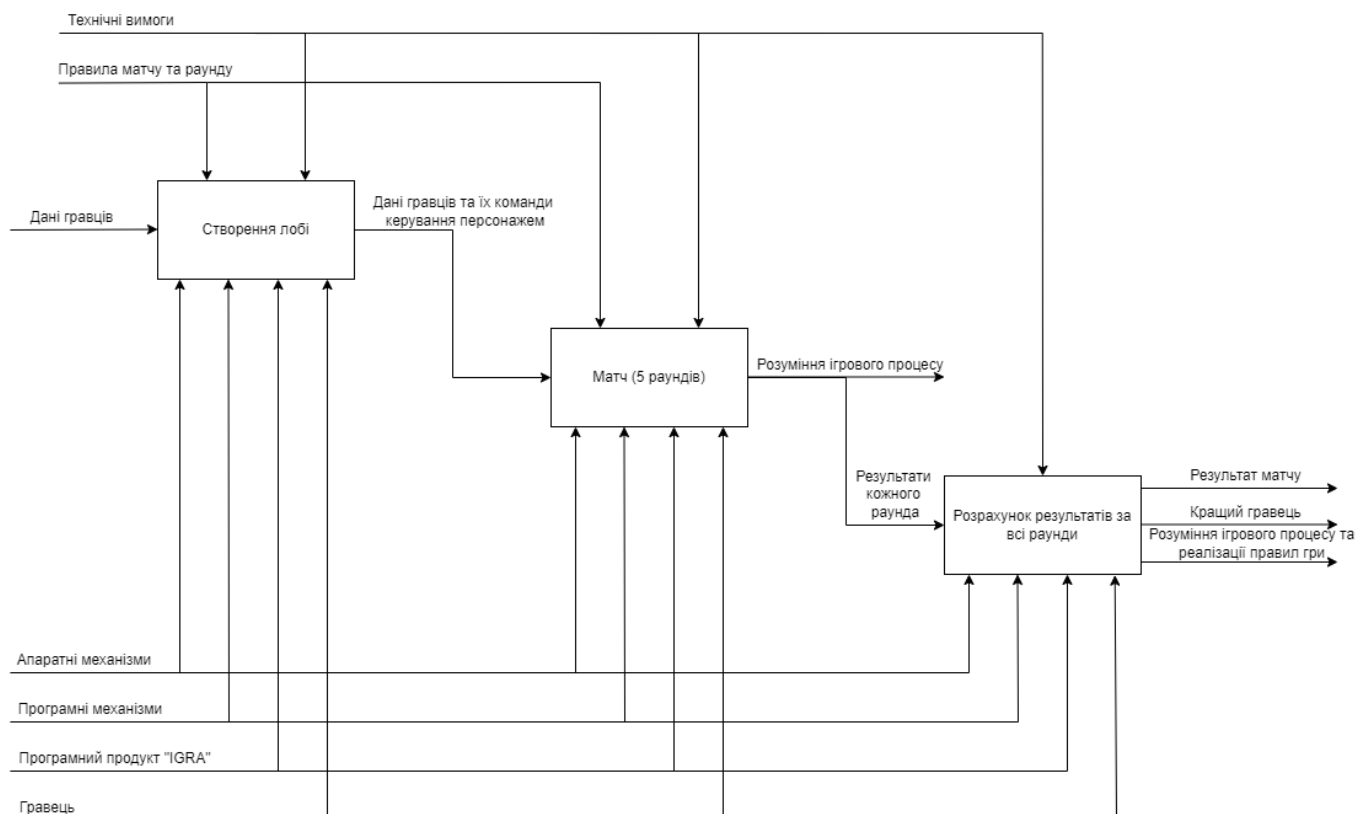


Рисунок 2.2 – Діаграма декомпозиції першого рівня демонстрації ігрового процесу

Для наглядної демонстрації взаємодії акторів з системою представимо майбутній продукт у вигляді моделі варіантів використання. Результат роботи сформовано в нотації UML, у вигляді Use-case діаграми (рис. 2.3). Актори системи детально представлені в таблиці 2.1. Для доступності розуміння діаграми представлена таблиця 2.2 з коротким описом варіантів використання.

Таблиця 2.1 – Актори системи

Назва актора	Короткий опис
<i>Актори-користувачі</i>	
Гравець	Користувач який створює лобі або приєднується до існуючого. В матчі керує одним з персонажів.
<i>Актори-зовнішні системи</i>	

Таблиця 2.1 – Актори системи

<b>Назва актора</b>	<b>Короткий опис</b>
Сервер матчу	Головна копія гри з де виконуються всі правила матчу, та від якої синхронізуються всі гравці. Сервер приймає вказівки гравців, що до керування їх персонажем.

Таблиця 2.2 – Опис варіантів використання програми

<b>Назва ВВ</b>	<b>Короткий опис</b>
Створення лобі	Користувач на своєму комп'ютері створює лобі та запускає серверну частину матчу
Приєднання до лобі	Користувач приєднується до лобі серверна частина якого вже запущена.
Проведення матчу	Організація 5 раундів матчу та демонстрації фінального результату.
Розподіл героїв по точкам спавну	Перед початком матчу кожному герою що приймає участь присвоюється точка спавна на ігровому рівні.
Призначення підконтрольного героя	Перед початком кожного раунду гравцю призначається підконтрольний герой в цьому раунді.
Проведення раунду	Під час раунду гравці намагаються знищити героїв суперників, або захопити центральну зону. Зберігається черга вибування з раунду, та контролюються спроби захопити центральну зону.
Розподіл балів	В залежності від черги вибуття з матчу, захоплення зони та кількості гравців, що вибули на момент завершення раунду, гравцям нараховуються бали.
Керування героєм	Кожен гравець надсилає команди керування своїм персонажем на сервер, у відповідь отримує інформацію про стан всіх об'єктів на сервері.

Таблиця 2.2 – Опис варіантів використання програми

<i>Назва ВВ</i>	<i>Короткий опис</i>
Переміщення	Дозволяє змінювати розташування власного персонажа на рівні.
Атака	Дозволяє атакувати персонажів суперника та знижувати їх рівень здоров'я.
Отримання пошкоджень	Дозволяє визначити власний рівень здоров'я та ефективність атак героїв суперників.
Захоплення центральної зони	Дозволяє завершити раунд та отримати максимальні бали за нього, якщо персонаж неперервно перебуватиме в центральній зоні визначений проміжок часу
Знищення героя	Дозволяє вибути з матчу, коли одиниці здоров'я рівні 0
Демонстрація результатів матчу	Дозволяє переглянути результати кожного з проведених раундів та сумарні результати за матч що до розподілу балів та визначити переможця.

