

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ
СІКОРСЬКОГО»**

Навчально-науковий інститут атомної та теплової енергетики

Кафедра цифрових технологій в енергетиці

"На правах рукопису"
УДК 004.8

«До захисту допущено»
Завідувач кафедри
Наталія АУШЕВА
“ ___ ” _____ 2023 р.

Магістерська дисертація

на здобуття ступеня магістра
за освітньо-професійною програмою
“Цифрові технології в енергетиці”
зі спеціальності 122 “Комп’ютерні науки”

на тему «Методи реалізації мережевого шутеру від першої особи на
основі вбудованих можливостей ігрового двигуна Unreal Engine 4.27»

Виконав: студент 2 курсу, групи ТР-22мп

Савонік Юрій Іванович

(прізвище, ім’я, по батькові)

_____ (підпис)

Науковий керівник д.т.н., професор Аушева Наталія Миколаївна

(посада, вчене звання, науковий ступінь, ім’я ПРИЗВИЩЕ)

_____ (підпис)

Рецензент _____

(посада, вчене звання, науковий ступінь, ім’я ПРИЗВИЩЕ)

_____ (підпис)

Засвідчую, що у цій
магістерській дисертації
немає запозичень з праць
інших авторів без
відповідних посилань.

Студент _____

(підпис)

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ АТОМНОЇ ТА ТЕПЛОВОЇ
ЕНЕРГЕТИКИ**

Кафедра ЦИФРОВИХ ТЕХНОЛОГІЙ В ЕНЕРГЕТИЦІ

Рівень вищої освіти другий (магістерський)

За освітньо-професійною програмою “Цифрові технології в енергетиці”

Спеціальність 122 “Комп’ютерні науки”

ЗАТВЕРДЖУЮ

Завідувач кафедри

Наталія АУШЕВА

(підпис)

« ____ » _____ 2023 р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Савоніку Юрію Івановичу

(прізвище, ім’я, по батькові)

1. Тема дисертації «Методи реалізації мережевого шутеру від першої особи на основі вбудованих можливостей ігрового двигуна Unreal Engine 4.27»

Науковий керівник Аушева Наталія Миколаївна, д.т.н., професор

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

затверджені наказом по університету від «06» листопада 2023 року № 5152-с

2. Термін подання студентом дисертації 18 грудня 2023 р.

3. Об’єкт дослідження: процес реалізації мережевого шутеру від першої особи на основі вбудованих можливостей ігрового рушія Unreal Engine 4.27.

4. Вихідні дані: відображення графіки, відтворення звуків у грі, статистика та результати гри.

5. Перелік завдань, які потрібно розробити: проаналізувати існуючі рішення; розробити архітектуру логіки власної гри; розробити програмне забезпечення гри; провести аналіз розробленої гри; зробити висновок за результатами гри.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу: схема структури гри; схема взаємодії модулів гри.

7. Дата видачі завдання «24» жовтня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання магістерської дисертації	Строки виконання етапів магістерської дисертації	Примітка
1	Опис предметної області	01.10.2022-14.02.2023	
2	Аналіз існуючих рішень	15.03.2023-17.07.2023	
3	Вибір засобів розробки	01.09.2023-03.09.2023	
4	Створення алгоритмів для подальшої реалізації	04.10.2023-20.10.2023	
5	Розроблення програмного рішення	21.10.2023-11.11.2023	
6	Тестування розробленої системи	12.11.2023-15.11.2023	
7	Розробка стартап-проекту	16.11.2023-24.11.2023	
8	Оформлення текстової документації	25.11.2023-03.12.2023	

Студент

_____ (підпис)

Юрій САВОНІК

(ім'я ПРІЗВИЩЕ)

Науковий керівник

_____ (підпис)

Наталія АУШЕВА

(ім'я ПРІЗВИЩЕ)

РЕФЕРАТ

Актуальність теми. Розробка мережевих шутерів від першої особи є актуальним завданням у галузі геймдеву, так як вони надають гравцям можливість взаємодії в реальному часі, завдяки чому створюється цікавий та захоплюючий процес гри та сприяє розвитку глобальних ігрових спільнот. Unreal Engine версії 4.27, як ігровий рушій, надає потужний інструментарій для реалізації і налаштування таких проєктів.

Метою роботи є створення програмного забезпечення, що буде працювати у мережевому режимі.

Завдання дослідження:

- проаналізувати існуючі рішення, визначити їх сильні та слабкі сторони, які можна використати;
- обрати засоби розробки для реалізації власної гри;
- розробити алгоритми для вирішення ключових задач у грі;
- розробити програмне забезпечення гри;
- провести тестування гри.

Об'єкт дослідження – комп'ютерні та графічні технології.

Предмет дослідження – комп'ютерні та графічні технології для розробки мережевого FPS шутеру.

Практична цінність отриманих в роботі результатів полягає в наданні можливості користувачам використовувати розроблене програмне забезпечення. Запропоновано найбільш оптимізований метод розробки мережевого шутеру та його архітектуру, що легко модифікується та доналаштовується з використанням вбудованих можливостей Unreal Engine 4.27.

Структура та обсяг роботи. Магістерська дисертація складається з вступу, п'яти розділів, висновків до розділів, загальних висновків, списку використаних джерел (х джерел, у тому числі у – іноземною мовою) та додатків.

Загальний обсяг дисертації – 90 сторінок. Основний зміст викладено на 88 сторінках. Роботу проілюстровано 38 рисунками, 4 формулами та 18 таблицями.

Ключові слова: Unreal Engine, C++, мережевий шутер, геймдев, онлайн, гра від першої особи.

ABSTRACT

Relevance of the Topic. The development of online first-person shooters is an urgent task in the field of game development, as they provide players with the opportunity to interact in real time, which creates an interesting and exciting gameplay and contributes to the development of global gaming communities. Unreal Engine version 4.27, as a game engine, provides a powerful toolkit for implementing and customizing such projects.

Research Objective. The aim of this work is to create software that operates in network mode. The research tasks include:

- analyze existing solutions, apply their strengths and weaknesses that can be used;
- apply development tools to implement your own game;
- develop algorithms for solving key tasks in the game;
- develop game software;
- test the game.

Research Subject. The subject of the research is computer and graphic technologies.

Practical Value. The practical value of the obtained results lies in providing users with the opportunity to utilize the developed software. The work proposes the most optimized method for developing a network shooter and its architecture, which can be easily modified and fine-tuned using the built-in features of Unreal Engine 4.27.

Structure and Scope of the Work. The master's dissertation consists of an introduction, four chapters, conclusions to chapters, general conclusions, a list of references (x sources, including y in foreign languages), and appendices.

Total volume of the dissertation: 89 pages. The main content is presented on 87 pages. The work is illustrated with 36 figures and 3 tables.

Keywords: Unreal Engine, C++, network shooter, game development, online, first-person game.

ЗМІСТ

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ	10
ВСТУП.....	12
1 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ ПОСТАВЛЕНОЇ ЗАДАЧІ.....	14
1.1 Загальні положення та аналіз предметної області.....	14
1.2 Відомі рішення	16
1.3 Порівняння існуючих рішень.....	21
1.4 Вибір засобів розробки для створення гри.....	23
Висновки до розділу 1	27
2 РОЗРОБЛЕННЯ АЛГОРИТМІЧНО-ПРОГРАМНИХ МЕТОДІВ ДЛЯ МЕРЕЖЕВОГО ШУТЕРУ ВІД ПЕРШОЇ ОСОБИ	28
2.1 Постановка задачі.....	28
2.2 Алгоритм роботи анімації переміщення.....	29
2.3 Алгоритм обчислення повороту камери.....	30
2.4 Алгоритм принципу поведінки ІІІ	34
2.5 Алгоритми обробки стрільби зі зброї та використання патрону .	37
2.6 Алгоритм обробки смерті персонажа	39
Висновки до розділу 2	40
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ГРИ	41
3.1 Архітектура гри	41
3.2 Розробка програмних модулів	43
Висновки до розділу 3	57

4 ЗАПУСК ТА ТЕСТУВАННЯ РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	58
4.1 Запуск програмного забезпечення.....	58
4.2 Тестування програмного забезпечення.....	64
Висновки до розділу 4	70
5 СТАРТАП ПРОЕКТ	71
5.1 Опис ідеї продукту	71
5.2 Технологічний аудит ідеї продукту	74
Кінець таблиці 5.3.	75
5.3 Аналіз ринкових можливостей стартап-проекту	75
5.4 Ринкова стратегія	85
Висновки до розділу 5	87
ВИСНОВКИ.....	89
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	90

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

Шутер (від англ. shooter — «стрілець» або стрілянка) — жанр відеоігор, піджанр action, основу ігрового процесу якого складає стрілянина по цілях. Часто завданням гравця є знищення усіх цілей на рівні, або/та набір найбільшої кількості очок, а не проходження сюжету

HUD – Heads-up display (частина графічного інтерфейсу користувача, яка дозволяє користувачу отримувати різноманітну інформацію, не викликаючи

додаткові меню).

Віджет (від англ. widget) у графічному інтерфейсі користувача — це елемент взаємодії, такий як кнопка або смуга прокрутки. Віджети призначені як інструмент для швидкого доступу до певної інформації чи сервісів.

Реплікація (від англ. Replication – реплікація) – є основною частиною мережі, мета її процесу полягає в синхронізації даних і викликів процедур між клієнтами та серверами, це акт передачі сервером інформації та даних клієнтам.

Reliable (від англ. reliable – надійний) – означає, що подія завжди буде викликатись на сервері або на стороні клієнта. Його слід вмикати лише для подій, які обов'язково повинні виконуватися.

RepNotify (від англ. Replication notify – сповіщення реплікації) – змінні, позначені як RepNotify, мають спеціальну функцію, яка автоматично викликається щоразу, коли значення змінюється, як на мережевому центрі, так і на віддалених машинах.

Геймдев (від англ. Video game development) — це процес створення відеоігри, яким займається розробник відеоігор (англ. video game developer), котрий може бути як однією людиною так і компанією з сотнями співробітників.

Геймплей – (від англ. *gameplay* – процес гри) — термін, яким називають особливості взаємодії людини з відеоігрою. Ці особливості створюються за допомогою правил, завдань та способів їх розв'язування, які пропонує гра.

LAN (від англ. *local area network* – локальна мережа) — це об'єднання певного числа комп'ютерів на відносно невеликій території.

UI (від англ. *User interface* - користувацький інтерфейс») – користувацький інтерфейс, що необов'язково може бути тільки графічним: тактильний, голосовий або звуковий.

III (від англ. *artificial intelligence*(AI) – штучний інтелект) — розділ комп'ютерної лінгвістики та інформатики, який швидко розвивається, і зосереджений на розробці інтелектуальних машин, здатних виконувати завдання, які зазвичай потребують людського інтелекту. Ці завдання можуть варіюватися від простих дій, як-от розпізнавання мови чи зображень, до більш складних завдань, як-от ігри чи водіння автомобіля.

Геймер (від англ. *gamer* — гравець) — людина, яка грає в інтерактивні ігри, особливо у відеоігри, настільні рольові ігри, а також картярські ігри, що базуються на навичках, і яка, як правило, грає протягом тривалого періоду часу.

FPS (від англ. *first-person shooter*) — шутер від першої особи.

ВСТУП

Створення мережевого FPS – це процес створення мережевого шутера[1] від першої особи на базі вбудованих можливостей ігрового двигуна Unreal Engine 4.27, який включає в себе використання різноманітних методів та технік для досягнення оптимальної продуктивності та реалізації високоякісного геймплею. Однією з ключових складових є ефективна обробка мережевої взаємодії, що дозволяє гравцям взаємодіяти між собою у режимі реального часу.

Для досягнення цієї мета використовується низка вбудованих функцій Unreal Engine 4.27, таких як система мережевого взаємодії, реплікаційний граф[15], використання ігрової сесії та інтеграція серверної та клієнтської логіки. Передові можливості двигуна дозволяють автоматизувати синхронізацію гравців, обробку подій та взаємодію об'єктів в мережевому середовищі.

Реалізація графічної частини включає в себе використання вбудованих інструментів для моделювання персонажів та оточення, створення текстур та використання освітлення для досягнення реалістичного вигляду гри. Зокрема, можливості Unreal Engine 4.27 у сфері обробки високоякісної 3D-графіки дозволяють створювати реалістичні світи, реалістичних персонажів та живе навколишнє середовище.

Для забезпечення високої продуктивності гри по мережі використовуються вбудовані інструменти оптимізації, такі як масштабування рівня деталей, використання LOD, а також оптимізоване управління ресурсами.

Відзначається також використання інструментів Unreal Engine 4.27 для реалізації різноманітних геймплейних механік, включаючи стрільбу, рух та інші взаємодії. Ігровий двигун дозволяє зручно і швидко імплементувати та

налаштовувати функціонал гри, зменшуючи час розробки та полегшуючи процес створення високоякісного мережевого шутера від першої особи.

Мета і завдання дослідження.

Метою роботи є створення онлайн гри від першого лиця на основі вбудованих функцій ігрового рушія Unreal Engine 4.27.

Для досягнення поставленої мети ставляться наступні завдання:

- проаналізувати існуючі рішення, визначити їх сильні та слабкі сторони, які можна використати;
- обрати засоби розробки для реалізації власної гри;
- розробити алгоритми для вирішення ключових задач у грі;
- розробити програмне забезпечення гри;
- провести тестування гри.

1 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ ПОСТАВЛЕНОЇ ЗАДАЧІ

В даному розділі було оглянуто існуючі рішення поставленої задачі та обрано засоби розробки.

1.1 Загальні положення та аналіз предметної області

На поточний час, відеоігри можна класифікувати за видом активностей, які гравець робить в іграх такого жанру найчастіше. Тож, звісно, в загальному, відеоігри можна розподілити на ігри руху, планування, сюжету або спілкування та дії або контролю. Однак найчастіше можна почути класифікацію ігор за жанрами. Жанр – сукупність ознак гри, яка визначається особливістю геймплею.

Найбільш популярними жанрами ігор є наступні:

- 1) стратегії (часто сюжетні ігри, що вимагають від гравця постійно аналізувати ситуацію для того, щоб приймати найкращі рішення. Самі по собі ігри часто щось більше, ніж просто логічні ігри, як ті ж шашки або шахи);
- 2) аркади (ігри, що, вимагають від геймера спритність та уважність);
- 3) симулятори (ігри, що є віртуальними прообразами реальних об'єктів та процесів, що пов'язані з отриманням віртуального, проте, часто, релевантного досвіду з певним транспортом, пристроєм і т.п.);
- 4) шутери (у цих іграх, основа геймплею полягає у стрільбі зі зброї по цілям, що можуть бути як об'єктами, так і віртуальними прототипами людей. У таких іграх часто головною задачею є знищення усіх цілей на рівні або отримання найбільшої кількості очок).[1]

Окрім цього, ігри також ділять на ті, у які грає лише один гравець (однокористувацькі ігри) та у які грає більше, ніж один користувач, завдяки мережі (мультиплеєрні або мережеві ігри).

Останнім критерієм класифікації ігор є поділ їх в залежності від платформи на яку гра випускається. Таким чином існує такий поділ: ігри на ПК, ігри на мобільні пристрої, ігри на ігрові консолі (сюди включають Nintendo, Sony PlayStation та Xbox) та мультиплатформні ігри (ігри, що випускаються на більше, ніж одну платформу).

Насьогодні, у гейміндустрії є тенденція на реалізацію міжплатформних мультиплеєрних ігор.

Наразі, по праву, одним з найпопулярніших і найприбутковіших можна вважати жанр шутеру(а якщо точніше, то мультиплеєрного шутеру). Підтвердженням правдивості тези про популярність і прибутковість може слугувати статистика онлайн крамниці ігор Steam (сервіс компанії Valve, відомого розробника відеоігор, який також надає послуги цифрової дистрибуції, багато-користувацьких ігор і спілкування гравців). Згідно з статистики з 100 найпопулярніших ігор 28 є онлайн шутерами і серед найприбутковіших 27 ігор є онлайн шутерами. Окрім цього необхідно також зважати на ринок ігор в кіберспорті. Адже, це окрім потенційних контрактів з різного виду компаніям робить також і пасивну рекламу гри та її розробнику. Проаналізувавши сайт на якому знаходяться топ 20 найпопулярніших кіберспортивних дисциплін на даний час можна побачити, що з них 12 – це онлайн шутери.

Онлайн шутери в свою чергу можна умовно розділити на 4 категорії:

- ігри від першої особи;
- ігри від другої особи;
- ігри від третьої особи;
- ігри від першої та третьої особи (тобто користувач може в будь-який момент переключитись у зручний для нього режим огляду).

Зважаючи на ці факти, можна прийти до висновку, що гравцям найбільше подобається грати у мультиплеєрні шутери від першої особи.

1.2 Відомі рішення

Існує дуже велика кількість онлайн шутерів від першої особи у ігровій індустрії, однак буде розглянуто 5 найпопулярніших представників цього жанру:

- Counter Strike 2;
- Valorant;
- Call of Duty: Warzone;

1.2.1 Counter Strike 2

Counter-Strike 2 (скорочено CS 2) — багатокористувацька гра в жанрі тактичного шутера від першої особи, розроблена компанією Valve. Є 5-ю грою в серії Counter-Strike.



Рис. 1.1 – Офіційний баннер гри Counter Strike 2

Гру було описано як «капітальний ремонт кожної системи, кожного елемента контенту і кожної частини ігрового процесу». У березні 2023 року було проведено тест гри. Покращення включають перероблені карти і поліпшені

аудіо-візуальні ефекти, а також зміни у використанні димових гранат і багато іншого. Освітлення, текстури та дизайн були змінені.

