

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра інженерії програмного забезпечення

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри інженерії
програмного забезпечення

_____ Євген ДАВИДЕНКО

«17» _____ червня _____ 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА
ІГРОВИЙ ЗАСТОСУНОК У ЖАНРІ АСТІОН-
SHOOTER З ВИКОРИСТАННЯМ ТРИВИМІРНОГО
МОДЕЛЮВАННЯ

Спеціальність 121 Інженерія програмного забезпечення
Освітня програма «Інженерія програмного забезпечення»

Здобувач

Артем ГРИШИН

«__» _____ 2025р.

Керівник роботи

канд. техн. наук,
доцент

Анжела БОЙКО

«__» _____ 2025р.

Миколаїв 2025

Завдання на виконання кваліфікаційної роботи

Чорноморський національний університет імені Петра Могили

Факультет	Комп'ютерних наук
Кафедра	Інженерії програмного забезпечення
Рівень вищої освіти	Перший (бакалаврський)
Освітній ступінь	Бакалавр
Спеціальність	121 Інженерія програмного забезпечення
Освітня програма	Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри інженерії
програмного забезпечення

_____ Євген Давиденко

«16» _____ січня _____ 2025 р.р.

ЗАВДАННЯ

на кваліфікаційну бакалаврську роботу здобувача

Гришин Артем

1. Тема кваліфікаційної роботи «Ігровий застосунок у жанрі action-shooter з використанням тривимірного моделювання» затверджена наказом ректора ЧНУ ім. Петра Могили № 315 від «13» листопада 2024 р.
2. Строк представлення кваліфікаційної роботи «24» червня 2025 р.
3. Очікуваний результат роботи та початкові дані якщо такі потрібні.
Очікуваним результатом є ігровий застосунок у жанрі action-shooter з використанням тривимірного моделювання.

4. Перелік питань, що підлягають розробці:

Аналіз предметної області, порівняльний аналіз аналогів, визначення вимог та функціоналу системи, моделювання, проєктування програмного застосунку, кодування та тестування застосунку.

5. Перелік графічних матеріалів:

Презентація

6. Консультанти:

Консультант	Кафедра (організація)	Частина роботи

Дата видачі завдання « 25 » листопада 2024 р.

КАЛЕНДАРНИЙ ПЛАН
виконання кваліфікаційної роботи

Тема: «Ігровий застосунок у жанрі action-shooter з використанням тривимірного моделювання.».

№	Найменування роботи	Початок	Закінчення
1.	Розробка та затвердження завдання на виконання КРБ	29.11.2024р.	30.11.2024р.
2.	Огляд джерел інформації за темою роботи	15.02.2025р.	27.02.2025р.
3.	Складання календарного плану КРБ	28.02.2025р.	29.02.2025р.
4.	Аналіз предметної області	01.03.2025р.	15.03.2025р.
5.	Моделювання та конструювання ПЗ	15.03.2025р.	04.04.2025р.
6.	Кодування, тестування розробленого ПЗ, аналіз результатів тестування, розробка керівництва користувача	05.04.2025р.	25.05.2025
7.	Розробка спеціальної частини з охорони праці	26.05.2025	30.05.2025
8.	Відгук керівника КРБ	30.05.2025	13.06.2025
9.	Оформлення КРБ та презентації	30.05.2025	12.06.2025
10.	Попередній захист	28.05.2025	28.05.2025
11.	Рецензування	30.05.2025	14.06.2025
12.	Захист кваліфікаційної роботи	24.06.2025	24.06.2025

Здобувач

Артем ГРИШИН

« 30 » січня 2025 р.

Керівник роботи

канд. техн. наук,

доцент

Анжела БОЙКО

« 30 » січня 2025 р.

АНОТАЦІЯ

до кваліфікаційної бакалаврської роботи

«Ігровий застосунок у жанрі action-shooter з використанням тривимірного моделювання»

Здобувач 409 гр.: Гришин Артем

Керівник: канд. техн. наук, доцент Бойко Анжела

У сучасному світі комп'ютерні ігри набувають все більшої популярності, а їх розробка постійно вдосконалюється. Жанр action-shooterів залишається одним із найзатребуваніших серед гравців завдяки динамічному геймплею та захоплюючим ігровим механікам. Використання тривимірного моделювання дозволяє створювати реалістичні ігрові світи, деталізованих персонажів та ефектну зброю, що підвищує занурення гравця у процес.

Український ігровий ринок потребує якісних вітчизняних проєктів у цьому жанрі. Бакалаврська робота спрямована на розробку ігрового прототипу з використанням сучасних інструментів 3D-моделювання та ігрових технологій. Результат може стати основою для подальшого розвитку повноцінної гри.

Мета: Створення ігрового прототипу в жанрі action-shooter з використанням тривимірного моделювання для демонстрації основних ігрових механік та залучення інтересу гравців.

Об'єкт роботи: процеси, що пов'язані із організацією та створенням ігрового застосунку.

Предмет роботи: інструментальні засоби та інформаційні технології розробки ігрового застосунку.

У вступі визначається актуальність теми, пояснено мету дослідження та надано короткий огляд поставленої задачі, предмету дослідження та об'єкта дослідження.

У першому розділі проведено аналіз існуючих аналогів, що дозволить визначити основні особливості ігор та сформулювати загальне розуміння предметної області. У другому розділі описано процес розробки проєктних рішень з метою відповіді на вимоги до програмного забезпечення. Цей процес

включатиме моделювання об'єкту та предмету дослідження, а також створення функціональних та інформаційних моделей програмного забезпечення.

У третьому розділі показано процес розробки, включаючи вибір мови програмування, вибір рушія, розробку концепції гри, її геймплею, а також створення діаграм та інтерфейсу. У висновках проводиться аналіз роботи та отриманих результатів.

КБР викладена на 48 сторінок, вона містить 4 розділи, 18 ілюстрацій, 4 таблиць, 15 джерел в переліку посилань.