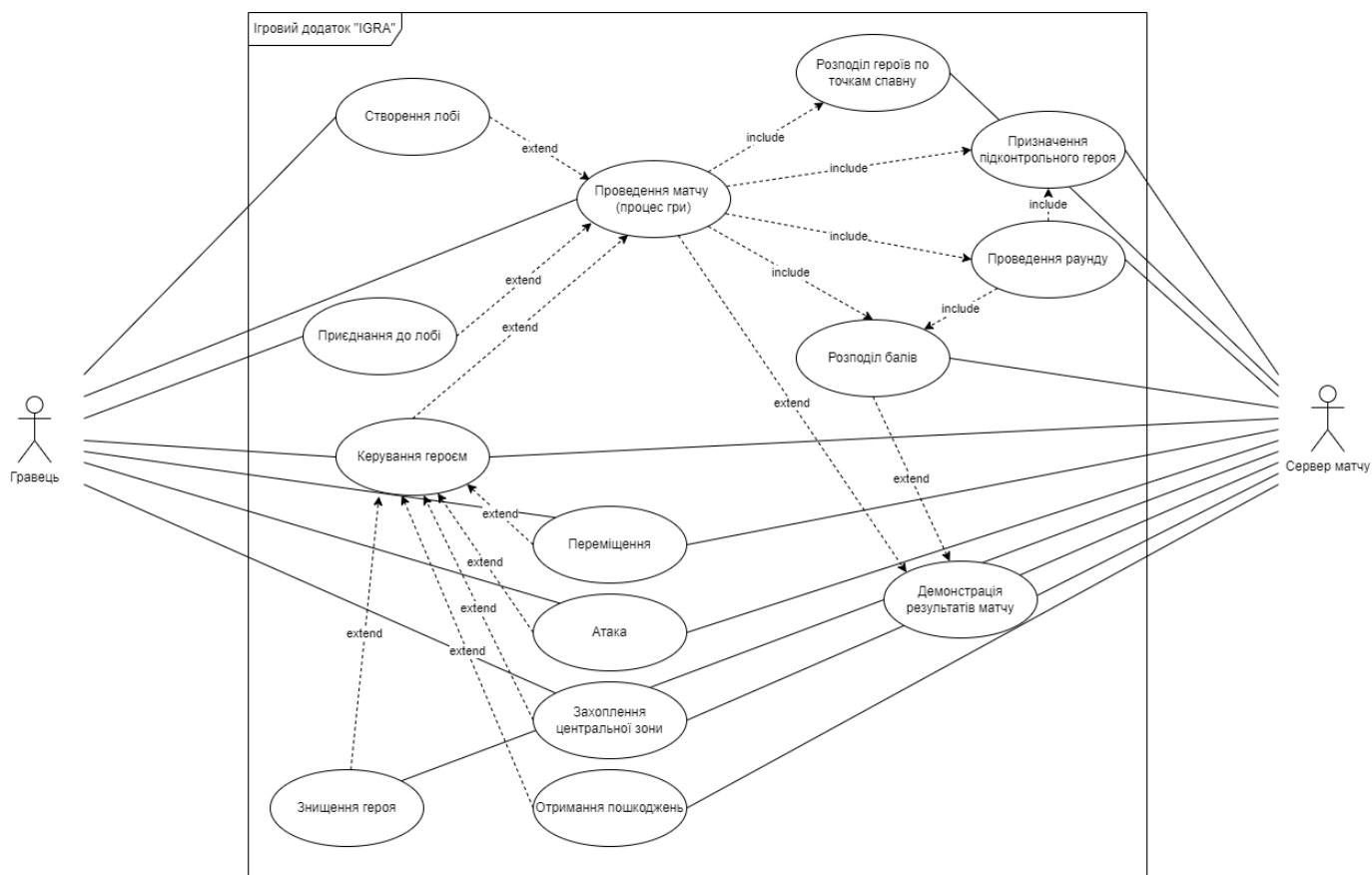


Рисунок 2.3 – Діаграма варіантів використання

## 2.2 Вибір та обґрунтування засобів реалізації

Розробка будь-якої гри починається з вибору ігрового рушія, будь то власна розробка студії чи загальнодоступне рішення. Розглянемо найбільш популярні та відомі загальнодоступні рушії, а саме: CryEngine, Unity, Unreal Engen 4 та Unreal Engen 5. Кожен з цих рушіїв може вільно використовуватися для навчання та розробки ігор (до отримання прибутку певного розміру). Провівши аналіз кожного з них та порівнявши їх можливості що до реалізації розробленої моделі (табл. 2.3), потрібно зробити висновок що до їх використання.

## **CryEngine**

Потужний ігровий рушій головною перевагою якого є якість графіки [3]. Програмування ігрового процесу відбувається за допомогою мови C++ та візуальної мови програмування. Не має вбудованих рішень для реалізації багатокористувацьких проектів. Має невелику спільноту користувачів та невелику кількість навчальних матеріалів.

## **Unity**

Потужне рішення для великої кількості проектів [16]. Рушій має невисокі системні вимоги і при правильному використанні досягає сучасних стандартів графіки. Використовує мову програмування C#. Не містить вбудованих рішень для мережевої роботи ігор, але існують сторонні плагіни що частково вирішують дану проблему. Велика спільнота користувачів що створила велику кількість навчальних матеріалів. Існує невеликий офіційний список загальнодоступних шаблонів. Шаблони від користувачів в більшості низької якості.

## **Unreal Engine 4**

Ігровий рушій що поєднує високу якість візуалізації та універсальність використання [19]. Підтримує мову програмування C++ та мову візуального програмування Blueprint. Має вбудований функціонал для створення багатокористувацьких ігор з гарною мережевою оптимізацією. Має велику спільноту користувачів що створюють достатню кількість навчальних матеріалів, також є досить детальна документація до рушія. Разом з рушієм доступні для використання та навчання різноманітні, якісні шаблони проектів з детальною документацією доступні для загального використання, деякі навіть для комерційного використання. Доступний внутрішній магазин високоякісних шаблонів користувачів, частина з яких безкоштовні.

## **Unreal Engine 5**

Покращена версія попереднього покоління рушія [18]. Нова система візуалізації високоякісних моделей та система освітлення. Містить велику кількість деталізованих загальнодоступних асетів. За якістю графіки може змагатися в першості з CryEngine. Мала кількість особистих навчальних матеріалів, але частина



навчальних матеріалів попередньої версії є актуальною. Разом з високими параметрами якості містить високі системні вимоги.

Таблиця 2.3 – Порівняння ігрових рушіїв

Параметри порівняння	Ігрові рушії			
	CryEngine	Unity	Unreal Engen 4	Unreal Engen 5
Доступність	+	+	+	+
Наявність документації у вільному доступі	+	+	+	+
Велика кількість навчальних матеріалів	-	+	+	±
Можливість створення багатокористувацьких ігор	+	+	+	+
Простота створення багатокористувацьких ігор	-	-	+	+
Наявність шаблонів для вільного використання	-	-	+	+
Задоволення системних вимог	+	+	+	-

Отже для розробки ігрового додатку змагального характеру доречно використовувати або Unreal Engine 4, або Unreal Engine 5 так як вони відразу пристосовані до розробки багатокористувацьких ігор. Так як системні вимоги у Unreal Engine 5 значно вищі за четверту версію рушія його використання становиться надто ускладненим, або, навіть, неможливим. Для розробки ігрового додатку обрано останню версію рушія Unreal Engine 4, а саме версію 4.27.2.

## 3 РОЗРОБКА ДОДАТКА

### 3.1 Архітектура програмного додатку

За основу розроблюваної гри взято загальнодоступний проект «ShooterGame» [11]. Перевагою використання даного проекту є завчасно створена клієнт серверна архітектура ігрового додатку, розроблені загальні механіки та інтерфейси «шутерів».

Додаток складається з головного меню, меню створення та підключення до серверу, та ігрового рівня на якому відбувається матч (рис. 3.1-3.3). При завершенні кожного раунду та матчу загалом гравцям демонструється таблиця результатів матчу.