Компанія Valve повідомила: «Дим тепер взаємодіє з іншими ігровими процесами, створюючи нові можливості. Кулі та гранати можуть виробляти дим, на короткий час закриваючи лінію огляду або розширюючи оклюзію». Дим також може «природним чином заповнювати простір» і реагувати на освітлення.



Рис. 1.2 – Порівняння вигляду димової гранати у CS 2 та CS:GO

Компанія також підтвердила перенесення ігрових предметів із Counter-Strike: Global Offensive у Counter-Strike 2. Також був проведений редизайн графічного інтерфейсу гри, зокрема головного меню. Вихід оновлення відбувся в обід 27 вересня за місцевим часом, а за київським часом близько 23:00.[3]

1.2.2 Valorant

Valorant (стилізовано як VALORANT) — безкоштовний інтелектуальний шутер від першої особи, розроблений компанією Riot Games для Microsoft Windows. Вперше був представлений під кодовою назвою «Project A» у жовтні 2019 року. Гра розпочала закриті бета-тестування з обмеженим доступом 7 квітня 2020 року, після чого 2 червня 2020 року відбувся офіційний випуск. Розробка гри почалася в 2014 році. Valorant черпає натхнення із серії тактичних шутерів Counter-Strike.



Рис. 1.3 – Офіційний баннер гри Valorant

Сюжет гри розгортається в найближчому майбутньому. Геймплей даної гри наступний: гравці грають за одного з набору агентів, персонажів, розроблених на основі кількох країн і культур по всьому світу.

У режимі основної гри гравці призначаються або до атакуючої, або до команди, що захищається, причому кожна команда має по п'ять гравців. Агенти мають унікальні здібності, кожна з яких вимагає зарядів, а також унікальну

кінцеву здатність, яка вимагає заряджання через вбивства, смерть та спеціальні кулі. Кожен гравець починає кожен раунд з «класичним» пістолетом і одним або кількома зарядами «спеціальних здібностей». Іншу зброю та заряди здібностей можна придбати за допомогою внутрішньоігрової економічної системи, яка присуджує гроші на основі результатів попереднього раунду, будь-яких вбивств, за які несе відповідальність гравець, і виконання будь-яких цілей.



Рис. 1.4 – Приклад геймплею Valorant

У грі є асортимент зброї, включаючи вторинну зброю, як-от стрілецьку зброю, і основну зброю, як-от пістолети-кулемети, дробовики, кулемети, штурмові гвинтівки та снайперські гвинтівки. Існує автоматична та напівавтоматична зброя, кожна з яких має унікальну схему стрільби, якою гравець повинен керувати, щоб мати можливість точно стріляти. Різні агенти дозволяють гравцям знаходити більше способів встановити «Спайк» (аналог бомби) і знешкодити ворогів. Наразі гра пропонує на вибір 21 агента.[4]

1.2.3 Call of Duty: Warzone

Call of Duty: Warzone — мультиплатформна безкоштовна гра в жанрі багатокористувацького шутера від першої особи і королівської битви, розроблена компаніями Infinity Ward і Raven Software і видана Activision 10 березня 2020 року для ПК, PlayStation 4 та Xbox One. Warzone дозволяє вести онлайн-битви на одному полі бою незалежно від платформи між 150—200 гравцями, які перебувають у вигаданому місті Верданськ, яке нагадує Донецьк.



Рис. 1.5 – Офіційний баннер гри Call of Duty: Warzone

У грі є три основні режими — «здобич», «королівська битва» і «супроводження грузу». Warzone використовує нову внутрішньоігрову валюту, якою можна розплатитися на станціях покупки всередині Верданську та на його околицях. Також гравці можуть використовувати валюту гри для «воскресіння» дружніх гравців, покупки предметів, наприклад, серій вбивств або протигазів. На момент запуску гри, Warzone дозволяв грати тільки в загоні з трьох чоловік. За тиждень, 17 березня 2020 року, в гру було додано соло-режим. Пізніше, 8 квітня 2020 року, одночасно з патчем третього сезону Modern Warfare, в гру було додано загін з чотирьох чоловік.[5]



Рис. 1.6 – Приклад геймплею Call of Duty: Warzone

1.3 Порівняння існуючих рішень

1.3.1 Counter Strike 2

Гра має свої переваги та недоліки.

Переваги:

- значне покращення графіки, по зрівнянню з попередньою версією гри;
- значне покращення аудіо-візуальних ефектів;
- було додано нові механіки гри, що дещо змінюють процес гри (Варто зазначити, що, хоча, ці механіки гравці і зустріли з позитивними відгуками, однак компанію звинуватили у копіюванні механік з гри Valorant).

Недоліки:

- спільнота гравців сприйняла перехід з Counter-Strike: Global Offensive на Counter Strike 2 з розчаруванням через те, що багато вмісту, доступного в Counter-Strike: Global Offensive, стало недоступно;

- деякі гравці втратили доступ до платних предметів, які були придбані в попередній частині серії;
- гра не має підтримки Mac OS;
- гра має достатньо погану оптимізацію;
- гра досі має погану систему боротьби проти читерів.

1.3.2 Valorant

У гри є певні переваги та недоліки.

Переваги:

- гра має хорошу оптимізацію;
- по зрівнянню з іншими представниками даного жанру, гра надає достатньо новий процес гри;
- гра має хорошу систему боротьби проти читерів.
- гра має хорошу систему боротьби з представниками токсичної спільноти.

Недоліки:

- гра має достатньо поганий внутрішньоігровий магазин, у зв'язку з чим гравець повинен чекати на певний предмет неозначений проміжок часу, замість того, щоб мати можливість купити його у будь-який момент;
- гра досі не має системи запису, збереження і відтворення попередніх зігранних матчів;
- гра має досить дивну систему реагування на пошкодження гравця;
- у грі наявний поганий застосунок з відкриття гри з великою кількістю неочікуваних помилок.

1.3.3 Call of Duty: Warzone

Розглянемо основні переваги та недоліки гри.

Переваги:

- гра надавала досить новий погляд на франшизу;
- гра надає дуже хорошу і реалістичну стрільбу;
- гра має дуже приємну спільноту гравців та цікаві мапи.

Недоліки:

- гра має погану систему боротьби проти читерів;
- гра розроблялась на старому ігровому рушію, в зв'язку з чим досі можуть виникати неочікувані помилки;
- хоча гра і непогано збалансована, однак вона вимагає великий обсяг вільної пам'яті.

1.4 Вибір засобів розробки для створення гри

На поточний час, можливість реалізації мережевої гри присутня у наступних ігрових рушіях:

- Unity 3D;
- Unreal Engine;
- Cry Engine.

Розглянемо коротко рушії Unity 3D та Unreal Engine. Cry Engine не буде розглянуто через те, що рушії має дуже мало інформації про інструментарій та скупі документацію.[7]

Unity 3D – це ігровий рушії, що виступає як інструмент для розробки додатків та ігор, що працює під операційними системами Android, Windows, Apple iOS, Linux, а також на гральних консолях PlayStation та Xbox.

Рендеринг зображення відбувається через віртуальну камеру огляду. В робочій області редактора ігрова сцена може розміщуватися як завгодно, а при рендерингу — так, як її видно з камери. В сцені може бути декілька камер, які рухаються за персонажем чи за вказаною траєкторією

Графічний рушій використовує DirectX, OpenGL, OpenGL ES та спеціальне власне API, в залежності від платформи.

Скриптова система ігрового рушія зроблена на Mono — вільному відкритому проєкті з реалізації .NET Framework. Програмісти можуть використовувати UnityScript (власна скриптова мова, подібна до JavaScript та ECMAScript) або C#.

Цей рушій має свої переваги та недоліки при створенні мережевої гри.[6]

Переваги:

- Unity 3D має велику кількість готових компонентів, ресурсів та інструментів, які полегшують розробку гри. Це дозволяє ефективно втілювати різноманітні ідеї і функціонал у гру.
- Unity має велику та активну спільноту розробників, а також обширну документацію, що значно полегшує вирішення проблем, шляхом отримання порад та обміном досвідом.
- Unity має вбудовані інструменти для створення мережевого геймплею, що спрощує реалізацію онлайн шутера.

Недоліки:

- Unity потребує більше зусиль для досягнення високої продуктивності в шутерах від першої особи, особливо при великій кількості гравців та складних сценах.
- Певні можливості Unity доступні тільки у платних версіях, що істотно підвищує витрати на розробку гри.
- Хоча Unity має вбудовані інструменти для мережевого геймплею, але для складних шутерів від першої особи може бути потрібно додатково налаштовувати мережевий код.

- У деяких випадках, при роботі з великим обсягом гравців та динамічним середовищем, можуть виникнути проблеми з продуктивністю та оптимізацією.

Загалом, вибір Unity 3D для розробки онлайн шутера від першої особи може бути доцільним у випадку, якщо гра буде мати невелику кількість гравців. Також, внаслідок того, що рушій використовує мову C#, швидкість роботи гри буде нижчою, ніж у випадку використання C++, який використовується у більшості ігрових рушіїв.

Unreal Engine - ігровий рушій, який розробляється та підтримується компанією Epic Games. Рушій, насамперед, пристосований насамперед для шутерів від першої особи, рушій використовувався і для створення ігор інших жанрів.

Написаний мовою програмування C++, рушій дає змогу створювати ігри для більшості операційних систем і платформ таких як Windows, Linux, Mac, Xbox, PlayStation, Wii, Dreamcast і Nintendo. У грудні 2009 Марк Рейн продемонстрував як працює рушій Unreal Engine 3 на iPod Touch і iPhone 3GS. Пізніше, робота рушія була продемонстрована на комунікаторі Palm Pre, що базується на мобільній платформі webOS.

Ігровий рушій, для спрощення портування, використовує модульну систему залежних компонентів: підтримує різні системи рендерингу (Direct3D, OpenGL, Pixomatic), відтворення звуку (EAX, OpenAL, DirectSound3D, засоби голосового відтворення тексту, розпізнавання мовлення (тільки для Xbox 360, PlayStation 3, Nintendo Wii та Microsoft Windows), модулі для роботи з мережею й підтримання різних пристроїв вводу.

Для багатокористувацького режиму гри підтримуються технології Windows Live, Xbox Live і GameSpy, що дає можливість під'єднати до 64 гравців (клієнтів) одночасно. Попри те, що офіційно засоби розробки не містять у собі

підтримки великої кількості клієнтів на одному сервері, рушій використовувався для створення MMORPG-ігор.[8]

Рушій дає можливість програмувати за допомогою мови C++ та мови візуального скриптингу Blueprint.

Використання UE4 для реалізації онлайн шутера від першої особи (FPS) також має свої переваги та недоліки.

Переваги:

- UE4 славиться своєю потужною графікою та можливостями візуалізації, що робить його ідеальним для створення реалістичних та вражаючих світів у шутерах від першої особи.
- UE4 має широкий набір готових компонентів, систем фізики, анімацій і модульну структуру, що спрощує розробку та реалізацію різноманітних ідей.
- UE4 має вбудовану систему мережевого геймплею, яка полегшує створення онлайн ігор та взаємодії гравців.
- UE4 має відкритий доступ до внутрішнього коду, тому можна окремо налаштувати його елементи.

Недоліки:

- Реалістична графіка та велика кількість функцій можуть ставити високі вимоги до ресурсів комп'ютера та графічної карти, якщо не працювати над оптимізацією гри.

Після того, як було розглянуто переваги та недоліки цих двох рушіїв було вирішено, що Unreal Engine 4 для кросплатформеного онлайн шутера від першої особи підходить краще, ніж Unity 3D. Також, Blueprint буде використовуватись лише для налаштування деяких елементів, адже він не зможе надати тієї ж швидкості роботи, що може бути при використанні C++. Також, банально, деякі функції для онлайн режиму у Blueprint просто немає.

Висновки до розділу 1

У першому розділі магістерської дисертації було розглянуто та проаналізовано предметну область. Також було розглянуто найпопулярніших представників жанру мультиплеєрного шутеру, наведено їх основні переваги та недоліки та наведено приклади геймплею. Насамкінець було обрано програмний засіб для реалізації гри, зважаючи на його переваги, над іншим засобами – ігровий рушій Unreal Engine 4 для розробки гри.

2 РОЗРОБЛЕННЯ АЛГОРИТМІЧНО-ПРОГРАМНИХ МЕТОДІВ ДЛЯ МЕРЕЖЕВОГО ШУТЕРУ ВІД ПЕРШОЇ ОСОБИ

В даному розділі було розроблено ключові алгоритми для гри.

2.1 Постановка задачі

Необхідно розробити алгоритмічно-програмні методи для реалізації мережевого FPS шутеру.

Графіка та елементи керування персонажем (налаштування різкості керування вхідним девайсом) у грі має бути змінною при налаштуванні геймера та зберігатись, щоб не було необхідності кожного разу встановлювати параметри заново.

Сесія у грі має мати можливість бути створеною з налаштовуваними параметрами (можна налаштовувати: мапи, часу та режиму гри, кількості гравців, типу з'єднання та заповнення вільних місць ботами, що керуються штучним інтелектом, запис геймплею).

У грі повинна бути можливість шукати активні та наявні у грі відкриті сесії та приєднуватись до одного з отриманих результатів.

Користувач повинен мати можливість заходити до внутрішньоігрового магазину з можливістю купівлі необов'язкового додаткового контенту.

Користувач повинен мати можливість переглядати та видаляти збережені записи процесу гри.

Гра містить окреме вікно для можливості перегляду статистики топ гравців у грі з наявною та правдивою інформацією.

Гра має наявну робочу та реалістичну систему штучного інтелекту, що може адекватно аналізувати та реагувати на зміну навколишнього середовища та дії гравців.

Гра має чесне реагування на дії гравців та працює стабільно. Також у гри повинна бути хороша оптимізація, як самого процесу гри, так і роботи серверу.

2.2 Алгоритм роботи анімації переміщення

У Unreal Engine є вбудований зручний редактор для створення та налаштування роботи логіки анімацій. Окрім цього, він надає можливість створювати, налаштовувати та використовувати внутрішні об'єкти, що значно пришвидшують процес налаштування анімацій. Одним з прикладів таких об'єктів є об'єкт змішування анімацій. В цьому об'єкті є можливість додати необхідну кількість анімацій та створити змінні, міняючи за допомогою коду, які одна з анімацій плавно переходить в іншу. Для того, щоб зрозуміти як використати цю можливість, варто спочатку зрозуміти логіку, за якою можна знайти поточний напрямок руху гравця.

Для того, щоб знайти нормалізований вектор необхідно використати наступні

формули:

$$Length = \sqrt{a_x^2 + a_y^2 + a_z^2} \quad (2.1)$$

$$a = \left(\frac{a_x}{Length}; \frac{a_y}{Length}; \frac{a_z}{Length} \right) \quad (2.2)$$

де $Length$ - довжина вектора,

a_x - координата вектора по осі x;

a_y - координата вектора по осі y;

a_z - координата вектора по осі z;

Для того, щоб знайти скалярний добуток використовують наступну формулу:

$$\vec{a} * \vec{b} = |a| * |b| * \cos(\vec{a} * \vec{b}) \quad (2.3)$$

Для того, щоб знайти векторний добуток використовують наступну формулу:

$$\vec{a} \times \vec{b} = a_y b_z - a_z b_y + a_z b_x - a_x b_z + a_x b_y - a_y b_x \quad (2.4)$$

Використовуючи формули (2.1) та (2.2) знаходиться нормалізований вектор швидкості. Напрямок буде рівний косинусу між нормалізований вектором швидкості та напрямним вектором гравця за формулою (2.3) помножену на синус координати z векторного добутку між нормалізований вектором швидкості та напрямний вектором актора за формулою (2.4). Далі алгоритм діє за схемою, що зображена на рисунку 2.1.

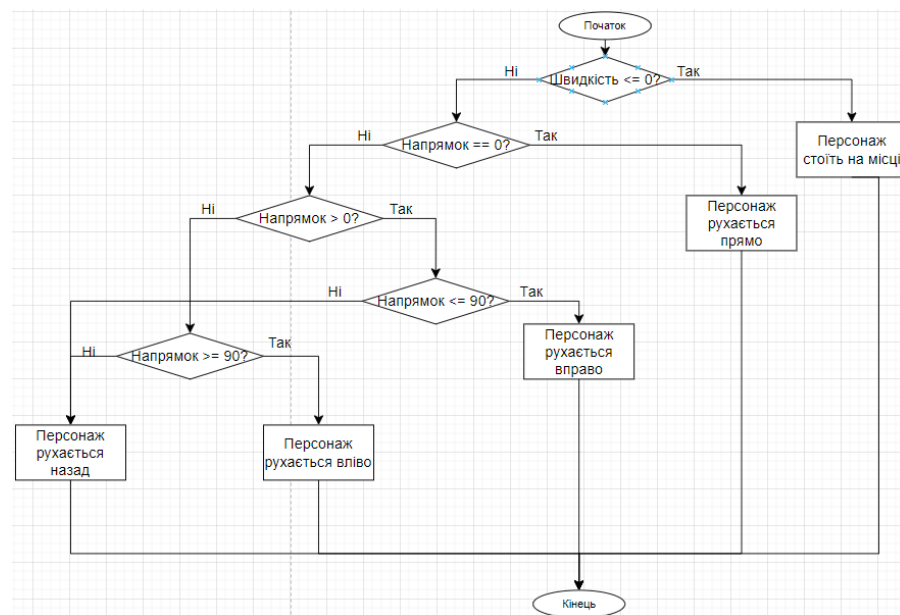


Рисунок 2.1 – Схема алгоритму роботи анімації переміщення

2.3 Алгоритм обчислення повороту камери

При створенні мультиплеєрного шутеру від першої особи потрібно вирішити наступну проблему. Гравець №1 повинен бачити на своєму екрані лише зброю і руки, в той час як для гравця №2, гравець №1 повинен

відобразитись з повною моделю гравця. Цю проблему можна вирішити створюючи на місці гравця дві моделі, кожна з яких буде анімована і коректно реагуватиме на зміну навколишнього середовища. Модель, яка буде використовуватись, коли гравець бачить себе, буде налаштована таким чином, щоб тільки гравець міг бачити цю модель, а інші гравці її бачити не будуть і ця модель не буде ніяк реагувати на фізичні взаємодії з нею. Другу модель, що відобразатиметься для інших гравців буде налаштовано прямо протилежно (тобто, тільки гравець не бачитиме цю модель, в той час як інші гравці будуть бачити її і модель буде реагувати на фізичні взаємодії з нею).