Ключові слова: Unreal Engine 5, тривимірне моделювання, action-shooter, C++, UML-діаграми

ABSTRACT

to the qualifying bachelor's thesis

«Game application in the genre of action-shooter using three-dimensional modeling»

Student 409 gr.: Grishin Artem

Supervisor: Candidate of Technical Sciences, Associate Professor Boyko Angela

In the modern world, computer games are becoming increasingly popular, and their development is constantly improving. The action shooter genre remains one of the most popular among players due to its dynamic gameplay and exciting game mechanics. The use of three-dimensional modeling allows creating realistic game worlds, detailed characters, and spectacular weapons, which increases the player's immersion in the process.

The Ukrainian gaming market needs high-quality domestic projects in this genre. The bachelor's thesis is aimed at developing a game prototype using modern 3D modeling tools and gaming technologies. The result can become the basis for further development of a full-fledged game.

Objective: To create a game prototype in the action-shooter genre using three-dimensional modeling to demonstrate the basic game mechanics and attract the interest of players.

Object of work: processes related to the organization and creation of a game application.

Subject of work: tools and information technologies for developing a game application.

The introduction defines the relevance of the topic, explains the purpose of the study and provides a brief overview of the task, subject of study and object of study.

The first section analyzes existing analogs to identify the main features of games and form a general understanding of the subject area. The second section describes the process of developing design solutions to meet software requirements. This process will include modeling the object and subject of research, as well as creating functional and information models of the software.

The third section shows the development process, including the choice of programming language, the choice of engine, the development of the game concept, its gameplay, and the creation of diagrams and interface. The conclusion analyzes the work and the results obtained.

The CBR is set out on 48 pages, it contains 4 sections, 18 illustrations, 4 tables, 15 sources in the list of references.

Keywords: Unreal Engine 5, three-dimensional modeling, action-shooter, C++, UML diagrams

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	3
ВСТУП.....	4
1 АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ.....	6
1.1 Огляд предметної області.....	6
1.2 Аналіз аналогів ігор жанру «Action-shooter».....	8
Висновки до розділу 1.....	12
2 ОПИС МЕТОДІВ ТА ТЕХНОЛОГІЙ	13
2.1 Аналіз інструментарію, моделей та методів для розробки ігрових застосунків	13
2.2 Аналіз інструментарію, моделей та методів для розробки ігрових застосунків	14
2.3 Опис технологій, що використовувались при розробці	15
2.4 Вимоги до програмного забезпечення	16
Висновки до розділу 2.....	20
3 МОДЕЛЮВАННЯ ПРОГРАМНОГО ЗАСТОСУНКУ	21
3.1 Створення сценаріїв використання	21
3.2 Побудова діаграм сценаріїв використання	23
3.3 Побудова діаграм станів	25
3.4 Побудова діаграм діяльностей	29
3.5 Побудова діаграми взаємодії.....	32
Висновки до розділу 3.....	33
4 ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	34
4.1 Створення трьохвимірних моделей	35
4.2 Тестування ігрової логіки.....	36
4.3 Тестування фізики та колізій.....	38
4.3 Тестування камер і візуальних ефектів	41
4.4 Тестування звуку	43
4.5 Функціональне тестування інтерфейсу	44
4.6 Установка, запуск, мінімальні вимоги	44
Висновки до розділу 4.....	45
ВИСНОВКИ.....	46
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	48

ПЕРЕЛІК СКОРОЧЕНЬ

ПЗ – програмне забезпечення

ПК – персональний комп'ютер

API – application programming interface

UE5 – Unreal Engine 5

UI – user interface

UML – unified modeling language

ВСТУП

Актуальність теми зумовлена тим, що комп'ютерні ігри вже заповнили весь світ, в них грають як діти, так і дорослі. Вони приваблюють користувачів своїм дизайном, сюжетом, персонажами та інтерактивністю у сучасному світі, де технології постійно розвиваються, комп'ютерні ігри залишаються важливим елементом розважальної індустрії, а жанр action-shooterів займає одну з провідних позицій завдяки своїй динамічності, інтерактивності та можливості занурення у віртуальні світи.

Метою роботи є розробка альфа-версії гри у жанрі action-shooter із використанням сучасного тривимірного моделювання для набуття захоплюючого ігрового досвіду.

Для досягнення поставленої мети необхідно виконати такі **завдання**:

- дослідити особливості жанру action-shooterів;
- проаналізувати сучасний ринок комп'ютерних ігор;
- сформулювати основні вимоги до майбутньої гри;
- спроектувати 3D-моделі для ігрового світу та персонажів;
- розробити анімації для ігрових об'єктів і персонажів;
- створити дизайн інтерфейсу користувача;
- реалізувати програмну частину гри на базі Unreal Engine 5;
- провести тестування гри для виявлення помилок і вдосконалення

ігрового процесу.

Об'єктом роботи є комп'ютерна гра у жанрі action-shooter, створена з використанням тривимірного моделювання на базі технологій Unreal Engine 5, так як в майбутньому буде продовжуватись розробка гри.

Предметом роботи є сучасні методи та технології створення комп'ютерних ігор з акцентом на жанр action-shooterів.

Ігровий рушій Unreal Engine 5 надає широкі можливості для створення високоякісних тривимірних ігор з реалістичною графікою, використовуючи сучасні технології рендерингу та закони фізики. Це робить його ідеальним

Ігровий застосунок у жанрі action-shooter з використанням тривимірного моделювання інструментом для розробки ігор у жанрі action-shooter, що забезпечують гравцям захопливий ігровий досвід.

Чому саме action-shooter? Жанр action-shooterів має декілька суттєвих переваг, які роблять його популярним серед гравців:

1. Динамічність. Гравець постійно перебуває в русі, виконує різноманітні завдання, приймає швидкі рішення, що робить гру захоплюючою.
2. Імерсивність. Завдяки сучасним технологіям графіки та звуку ігри в жанрі шутер занурюють гравців у реалістичні світи, де кожна деталь створює ефект присутності.
3. Різноманіття сценаріїв. Від науково-фантастичних до реалістичних сюжетів, жанр action-shooterів охоплює широкий спектр тем, що дозволяє задовольнити інтереси різних аудиторій.