Рисунок 3.1 – Головне меню



Рисунок 3.2 – Меню приєднання до сервера

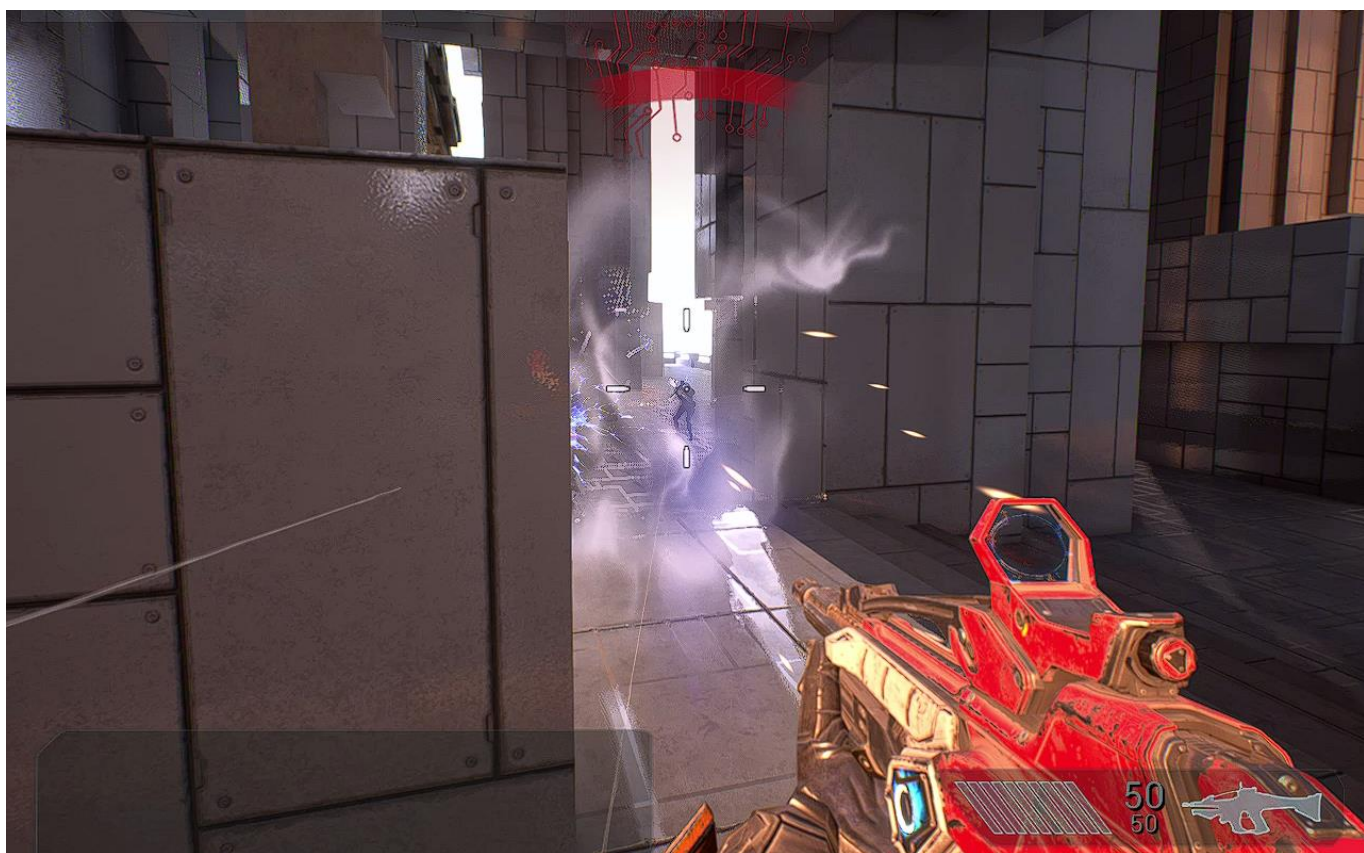


Рисунок 3.3 – Матч на ігровому рівні

### 3.2 Програмна реалізація

Розробка ігрового додатку проводилася за допомогою ігрового рушія Unreal Engine 4. Для редагування класів написаних мовою програмування C++ використовувалась Microsoft Visual Studio 2019.

Проведена підготовка та адаптація ігрового рівня до нових правил гри. Для цього спеціально розроблено новий ігровий клас центральної зони. Даний клас представляє собою Blueprint екземпляр якого розміщений на рівні. Він складається з платформи та боксу колізії який викликає виконання ігрової логіки (рис. 3.4).

На рівні розміщено точки старту та запрограмована логіка їх використання в рамках правил гри (рис. 3.5).

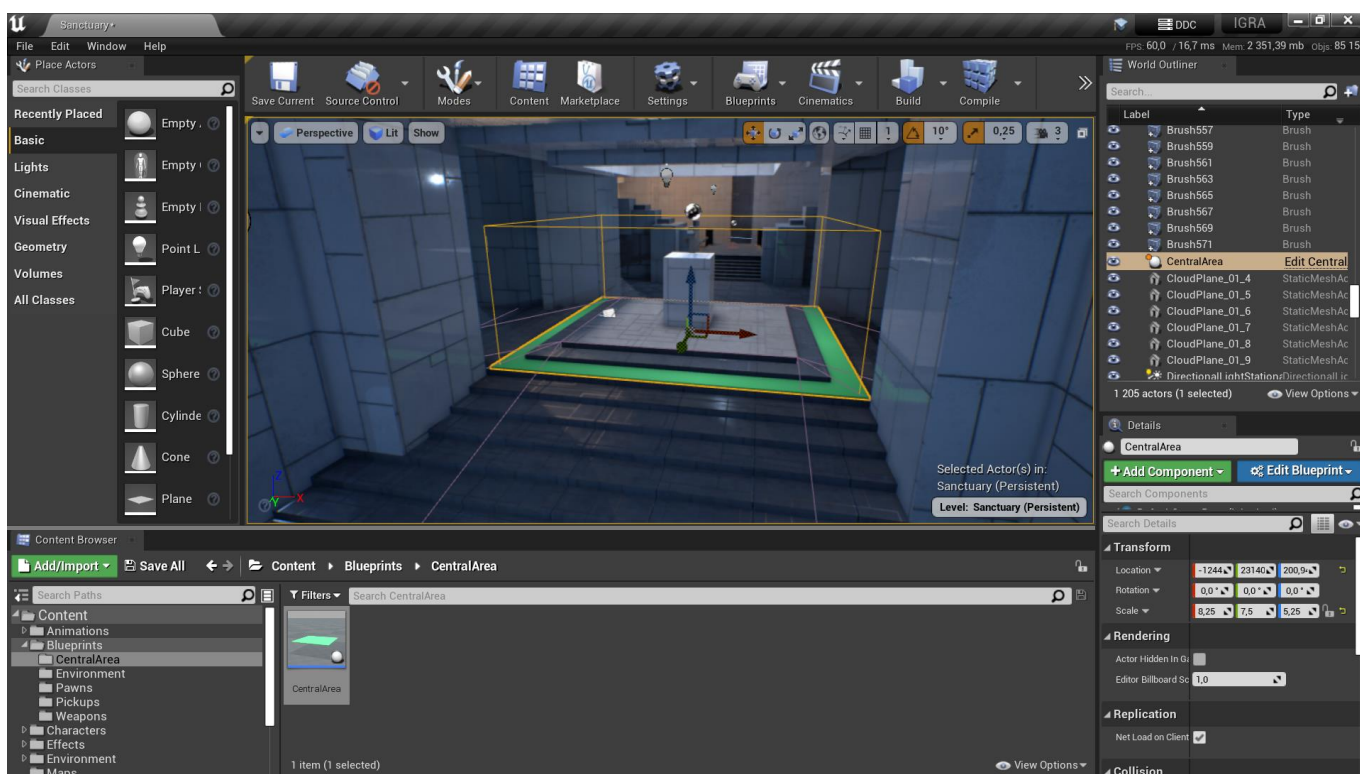


Рисунок 3.4 – Центральна зона



Рисунок 3.5 – Розміщення стартових позицій

Різноманіття персонажів створене методом наслідування від розробленого в шаблоні класу персонажа і модифікуванням його параметрів та параметрів його озброєння. Створено 5 персонажів з унікальними наборами параметрів.

В ігровий мод імплементовано правила гри. Розроблено функції початку матчу, початку раунду, завершення раунду та завершення матчу. Реалізовану функцію розподілу героїв для кожного раунду.

Розроблено інтерфейс демонстрації результатів кожного раунду та всього матчу (рис. 3.6).

	Round 1	Round 2	Round 3	Round 4	Round 5	Sum
DESKTOP-2...	2	6	0	4	2	14
DESKTOP-2...	4	2	6	2	1	15
DESKTOP-2...	3	2	0	1	3	9
DESKTOP-2...	5	1	0	3	5	14
DESKTOP-2...	1	2	0	5	4	12

Рисунок 3.6 – Результати проведено матчу

Після завершення процесу розробки, проект було зібрано для платформ на Windows (64 bit).

### 3.3 Використання програмного додатку

Розроблений ігровий додаток придатний для використання в локальних мережах. Зважаючи на те що основна ціль його розробки це тестування правил гри та демонстрація ігрового процесу, даний недолік не є критичним. За допомогою сторонніх програмних додатків дана проблема може бути вирішена шляхом створення віртуальної локальної мережі між комп'ютерами що в такій не знаходяться.

Проведене тестування показало, що правила гри покладені в основу ігрового процесу мають право на повноцінне існування в різних форматах. Недоліком даної реалізації ігрового процесу стала недостатня різноманітність героїв.

Загальним результатом використання розробленого ігрового додатку змагального характеру стали наступні висновки:

- ігровий процес може використовуватися як основа для розробки потенційно успішної, багатокористувацької гри;
- правила гри потребую значного різноманіття героїв;
- «центральна зона» вирішує проблему навмисного «затягування» раундів;
- правила гри можуть бути запропоновані для імплементації в одну з популярних ігор як режим матча: «Paladins», «Team Fortress 2» або «Overwatch» [7];

## ВИСНОВКИ

Під час виконання роботи було проведено аналіз існуючих відеоігор, та виявлено найбільш популярні жанри. Проведено детальний аналіз обраних представників, виділено основні механіки ігрового процесу для кожної гри. На основі отриманих даних та наявної ідеї зроблено детальний опис ігрового процесу що відрізняється від всіх існуючих аналогів та міг би сподобатись користувачам. Для перевірки ігрового процесу, на актуальність, серед гравців, прийнято рішення створити ігровий додаток що буде містити основний ігровий процес та правила гри.