Також необхідно, щоб камера коректно працювала для цих моделей у вигляді одного алгоритму. Це може бути вирішеним наступним чином.

У нас є окремо клас гравця та клас контролеру гравця (де налаштовується логіка реагування гравця на певні натиски кнопок на елементі управління, що трапляється з гравцем при його смерті і т.п.). Цей клас контролеру містить в собі можливість налаштування класу менеджера камери. І в ньому можна реалізувати наступний алгоритм, що зображений на рисунку 2.2.

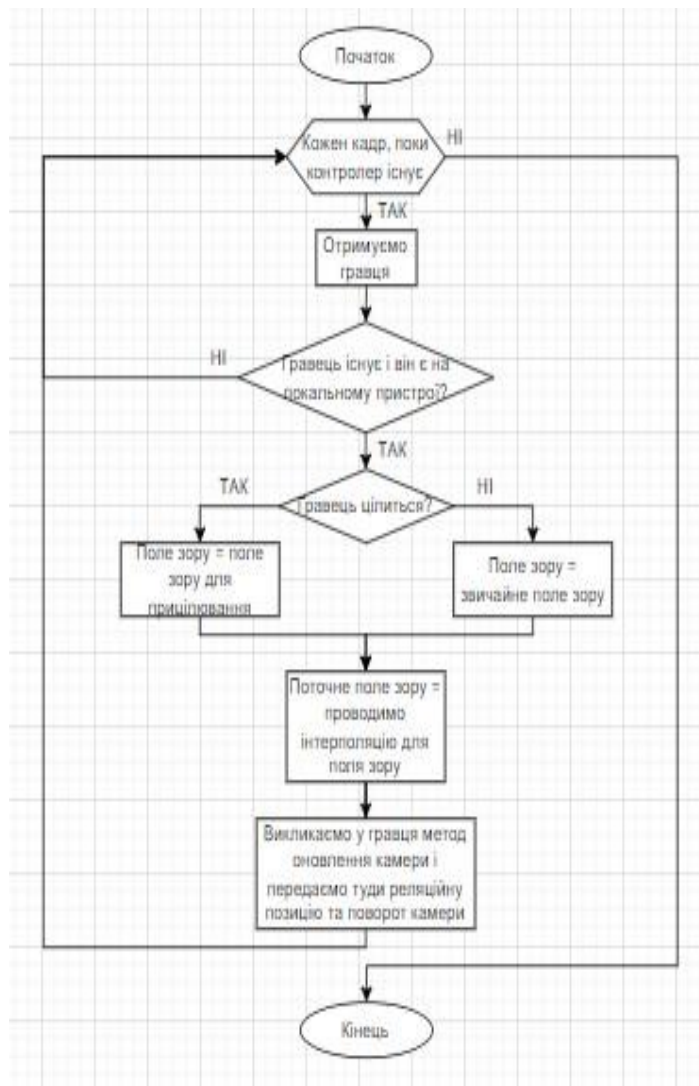


Рисунок 2.2 – Схема алгоритму оновлення менеджера камери

У свою чергу для того, щоб правильно обробити реляційну позицію та поворот камери та повернути модель гравця, у класі гравця буде реалізовано алгоритм, що зображений на рисунку 2.3.

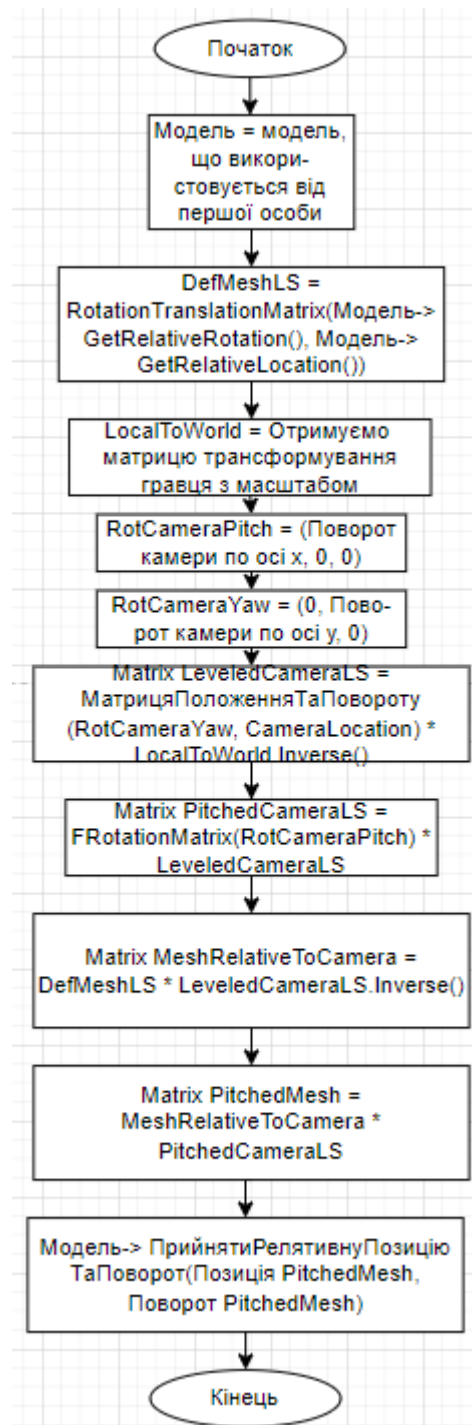


Рисунок 2.3 – Схема алгоритму оновлення позиції та повороту моделі гравця

2.4 Алгоритм принципу поведінки ШІ

У Unreal Engine є вбудований зручний редактор для створення та налаштування роботи логіки ШІ. Також, звісно ж, він містить також внутрішні об'єкти, що суттєво допомагають процесу розробки штучного інтелекту. Насамперед, для того, щоб зрозуміти, який саме необхідно розробити алгоритм, потрібно визначити, які можливості повинен мати та які дії повинен виконувати бот, що керується ШІ. Після аналізу загальних механік, було визначено, що бот матиме наступні можливості:

- аналізувати свій стан здоров'я та, при необхідності, шукати та йти до бонусу здоров'я, однак лише у випадку, якщо до цього бонусу є можливість дійти;
- аналізувати кількість боєприпасів у поточній зброї і, при необхідності, йти до бонусу патронів, однак лише у випадку, якщо до цього бонусу є можливість дійти;
- розрізняти об'єкти, що знаходиться перед ним в полі зору (супротивник чи гравець тієї ж команди);
- якщо був знайдений супротивник, то бот повинен проаналізувати поточну ситуацію (кількість здоров'я та кількість боєзапасу, що доступний боту) та вирішити чи скорочувати дистанцію з ворогом, чи краще тримати дистанцію;
- патрулювати місцевість.

Проаналізувавши всі можливості ШІ було розроблено алгоритм поведінки ШІ, що зображений на рисунку 2.4.

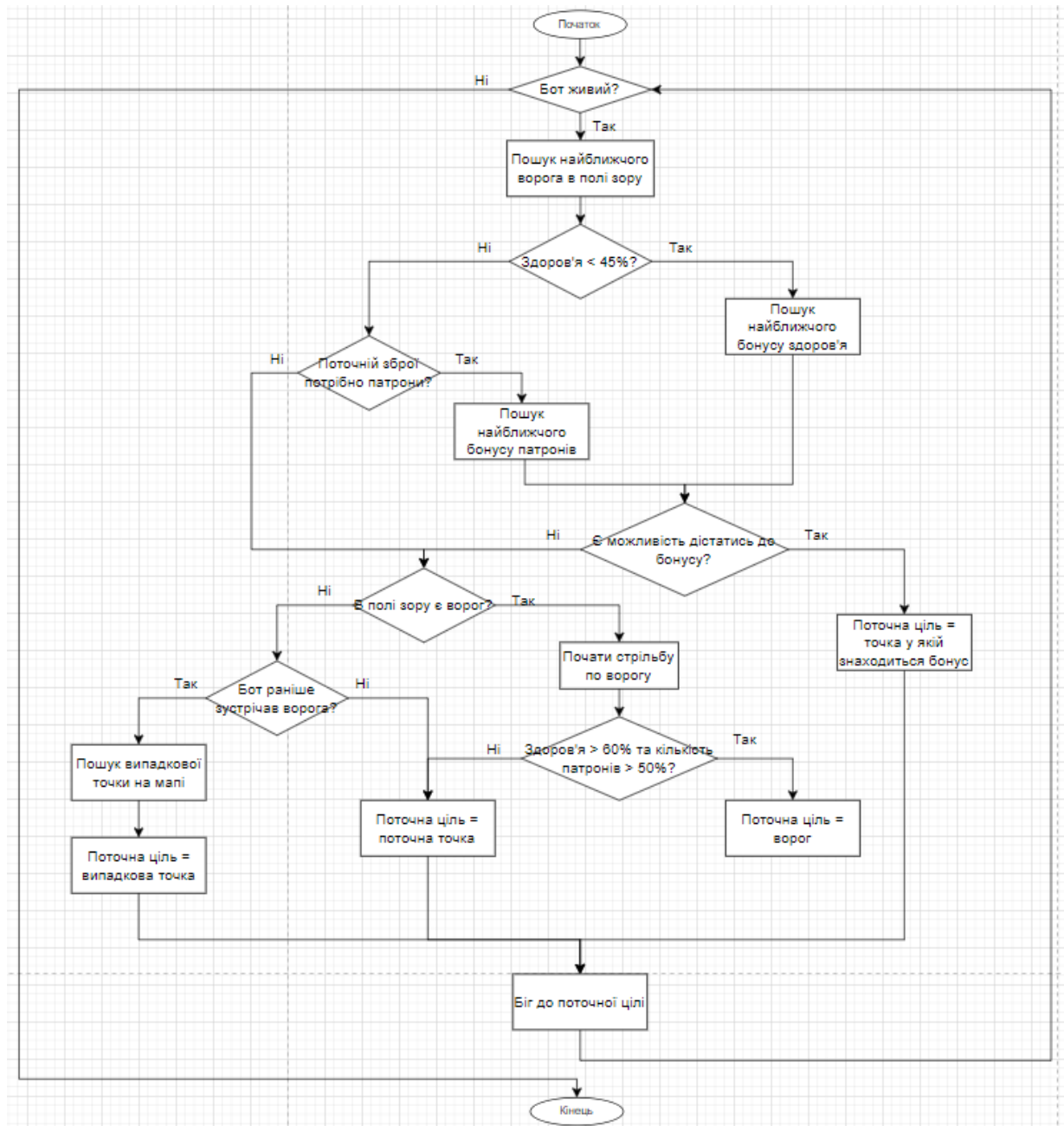


Рисунок 2.4 – Схема алгоритму принципу поведінки ШІ.

Також необхідно розробити алгоритм, що буде відповідати за аналіз об'єктів, що знаходяться перед ботом, який керується ШІ. Для цього, звісно ж можна використати спеціальний компонент зору для штучного інтелекту, однак це буде не зовсім виправданим для такої тривіальної задачі. Натомість було вирішено, що достатнім буде, статично записати відстань, на яку бачить бот та,

при початку сесії, динамічно записувати тип супротивника до інформації, яку використовує бот. Далі достатнім буде запускати промінь на необхідну відстань та встановлювати результат. Принцип такого алгоритму продемонстровано на рисунку 2.5.



Рисунок 2.5 – Схема алгоритму аналізу об'єкту, що знаходиться перед ботом.

2.5 Алгоритми обробки стрільби зі зброї та використання патрону

Під час процесу розробки алгоритмів для гри гри, потрібно також враховувати декілька факторів при використанні патронів: чи у зброї нескінченна кількість патронів, чи у зброї нескінченна кількість обойм, чи персонаж, що вистрілив з зброї гравець чи бот, що керується штучним інтелектом та який тип патронів використовується. Для цього був розроблений наступний алгоритм, що зображений на рисунку 2.6.

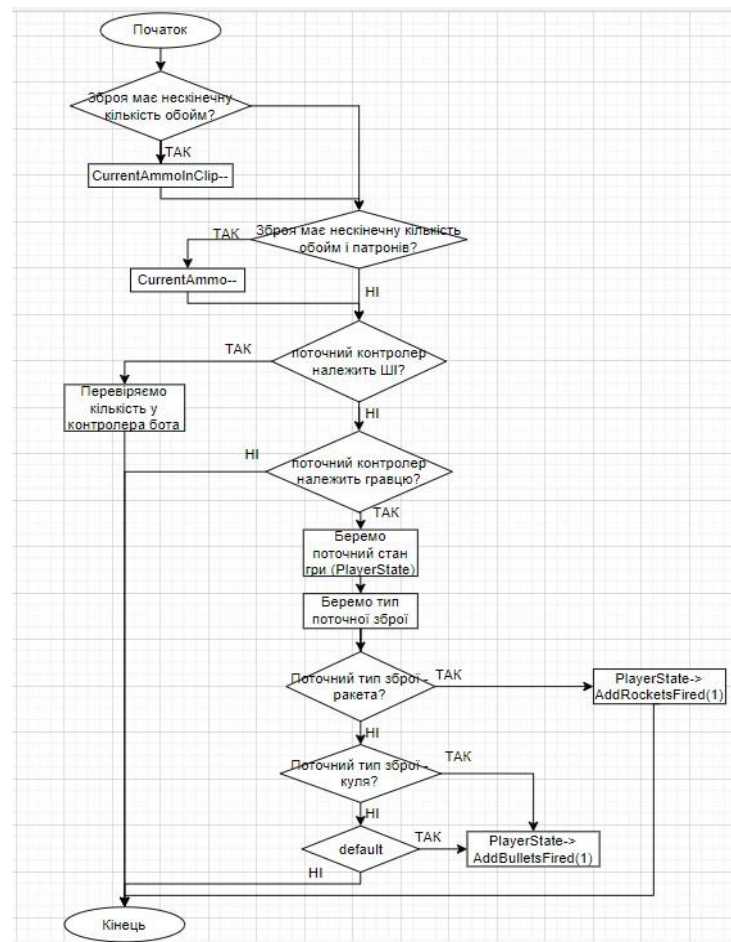


Рисунок 2.6 – Схема алгоритму аналізу об'єкту, що знаходиться перед ботом.

2.6 Алгоритм обробки смерті персонажа

Так як гра повинна бути не у автономному, а в мережевому режимі, то, у такому випадку, необхідно прописати логіку смерті у персонажа, в залежності від того, де викликається ця подія. Для цього був створений наступний алгоритм, що зображений на рисунку 2.8.

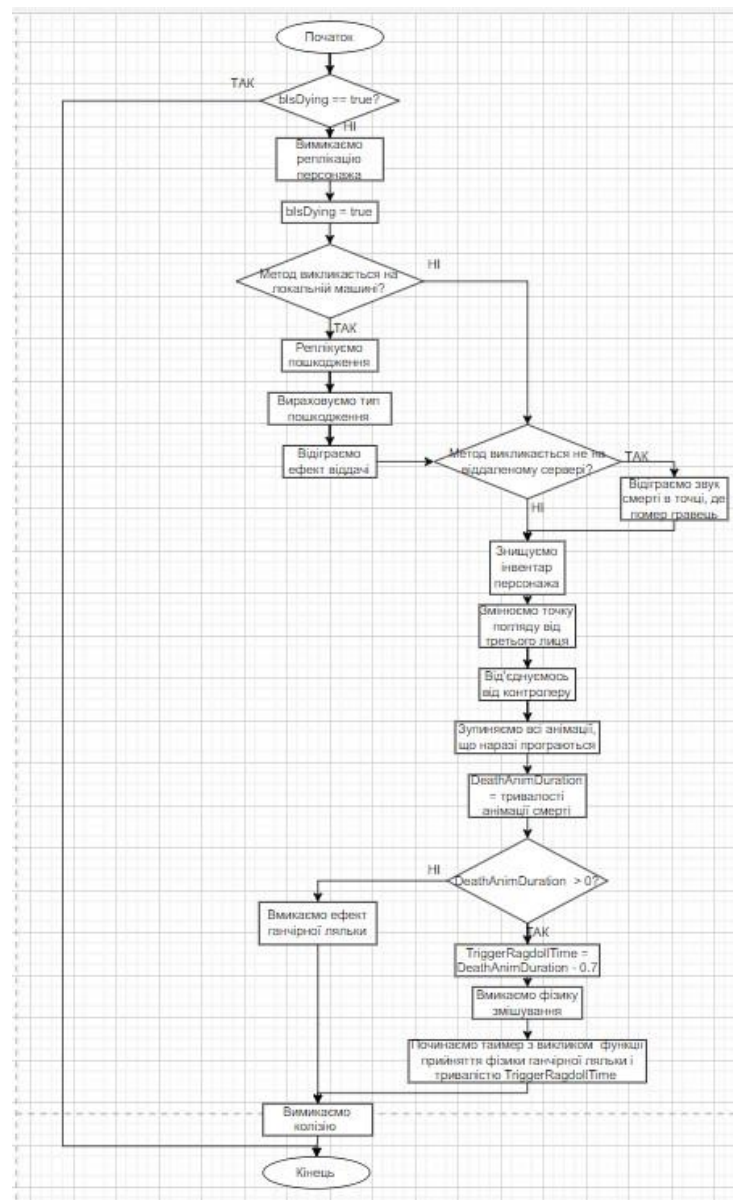


Рисунок 2.8 – Схема алгоритму реагування на смерть

Висновки до розділу 2

У даному розділі було описано постановку задачі та вимоги до розроблюваного продукту. Було запропоновано розроблені власні алгоритми, для подальших їх реалізацій, для таких процесів/дій, як:

- роботи анімації переміщення
- обчислення повороту камери
- принципу поведінки ШІ
- обробки стрільби зі зброї та використання патрону
- обробки смерті персонажа.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ГРИ

В даному розділі було створено архітектуру та програмні модулі для гри.