1 АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ

1.1 Огляд предметної області

У сучасному цифровому світі відеоігри перетворилися на невід'ємну частину повсякденного життя мільйонів людей – незалежно від віку, професії чи соціального статусу. Особливо швидко розвивається сегмент action-shooter ігор з використанням 3D-моделювання, що пояснюється рядом чинників:

- доступність – завдяки широкому розповсюдженню комп'ютерів, консолей, смартфонів та хмарних платформ, шутери стають доступними широкому загалу користувачів;
- різноманіття жанрів – action-shooter ігри мають безліч піджанрів: від військових симуляторів до науково-фантастичних бойовиків, дозволяючи гравцю знайти щось на власний смак;
- суспільне визнання – якщо раніше ігри сприймалися як розвага без користі, то сьогодні вони вважаються інструментом розвитку логіки, реакції та навіть командної взаємодії;
- новітні технології – інтеграція віртуальної та доповненої реальності, а також глибоке 3D-моделювання й фізичні рушії створюють реалістичне, захоплююче ігрове середовище.

Жанр action-shooter є одним із найдинамічніших та найпопулярніших у сучасній ігровій індустрії. Його популярність зумовлена високим рівнем адреналіну, активною взаємодією з ігровим середовищем та широкими можливостями геймплейної реалізації [1].

Основою таких ігор є постійна участь гравця у бойових діях із застосуванням різних видів зброї та тактичних підходів. Завдяки використанню тривимірного моделювання, ігрове середовище виглядає реалістично та деталізовано, що сприяє глибокому зануренню в атмосферу подій. У деяких проектах додається механіка обмежених ресурсів, яка надає геймплею стратегічної глибини та вимагає від гравця розумного управління доступними засобами.

Важливою частиною є й сюжетна складова: шутери часто включають інтерактивні місії, завдання на виживання або захоплення територій, що доповнюються елементами драматургії. Атмосферність відіграє ключову роль — освітлення, звукові ефекти та музичний супровід створюють унікальний емоційний настрій, який тримає гравця в постійній напрузі[2].

Розробка тривимірної гри у жанрі action-shooter вимагає використання потужних інструментів, і одним із найефективніших рішень є Unreal Engine 5. Цей рушій забезпечує високоякісну графіку, динамічне освітлення та широкі можливості для створення складних ігрових механік. Важливою частиною процесу є створення 3D-моделей персонажів, об'єктів та середовища з подальшим додаванням анімації, що забезпечує плавність рухів і взаємодій.

Значну роль відіграє програмування, оскільки реалізація геймплейних функцій, штучного інтелекту та користувацького інтерфейсу потребує знання мов C++, Blueprints або Python. Не менш важливою є робота над геймдизайном — це охоплює розробку ігрових рівнів, балансування механік та побудову ігрового процесу в цілому. Звуковий супровід також є важливою складовою: якісно підібрані звуки бою, фонове озвучення та музика значно підсилюють емоційне сприйняття гри. Після завершення основної розробки необхідно провести всебічне тестування на різних платформах і здійснити оптимізацію, щоб забезпечити стабільну роботу гри та високу продуктивність.

Приклад комп'ютерних ігор у жанрі action-shooter: Call of Duty (серія), Far Cry (серія).

1.2 Аналіз аналогів ігор жанру «Action-shooter»

Аналіз аналогічних ігор у жанрі action-shooter є важливою складовою процесу розробки, оскільки дозволяє виявити сильні та слабкі сторони вже існуючих проєктів, а також знайти вільні ніші для створення конкурентоспроможного продукту. Це дає можливість врахувати чужі помилки, краще зрозуміти потреби гравців і підвищити загальну якість нового застосунку, особливо в аспектах тривимірного моделювання, геймплею та візуальної подачі.

У сучасному ігровому середовищі, де конкуренція надзвичайно висока, розуміння того, як функціонують і сприймаються подібні ігри, стає критично важливим. Завдяки детальному аналізу можна визначити ефективні дизайнерські рішення, які сприяють зануренню гравця у гру, виявити недоліки, що знижують якість ігрового досвіду, а також сформулювати бачення того, яким повинен бути новий продукт, щоб він відповідав очікуванням сучасної аудиторії.

Особливе значення має тривимірне моделювання, оскільки саме воно формує основний візуальний образ гри. Від якості моделей, текстур, анімацій і візуальних ефектів залежить не лише естетика, а й загальна зручність сприйняття ігрового простору. Оцінка підходів до тривимірної графіки в інших проєктах дозволяє розробникам виявити успішні технічні рішення та вдосконалити власні методи реалізації візуального контенту.

Кафедра інженерії програмного забезпечення
Ігровий застосунок у жанрі action-shooter з використанням тривимірного моделювання
Таблиця 1.1 – Опис Call of Duty

Назва	Call of Duty
Виробник	Activision (розробники різних частин: Infinity Ward, Treyarch, Sledgehammer Games тощо)
Архітектура	PC (Microsoft Windows), консолі (PlayStation, Xbox)
Мова реалізації	C++
Функції та характеристики	– динамічний геймплей: інтенсивні бойові сцени з різноманітною зброєю та технікою; – сюжетна кампанія: кінематографічні історії, що охоплюють різні історичні періоди або вигадані конфлікти; – мультиплеєр: розвинений багатокористувацький режим із різними картами, режимами гри та системами прогресії;
Переваги	– кінематографічність: висока якість постановки, спецефектів та сюжету; – різноманітність режимів: сюжет, мультиплеєр, кооперативні місії; – якість стрілянини: плавна анімація та реалістична механіка зброї.
Недоліки	– коротка тривалість сюжетних кампаній; – повторюваність у деяких частинах серії; – мікротранзакції у мультиплеєрі.
Джерело інформації	https://en.wikipedia.org/wiki/Call_of_Duty

Кафедра інженерії програмного забезпечення
 Ігровий застосунок у жанрі action-shooter з використанням тривимірного моделювання
 Таблиця 1.2 – Опис Far Cry