На основі опису зроблено технічне завдання для програмного продукту майбутньої гри представлене в додатку А. Для поставленої задачі проведено планування робіт. Результат планування представлений в додатку Б.

Проведено моделювання ігрового додатку, результати представлені за допомогою нотацій UML та IDEF0. На основі створених моделей та за допомогою шаблону «Shooter Game» було розроблено ігровий додаток змагального характеру.

Використання даного додатку допомогло зробити висновки щодо потрібності створених правил гри та можливості їх подальшого використання. Одним з неочевидних результатів даної роботи стало виникнення сюжетного пояснення правил гри.

Отже розроблений ігровий процес має право на подальше існування та розвиток як в форматі окремого ігрового додатку, так і в форматі режиму матчу для існуючих ігор. Розроблений ігровий додаток змагального характеру виконав поставлену перед ним основну задачу, але для повноцінного самостійного існування потребує значних змін та об'єму роботи.



## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бой насмерть. Counter-Strike Wiki. URL: [https://counterstrike.fandom.com/ru/wiki/Бой\\_насмерть](https://counterstrike.fandom.com/ru/wiki/Бой_насмерть) (дата звернення: 10.06.2022).
2. Counter-Strike: Global Offensive. Counter-Strike: Global Offensive. URL: <https://blog.counter-strike.net/> (дата звернення: 10.06.2022).
3. CRYENGINE | The complete solution for next generation game development by Crytek. CRYENGINE. URL: <https://www.cryengine.com/> (дата звернення: 10.06.2022).
4. Dealessandri M. Saudi Arabia, United Arab Emirates and Egypt markets worth \$1.76bn in 2021. GamesIndustry.biz. URL: <https://www.gamesindustry.biz/articles/2022-01-10-saudi-arabia-united-arab-emirates-and-egypt-markets-worth-usd1-76bn-in-2021> (дата звернення: 10.06.2022).
5. Dota 2. Dota 2. URL: <https://www.dota2.com/home?l=ukrainian> (дата звернення: 10.06.2022).
6. Flowchart Maker & Online Diagram Software. Flowchart Maker & Online Diagram Software. URL: <https://app.diagrams.net/> (date of access: 15.06.2022).
7. Overwatch. Overwatch. URL: <https://playoverwatch.com/ru-ru/> (дата звернення: 10.06.2022).
8. Paladins® on Steam. Welcome to Steam. URL: <https://store.steampowered.com/app/444090/Paladins/> (дата звернення: 10.06.2022).
9. PUBG: BATTLEGROUNDS on Steam. Welcome to Steam. URL: [https://store.steampowered.com/app/578080/PUBG\\_BATTLEGROUNDS/](https://store.steampowered.com/app/578080/PUBG_BATTLEGROUNDS/) (дата звернення: 10.06.2022).

10. Rousseau J. China video games market revenue hits \$45.5bn during 2021. GamesIndustry.biz. URL: <https://www.gamesindustry.biz/articles/2022-05-23-china-video-games-market-revenue-hit-usd45-5bn-during-2021> (дата звернення: 10.06.2022).
11. Shooter Game. Unreal Engine 5 Documentation | Unreal Engine 5.0 Documentation. URL: <https://docs.unrealengine.com/4.27/en-US/Resources/SampleGames/ShooterGame/> (дата звернення: 10.06.2022).
12. Skillbox. Убей ближнего своего: почему Deathmatch – вечнозелёный режим. skillbox.ru. URL: <https://skillbox.ru/media/gamedev/pochemu-deathmatch-vechnozelyenyu-rezhim/> (дата звернення: 10.06.2022).
13. Steam most played games by peak players 2022 | Statista. Statista. URL: <https://www.statista.com/statistics/1179973/steam-games-peak-concurrent-players/> (дата звернення: 10.06.2022).
14. Team Fortress 2 on Steam. Welcome to Steam. URL: [https://store.steampowered.com/app/440/Team\\_Fortress\\_2/](https://store.steampowered.com/app/440/Team_Fortress_2/) (дата звернення: 10.06.2022).
15. The most powerful real-time 3D creation tool - Unreal Engine. URL: <https://www.unrealengine.com/marketplace/en-US/learn/shooter-game> (дата звернення: 10.06.2022).
16. Unity. Unity. URL: <https://unity.com/ru> (дата звернення: 10.06.2022).
17. Unreal Engine 4 Documentation. Unreal Engine 5 Documentation | Unreal Engine 5.0 Documentation. URL: <https://docs.unrealengine.com/4.27/en-US/> (дата звернення: 10.06.2022).
18. Unreal Engine 5. Unreal Engine. URL: <https://www.unrealengine.com/en-US/unreal-engine-5> (дата звернення: 10.06.2022).
19. Unreal Engine | The most powerful real-time 3D creation tool. Unreal Engine. URL: <https://www.unrealengine.com/en-US> (дата звернення: 10.06.2022).

## ДОДАТОК А

### Технічне завдання

#### **ТЕХНІЧНЕ ЗАВДАННЯ на розробку програмного продукту «Ігровий додаток змагального характеру»**

#### **ПОГОДЖЕНО:**

Доцент кафедри комп'ютерних наук

\_\_\_\_\_ Федотова Н.А.

Студент групи ІТ-81

\_\_\_\_\_ Дачко А.В.

## **1 Призначення й мета створення додатку**

### **1.1 Призначення додатку**

Ігровий додаток має реалізовувати ігровий процес та правила гри, для демонстрації та випробування їх гравцями. Гравці повинні мати змогу запустити сервер та провести матч.

### **1.2 Мета створення**

Демонстрація ігрового процесу гравцям, та перевірка їх зацікавленості в повномасштабній грі за описаними правилами.

### **1.3 Цільова аудиторія**

Геймери що добровільно згодні прийняти участь в тестуваннях та опитуваннях.

## **2 Вимоги до додатку**

### **2.1 Вимоги до додатку в цілому**

#### **2.1.1 Вимоги до структури й функціонування додатку**

Ігровий додаток має бути доступний для простого встановлення та використання. Інтерфейс повинен надати можливість розпочати матч будучи для нього сервером та приєднання до вже існуючого сервера. Матч повинен складатися з кількості раундів рівній кількості гравців у ньому. У кожного гравця в кожному раунді має бути під контролем новий герой.

#### **2.1.2 Вимоги до користувачів**

Від користувачів не вимагається нічого більшого окрім вміння чітко слідувати інструкції для створення та початку матчу.

#### **2.1.3 Вимоги до розмежування доступу**

Всі гравці мають рівний доступ до гри. Для початку матчу один з гравців створює матч з серверною частиною на своєму комп'ютері, а інші приєднуються до нього.

## **2.2 Структура додатку**

### **2.2.1 Загальна інформація про структуру додатку**

Структура додатку являє собою набір рівнів, частина з яких меню та арени для проведення раундів матчу:

- Головне меню
- Меню створення лобі
- Меню приєднання до лобі
- Сторінка інформації про розробника
- Арена для проведення матчу

### **2.2.2 Навігація**

Перехід між відповідними меню відбувається за допомогою навігаційних кнопок. З головного меню можна потрапити до: меню створення лобі, меню приєднання до лобі, сторінки інформації про розробника. На арену для проведення матчу можна потрапити з меню створення лобі та меню приєднання до лобі. З кожного рівня можна повернутися до рівня головного меню.

### **2.2.3 Дизайн та структура додатку**

Вимоги до стилю додатку відсутні, дизайн має бути простий та зрозумій користувачам. Шаблон дизайну основних меню наведено на рисунках А.1-А.6.