3.1 Архітектура гри

Під час того, як гра розроблялася, було прийнято рішення розділити програмне забезпечення на дві бібліотеки:

- ігрові механіки;
- користувацький інтерфейс.

Зі свого боку, кожна з бібліотек містить в собі модулі, що відповідають за певні складові гри.

Так, бібліотека з ігровими механіками містить модулі, які відповідають за стабільну роботу серверу, симуляції на сервері усіх ігрових об'єктів, взаємодію між такими об'єктами та гравцем на сервері, передачу активацій подій і інших елементів ігрового процесу.

У свою чергу, бібліотека з користувацьким інтерфейсом має модулі, які відповідають за коректне графічне відображення меню, анімацій, текстур, візуальних ефектів, статистики гравців, чату та переходу між рівнями.

Так, наприклад, у користувацькому інтерфейсі знаходиться модуль «Меню», що містить наступні елементи:

- налаштування матчу;
- приєднання до матчу;
- топ гравців;
- налаштування гри;
- внутрішньоігровий магазин;
- вихід з гри.

Все це відображає схема структури програмного забезпечення, що зображена на рисунку 3.1.

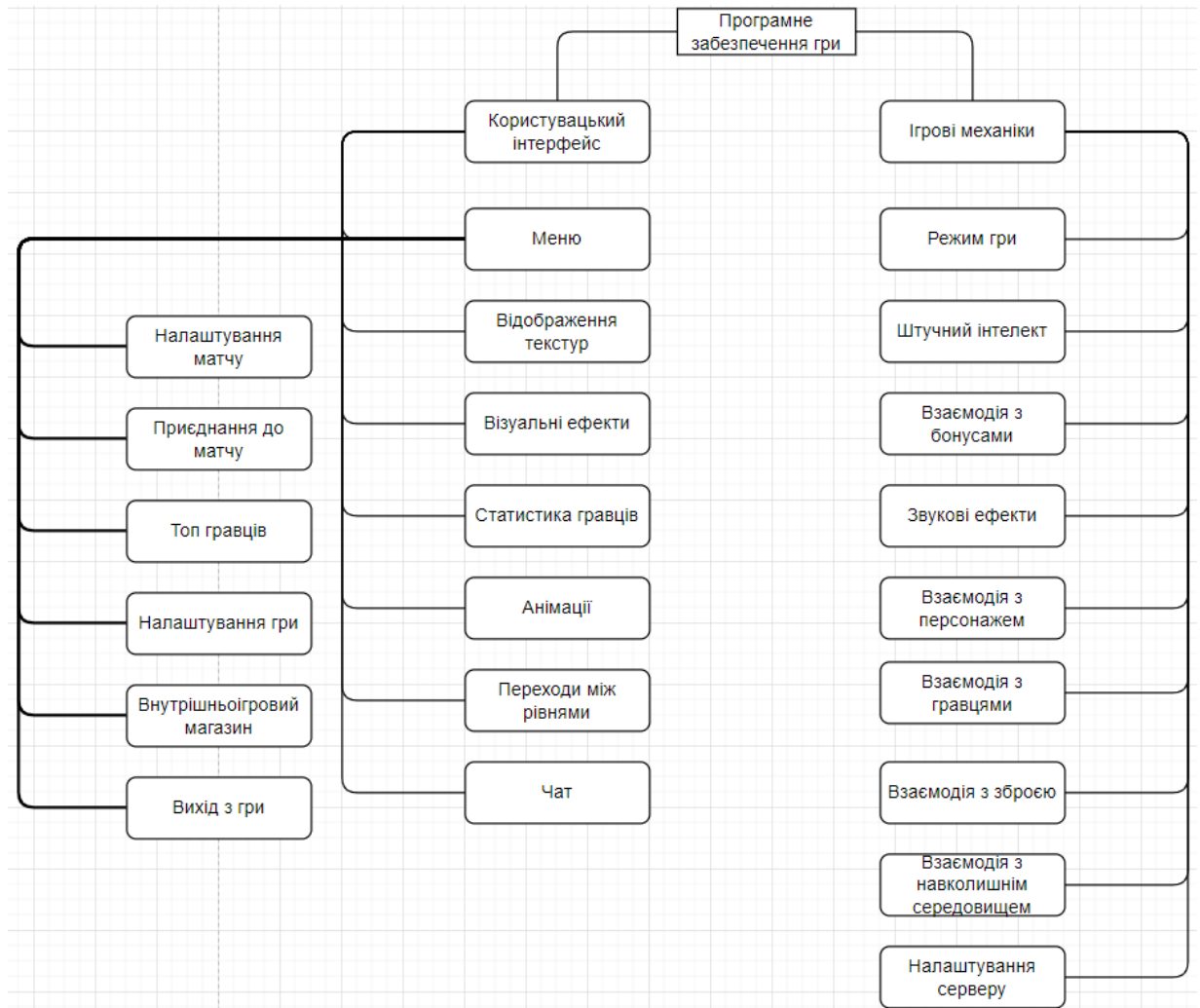


Рисунок 3.1 – Схема структури програмного забезпечення.

За таких обставин, дуже важливою необхідністю є продумання взаємозв'язку цих модулів, іншими словами, яким чином модулі бібліотеки взаємодіють між собою у грі. Для цього була розроблена схема взаємодії модулів, що зображена на рисунку 3.2.

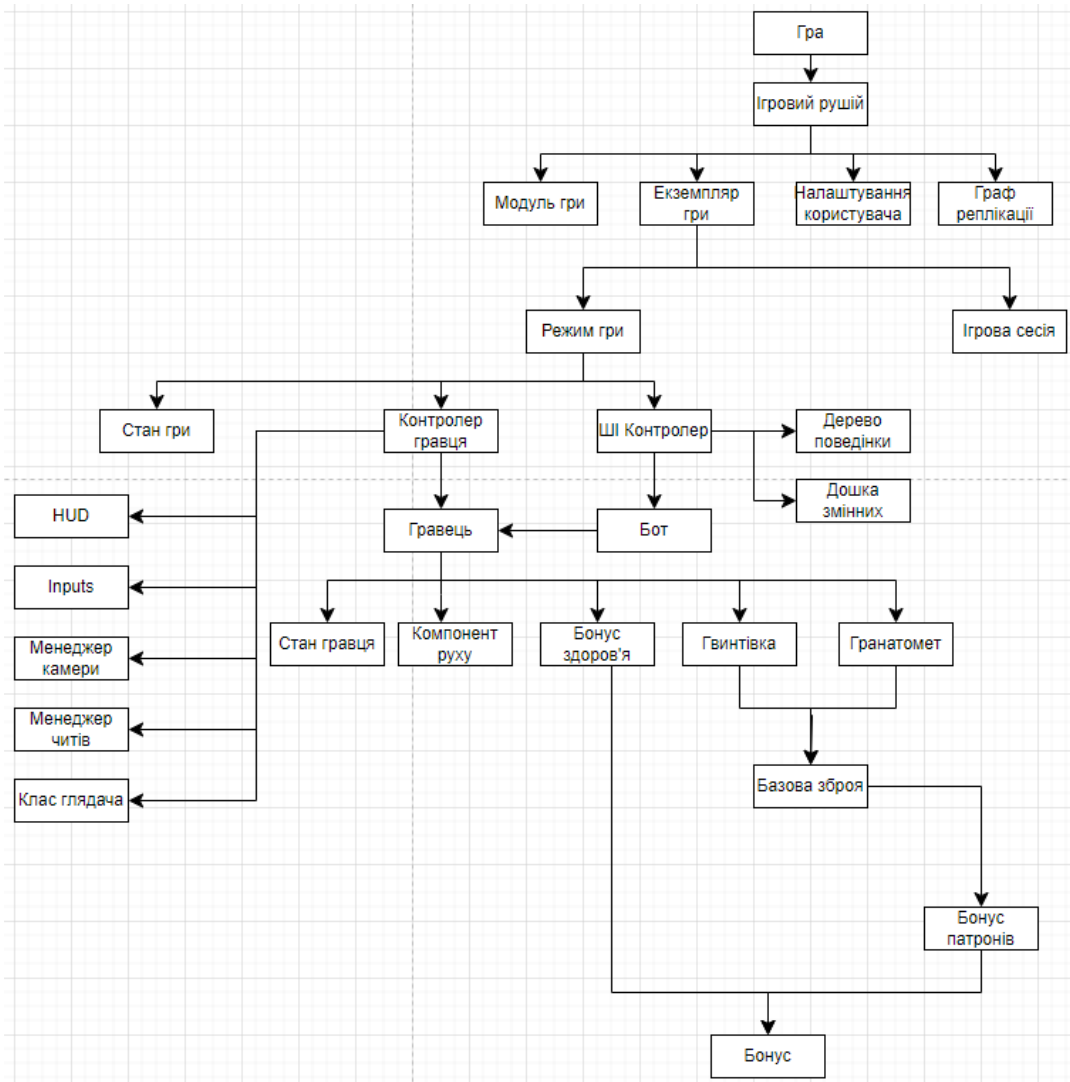


Рисунок 3.2 – Схема взаємодії програмних модулів гри

3.2 Розробка програмних модулів

3.2.1 Імпортування необхідних для розробки елементів

На поточний час, інді-розробник дуже рідко створює всі елементи гри самотужки. Часто, розробник користується уже готовими елементами, які він може знайти або на різних сайтах, або ж у офіційному маркеті ігрового рушія.

Отож, для гри було взято базові анімації, звукові та візуальні ефекти, готові меші об'єктів та скелети персонажів у безкоштовному проєкті від компанії Unreal Engine.

Для звуків було створено та налаштовано класи звуків[9], які залежать від головного класу. Завдяки цьому можна буде відключати всі, що буде корисним для налаштувань гравця. Повну ієрархію продемонстровано на рисунку 3.3.

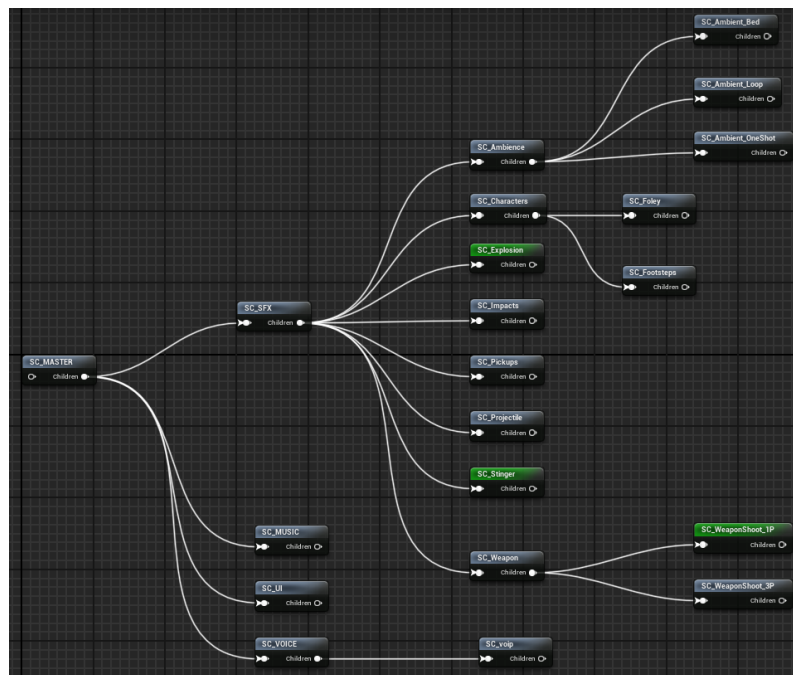


Рисунок 3.3 – Ієрархія класів звуків

Також, було створено затухання звуків при віддалені від джерела звуку для реалістичності[10]. Після цього, було створено звукові ефекти для рівнів та подій, що відбуваються у грі, від яких очікується реагування у вигляді відтворення звуку.

Для візуальних ефектів було створено класи, що відповідають за реалізацію відтворення впливу на поверхню у яку влучили різних візуальних та аудіо ефектів

3.2.2 Створення рівнів

У загальному випадку, у рівень додаються всі елементи відразу (меші навколишнього середовища, об'єкти звуку, колізія мешів і т.п.). Якщо створюється невеликий рівень, то такий підхід є більш, ніж виправданим. Але, у випадку, якщо рівні створюються для онлайн гри, то краще використати інший підхід для оптимізації використання ресурсів. Тоді, зазвичай, використовують, так звані, підрівні. Тобто спочатку будуть створюватись підрівні окремо для: звуків, мешів, колізії мешів, освітлення, елементів з якими може взаємодіяти гравець та елементи навколишнього середовища. Потім створюється загальний рівень, який об'єднує ці всі підрівні[11].

З використанням підходу підрівнів було створено 2 рівня та без використання один рівень (рівень для головного меню, що містить лише плитку з картинкою головного меню).

Насамкінець, для цих двох карт створено різну гравітацію, для того, щоб геймплей на кожному рівні відрізнявся.

3.2.3 Створення основної серверної логіки

Для того, щоб була можливість створювати та приєднуватися до сесії, необхідно створити власний клас екземпляру гри, де знаходиться вся логіка на найвищому рівні.

Екземпляр гри – це об'єкт менеджера високого рівня для екземпляра запущеної гри. Виникає під час створення гри та не знищується, допоки екземпляр гри не буде закрито. Якщо гра запущена як кінцевий продукт, то він буде один для всіх. Запуск у редакторі рушія згенерує один екземпляр для кожного екземпляру редактора.[12]

У створеному класові екземпляру гри знаходяться функції, що відповідають за: підключення та створення ігрової сесії, пошук активних ігрових сесій, перехід між рівнями, отримання або створення мережевого безпекового токена, обробку отримання запиту на приєднання до гри, переходу між сесіями гри та рівнів, відключення гравця від сесії, встановлення режиму гри, встановлення параметрів для гри. Після цього цей клас був встановлений у UE 4, як екземпляр класу за замовчуванням. Для цього необхідно було перейти в редактор рушія, де відкрити Налаштування Проєкту. В Налаштуваннях необхідно перейти в категорію Рівні & Режими, де у поле Клас Екземпляру гри вибрати створений власний клас екземпляру гри.

Він використовує так звану ігрову сесію, власний клас якої також був створений.

Ігрова сесія діє як обгортка для гри навколо інтерфейсу сеансу. Ігровий код викликає його, коли йому потрібно взаємодіяти з інтерфейсом сеансу. Ігровий сеанс існує лише на сервері, поки працює онлайн-гра.[13] Саме тут була створена логіка створення та приєднання до гри, а також налаштування параметрів при створенні та пошуку ігор.

Насамкінець, для оптимізації, був створений власний граф реплікації.

Граф реплікації - це система для мережевої реплікації в багатокористувацьких іграх, розроблена для гарного масштабування з великою кількістю гравців і реплікованих акторів. Стандартна стратегія мережевої реплікації, яка полягає у вимаганні від кожного реплікованого актора визначати, чи повинен він надсилати оновлення кожному підключеному клієнту, у подібних випадках працює погано та призведе до великої навантаженості на ЦП сервера. Граф реплікації позбавляє акторів необхідності оцінювати кожного підключеного клієнта окремо, вирішуючи проблему продуктивності ЦП без шкоди для роботи клієнта. [14]

У власному класі графу реплікації не використовується звичайний параметр `IsNetRelevantFor`. Натомість клас містить власні вузли реплікації. Ці вузли відповідають за створення списків об'єктів для реплікації для кожного, хто під'єднаний. Більшість об'єктів з цих списків будуть постійними між кадрами. Це дає змогу спільно/повторно використовувати більшу частину роботи зі збору даних (тобто виявляти, які об'єкти у грі слід розглядати для реплікації). Ці вузли можуть бути глобальними (тобто ці вузли використовують всі), специфічними для з'єднань (кожен, хто під'єднаний до гри має власний вузол) або спільними (наприклад, при гри у команді усі, хто під'єднаний до гри мають спільне користування вузлом).

3.2.4 Створення та налаштування зброї

На початку, було створено та налаштовано сокети (точки кріплення, які можна розмістити на меші, після налаштування якого можна під'єднувати об'єкти до створеного сокету)[15] для мешів зброї, що необхідно для створення ефекту вистрілу з дула та взяття позиції сокету, як початкову точку для симуляції стрільби.

Спочатку було створено структуру, що містить дані про зброю: чи зброя має нескінченну кількість патронів, чи зброя має нескінченну кількість обойм, максимальну кількість патронів, скільки у одній обоймі патронів, початкову кількість обойм та час між вистрілами.

Також була створена структура, що містить дані про анімацію, що буде відтворюватись від першого та від третього лиць.