Назва	Far Cry
Виробник	Ubisoft (першу частину розробила Crytek)
Архітектура	PC (Microsoft Windows), консолі (PlayStation, Xbox)
Мова реалізації	C++, власні движки (CryEngine для першої частини, Dunia Engine для наступних)
Функції та характеристики	– відкритий світ: масштабні та докладно опрацьовані локації з можливістю вільного дослідження; – динамічна бойова система: використання різноманітної зброї, техніки, а також можливість стелсу; – сюжет: сильний акцент на історії з яскравими антагоністами та несподіваними поворотами; – система крафту: можливість створення боєприпасів, лікувальних засобів та спорядження з матеріалів, знайдених у світі гри.
Переваги	– відкритий світ: багаті, живі світи, які стимулюють дослідження; – глибина геймплею: можливість вирішувати завдання різними способами; – яскраві персонажі: харизматичні антагоністи і глибокі історії; – графіка: високий рівень деталізації оточення і динамічне освітлення.
Недоліки	– повторюваність: подібна структура місій у різних частинах серії; – технічні проблеми: можливі баги та збої, особливо на релізі; – сюжетні кліше: іноді історії можуть здаватися передбачуваними.
Джерело інформації	https://en.wikipedia.org/wiki/Far_Cry

Кафедра інженерії програмного забезпечення
Ігровий застосунок у жанрі action-shooter з використанням тривимірного моделювання
Таблиця 1.3 - Опис Metro Exodus

Недоліки	– технічні проблеми: на старті гри спостерігались баги та оптимізаційні труднощі; – відкритий світ: деякі гравці відзначали, що рівні здавалися менш насиченими подіями порівняно з більш лінійними частинами; – обмеження свободи: незважаючи на великі рівні, структура гри частково залишається досить скриптованою.
Джерело інформації	https://en.wikipedia.org/wiki/Metro_Exodus
Назва	Metro Exodus
Виробник	4A Games
Архітектура	PC (Microsoft Windows), консолі (PlayStation, Xbox)
Мова реалізації	C++, власний ігровий рушій 4A Engine
Функції та характеристики	– напіввідкритий світ: великі, відкриті рівні з можливістю вільного дослідження; – виживання: необхідність контролю за ресурсами, такими як фільтри для протигазів і боєприпаси; – сюжетна кампанія: глибока історія, яка продовжує події попередніх частин серії; – система крафту: можливість модифікувати зброю та спорядження безпосередньо у польових умовах.
Переваги	– атмосфера: сильне занурення у постапокаліптичний світ завдяки графіці, музиці та дизайну рівнів; – сюжет: емоційна історія з добре прописаними персонажами; – графіка: реалістичне освітлення, деталізовані моделі й ефекти (особливо з підтримкою технології Ray Tracing); – геймплей: баланс між відкритим світом та лінійними епізодами.

Кінець таблиці 1.3

Висновки до розділу 1

У процесі формування першого розділу було досліджено ігри у жанрі «Action-Shooter» із використанням ігрового рушія Unreal Engine 5. Було проведено аналіз предметної області застосунку та обґрунтовано актуальність обраної теми. Основна увага була приділена вивченню сучасного стану ігрової індустрії у світі та аналізу чинників, що сприяють її стрімкому розвитку і зростанню популярності серед людей різного віку та соціальних груп.

Додатково здійснено детальний огляд жанру «Action-Shooter», в якому визначено його основні особливості, такі як інтенсивний бойовий процес, реалістичне середовище, інтерактивність місій, атмосфера та роль обмежених ресурсів у геймплеї. Цей жанр відрізняється швидкісним і напруженим геймплеєм, де гравець має швидко реагувати на загрози. Важливу роль відіграють деталізовані ігрові світи, які посилюють відчуття реальності. Обмежені ресурси, такі як боєприпаси або аптечки, змушують гравця діяти стратегічно. Наведено приклади відомих ігор, що представляють цей жанр.

Було досліджено ключові аспекти, необхідні для розробки ігор на базі Unreal Engine 5, зокрема 3D-моделювання та анімацію, програмування ігрової логіки, дизайн геймплею, створення звукового супроводу, а також тестування та оптимізацію проєкту. Цей дозволяє створювати високоякісну графіку завдяки технологіям Nanite та Lumen. Також важливим етапом є аналіз аналогічних ігрових застосунків: досліджено мови розробки, функціональні можливості, основні характеристики, переваги й недоліки існуючих рішень, зокрема тестування та оптимізація проєкту.

2 ОПИС МЕТОДІВ ТА ТЕХНОЛОГІЙ

2.1 Аналіз інструментарію, моделей та методів для розробки ігрових застосунків

Сучасна розробка ігор базується на поєднанні інженерних практик, дизайнерських рішень і творчого підходу. З огляду на міждисциплінарний характер ігрових проєктів, традиційні підходи до розробки програмного забезпечення були адаптовані до потреб галузі. Це призвело до формування специфічного процесу — Game Development Software Engineering (GDSE), який охоплює всі етапи створення гри: від концепту до тестування та виходу на ринок.

Методологічні підходи

Переважає більшість студій та незалежних розробників віддає перевагу **гнучким (Agile) підходам** до управління розробкою. Методології Scrum, Kanban та Extreme Programming дозволяють ефективно працювати в умовах динамічно змінних вимог. Проєкти розбиваються на короткі ітерації (спринти), в межах яких команда розробляє певний функціонал, тестує його і презентує. Це забезпечує прозорість процесу, швидкий зворотний зв'язок і гнучкість прийняття рішень.

Прототипування є ще одним критично важливим методом. На ранніх етапах розробники створюють ігрові прототипи для перевірки ідей, механік та взаємодії з користувачем. Швидке прототипування дозволяє заощадити час і ресурси, оскільки дає змогу виявити недоліки до початку повномасштабної розробки.

У великих проєктах може частково застосовуватись **каскадна (waterfall) модель**, особливо на етапах документації та формалізації ігрового дизайну. Проте її використання обмежене через низьку гнучкість, що не відповідає сучасному темпу ігрової індустрії [15].

2.2 Аналіз інструментарію, моделей та методів для розробки ігрових застосунків

Розробка ігрових застосунків у жанрі action-shooter потребує застосування потужного інструментарію, здатного забезпечити якісну візуалізацію, динамічну побудову ігрового процесу та інтеграцію тривимірних моделей. Аналіз актуальних наукових джерел, розміщених у фахових виданнях та наукометричних базах за останні п'ять років, засвідчив, що одними з найефективніших засобів у створенні 3D-ігор є рушій Unreal Engine 5 (UE5) та система тривимірного моделювання Blender [7].