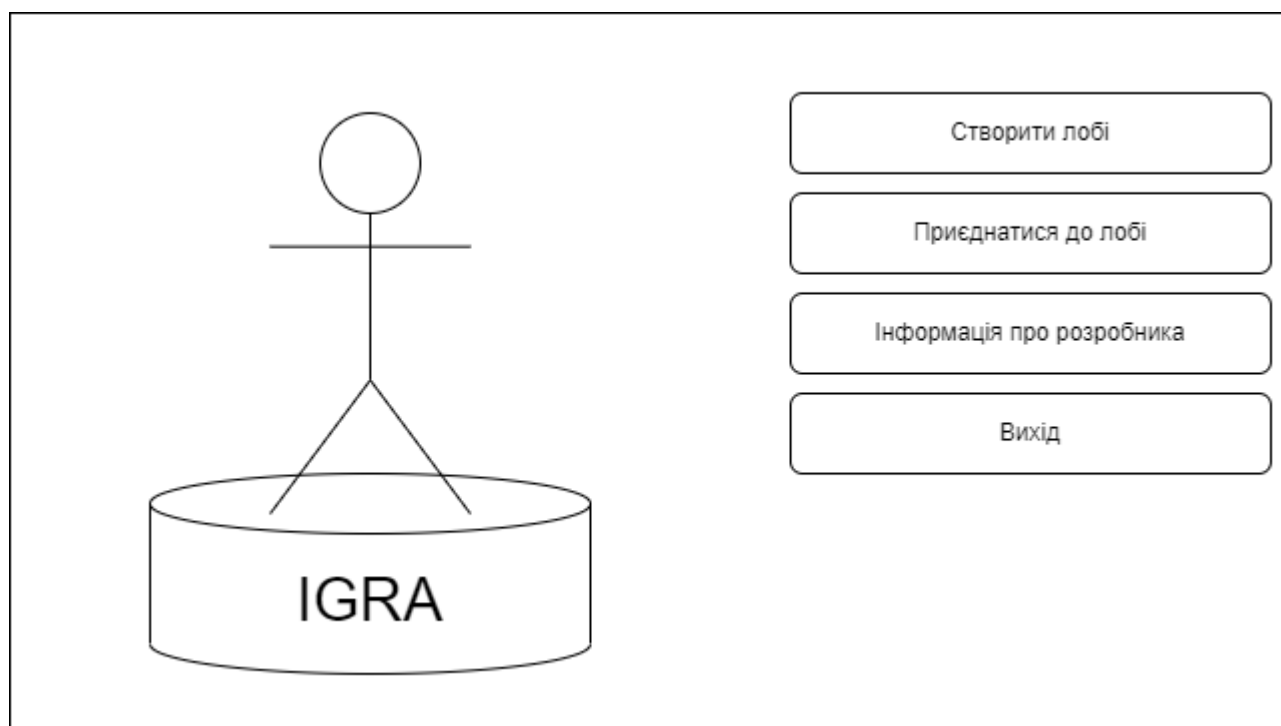


Рисунок А.1 – Головне меню

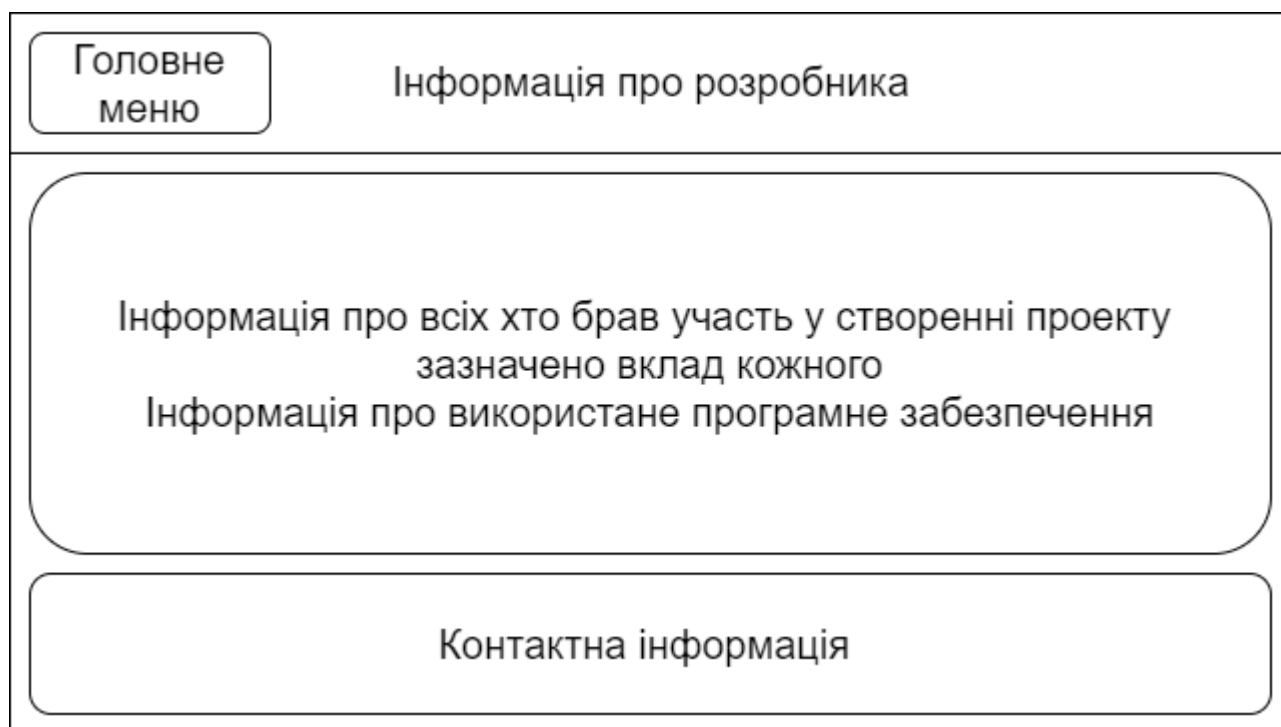


Рисунок А.2 – Інформація про розробника

Головне меню	Код лобі	ZX1548	Статус роботи сервера
Список приєднаних гравців			
Запустити сервер		Розпочати матч	

Рисунок А.3 – Створення лобі

Головне меню	Приєднатися до лобі		
Введіть код лобі			
ZX1548		Приєднатися	
Місце для повідомлення про статус приєднання до лобі			

Рисунок А.4 – Приєднання до лобі



Рисунок А.5 – Завершення роботи ігрового додатку

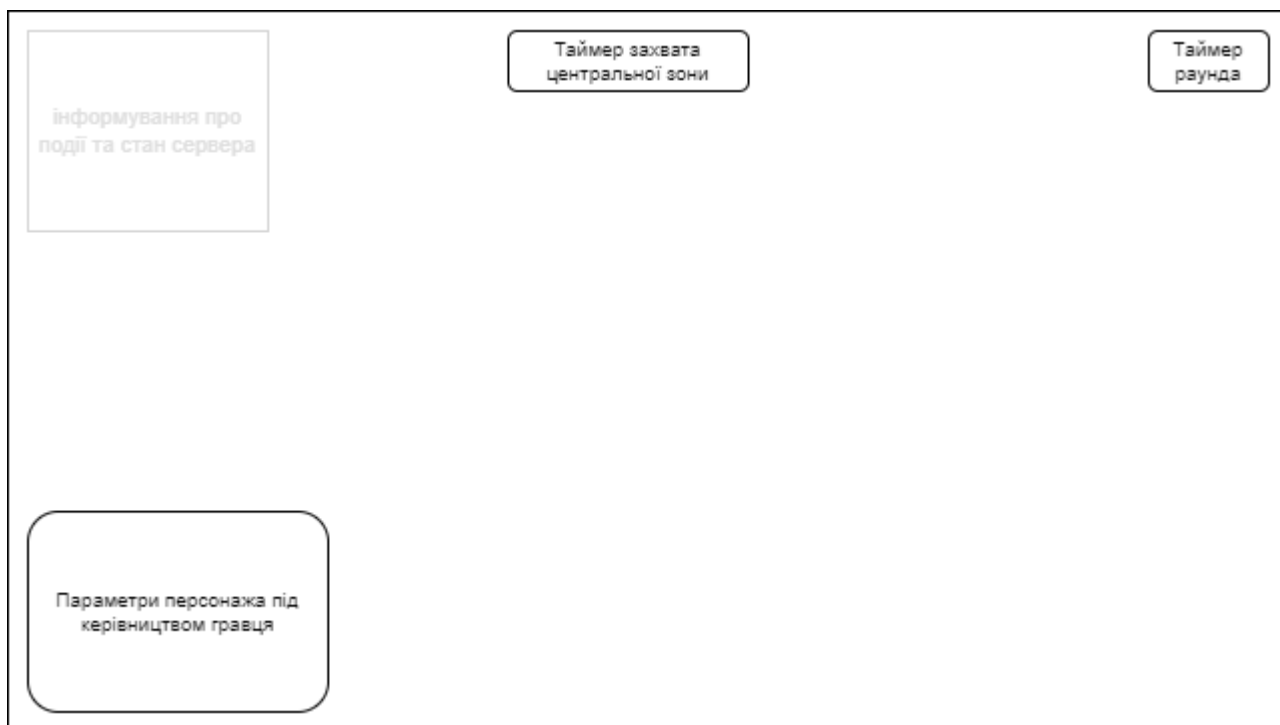


Рисунок А.6 – Інтерфейс матчу



## 2.3 Вимоги до функціонування системи

### 2.3.1 Потреби користувача

Потреби користувача, визначені на основі рішення замовника, представлені у таблиці А.1.