Далі був створений базовий клас зброї, який реалізовує логіку за паттерном стану. Він має 4 стани: зброя у стані спокою, зброя стріляє, зброя перезаряджається та зброя екіпується. Також він має структуру, що відповідає за тип патрону (куля чи ракета). Тут реалізовані наступні функції зброї: екіпування зброї, сховання зброї, додавання та вилучення зброї до інвентаря, процесу

стрільби та перезарядки зброї. Також клас містить інформацію про: ікони зброї для HUDa, поточну кількість патронів та обойм, використання прицілу, аудіокомпоненти для стрільби з зброї та її перезарядки, візуальні компоненти, що будуть відтворюватись при стрільбі, клас, що буде трясти екран для симуляції віддачі.

Від даного класу було відспадковано два інших класи: гвинтівка та гранатомету.

У клас гвинтівки було додано наступну інформацію: кучність стрільби, кучність стрільби при прицілюванні, дистанція пострілу зі зброї, тип пошкодження. Також перевизначено процес стрільби зі зброї, який тепер реалізований на основі випускання променю від дула у напрямку, в якому знаходиться приціл на максимальну відстань з поправкою на кучність зброї).

Перед тим як було реалізовано клас гранатомету, було створено клас ракети, яку запускатиме гранатомет. Клас ракети містить наступну інформацію: початку швидкість, подію, що відбудеться при влучанні в якийсь об'єкт, компоненти руху, колізії та візуальних ефектів, інформацію про зброю. Клас містить наступні функції: взриву, знищення та руху.

У класі гранатомету знаходиться наступна інформація: снаряд, яким стріляє гранатомет, тривалість польоту снаряду до взриву, кількість та радіус пошкодження, яке надає снаряд та типу пошкодження. Також клас перевизначає процес стрільби, який реалізований на основі створення реального об'єкта та передачі йому початкової швидкості.

3.2.5 Створення гравця

Так як це клас гравця, то він відповідає за багато функцій та містить багато інформації. Умовно розділим ці дані на чотири частини:

- 1) здоров'я;
- 2) управління зброєю;

- 3) переміщення;
- 4) анімації;

Для здоров'я гравець містить наступні функції: отримання пошкодження, опрацювання смерті, коли гравця вбив інший гравець та коли гравець скоїв самогубство, визначення ким був убитий гравець. Також у цій частині знаходяться наступна інформація: максимальна кількість здоров'я, поточне значення здоров'я, значення здоров'я, що вважається низьким. Для кращого відображення останнього параметру реалізовано також звук та візуальний ефекти.

Для управління зброєю гравець містить наступні функції: стрільби, перезарядки, прицілювання та зміни з наявної зброї, а у якості інформації: масив зі зброї та наявну в руках зброю. Варто зазначити, що ці дані викликаються і на сервері, і на стороні клієнта, тобто у разі пострілу відбувається його симуляція на сервері. Далі сервер сповіщає всіх про вистріл і пристрої клієнтів відтворюють необхідні дії.

Після цього було розроблено клас таким чином, щоб він виконував свої дії, відповідно до того, як гравець натискає клавіші. Тому було реалізовано: ходьбу персонажа, за допомогою клавіш W,A,S,D та клавіш стрілок, бігу персонажа, при утриманні клавіші Shift , стрибок персонажа, при натиску клавіші Space, управління

камерою ,відповідно до проведеної миші (при цьому, часткову роботу виконує менеджер камери, стрільбу, за допомогою ЛКМ (лівої кнопки миші), прицілювання, за допомогою ПКМ (правої кнопки миші), відкритті меню, за допомогою клавіші Escape, відкритті статистики гри, при натиску на клавішу Tab, зміну зброї, за допомогою клавіш Q та колеса миші, перезарядку зброї, за допомогою клавіші R, відображення текстового та голосового чату за допомогою клавіш C та K, відповідно.

Для налаштування анімацій для мешу скелету, що використовується як той, який бачить тільки гравець було використано анімований граф, що зображений на рисунку 4.5.

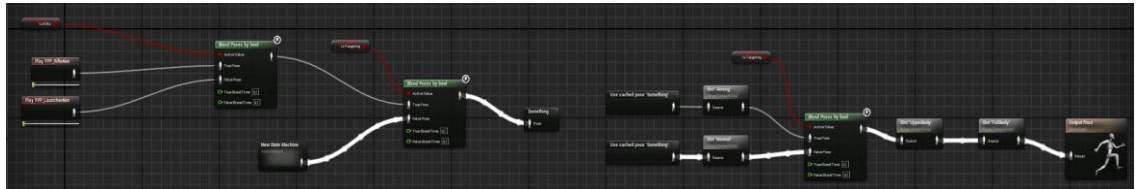


Рисунок 4.5. Анімований граф для мешу «від першого лиця»

Граф у свою чергу використовує машину станів, що реалізує алгоритм продемонстрований на рисунку 2.1. Машина стану зображена на рисунку 4.6.

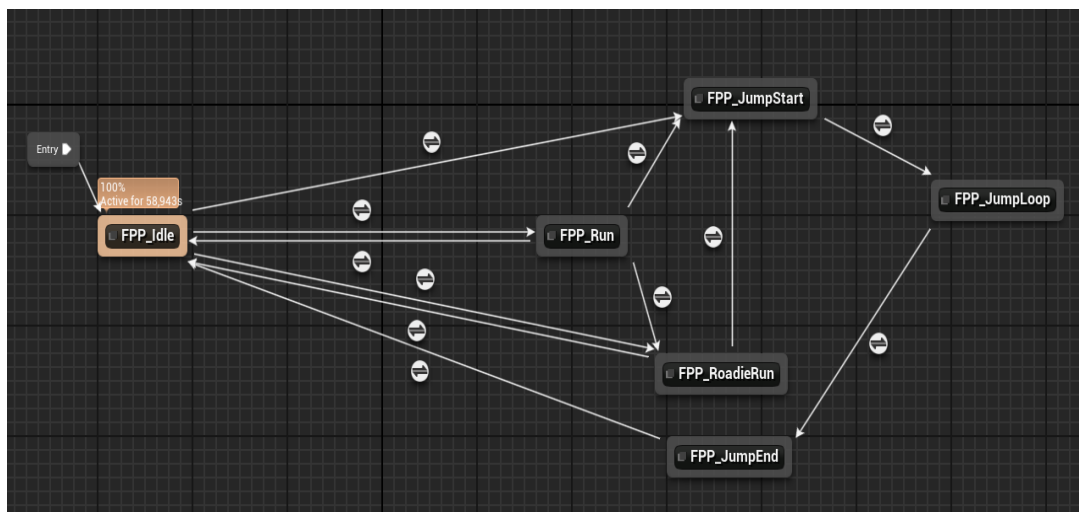


Рисунок 4.6. Машина станів для мешу «від першого лиця»

Після цього було створено анімований граф для мешу, який бачать всі інші гравці, що зображений на рисунку 4.7.

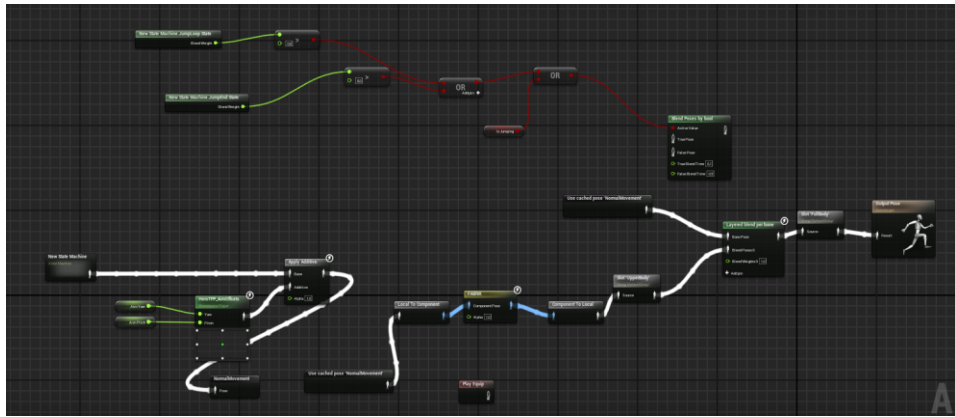


Рисунок 4.7. Анімований граф для мешу «від третього лиця»

Даний граф також використовує машину станів, що реалізовує алгоритм продемонстрований на рисунку 2.1. Машина стану зображена на рисунку 4.6.

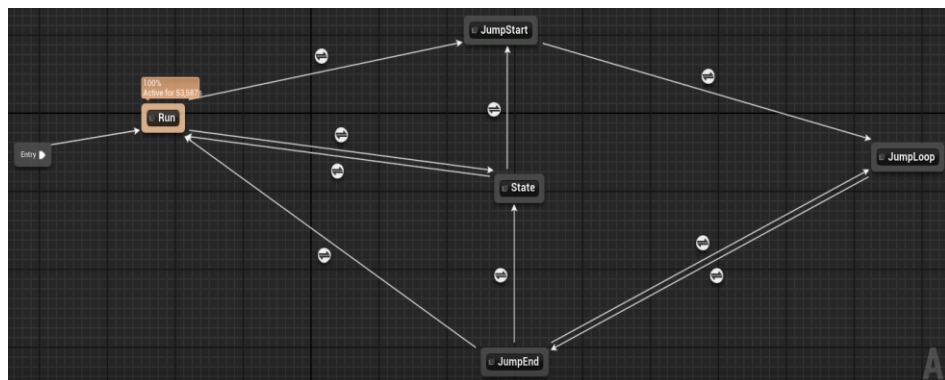


Рисунок 4.8. Машина стану для мешу «від третього лиця»

Хоча і здається, що ці машини станів ідентичні, однак насправді так здається лише тому, що стани названі однаково. Насправді, кожен стан для мешу від першого та від третього лиця відрізняється.

Також, для більшої реалістичності, значення пошкодження змінюється в залежності від яку область тіла було влучено. Для цього необхідно було

налаштувати ці дані в меші скелету персонажа «від третього лица». Результат такого налаштування зображений на рисунку 4.7.

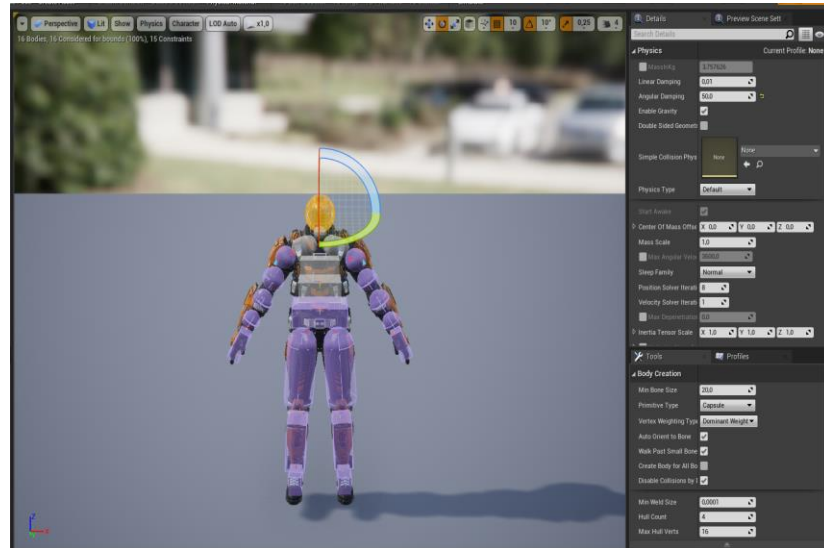


Рисунок 4.7. Налаштування мешу «від третього лица на різне значення пошкодження»

3.2.6 Створення бонусів

Так як загальна логіка бонусів не відрізняється, то спочатку був створений базовий клас бонусу, у якого є компонент колізії, час, за який буде поновлюватися бонус, звук взятого та знову активного бонусу, візуальний ефект взятого та знову активного бонусу та інформацію про того, хто підняв бонус.

Після чого було створено бонус здоров'я, що містить лише значення, на яке піднімається здоров'я того, хто взяв бонус, власне, взяття бонусу та інформацію про те, чи може бонус бути, впринципі, взятим.

Також, на основі базового бонусу, було створено бонус патронів, що містить функції вирахування, хто може взяти бонус, для якої зброї цей бонус назначений та кількість патронів, яку додає бонус.

3.2.7 Створення ШІ

Як вже було вказано вище, редактор UE 4 надає можливість налаштовувати ШІ візуально. Для цього, на початку, було створено дошку значень, у якій зберігаються змінні для подальшого використання[17].

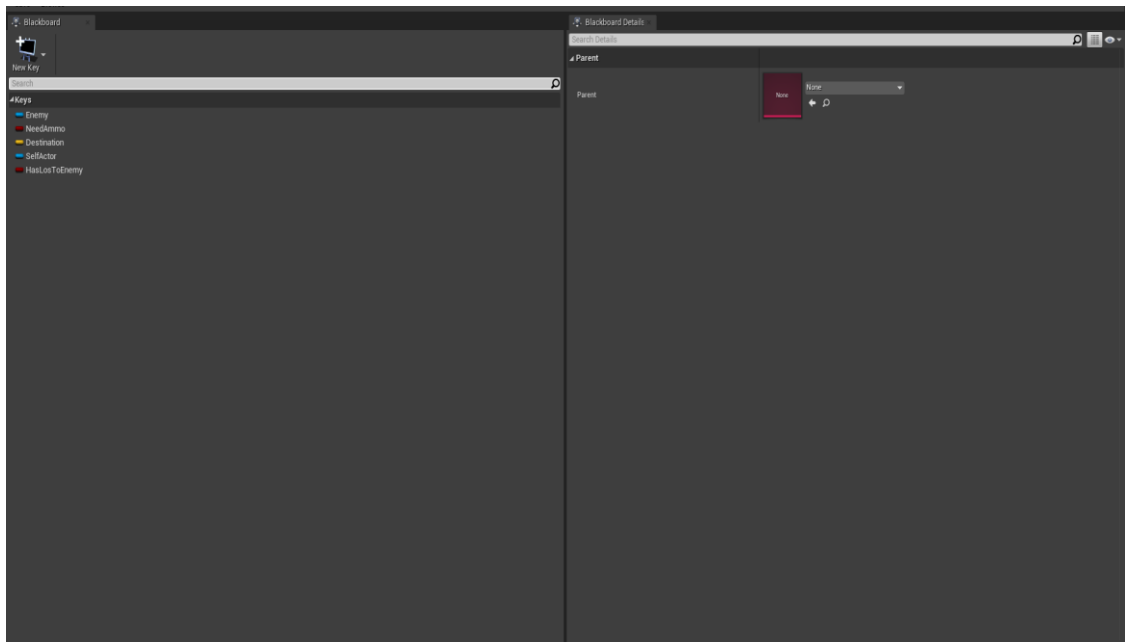


Рисунок 3.2 – Змінні у дошці змінних

Після цього було створено клас бота, що успадковується від класу гравця та відрізняється тільки тим, що обчислює поворот іншим чином.

Також було створено контролер для штучного інтелекту. Цей контролер містить в собі інформацію про дошку змінних, дерево поведінки та має функції респауну, перевірки здоров'я та патронів, перевірку на закінчення гри та на пошук найближчого ворога..

Після цього було створено логіку двох Завдань. Завдання – це листові вузли дерева поведінки, які виконують фактичні дії. Таким чином, перше

завдання відповідає за пошук найближчого ворога, а друге – за пошук найближчого бонуса.

Наступними корисними для створення AI елементами було створено декоратор. Декоратори – це умови в дереві поведінки, що приєднуються до композитного вузла або вузла завдання і визначає, чи може бути виконана гілка в дереві або навіть один вузол. Створений декоратор відповідає за перевірку чи знаходиться якийсь об’єкт перед ботом, що керується ШІ.

Після цього було створено та налаштовано Дерево Поведінки для бота. Перед тим, як відобразити реалізацію цього об’єкту, необхідно пояснити деякі базові концепти. Обирач – це елемент, що проходить між залежними від нього вузлами зліва направо в пошуках успішного вузла. Якщо перший вузол не спрацьовує, то він обирає наступний. У разі успіху вузол завершується і ми повертаємося вгору по дереву.

Послідовник – це елемент, що проходить між залежними від нього вузлами зліва направо, поки вузол не перестане спрацьовувати. Якщо вузол виходить з ладу, то повертаємося вгору по дереву.[18]

Тоді Дерево поведінки було створене, згідно з алгоритмом, що зображений на рисунку 2.4, таким чином, як це вказано на рисунку 3.4.