Unreal Engine 5 забезпечує високий рівень деталізації та продуктивності завдяки новітнім технологіям рендерингу, освітлення й оптимізації сцен. Інструменти рушія дозволяють реалізовувати ігрову логіку, взаємодію між об'єктами, систему управління персонажами, а також забезпечують підтримку фізичних симуляцій та візуальних ефектів [5]. Blender, як потужний та безкоштовний інструмент 3D-моделювання, дає можливість створювати і редагувати моделі персонажів, об'єктів оточення, а також здійснювати UV-розгортку, текстуркування та базову анімацію. Зв'язок між Blender і UE5 реалізується через стандартизовані формати обміну, зокрема FBX, що дозволяє легко імпортувати активи до рушія.

Методологічно проект базується на об'єктно-орієнтованому підході до розробки програмного забезпечення, а для побудови логічної та функціональної структури застосовуються стандарти UML (Unified Modeling Language). Використання діаграм класів, прецедентів, активностей тощо дозволяє формалізовано описати архітектуру системи, процеси взаємодії між її компонентами та логіку геймплею. Це забезпечує послідовний перехід від проєктування до реалізації програмного продукту.

2.3 Опис технологій, що використовувались при розробці

Під час реалізації ігрового застосунку було використано два ключові технологічні інструменти — Unreal Engine 5 для розробки основного функціоналу та реалізації ігрової логіки, а також Blender для створення тривимірного візуального контенту. Обидва інструменти взаємодіють як частини єдиного процесу проєктування, моделювання та інтеграції ігрових об'єктів у загальну систему.

Технології, що використовуються у процесі розробки, мають вирішальне значення для успіху гри. Центральне місце займають ігрові рушії — спеціалізовані платформи, які забезпечують базову архітектуру гри. Найпоширенішими є Unreal Engine (UE) та Unity. Вони включають засоби для 3D/2D візуалізації, фізичні модулі, систему обробки звуку, мережеву підтримку, а також системи скриптів і поведінки персонажів.

Unreal Engine 5 забезпечив повноцінне середовище для побудови ігрових рівнів, розміщення об'єктів, налаштування камер і освітлення, а також написання логіки за допомогою Blueprints та C++. Завдяки вбудованим можливостям рушія були реалізовані ключові компоненти ігрового процесу: рух і керування персонажем, обробка зіткнень, анімація дій, звуковий супровід та взаємодія з користувачем.

Вибір мов програмування залежить від рушія та цільової платформи. Наприклад, Unreal Engine використовує C++ і Blueprint (візуальне скриптування), тоді як Unity — переважно C#. Мови програмування відповідають за реалізацію логіки гри, фізики, UI, штучного інтелекту та мережевих функцій.

Іншою важливою складовою є інструменти для створення контенту: 3D-моделі, анімації, текстури, звуки. Для цього використовуються професійні графічні та анімаційні пакети — Blender, Autodesk Maya, Adobe Photoshop, Substance Painter тощо. Вони тісно інтегруються з рушіями, що дозволяє прискорити цикл виробництва активів.

Blender використовувався для створення 3D-моделей персонажів, зброї, об'єктів середовища та реквізиту. Моделі були розроблені відповідно до вимог щодо оптимізації полігональності, що забезпечує належну продуктивність у рушії. Після створення геометрії здійснювалося текстурування, запікання нормалей та підготовка моделей до експорту у форматі FBX, який є сумісним із UE5.

Узгоджене використання цих технологій дозволило досягти високого рівня інтеграції графічного контенту з програмною частиною, а також забезпечити ефективну реалізацію вимог до ігрового застосунку у жанрі action-shooter.

2.4 Вимоги до програмного забезпечення

2.4.1 Призначення та межі проєкту

1. Призначення програмного забезпечення – створення захоплюючого та динамічного ігрового досвіду в жанрі «Action-Shooter» з використанням тривимірного моделювання. Гра розроблена для розваги гравців, які шукають інтенсивні бойові пригоди, де необхідно демонструвати майстерність стрільби, тактику та швидку реакцію в умовах постійної небезпеки.

2. Було погоджено, що для створення гри буде використано ігровий рушій Unreal Engine 5.

3. Крайня дата завершення роботи над ПЗ – 13.06.2025р.

2.4.2 Загальний опис

1. Застосунок розроблений для розваги користувачів, які захоплюються динамічними та занурюючими ігровими досвідами, що дають можливість відчувати напружену атмосферу бойових дій і боротьби за виживання.

2. Основні характеристики користувачів: наявність ПК.

3. Загальних обмежень немає.

2.4.3 Функції системи

1. Функція досягнень створює додатковий стимул для гравців до виконання певних завдань або досягнення певних результатів в грі.
2. Вхідна інформація – назви локації, виконанні завдання, вихідна локація – отримання досягнення у головному меню.
3. Файл з досягненнями, які виконанні повністю або не до кінця.
4. Гравець має можливість збирати предмети, переглядати їх у інвентарі та використовувати у відповідних ситуаціях гри.
5. Вхідна інформація – взаємодія з предметами, відкриття меню інвентаря.
6. Вихідна інформація – відображення списку предметів та їх описів на екрані.
7. Зберігання та редагування масиву предметів у пам'яті гри.
8. Система квестів дозволяє гравцю отримувати сюжетні або побічні завдання та виконувати їх для просування в грі.
9. Вхідна інформація – активація квесту через діалог чи об'єкт.
10. Вихідна інформація – оновлення статусу завдання, отримання винагороди, відображення прогресу.
11. Структура для зберігання інформації про стан виконання квестів, можливість їхнього динамічного відслідковування.
12. Карта дозволяє гравцю орієнтуватися у світі гри, переглядати місце розташування та важливі позначки.
13. Вхідна інформація – натискання клавіші або відкриття меню карти;
14. Вихідна інформація – відображення мапи з динамічними позначками локацій.
15. Підтримка масштабування карти, оновлення інформації у реальному часі.
16. Гравець може взаємодіяти з об'єктами, які не додаються до інвентаря, такими як механізми, записки або двері.
17. Вхідна інформація – підхід до об'єкта та натискання клавіші взаємодії.