Таблиця А.1 – Потреби користувачів

ІД	Потреби користувача	Джерело
UN-01	Можливість запустити сервер для проведення матчу	Гравець
UN-02	Можливість приєднатися до створеного серверу	Гравець
UN-03	Провести матч за правилами гри	Гравець
UN-04	Визначити переможця матчу	Гравець
UN-05	Зрозуміла ідентифікація гравців	Гравець

### 2.3.2 Функціональні вимоги

На основі вимог замовника були визначені такі функціональні вимоги:

- Форма запуску серверу для проведення матчу
- Форма для ведення нікнейму та приєднання до серверу
- Функціонал правил матчу та проведення гри (переміщення, атака, механіка здоров'я, зміна підконтрольного персонажа в кожному раунді)
- Форма виведення результату матчу

### 2.3.3 Системні вимоги

Перелік системних вимог наведений в таблиці А.2.

Таблиця А.2 – Системні вимоги

ІД	Системні вимоги	Пріоритет	Опис
SR-01	Інтерфейс запуску серверу	М	Надає можливість користувачу запустити сервер матчу на власному

			пристрої. Всі обчислення стосовно перебігу подій проводяться на пристрої-сервері
SR-02	Інтерфейс приєднання до серверу	М	Надає можливість приєднатися до вже створеного сервера в локальній мережі
SR-03	Реалізувати керування персонажем	М	Переміщення, атака, механіка здоров'я та знищення
SR-04	Створити арену	М	Створити рівень на якому буде відбуватися ігровий процес
SR-05	Реалізувати правила раунду	М	Вести облік тих хто вибув з раунду гравців, та відстежувати захват центральної зони, розподілити бали по завершенню раунду
SR-06	Реалізувати правила матчу	М	Провести необхідну кількість раундів, реалізувати зміну підконтрольних героїв по завершенню раунду, вести загальний підрахунок балів за сумою всіх раундів
SR-07	Демонстрація результатів матчу	М	По завершенню всіх раундів представити користувачам

			таблицю з результатами за кожен раунд та результатом за матч.
SR-08	Налаштування правил матчу	C	Користувач що запускає сервер має можливість налаштувати основні ігрові параметри матчу
SR-09	Встановлення нікнейму гравця	S	Можливість гравця встановити нікнейм, що буде відображатися в таблиці статистики
SR-10	Вибір персонажів на матч	C	Перед початком матчу Гравці із великого різновиду героїв обирають одного, за якого будуть грати в першому раунді. Усі вибрані гравцями персонажі формують набір персонажів що бере участь у матчі

Умовні позначення в таблиці А.2:

Must have (M) – вимоги, які повинні бути реалізовані в системі;

Should have (S) – вимоги, які мають бути виконані, але вони можуть почекати своєї черги;

Could have (C) – вимоги, які можуть бути реалізовані, але вони не є центральною метою проекту.

## 2.4 Вимоги до видів забезпечення

### 2.4.1 Вимоги системного забезпечення

Уся ігрова діяльність пов'язана з роботою сервера матчу та приєднаних гравців повинна працювати в одній локальній мережі без перебоїв.

### 2.4.2 Вимоги до лінгвістичного забезпечення

Додаток має бути виконаний українською мовою. Для уникнення проблем при розробці призначити назву додатку англійською мовою, наприклад, «IGRA».

### 3 Склад і зміст робіт зі створення додатку

Докладний опис етапів роботи зі створення додатку наведено в таблиці А.3.

Таблиця А.3 – Етапи створення ігрового додатку

№	Склад і зміст робіт	Строк розробки (у робочих днях)
1	Загальний опис ігрового процесу	1 день
2	Пошук аналогів та аналіз можливих конкурентів	3 дні
3	Аналіз реалізації окремих ігрових механік	1 день
4	Виділення головних ігрових механік	2 дні
5	Детальний опис ігрового процесу	1 день
6	Формування завдання на розробку	3 дні
7	Розробка клієнт-серверної роботи додатку	10 днів
8	Реалізація класу гравця та розробка різновиду персонажів	10 днів
9	Моделювання рівня (арени)	5 днів
10	Розробка інтерфейсу користувача	5 дні
11	Реалізація правил гри	10 днів
12	Тестування	5 днів
13	Оформлення супровідної документації	5 днів
14	Створення спільноти гравців та незалежне тестування	2 дні
	Загальна тривалість робіт	63 дні

### 4 Вимоги до складу й змісту робіт із введення додатку в експлуатацію

Для проведення відкритого і незалежного тестування потрібно організувати локальну мережу в якій будуть знаходитись всі учасники конкретного матчу. За

неможливості провести тестування в одній локальній мережі використати сторонні розробки що створюють між пристроями віртуальну локальну мережу.

## ДОДАТОК Б

### Планування робіт

**Деталізація мети проекту методом SMART.** Щоб не перевантажувати проект розробки ігрового додатка зайвими процесами, потрібно однозначно визначити головну мету. Визначена методом SMART мета встановлює конкретну ціль якої треба досягти за відведений строк. Результати деталізації наведені в таблиці Б.1.

Таблиця Б.1 – Деталізація мети проекту методом SMART

Specific (конкретна)	Розробити гру на основі придуманих механік
Measurable (вимірювана)	Користувач може створити лобі та провести матч з 5 іншими користувачами на 5 різних героях, в межах однієї локації(арени).
Achievable (досяжна, узгоджена)	Для реалізації потрібно освоїти Unreal Engine 4 та розробку з використанням Blueprints. Зможуть полегшити задачу навички моделювання, анімації, володіння «Adobe Photoshop», знання мови C++ та повне уявлення кінцевого ігрового процесу. Зважаючи на всі наявні ресурси, вміння та обмеження – мета є досяжною.
Relevant (реалістична)	Створений ігровий додаток дасть можливість ознайомитись з ігровим процесом, перевірити механіки на сумісність, перевірити зацікавленість гравців в подальшому розвитку гри, та представлення її у форматі кіберспортивної дисципліни.
Time-framed (обмежена в часі)	до строку здачі дипломного проекту до 13 червня

**Планування змісту робіт.** Отримавши кінцеву мету проекту потрібно визначити, що саме треба зробити для її досягнення. Для цього проведемо декомпозицію кінцевої мети на більш прості етапи: формування концепції та проекту гри, розробка ігрового додатку, тестування, впровадження, створення документації. Майже кожен з отриманих підпроцесів, не має конкретно одної цілі, і одного результату, що міг би символізувати його завершення. Отже продовжуємо декомпозицію до тих пір, поки не отримаємо роботи, час виконання яких можна передбачити, а результат є чітким і однозначним.

В результаті виконання декомпозицій отримуємо ієрархічну структуру робіт (WBS – Work Breakdown Structure) для розробки ігрового додатку «IGRA» представлену на рисунку Б.1.

**Планування структури виконавців.** Розібравши проект на елементарні роботи, потрібно призначити кожній роботі виконавця, та відповідального за її завершення. Зазвичай це одна особа, вузькопрофільний спеціаліст, що ефективно виконує свою роботу.

На рисунку Б.2 представлено організаційну структуру планування проекту (OBS – Organizational Breakdown Structure). Список виконавців, що функціонують в проекті наведено в таблиці Б.2.