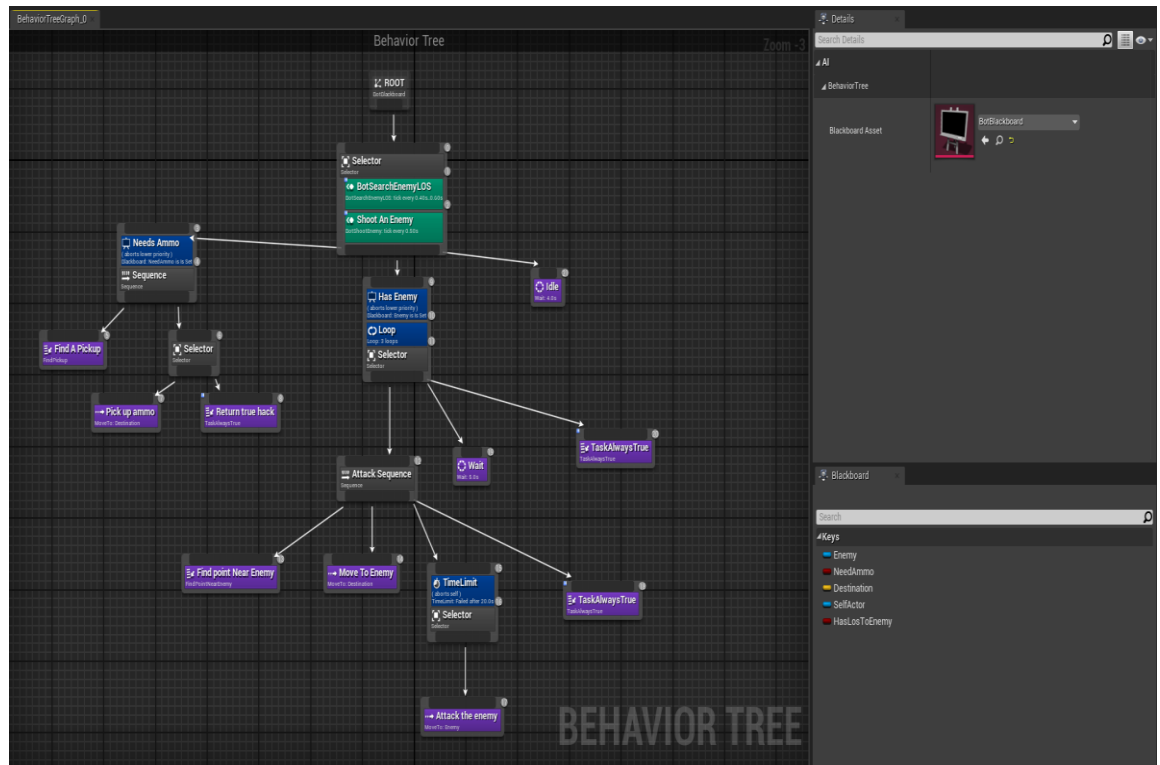


Рисунок 3.4 – Дерево Поведінки для ІІІ

3.2.8 Створення Режимів гри

Так як в майбутньому передабачається, що режимами гри будуть лише додаватись, то було створено базовий режим гри, від якого будуть успадковуватися всі інші.

Базовий режим гри зберігає в собі наступні функції: можливість додавати ботів до гри, фіксацію початку та закінчення матчу, перезапуску гравця, розташування відродженого або під'єданого гравця в найкращу точку, з огляду на ситуацію, що відбувається під час матчу, модифікування значення пошкодження, фактичне знищення одного гравця іншим, дозвіл на використання читів, інформацію про те чи може на даний час один гравець нанести пошкодження іншому, перезапуску матчу, виявлення переможця та закінчення

матчу. Також базовий режим гри містить наступну інформацію: масив класів для гравця та бота, час гри, матчу та між матчами, лічильник знешкоджень, модифікатор для того, щоб наносилось зменшене значення пошкодження, при завданні шкоди самому собі, максимальну кількість ботів та інформацію про бонуси, що є на рівні.

Від цього класу успадковуються два інших: кожен сам за себе та командний матч смерті. Перший клас змінює логіку виявлення переможця таким чином, що переможцем може бути лише один гравець з найбільшою кількістю знешкоджень.

Другий клас змінює логіку функцій базового режиму гри: перевірку чи може бути нанесене пошкодження, вибору команди, виявлення переможця, таким чином, що виграє команда з найбільшою кількістю знешкоджень, перевірку чи може бути в певній точці створений гравець та ініціалізацію ботів. Також цей клас містить у собі інформацію про кількість команд та ідентифікатор команди, що перемогла.

Для того, щоб розподіляти по командам гравців, що на даний час мають один і той же меш скелету, було вирішено змінити його логіку відтворення матеріалу.

Для цього було додано до звичайної кількості налаштувань матеріалу свої власні, якізначаються перед початком гри завдяки відкритому ключу, який називається «Індекс кольору команди». Це дозволяє логічно пов'язати ідентифікатор у режимі гри та дані у матеріалові. Також, таке рішення є хорошим, так як, щоб додати нову команду буде достатнім додати кольори та підписати їх, як цього вимагає логіка обробки вибору команди.

3.2.9 Створення UI

Насамкінець, було створено наступні UI елементи: віджет чату, віджет статистики гравця в матчі, віджет відтворення попередніх матчів, віджет налаштування графіки, віджет внутрішнього магазину, віджет «Хост», віджет приєднання до матчу, віджет списку лідерів у грі, віджет чіт меню, віджет головного меню та клас HUD.

Ці всі віджети було вирішено написати в одному розділі, так як вони всі просто беруть вже зроблені раніше функції з Ігрової сесії або з екземпляру гри. Клас HUDa, в свою чергу, просто міняє активний віджет на той, що буде викликаний при натиску на кнопку.

Висновки до розділу 3

У даному розділі було описано структуру бібліотек гри, що містять в собі модулі та схему взаємодії цих модулів. Також було описано розробку елементів програмного забезпечення, а точніше: звуків, візуальних ефектів, рівнів, основної серверної логіки, зброї, персонажа, ШІ, бонусів, режимів гри та UI елементів.

4 ЗАПУСК ТА ТЕСТУВАННЯ РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

В даному розділі було відображено результат запуску програмного продукту та обрано і протестовано складових гри.

4.1 Запуск програмного забезпечення

Одразу після запуску комп'ютерної гри гравець бачить головне меню гри, що зображене на рисунку 4.1.

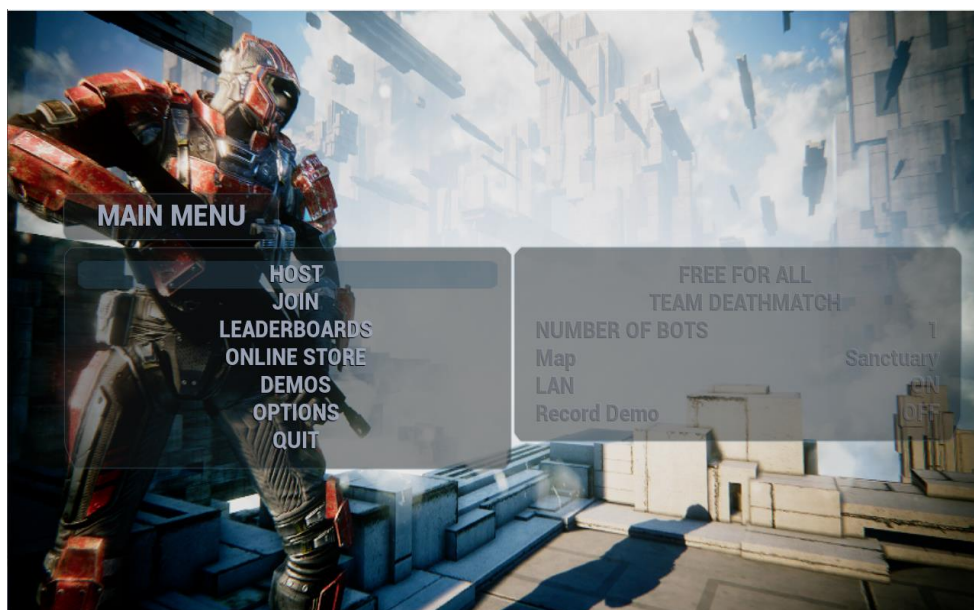


Рисунок 4.1 – Головне меню гри

Розглянемо кожен з елементів головного меню.

Елемент «Хост» необхідний для створення та налаштування нової онлайн сесії, де є можливість налаштування: кількості гравців, рівень, тип підключення, можливість записати демо та режим гри. Цей елемент зображено на рисунку 4.2.

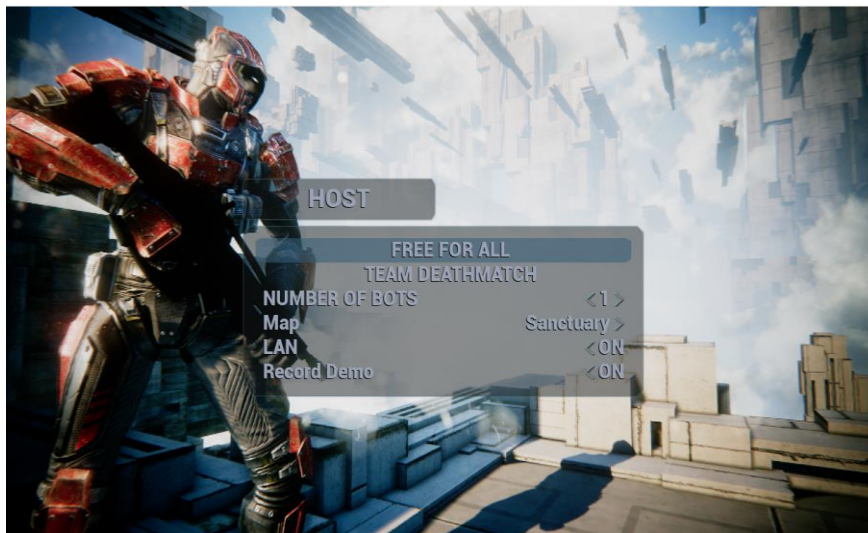


Рисунок 4.2 – Приклад налаштування сесії гри у елементі «Хост»

Елемент «Підключитись» відповідає за можливість підключитись до однієї з наявних сесій. У цьому елементі є можливість обрати одну з наявних сесій в одному з типів підключення. Результат пошуку зображено на рисунку 4.3.

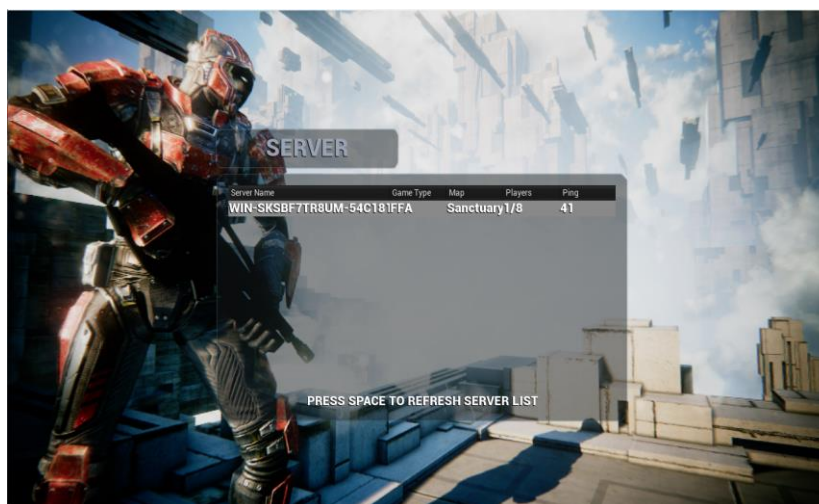


Рисунок 4.3 – Приклад результату пошуку у елементі «Підключитись»

У елементі «Дошка лідерів» знаходиться список найкращих гравців у грі. Приклад інформації, що може бути відображено у списку зображено на рисунку 4.4.



Рисунок 4.4 – Приклад лідерів у «Дошці лідерів»

У елементі «Онлайн магазин» відображаються необов'язкові для гри товари, що гравець може купити за справжню валюту. Приклади тестових товарів зображено на рисунку 4.4.

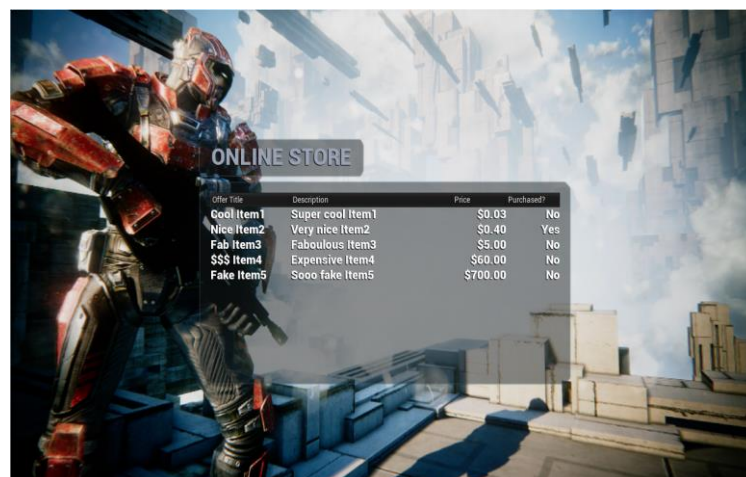


Рисунок 4.5 – Приклад товарів у «Онлайн магазині»

Елемент «Демо» дозволяє переглядати користувачам матчі, які були при створенні налаштовані як ті, де записується демо. Приклад перегляду матчу зображено на рисунку 4.4.



Рисунок 4.6 – Знімок екрану з матчу, що був записаний раніше та відтворюється в «Демо»

Насамкінець, елемент «Налаштування» містить можливість користувача налаштувати графіку та деякі інші налаштування у грі. Приклад такого налаштування зображено на рисунку 4.7.



Рисунок 4.7 – Приклад обраних налаштувань в «Налаштуваннях»

Перед початком матчу відображається кількість часу, що залишився до початку матчу та за цей час геймер може ознайомитись з грою. Приклад такого ігрового елемента зображено на рисунку 4.8.

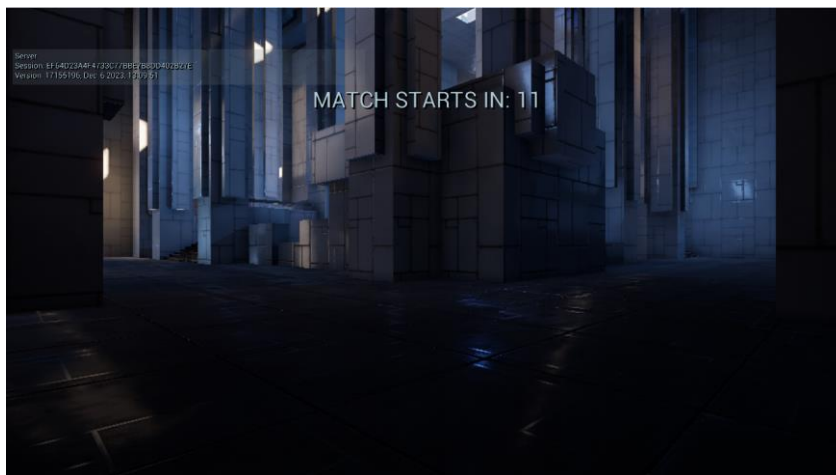


Рисунок 4.8 – Приклад гри перед початком матчу

Гравцю, що запускає сервер було вирішено дати можливість додатково налаштувати деякі елементи геймплею за допомогою чит менеджера, інтерфейс якого зображено на рисунку 4.9.

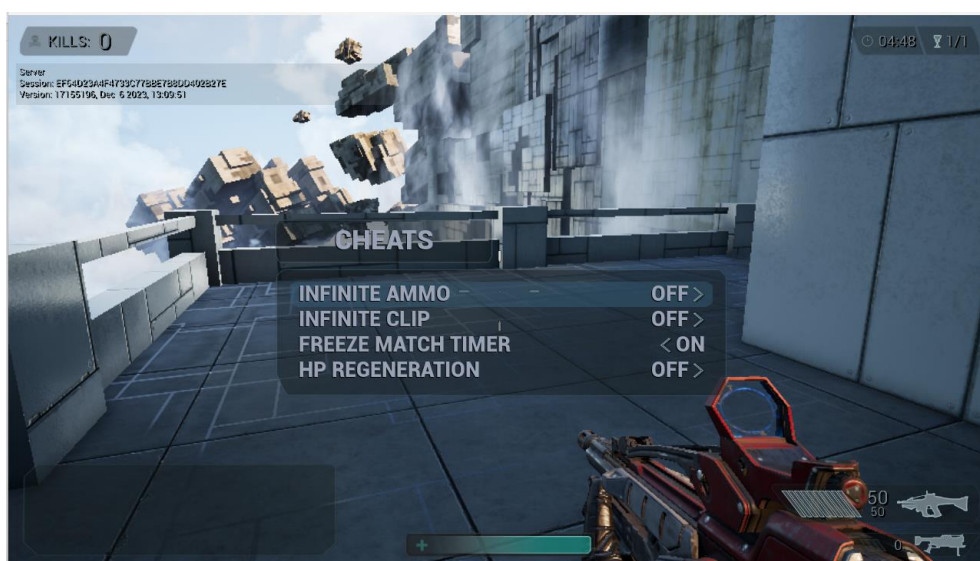


Рисунок 4.9 – меню читів у грі

Під час гри, гравець може переглядати статистику гри. Приклад статистики зображено на рисунку 4.10.

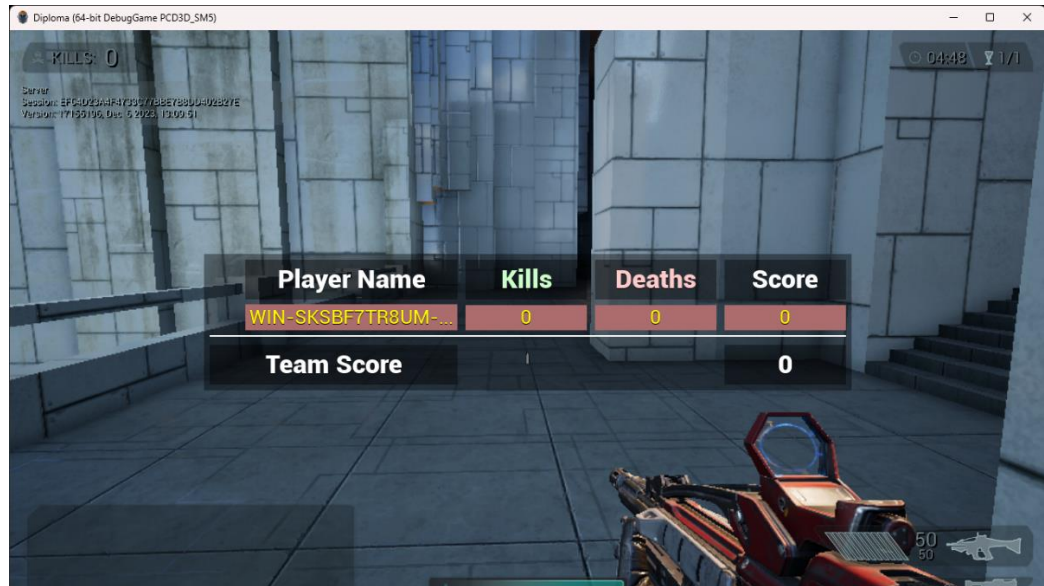


Рисунок 4.10 – приклад статистики гравця

Насамкінець, при смерті гравця програється анімація смерті та вмикається режим глядача. Приклад такої поведінки зображено на рисунку 4.11.