18. Вихідна інформація – зміна стану об'єкта (наприклад, відкриття дверей, відображення тексту).

19. Наявність механізмів обробки взаємодій та відповідних анімацій чи текстових елементів.

2.4.4 Вимоги до інформаційного забезпечення

1. Основним джерелом вхідної інформації є користувач.
2. Вимоги до нормативно довідкової інформації відсутні.
3. Усі данні зберігаються в файл збереження JSON.

2.4.5 Вимоги до технічного забезпечення

- ОС: Windows 7 і вище
- Оперативна пам'ять: 16 GB
- Відеокарта: відеопам'ять 2 GB і більше

2.4.6 Вимоги до програмного забезпечення

1. Архітектура програмної системи складається з клієнтської частини.
2. Гра розроблена на платформі Unreal Engine 5 з використанням системи Blueprints та мови C++.
3. Для створення програмного забезпечення було використано ОС Windows 10.
4. Гра розроблена на платформі Unreal Engine 5 з використанням системи Blueprints та мови C++.

2.4.7 Вимоги до зовнішніх інтерфейсів

1. Інтерфейс має задовольняти усі вимоги дизайну, він є доволі зручним у використанні.
2. Апаратним інтерфейсом є ПК користувача, з операційною системою Windows 10.
3. У ході розробки було використано дві категорії Unreal Engine API: UE5 Audio API, UE5 UI API.
4. Комунікаційний протокол відсутній.

2.4.8 Властивості програмного забезпечення

1. Гра є доступною для будь-якого користувача, за умови наявності апаратного інтерфейсу.
2. Гра не потребує супроводжуваності.
3. Програмне забезпечення може працювати на ОС Windows (7 версія та вище).
4. Продуктивність роботи ПЗ залежить від характеристик ПК.
5. Двигун гри забезпечує стабільну роботу та внутрішній обмін даних.
6. Усі вимоги сформовано та описано вище, доповнення не вимагається.

Висновки до розділу 2

У процесі підготовки другого розділу було проведено комплексний аналіз інструментарію, методів і технологій, необхідних для розробки ігрового застосунку у жанрі «Action-Shooter». Обґрунтовано вибір сучасних засобів розробки, зокрема Unreal Engine 5 та Blender, які забезпечують високий рівень якості графіки, інтерактивності та продуктивності. Unreal Engine 5 надає широкі можливості для створення ігрової логіки, управління персонажами, налаштування візуальних ефектів і фізичних симуляцій, тоді як Blender використовується для моделювання тривимірного контенту з урахуванням оптимізації та підготовки до імпорту в рушій. Розробка базується на об'єктно-орієнтованому підході із застосуванням UML-діаграм, що дозволяє чітко структурувати архітектуру проєкту та взаємодію компонентів. Також визначено функціональні, технічні та інформаційні вимоги до програмного забезпечення, описано архітектуру системи, особливості інтерфейсів, вимоги до апаратного забезпечення та середовища розробки. Узгоджене використання сучасних технологій і дотримання вимог забезпечує ефективність процесу розробки, стабільність ігрового застосунку та відповідність його функціональності жанровим очікуванням користувачів.

3 МОДЕЛЮВАННЯ ПРОГРАМНОГО ЗАСТОСУНКУ

Моделювання програмного застосунку є важливим етапом у процесі проєктування та розробки програмного забезпечення. У цьому розділі окреслюються ключові концептуальні та архітектурні рішення щодо створення застосунку, визначаються його основні функції, принципи роботи та вимоги, які він має задовольняти.

3.1 Створення сценаріїв використання

Короткий usecase. Користувач запускає ігровий застосунок та завантажує свій збережений прогрес. Система переносить його у тривимірний ігровий світ, розміщуючи на стартовій базі. Гравець отримує повідомлення від персонажа-оператора про нову місію, яка передбачає дослідження активних аномалій. Під час виконання завдання користувач просувається територією, стикаючись із ворогуючими фракціями, ворожими бійцями або небезпечними істотами [11]. Досягнувши цільової зони, гравець активує спеціальний пристрій для виявлення скарбу та здобуває цінні ресурси. Після успішного виконання завдання він повертається на базу, передає скарб командирю та обмінює здобуті ресурси на нове спорядження чи вдосконалення. Після завершення взаємодії система пропонує обрати наступну місію або вийти з гри. Користувач обирає вихід, зберігає прогрес та завершує сеанс.

Поверхневий usecase:

Основний сценарій (успішний): користувач запускає гру та завантажує збереження. У ігровому світі він підходить до торговця [12]. Система перевіряє наявний баланс валюти та рівень репутації гравця. Відповідно до цих параметрів користувачу відображаються доступні для покупки товари. При покупці система генерує унікальні ідентифікатори для кожного придбаного предмета, які автоматично додаються до інвентаря гравця. Отримані речі можна одразу використовувати у подальшому ігровому процесі.

Таблиця 2.1 – Повний usecase

Scope	Система взаємодії користувача з торговцем для купівлі та продажу ігрових предметів.
Level	Рівень цілей користувача (User-goal).
Primary Actor	Користувач, торговець
Stakeholders and interests	1. Користувач: зацікавлений у придбанні спорядження та зброї для покращення ефективності в бою. 2. Торговець: прагне реалізувати товари за вигідну для себе ціну та заробити ресурси.
Preconditions	1) Користувач має бажання придбати або продати товари. 2) Користувач володіє достатньою кількістю валюти. 3) Торговець має доступні для купівлі товари.
Main Success Scenario:	1) Користувач наближається до торговця. 2) Використовує команду взаємодії. 3) Система відкриває інтерфейс торгівлі. 4) Система перевіряє гроші та репутацію користувача. 5) Користувач переглядає доступний асортимент. 6) Вибирає потрібний товар. 7) Система перевіряє наявність обраного товару. 8) Користувач підтверджує купівлю. 9) Система генерує ідентифікатор предмета та додає його до інвентаря. 10) Оновлюється баланс валюти. 11) Користувач завершує торгівлю.