Таблиця Б.2 – Виконавці проекту

<b>Роль</b>	<b>Ім'я</b>	<b>Проектна роль</b>
Проектувальник	Дачко А.В.	Формує концепцію гри, проектує ігрову арену та персонажів.
Розробник	Дачко А.В.	Виконує розробку ігрового додатку.
Тестувальник	Дачко А.В. Федотова Н.А.	Відповідає за тестування ігрового процесу, та додатку в цілому.
Керівник проекту	Федотова Н.А.	Формує завдання на розробку проекту.
Менеджер проекту	Дачко А.В.	Відповідає за виконання термінів, розподіл ресурсів та завдань між

		учасниками. Виконує збір та аналіз даних.
--	--	---

Діаграма Ганта. Важливим етапом планування проекту є побудова діаграми Ганта (календарного графіку), що визначає чітку послідовність робіт, з конкретними датами початку та завершення етапів проекту. Графік дає можливість оцінювати стан справ на проекті, в будь який момент часу, і приймати управлінські рішення, що до розподілу ресурсів, для дотримання дедлайнів. Календарний графік проекту представлено на рисунку Б.3.



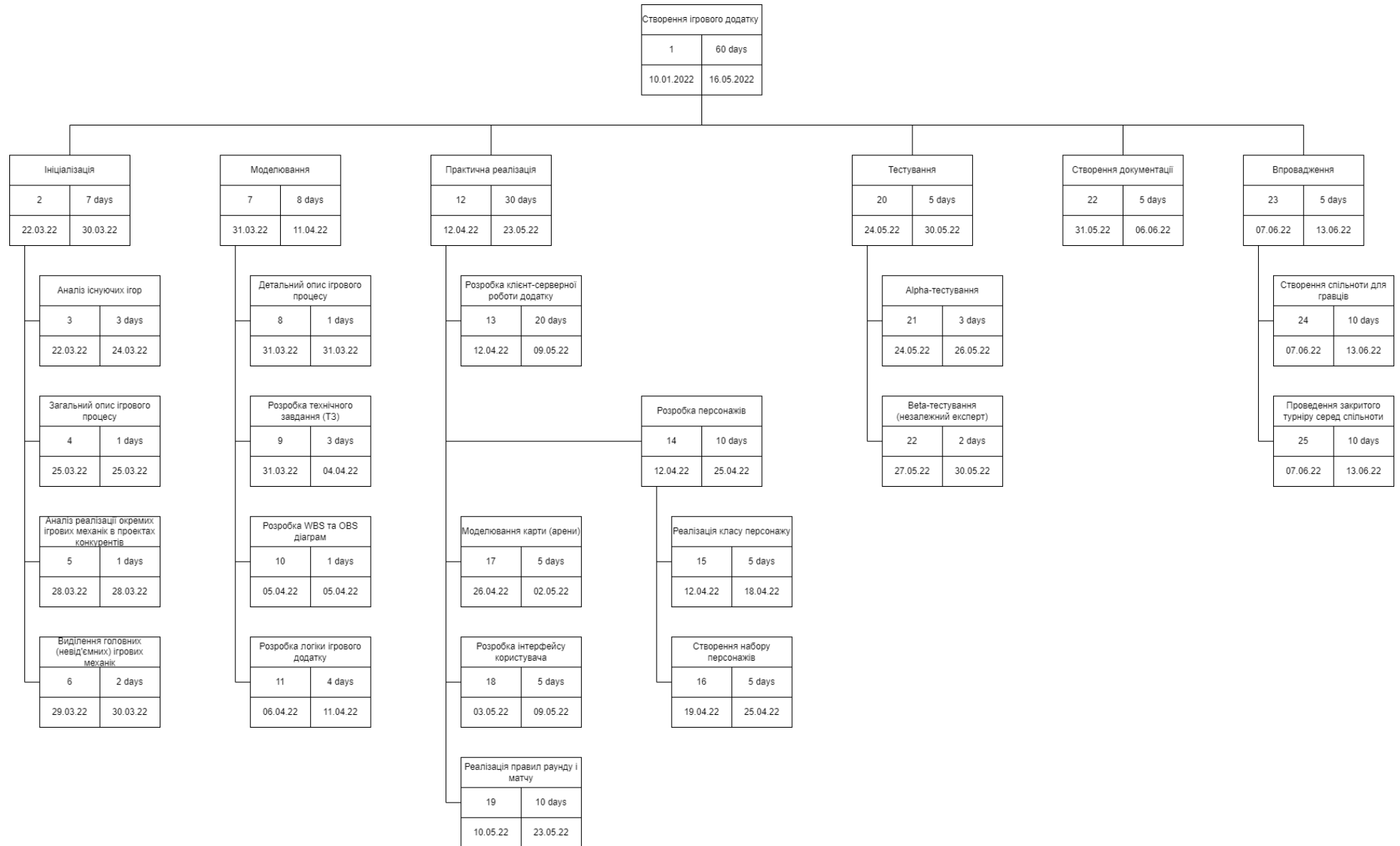


Рисунок Б.1 – WBS-структура робіт проекту

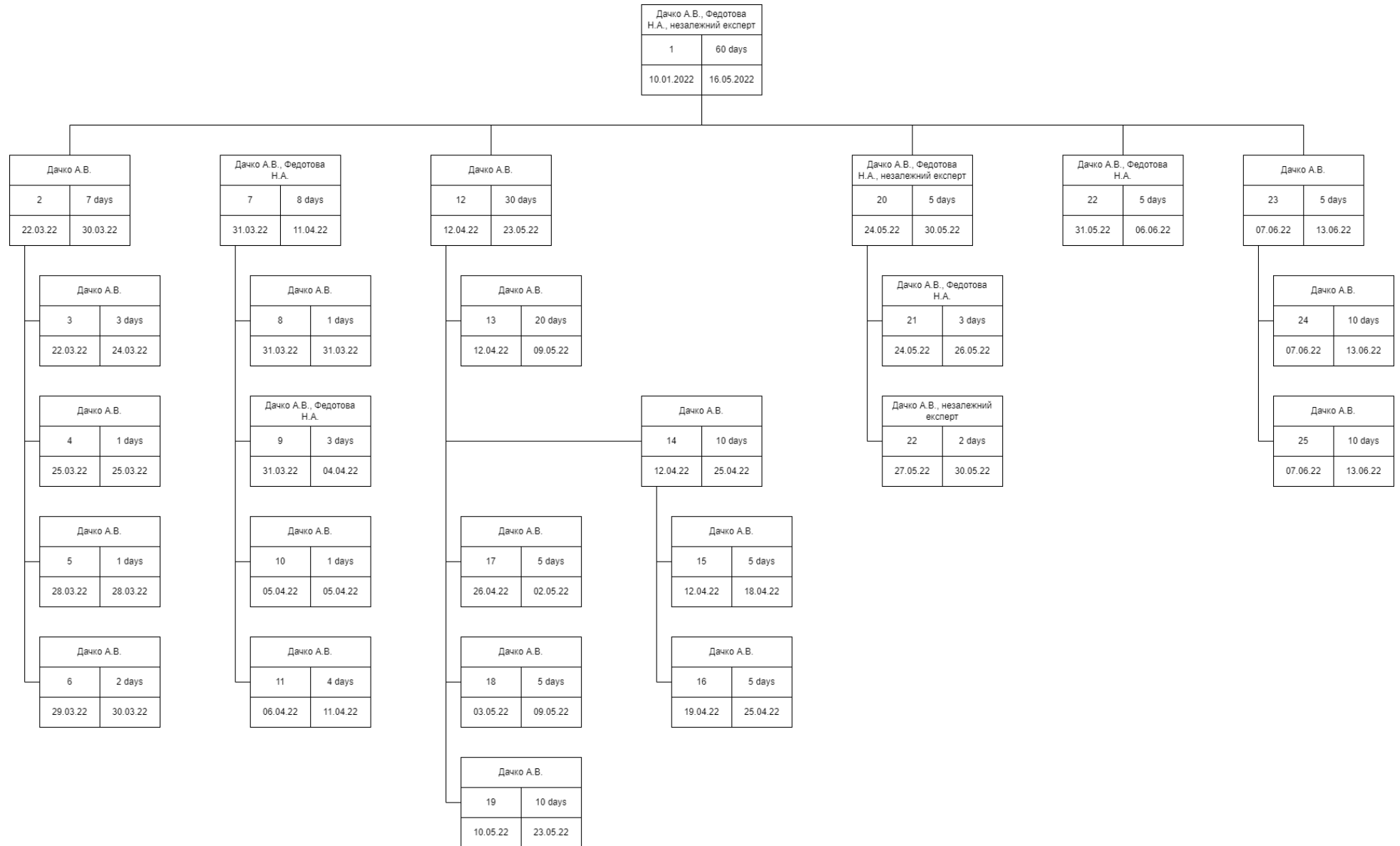


Рисунок Б.2 – OBS-структура робіт проекту

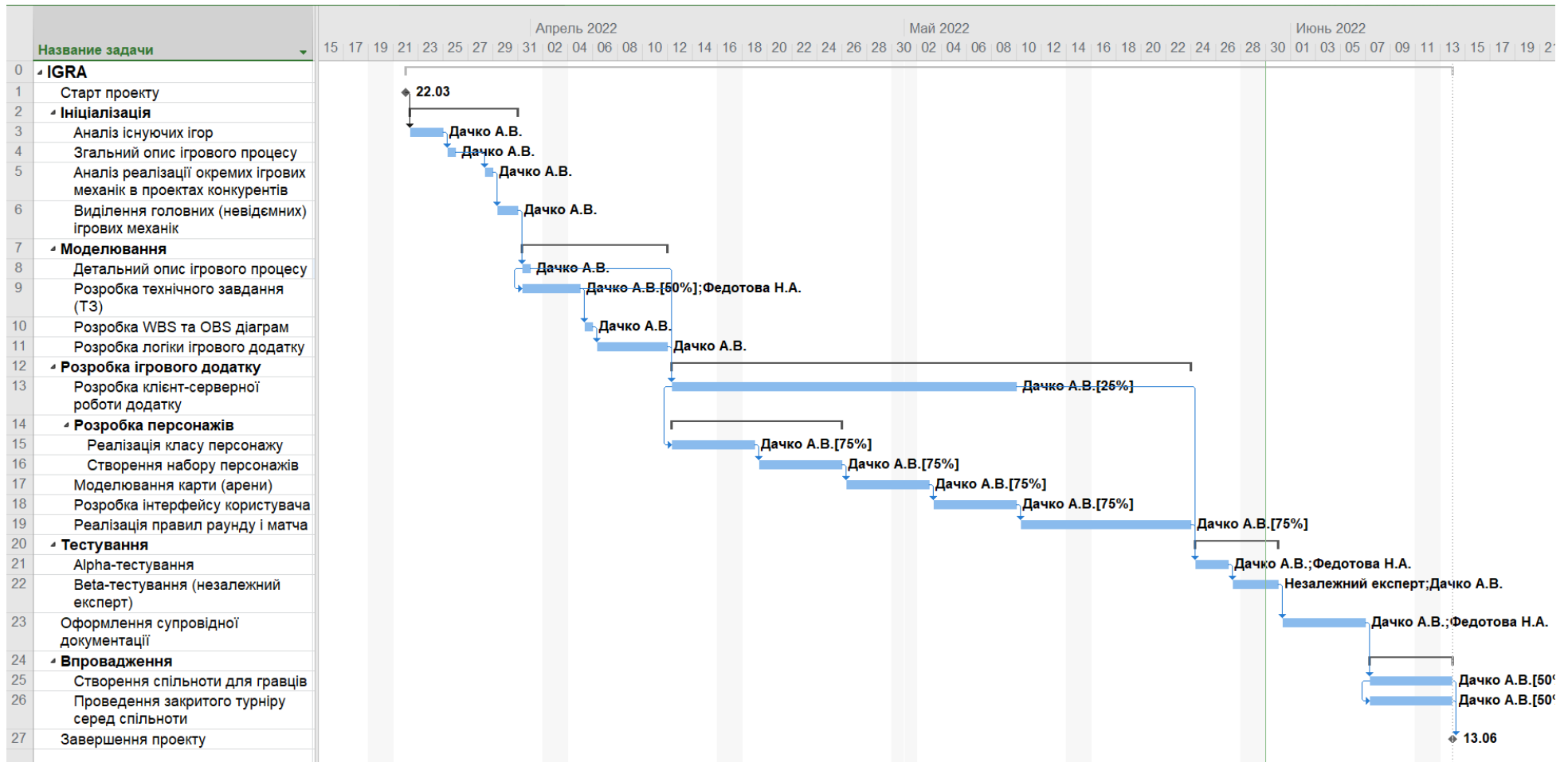


Рисунок Б.3 – Календарний графік проекту

**Управління ризиками проекту.** Під час планування проекту потрібно не забувати і про можливі ризики. Для їх виявлення потрібно провести якісну та кількісну оцінку ризиків. Для цього потрібно створити перелік всіх можливих ризиків для проекту. Кожному ризику потрібно визначити ймовірність виникнення та можливий вплив на проект (табл. Б.3).

Таблиця Б.3 – Шкала оцінювання ризиків за ймовірністю виникнення та величиною впливу

Оцінка	Ймовірність ризику	Вплив ризику	Типи ризику
1	Низька	Низький	Прийнятні
2	Середня	Середній	Виправдані
3	Висока	Високий	Недопустимі

Для зниження негативного впливу ризиків на проект потрібно розробити план реагування на них. У результаті планування для кожного ризику визначено ймовірність виникнення, вплив та ранг ризику. Для наочності створено матрицю ймовірностей та впливу ризиків (табл. Б.4). У відповідності до рангу розроблені стратегії реагування на ризики. Результати роботи з оцінювання ризиків наведені у таблиці Б.5.

Таблиця Б.4 – Матриця ризиків

Ймовірність ризику	Вплив загрози ризику				
	0,05	0,1	0,2	0,4	0,8
0,9					
0,7				RS_11	
0,5			RS_3, RS_4	RS_7,	
0,3		RS_10		RS_2, RS_5, RS_9	
0,1			RS_1, RS_6, RS_8		

Таблиця Б.5 – Ризики та стратегії реагування

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	План Б