Рисунок 4.11 – Смерть персонажа у грі

4.2 Тестування програмного забезпечення

При тестуванні, було вирішено зробити тестування наступних складових гри: роботи гри при їх відтворенні на різних типах з'єднання, продуктивності гри та коректності роботи ключових механік гри.

4.2.1 Результати тестування онлайн складової гри

Так як гра написана у мультиплеєрному режимі, то необхідно було написати тести для перевірки коректності входу у гру, пошуку матчу, приєднання до матчу та оновлення статусу матчу (гра в процесі, призупинена чи закінчена). При тому ж, гру можна запускати з можливістю LAN-підключення та приєднання до сервера (при тому ж, можна приєднуватись до сервера, що прослуховується та до віддаленого серверу).

Для тестування було створено базовий контролер, який має в собі весь необхідний функціонал (входу у гру, пошук матчу, створення матчу, процесу гри, результати тестування онлайн складової гри).

Спочатку, було протестовано чи правильно гра працює при роботі на віддаленому сервері. У випадку, якщо гра не запуститься на протязі 300 секунд буде отримано наступну помилку у журналі помилок: «Failing boot test after 300 secs!». Результат запуску тесту зображено на рисунку 4. 12.

```
• Server logged in
• Play in editor total start time 8.812 seconds.
• Client logged in
• Play in editor total start time 10.192 seconds.
• Client logged in
• Play in editor total start time 10.49 seconds.
• Client logged in
• Play in editor total start time 10.942 seconds.
```

Рисунок 4.12 – Результат тесту при грі на віддаленому сервері

Як видно з логування, тест успішно пройдено.

Далі було протестовано гру на прослуховуваному сервері. При невдалому сценарії буде відображатись помилка «Failing boot test!». Результат запуску тесту зображено на рисунку 4.13.

```
• Server logged in
• Play in editor total start time 4.141 seconds.
• Client logged in
• Play in editor total start time 4.232 seconds.
• Client logged in
• Play in editor total start time 4.524 seconds.
```

Рисунок 4.13 – Результат тесту при грі на прослуховуваному сервері

Як видно з логування, тест успішно пройдено.

4.2.2 Результати тестування статистики продуктивності гри

Гравці, безсумнівно, з набагато більшим бажанням будуть грати у ігри, що працюють стабільно та є оптимізовані. Окрім того, чим краще гра оптимізована, тим більше ПК зможуть запусити гру і обробляти її стабільно, що збільшить потенційну кількість геймерів, що буде грати у розроблений програмний продукт. Переглянемо результат статистики нашої гри. Саме тому було вирішено витратити час на постійні мініоптимізації процесів у гру. Результат статистики гри відображено на рисунку 4.14.

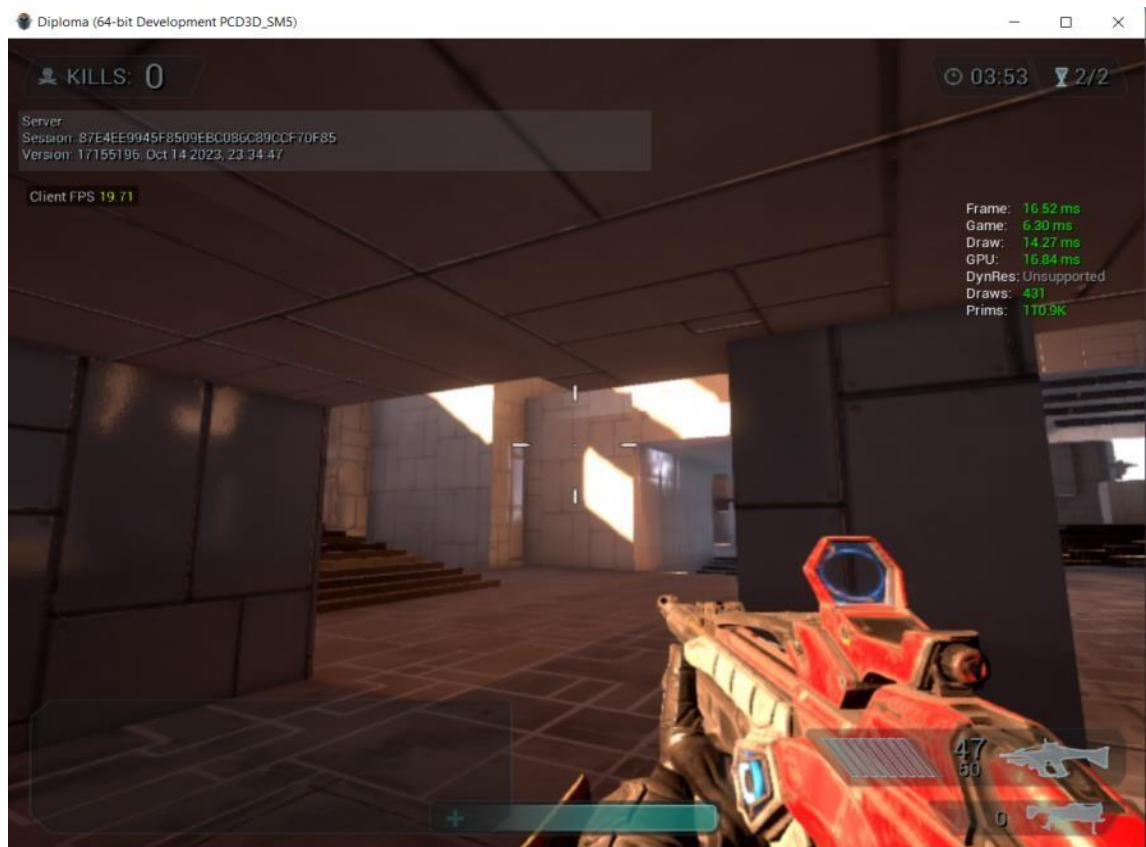


Рисунок 4.14 – Статистика продуктивності гри

Як видно з інформації, що показана на рисунку 4.14 , гра має хороші показники роботи і всі параметри позначені зеленим кольором, що означає, що було отримано хороший результат оптимізації у грі.

4.2.3 Результати тестування коректної роботи ігрових механік

Для того, щоб зрозуміти, які механіки потрібно проаналізувати, необхідно, перш за все, визначити у випадку несправності яких механік, продукт не буде працювати у нормальному режимі.

У поточному випадку це будуть наступні модулі: створення сесії та приєднання до сесії, віднімання здоров'я у гравця, віднімання боєприпасів та підняття бонусів. В такому випадку тестування буде виконуватись наступним чином, що відображений у таблиці 4.1.

Таблиця 4.1 – Тести для ігрових механік

Модуль продукту	Вихідні дані, що необхідні для тест-кейсу	Результат, що очікується по кожному кроці тест-кейсу
Створення сесії	<ol style="list-style-type: none"> 1. Зайти в гру 2. Обрати «Хост» 3. Обрати режим гри. 	<ol style="list-style-type: none"> 1. Виконується вхід в гру. 2. Відкривається елемент «Хост» з налаштуваннями. 3. Створюється ігрова сесія, до якої приєднується гравець, як хост.

Продовження таблиці 4.1.

Приєднання до сесії	<ol style="list-style-type: none"> 1. Зайти в гру 2. Обрати «Підключитися» 3. Обрати сервер, до якого відбудеться підключення. 	<ol style="list-style-type: none"> 1. Виконується вхід у гру. 2. Відкривається елемент «Підключитися» х налаштуваннями підключення. 3. Відбувається приєднання гравця до сесії, як клієнта.
Віднімання боєприпасів.	<ol style="list-style-type: none"> 1. Переконатися, що відключений у хоста чит на нескінченну кількість патронів. 2. Вистрілити один раз з поточної зброї. 3. Перезарядити зброю. 	<ol style="list-style-type: none"> 1. У хоста вимкнений чит на нескінченну кількість патронів. 2. Відбувається вистріл з зброї з переобчисленням кількості патронів. 3. Відбувається перезарядка зброї. 4. Віднімається 1 патрон з кількості патронів, що є запасними та оновлюється початкова кількість патронів у обоймі.
Віднімання здоров'я	<ol style="list-style-type: none"> 1. Створити або приєднатись до матчу. 2. Очікувати атаки бота або іншого гравця. 3. Кількість здоров'я зменшується. 	<ol style="list-style-type: none"> 1. Відбувається приєднання до матчу. 2. Бот або інший гравець атакує персонажа. 3. Шкала здоров'я зменшується.

Кінець таблиці 4.1.

Підняття бонусу	<ol style="list-style-type: none"> 1. Переконавшись, що відключений у хоста чит на нескінченну кількість патронів. 2. Вистрілити один раз з поточної зброї. 3. Підійти до бонусу на патрони для поточної зброї 	<ol style="list-style-type: none"> 1. У хоста вимкнений чит на нескінченну кількість патронів. 2. Віднімається патрону у зброї. 3. Бонус зникає і до зброї додається патрон.
-----------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Після того, як було сформовано основні тести для ігрових механік, було перевірено їх коректність роботи. Результати тестів вказані у таблиці 4.2.

Таблиця 4.2 – Результати тестування для ігрових механік

Модуль продукту	Вихідні дані, що необхідні для тест-кейсу	Результат по кожному кроці тест-кейсу
Створення сесії	<ol style="list-style-type: none"> 1. Зайти в гру 2. Обрати «Хост» 3. Обрати режим гри. 	Коректна робота
Приєднання до сесії	<ol style="list-style-type: none"> 1. Зайти в гру 2. Обрати «Підключитися» 3. Обрати сервер, до якого відбудеться підключення. 	Коректна робота

Кінець таблиці 4.2.

Віднімання боєприпасів.	<ol style="list-style-type: none"> 1. Переконатися, що відключений у хоста чит на нескінченну кількість патронів. 2. Вистрілити один раз з поточної зброї. 3. Перезарядити зброю. 	Коректна робота
Віднімання здоров'я	<ol style="list-style-type: none"> 1. Створити або приєднатись до матчу. 2. Очікувати атаки бота або іншого гравця. 3. Кількість здоров'я зменшується. 	Коректна робота
Підняття бонусу	<ol style="list-style-type: none"> 1. Переконатися, що відключений у хоста чит на нескінченну кількість патронів. 2. Вистрілити один раз з поточної зброї. 3. Підійти до бонусу на патрони для поточної зброї 	Коректна робота

Висновки до розділу 4

У даному розділі було здійснено опис програмного продукту та проведено тестування розробленого програмного забезпечення, зроблено набір окремого тестування для наступних складових гри: сервер, ігрових механік та статистики продуктивності гри.

5 СТАРТАП ПРОЕКТ

В даному розділі описується рішення для монетизації гри, що було розроблена в магістерській дисертації.

5.1 Опис ідеї продукту

Головна мета даного стартапу – це розробка онлайн шутеру від першої особи на основі ігрового рушія Unreal Engine 4. Дане рішення може використовуватися гравцями на будь-якому пристрої, при цьому геймеру не доведеться окремо встановлювати додаткові програми. Зміст ідеї, вигоди продукту по зрівнянню з існуючими рішеннями та цільові сфери використання розглянуто в таблиці 5.1.

Таблиця 5.1 – Загальні ідеї стартапу

Зміст ідеї	Сфери використання	Вигоди використання продукту для користувача
Реалізація онлайн шутеру від першої особи, на основі ігрового рушія UE 4	Геймінгова індустрія	Продукт може стати популярним в індустрії розважальних відеоігор.
	Кіберспорт	Залучення гравців у високорівневі професійні змагання.
	Спільнота гравців	Використання гри для спілкування з іншими гравцями

Огляд ринку показав, що фактично - ідентичних за загальним набором компонентів гри у розробленого програмного продукту є потенційні конкуренти. Одними з них є CS 2 та Valorant. На основі цих ігор, було проведено аналіз сильних, слабких та нейтральних сторін, результати яких наведено в таблиці 5.2.

Таблиця 5.2 – Опис сильних, слабких та нейтральних сторін основної ідеї стартап-проекту

Техніко-економічні ознаки ідеї	Концепції конкурентів			W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
	Поточний проект	CS 2	Valorant			
Мультиплатформенність	Так	Ні	Ні			+
Тип з'єднання	LAN, Прослуховуваний сервер, Віддалений сервер	LAN, Віддалений сервер	LAN, Віддалений сервер			+
Спосіб зберігання	ZIP, Github, Маркет	Маркет	Маркет			+
Оптимізація	Хороша	Середня	Хороша		+	

Кінець таблиці 5.2.

Наявність можливості перегляду зіграних раніше ігор	Є	Є	Немає		+	
Наявність ШІ у грі	Є	Є	Немає		+	
Наявність кредиту довіри спільноти	Немає	Середні й кредит довіри	Середні й кредит довіри	-		
Наявність активної спільноти	Немає	Є	Є	-		

З аналізу результатів з таблиці можна дійти висновку, що перелік з сильних сторін продукту, що був розроблений, становить основу для створення конкурентноспроможної бізнес-моделі для світового ринку ігор розроблення програмного забезпечення, так як ці критерії є хорошими перевагами для ігор будь-яких жанрів. Основними завданнями для цього продукту є лише у вдосконаленні спільноти користувачів і отримання їх довіри, що вирішується з часом з мінімальною рекламою (платною, шляхом розміщення її на різних платформах або безкоштовною, шляхом рекламування даного продукту блогерами).

5.2 Технологічний аудит ідеї продукту

У таблиці 5.3 наводиться технологічний аудит ідеї стартапу, де було визначено набір технологій для розробки поточно рішення, проводиться аналізування технологій, що необхідні та є у вільному доступі, з огляду на вказані ліцензії.

Серед основних функцій проекту:

- налаштування онлайн режиму, що включає в себе можливість створювати та підключатись до ігрової сесії;
- управління ігровою логікою, що включає в себе можливість налаштування графіки, купування онлайн товарів, перегляд зіграних раніше матчів, перегляд таблиці лідерів, та, безпосередньо, ігрова логіка під час матчу.

Таблиця 5.3 – Технологічна здійсненість ідеї проекту

№ п/п	Ідея продукту	Технології реалізації	Наявність технологій	Доступність технологій
1	Налаштування онлайн режиму	UE 4, C++	Реалізація у ігровому рушії з власними реалізаціями деяких елементів	Доступні
2	Управління ігровою логікою	Unreal Engine 4, C++, Blueprint	Наявність власної реалізації	Доступні

Кінець таблиці 5.3.

Технологія, обрана для реалізації ідеї проекту: Гра розробляється на основі Unreal Engine 4 з використанням мови програмування C++. Косметичні налаштування здійснюються за допомогою візуальної мови створення програм Blueprint

5.3 Аналіз ринкових можливостей стартап-проекту

У даному розділі проводиться аналіз можливостей на ринку інформаційних технологій у сфері розробки ігор. Також у розділі розглядаються загрози, які потенційно можуть мати негативні наслідки.

Спочатку було розглянуто аналіз загроз на ринку для стартап-проекту. Висновки аналізу знаходяться у таблиці 5.4.

Таблиця 5.4 – Аналіз загроз на ринку для стартап-проекту

№ п/п	Характеристика ринку	Критерій
1	Кількість головних конкурентів, од	Важко оцінити
2	Обсяг продаж, грн/ум.од.	Важко оцінити
3	Динаміка ринку (якісна оцінка)	Зростає дуже швидко
4	Умови для розвитку	+
5	Обмеження для старту проекту	Немає
6	Проходження сертифікації для запуску	Немає
7	Рентабельність проектів в галузі, %	95%

Як видно з результатів проведеного аналізу, можна однозначно ствердити, що дана технологічна ніша є дуже привабливою для запуску стартап-проекту.

Далі необхідно визначити потенційні групи клієнтів та їх характеристики, після чого необхідно сформувавши орієнтований перелік вимог до товару з визначених груп клієнтів. Результат характеристики наводиться у таблиці 5.5.

Таблиця 5.5 – Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Розважальні ігри	Молодь віком від 18 до 35 років.	Ця аудиторія може бути більш зацікавлена відповідно налаштованою графікою та високою швидкістю гри, а також може бути налаштована на конкуренцію та перемогу в іграх	Висока якість графіки, хороша оптимізація гри
2	Ігри на розвиток мислення	Професіонали, які шукають відпочинок після робочого дня та можливість покращити свої навички у розв'язанні завдань.	Студенти та професіонали можуть бути більш зацікавлені в іграх з високою складністю, які вимагають від них зосередженості та стратегічного мислення	Можливість зберігання результатів у лідербордах

На основі отриманих результатів аналізу груп потенційних клієнтів проводиться аналіз середовища ринку. Зокрема необхідно визначити фактори, що не сприяють впровадженню проекту. У таблиці 5.6 подано ці фактори, а також можливу реакцію компанії.

Таблиця 5.6 – Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Проблеми з мережевим з'єднанням та лаги	Нестабільне мережеве з'єднання або проблеми з лагами можуть значно вплинути на геймплей гравців, знижуючи задоволення від гри та спричиняючи негативний вплив на репутацію гри	Підтримка стабільного мережевого з'єднання, регулярні оновлення для оптимізації та виправлення помилок, а також використання серверів з високою швидкістю можуть допомогти зменшити ці проблеми
2	Безпека та чесність гри	Можливість читерства, використання недозволених програм чи нелегітимних методів гри може порушити баланс гри та знизити довіру гравців до продукту	Регулярне оновлення античит системи, моніторинг та швидке реагування на скарги щодо читерства, а також створення прозорих правил та суворих санкцій для порушників можуть

Кінець таблиці 5.6.