Extensions	1) Недостатньо грошей: торговець відмовляє в угоді та пропонує альтернативи. 2) Відсутність товару: користувач може обрати інший предмет або зачекати оновлення асортименту. 3) Продаж користувачем товарів: користувач переносить товар у зону продажу та підтверджує операцію, отримуючи оплату. 4) Шахрайство: при виявленні спроб читерства система блокує дію користувача.
Special Requirements	Необхідно реалізувати захист від маніпуляцій із валютними ресурсами або інвентарем.

Кінець таблиці 3.1

Альтернативні сценарії:

1. Користувач не має достатньо ресурсів для купівлі обраного товару.
2. Торговець не має в наявності певних товарів через непередбачувані події в грі, наприклад, напад ворожих сил або зникнення каравану.
3. Користувач застосовує сторонні засоби для отримання необмежених ресурсів; система виявляє порушення та блокує акаунт гравця.
4. Виникає технічна помилка у грі, через що інвентар торговця не завантажується.

3.2 Побудова діаграм сценаріїв використання

Діаграми сценаріїв використання відіграють вирішальну роль у процесі моделювання ігрового застосунку, оскільки вони надають чітке графічне представлення всіх можливих варіантів взаємодії між гравцем та ігровою системою. Цей інструмент дозволяє не лише візуалізувати, але й детально проаналізувати кожен потенційний шлях розвитку подій, що особливо важливо для створення гнучкого та захоплюючого ігрового процесу. Основне призначення

таких діаграм полягає у точному визначенні механізмів, за допомогою яких користувач буде залучатись до ігрового світу, взаємодіяти з його елементами та впливати на розвиток сюжету в умовах реального геймплею[8].

Використання діаграм сценаріїв є невід'ємною частиною процесу створення сучасних відеоігор, оскільки вони дозволяють розробникам створити інтуїтивно зрозумілу систему управління та інтерфейс, який повністю відповідає очікуванням цільової аудиторії. Завдяки детальній візуалізації всіх можливих сценаріїв взаємодії, команда розробників може заздалегідь виявити потенційні проблеми юзабіліті та оптимізувати користувацький досвід. Це безпосередньо впливає на якість кінцевого продукту, роблячи ігровий процес більш зручним, логічним і захоплюючим для гравців різного рівня підготовки.

Крім того, діаграми сценаріїв використання слугують фундаментом для ефективного планування процесу тестування гри. Вони дозволяють QA-фахівцям розробити комплексний набір тестових кейсів, які повністю охоплюють як основні, так і альтернативні шляхи проходження гри. Таке систематичне тестування забезпечує виявлення критичних помилок на ранніх етапах розробки, зменшує витрати на подальші доробки та значно підвищує стабільність робочої версії продукту.

Важливо відзначити, що якісно розроблені діаграми сценаріїв дозволяють створити більш реалістичні тестові ситуації, що максимально наближені до умов реального використання гри кінцевими користувачами.

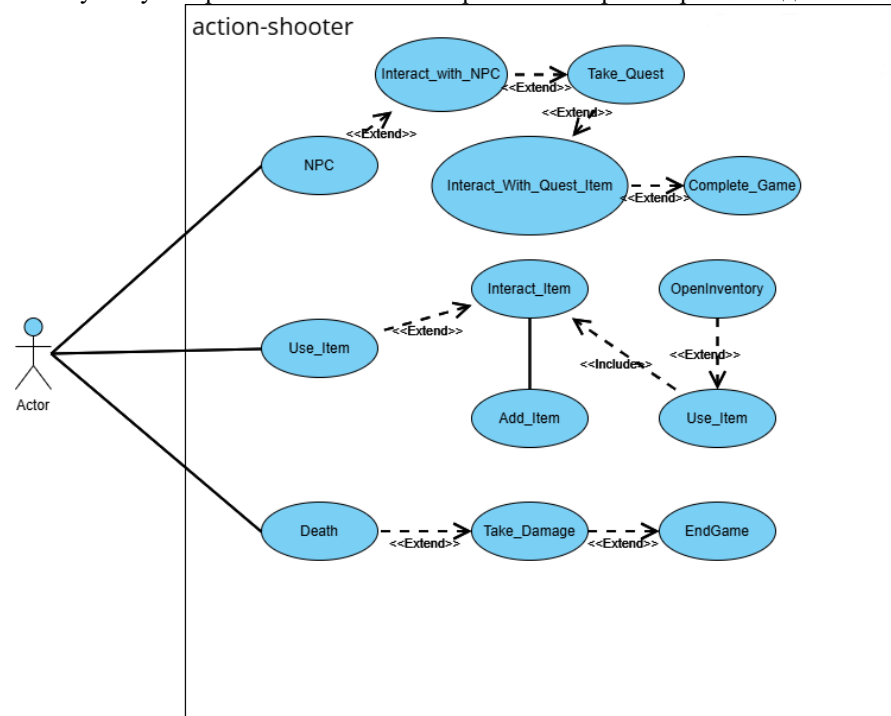


Рисунок 3.1 – Діаграма сценаріїв використання

Дана діаграма відображає систему взаємодій у грі, яка складається з трьох ключових компонентів. Взаємодія гравця з предметами включає механіку інвентаря, комбінування елементів, обмеження ресурсів та динамічну зміну ігрового середовища через взаємодію з об'єктами. Взаємодія з персонажами охоплює систему діалогів з гілками вибору, механізми репутації, поведінку NPC-компаньйонів з унікальними навичками та адаптивних противників. Сценарії подій з персонажем гравця містять основні сюжетні повороти, випадкові події, систему морального вибору з наслідками та механіки розвитку персонажа. Усі ці елементи формують цілісну систему, де кожен компонент впливає на інші, створюючи глибокий і динамічний ігровий досвід [13].

3.3 Побудова діаграм станів

Діаграми станів є графічними моделями, що використовуються для опису поведінки ігрового застосунку в жанрі action-shooter, зокрема для відображення змін станів персонажів, об'єктів або ігрового середовища у відповідь на різні події. Основна мета таких діаграм — візуалізація всіх можливих станів, у яких може

перебувати персонаж або об'єкт у тривимірному середовищі, та визначення умов, що спричиняють зміну цих станів.