RS_1	Відкритий	Поява альтернативного продукту	Низька	Середній	2	Провести попереднє дослідження альтернативних продуктів. Виділити основні унікальні властивості власного продукту.	Захистити власну розробку авторським правом.
RS_2	Відкритий	Нечітке завдання на розробку	Низька	Високий	3	Якісно та однозначно скласти завдання розробку.	Документація конкретних недоліків та їх виправлення.
RS_3	Відкритий	Низька кваліфікація розробника	Середня	Середній	4	Підвищити кваліфікацію, шляхом вивчення онлайн ресурсів та проходження онлайн курсів.	Врахувати час на додаткове вивчення ресурсів. Скорегувати графік розробки.
RS_4	Відкритий	Неоптимальний розподіл часу	Середня	Середній	4	Корегувати графік роботи. Виділити додаткові робочі дні за рахунок вихідних (до повернення в темп графіку)	Кількісно зменшити об'єм роботи (де це можливо, наприклад тимчасово зменшити різноманітність персонажів) до

Таблиця Б.5 – Ризики та стратегії реагування

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	План Б
							повернення в темп графіку.
RS_5	Відкритий	Вибір не ефективної технології розробки	Середня	Високий	6	Проаналізувати методи та засоби, для виконання проекту. Обрати зрозумілу та легку в використанні технологію розробки.	
RS_6	Відкритий	Неправильна оцінка в масштабі проекту	Низька	Середній	2	Провести детальний аналіз проекту. Визначити основні етапи проекту, виділити час на їх виконання. Проаналізувати масштаби проекту на основі додаткових джерел.	Перебудова стратегії реалізації проекту.
RS_7	Відкритий	Помилка проектування	Середній	Високий	6	На ключових етапах виконувати демонстрацію поточних результатів.	Здійснювати проміжний контроль результатів у ході виконання проекту.

Таблиця Б.5 – Ризики та стратегії реагування

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	План Б
RS_8	Відкритий	Збої в роботі програмного забезпечення	Низька	Середній	2	Підготувати резервні копії проекту.	Перенести проект на іншу версію програмного забезпечення
RS_9	Відкритий	Випадкова втрата даних	Низька	Високий	3	Тримати резервну копію проекту та всієї документації у хмарних сховищах та на окремих носіях. Періодично синхронізувати дані на різних носіях.	Відтворити не синхронізовану частину роботи.
RS_10	Відкритий	Реалізація непотрібного функціоналу	Низька	Низький	1	Пропустити реалізацію, якщо це можливо та функціонал не несе особливої цінності. Залишити все, що не заважає проекту	
RS_11	Відкритий	Недостатнє технічне забезпечення	Висока	Високий	9	Провести апгрейд та оновлення технічного забезпечення.	Знайти тимчасове технічне забезпечення для виконання розробки проекту.