3	Низька активність гравців та втрата інтересу	може призвести до низької активності і навіть до втрати аудиторії	Регулярне оновлення контенту, введення нових карт, персонажів, режимів гри та евентів
---	----------------------------------------------	-------------------------------------------------------------------	---------------------------------------------------------------------------------------

Було сформовано також фактори, що мають сприятливі можливості для продукту. У таблиці 5.7 подано ці фактори, а також можливу реакцію компанії.

Таблиця 5.7 – Фактори можливостей

№ п/п	Фактор	Можливість	Можлива реакція компанії
1	Кросплатформеність	У гру можна грати на усіх платформах	Вивід продукту на усі платформи
2	Розширення географії аудиторії	Збільшення бази гравців за рахунок привернення гравців з різних країн, які зацікавлені в науково-фантастичних шутерах	Переклад гри на різні мови, рекламні кампанії в різних країнах, організація турнірів та подій, що привернуть увагу гравців з різних куточків світу

Кінець таблиці 5.7.

3	Створення спільноти та екосистеми гравців	Формування активної спільноти гравців, яка сприяє обміну досвідом, організацією турнірів та спільних подій	Створення форумів, соціальних мереж, розвиток онлайн-спільнот, регулярне спілкування з гравцями через стрімінги, блоги та інші платформи для підтримки спільноти
4	Регулярне оновлення контенту та геймплею	Забезпечення постійного інтересу гравців через нові картини, режими гри, персонажі, предмети та ігрові події	Проведення регулярних оновлень з новими функціями, контентом, запуск сезонних оновлень або ігрових подій

Після цього необхідно провести ступеневий аналіз конкуренції на ринку, результати якого вказані у таблиці 5.8.

Таблиця 5.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
Чиста конкуренція	На ринку присутні багато невеликих компаній, які пропонують схожі продукти або послуги	1) Визначити унікальні конкурентні переваги

Кінець таблиці 5.8.

Міжнародна	Проявляється в тому, що на ринку існує багато компаній-розробників і видавців ігор з різних країн світу, які також пропонують свої продукти на глобальному ринку.	1) Розробка стратегії маркетингу та реклами 2) Вивчення міжнародних ринків та потреб клієнтів з різних країн,
Внутрішнього алузева	Проявляється у боротьбі за лідерство на ринку, приваблюванням і утриманням клієнтів, підвищенням якості та інноваційністю продукту, зниженням вартості виробництва та покращенням ефективності бізнес-процесів.	Підтримувати своїх клієнтів і підвищувати їх лояльність до проекту
Товарно-видова	Проявляється у конкуренції між компаніями, що пропонують схожі товари чи послуги	1) Покращення якості продукту 2) Розширення функціоналу
Нецінова	У сфері розробки ігор конкуренція за ціною не є основною перевагою, оскільки ціни на ігри на ринку майже однакові.	1) Покращення якості продукту 2) Підтримка користувачів
Марочна	Компанії конкурують за своє ім'я, бренд, репутацію та інші характеристики, які допомагають їм вирізнитися на ринку серед інших компаній.	1) Рекламувати бренд 2) Запускати нові ігри

Після цього було проведено аналіз конкуренції в галузі за М. Портером, що відображено в таблиці 5.9.

Таблиця 5.9 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товаризамінники
	Прямими конкурентами є інші розробники ігор.	У цій галузі досить легко вийти на ринок, тому потенційні конкуренти присутні.	Рекламні мережі та сервіси платежів.	Геймери різних платформ	Схожі ігри на інших платформах (комп'ютери, консолі)
Висновки:	Висока	У цій галузі наявність потенційних конкурентів є досить високою. Строки виходу на ринок потенційних конкурентів є невідомими	Вплив постачальників на умови роботи на ринку може бути значним, особливо якщо вони мають монопольну позицію на ринку.	Споживачі можуть вимагати від компанії певні можливості	наявність товарів-замінників може обмежувати ринкові можливості

В даному секторі ринку наявна дуже досить велика конкуренція, бо на сьогоднішній день компанії розвивають цей напрям. На основі таблиці 5.9, можна зробити наступне твердження, що в даній галузі велика конкуренція, тому є ризики провалу або проблем при виході продукту на ринок.

Після цього необхідно сформувати фактори конкурентоспроможності, невраховуючи особливості ідеї, критерії вибору споживачами продукту та маркетингові фактори, а також визначити перелік факторів конкуренції що показані у таблиці 5.10.

Таблиця 5.10 – Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування
1	Можливості індивідуалізації персонажів	Гра повинна мати можливості для кастомізації персонажів, такі як різні кольори та аксесуари, що можуть заохочувати гравців до покупки додаткових предметів
2	Цікавий сеттинг	Наявність унікального сеттингу значно підвищує зацікавленість гравців у грі, адже тоді гру цікавіше досліджувати
3	Стабільна робота серверу	Ігри у жанрі шутеру вимагають підтримки мультиплеєра, щоб грати з друзями або іншими гравцями зі всього світу. Тому гра повинна мати стабільну роботу мультиплеєру
4	Не жадна монетизація	Цей стартап має ефективну стратегію монетизації, що буде відповідати вимогам гравців і забезпечувати прибуток для компанії

За цими даними факторами конкурентноспроможності було проведено характеристику різних сторін проекту, який наведено у таблиці 5.11.

Таблиця 5.11 – Порівняльний аналіз сильних та слабких сторін проекту

п/п	Фактор конкурентноспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з запропонованими					
			3	2	1			
	Можливості індивідуалізації персонажів	19						
	Цікавий сеттинг	15						
	Стабільна робота серверу	6						
	Не жадна монетизація	5						

Останнім етапом ринкового аналізу можливостей впровадження стартапу – це проведення SWOT-аналізу, дані якої зображено у таблиці 5.12.

Таблиця 5.12 – SWOT-аналіз стартапу

<p>Сильні сторони:</p> <ol style="list-style-type: none"> 1) Унікальні геймплейні механіки гри, що можуть привернути увагу геймерів 2) Можливість розвитку та розширення гри 3) 3) Можливість створити велику спільноту гравців та проводити турніри та змагання, що приверне увагу більш широкої аудиторії. 	<p>Слабкі сторони:</p> <ol style="list-style-type: none"> 1) Жанр шутерів є досить насиченим, що може зробити конкурентну боротьбу складнішою. 2) Недостатня популярність у порівнянні з відомими гравцями ринку, що може скласти складнощі при залученні інвесторів та партнерів. 3) Потребується значний час та ресурси на розробку та випуск гри
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Кінець таблиці 5.12.

<p>Можливості:</p> <p>1) Потенціал для залучення широкої аудиторії гравців</p> <p>2) Можливість співпраці з відомими ігровими платформами для розширення аудиторії.</p> <p>3) Можливість проведення змагань та турнірів, що може допомогти підняти популярність гри та залучити нових гравців.</p>	<p>Загрози:</p> <p>1) Конкуренція в жанрі, яка може скласти складнощі .</p> <p>2) Швидкий темп розвитку ігрової індустрії та вихід нових ігор, що може зменшити інтерес гравців.</p> <p>3) Залежність від технічних проблем та багів, які можуть негативно впливати на враження до відходу частини аудиторії.</p>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

На основі отриманих результатів були розроблені альтернативи ринкової поведінки для виведення стартап-проекту на ринок та приблизний час реалізації їх на ринку, з огляду на потенційні проекти конкурентів. Результат поданий у таблиці 5.13.

Таблиця 5.13 – Альтернативи ринкового впровадження стартап-проекту

№ п/п	Орієнтований комплекс заходів ринкової поведінки	Ймовірність отримання ресурсів	Термін реалізації
1	Використання маркетингової компанії для збільшення аудиторії	30% (збільшиться кількість потенційних клієнтів)	6 місяців
2	Випуск гри на одній з відомих платформ магазинів як безкоштовний продукт	60% (кількість потенційних клієнтів збільшиться, через пасивну рекламу)	3 місяці

Кінець таблиці 5.13.

3	Випуск гри мультиплатформною	80% (набагато збільшиться кількість користувачів)	збільшиться потенційних	2 місяці
---	------------------------------	---------------------------------------------------	-------------------------	----------

Серед способів реалізації стартап проекту на ринку найвигіднішим варіантом на початку є пункт №3.

5.4 Ринкова стратегія

Для того, щоб мати змогу об'єктивно оцінити ринкову діяльність та розробку стратегії ринкової політики, необхідно врахувати такі умови: ринок, у сфері якого відбувається розробка продукту, які сегменти ринку є цільовими, потенційних та прямих конкурентів.

На етапі аналізу ринку необхідно в першу чергу визначити його продуктові та географічні кордони, суб'єкти – продавці та покупці, поточний об'єм та емність ринку, кількісні та якісні показники структури ринку.

Опис цільових груп потенційних клієнтів наведені у таблиці 5.14.

Таблиця 5.14 – Обрання потенційних груп потенційних користувачів

№ п/п	Цільова група потенційних клієнтів	Зацікавленість введення проекту на ринок для споживача	Приблизний відсоток в межах цільової групи	Конкуренція в сегменті	Легкість входу у сегмент
1	Геймдев студії	Середня	40%	Висока	Низька
2	Видавці ігор	Середня	60%	Середня	Середня

Кінець таблиці 5.14.

3	Виробники апаратного забезпечення	Висока	30%	Низька	Середня
4	Організації, які спеціалізуються на турнірах	Висока	90%	Висока	Висока
В результаті аналізу була обрано 2 група					

Після того, як було проаналізовано групи потенційних клієнтів, необхідно обрати цільову групу, для якої буде пропонуватись продукт та обрати стратегію охоплення ринку, що буде базуватись на одній з наступних: стратегії масового маркетингу (коли не враховується відмінність між сегментами ринку та ринок розглядається, як одне ціле), стратегії диференційного маркетингу (коли йде охоплення якомога більшої можливої кількості сегментів ринку зі спеціально для них розробленими продуктами і окремою маркетинговою політикою для кожного з сегментів) та стратегія концентрованого маркетингу (коли йде зосередження всієї уваги та ресурсів на одному з сегментів ринку і пропонується продукт лише для обраної групи потенційних клієнтів).

Таким чином, було розроблено стратегію розвитку, що описана у таблиці 5.15.

Таблиця 5.15 – Обрання потенційних груп потенційних користувачів

№ п/п	Альтернатива розвитку	Стратегія охоплення ніші на ринку	Головні конкуренти стартап проекту, відповідно до обраної альтернативи	Початкова стратегія розвитку

Кінець таблиці 5.15.

1	Випуск гри мультиплатформною	Клієнти не завжди мають ПК, таким чином можливо охопити більше клієнтів	Важко підрахувати	Диференції
---	------------------------------	-------------------------------------------------------------------------	-------------------	------------

Далі було сформовано початкову стратегію конкурентної поведінки, опис якої наведено у таблиці 5.16.

Таблиця 5.16 – Початкову стратегію конкурентної поведінки

№ п/п	Чи є проект «першопроходцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
1	Ні	Забирати у існуючих	Не буде	Стратегія лідера

Висновки до розділу 5

У даному розділі було проведено аналіз вірогідної маркетингової стратегії для розробки рішення, а саме шутеру від першої особи на основі вбудованих можливостей ігрового двигуна Unreal Engine 4.27, а також перспективи впровадження цього продукту на ринку розробки ігор.

Рентабельність даної системи знаходиться на дуже високому рівні, але надалі необхідно розвивати продукт як в плані технологій, так і маркетингу, адже як показав аналіз, ця сфера змінюється дуже швидко.

На основі результатів, що були отримані при проведенні аналізу можна дійти до висновку, що подальша підтримка даного продукту є доцільною, оскільки в галузі є потреба в рішеннях такого роду.

ВИСНОВКИ

У дисертації розглянуто проблему реалізації онлайн шутеру з використанням вбудованих можливостей ігрового рушія Unreal Engine 4. Проведено аналіз існуючих рішень, виділено їх переваги і недоліки та актуальність проблеми і було сформовано вимоги до програмного забезпечення та виділено основні напрямки дослідження.

Представлено власні алгоритми для рішення проблеми реалізації ігрових механік та, власне, створено програмний продукт. Розроблене рішення надає можливість приєднуватися та створювати онлайн сесії, будь-яких видів, після чого грати у, безпосередньо, онлайн гру. Перевагою рішення є гнучкість налаштування створення онлайн сесії, що стає точками розширення і надає можливість використовувати даний програмний застосунок, або як кінцевий застосунок, або як базу для інших ігор.

В ході було розглянуто процес тестування різних аспектів гри, досліджено результати, які дозволяють однозначно ствердити, що гра має хорошу оптимізацію та всі її елементи працюють коректно.

Отже, розроблено програмне забезпечення, яке надає можливість приєднуватися та створювати онлайн сесії, будь-яких видів, після чого грати у, безпосередньо, онлайн гру.

Також описано програмну реалізацію та архітектуру гри. Описано її основні бібліотеки та модулі та як ці модулі взаємодіють між собою, аргументовано вибір технологій. На прикладі показано основні сценарії використання системи.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Жанри відеоігор: веб-сайт. URL: https://uk.wikipedia.org/wiki/%D0%96%D0%B0%D0%BD%D1%80%D0%B8_%D0%B2%D1%96%D0%B4%D0%B5%D0%BE%D1%96%D0%B3%D0%BE%D1%80 (Дата звернення: 28.11.2023).
2. Шутер: веб-сайт. URL: <https://uk.wikipedia.org/wiki/%D0%A8%D1%83%D1%82%D0%B5%D1%80> (Дата звернення: 10.09.2023).
3. Counter-Strike 2: веб-сайт. URL: https://uk.wikipedia.org/wiki/Counter-Strike_2 (Дата звернення: 28.11.2023).
4. Valorant: веб-сайт. URL: <https://uk.wikipedia.org/wiki/Valorant> (Дата звернення: 28.11.2023).
5. Call of Duty: Warzone: веб-сайт. URL: https://ru.wikipedia.org/wiki/Call_of_Duty:_Warzone (Дата звернення: 28.11.2023).
6. Unity: веб-сайт. URL: [https://uk.wikipedia.org/wiki/Unity_\(%D1%96%D0%B3%D1%80%D0%BE%D0%B2%D0%B8%D0%B9_%D1%80%D1%83%D1%88%D1%96%D0%B9\)](https://uk.wikipedia.org/wiki/Unity_(%D1%96%D0%B3%D1%80%D0%BE%D0%B2%D0%B8%D0%B9_%D1%80%D1%83%D1%88%D1%96%D0%B9)) (Дата звернення: 28.11.2023).
7. CryEngine documentation: веб-сайт. URL: <https://docs.cryengine.com/> (Дата звернення: 28.11.2023).
8. Unreal Engine: веб-сайт. URL: https://uk.wikipedia.org/wiki/Unreal_Engine (Дата звернення: 28.11.2023).
9. Sound Classes: веб-сайт. URL: <https://docs.unrealengine.com/4.27/en-US/WorkingWithAudio/SoundClasses/> (Дата звернення: 01.12.2023).
10. Sound Cue Reference: веб-сайт. URL: <https://docs.unrealengine.com/4.27/en-US/WorkingWithAudio/SoundCues/NodeReference/> (Дата звернення: 01.12.2023).

11. Collaborate with Sublevels: веб-сайт. URL: <https://docs.unrealengine.com/4.27/en-US/Basics/Levels/CollaborateWithSublevels/> (Дата звернення: 01.12.2023).
12. UGameInstance: веб-сайт. URL: <https://docs.unrealengine.com/4.27/en-US/API/Runtime/Engine/Engine/UGameInstance/> (Дата звернення: 01.12.2023).
13. AGameSession: веб-сайт. URL: <https://docs.unrealengine.com/4.27/en-US/API/Runtime/Engine/GameFramework/AGameSession/> (Дата звернення: 02.12.2023).
14. Replication Graph: веб-сайт. URL: <https://docs.unrealengine.com/4.27/en-US/InteractiveExperiences/Networking/ReplicationGraph/> (Дата звернення: 02.12.2023).
15. Skeletal Mesh Sockets: веб-сайт. URL: <https://docs.unrealengine.com/4.27/enUS/WorkingWithContent/Types/SkeletalMeshes/sockets/> (Дата звернення: 02.12.2023).
16. Animation System Overview: веб-сайт. URL: <https://docs.unrealengine.com/4.27/enUS/AnimatingObjects/SkeletalMeshAnimation/Overview/> (Дата звернення: 02.12.2023).
17. Blackboard Documentation: веб-сайт. URL: <https://forums.unrealengine.com/t/blackboard-documentation/1795> (Дата звернення: 03.12.2023).
18. Behavior Tree Overview: веб-сайт. URL: <https://docs.unrealengine.com/4.27/enUS/InteractiveExperiences/ArtificialIntelligence/BehaviorTrees/BehaviorTreesOverview/> (Дата звернення: 03.12.2023).
19. UBTTaskNode: веб-сайт. URL: <https://docs.unrealengine.com/4.26/en-US/API/Runtime/AIModule/BehaviorTree/UBTTaskNode/> (Дата звернення: 03.12.2023).
20. Online Subsystem: веб-сайт. URL: <https://docs.unrealengine.com/4.27/en-US/ProgrammingAndScripting/Online/> (Дата звернення: 03.12.2023).

21. Networking Overview: веб-сайт. URL:

<https://docs.unrealengine.com/4.27/enUS/InteractiveExperiences/Networking/>

[Overview/](#) (Дата звернення: 03.12.2023).