Використання діаграм станів у розробці 3D action-shooter гри допомагає чітко описати поведінку системи у відповідь на входні події — наприклад, атаки ворогів, взаємодію з об'єктами, або зміну режиму руху персонажа (біг, стрибок, ухилення тощо). Це дозволяє краще розуміти, як ігровий застосунок реагує на дії гравця та події в ігровому світі.

Для побудови діаграм станів спершу потрібно визначити всі можливі стани персонажів та об'єктів (наприклад: стоїть, біжить, стріляє, отримує ушкодження, мертвий). Далі ідентифікуються події чи дії, які змінюють ці стани (наприклад: натискання кнопки стрільби, попадання кулі). Після цього встановлюються переходи між станами, а також відповідні дії або анімації, що виконуються під час переходу. Завершальним етапом є побудова самої діаграми за допомогою умовних позначень і стрілок для наочного представлення станів, переходів і супутніх дій.

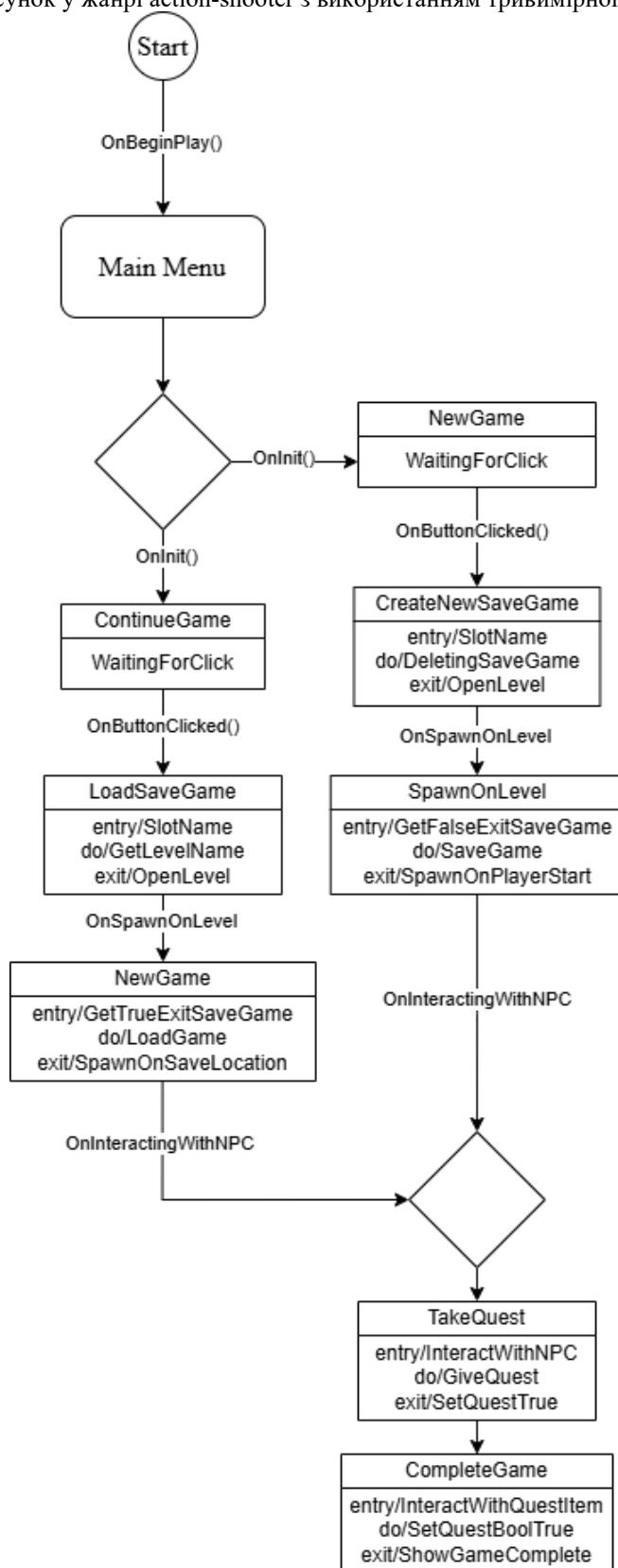


Рисунок 3.2 – Діаграма стану системи в цілому

На діаграмі станів для ігрового застосунку в жанрі action-shooter загалом відображено послідовність переходів між основними етапами гри. Серед цих

етапів: головне меню, початок нової гри або продовження вже збереженої сесії.

Після вибору опції нової гри або продовження, гравець завантажується на певний рівень у тривимірному ігровому середовищі. Далі користувач проходить рівень, виконуючи завдання та взаємодіючи з об'єктами та супротивниками. Після завершення гри система відображає результати проходження.

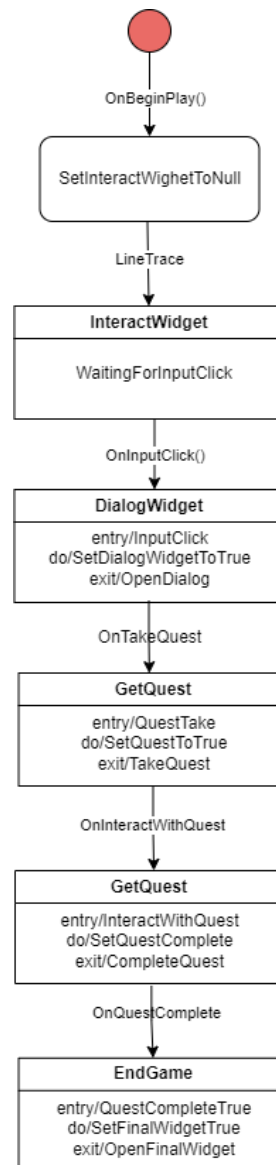


Рисунок 3.3 – Діаграма станів квесту

На другій діаграмі станів представлено процес проходження окремого завдання у грі. Після успішного виконання завдання гравцем, ігровий процес завершується, і на екрані з'являється фінальне вікно з результатами проходження.

3.4 Побудова діаграм діяльностей

Діаграма діяльностей є графічним відображенням послідовності дій або процесів у межах ігрового застосунку в жанрі action-shooter. Вона використовується для моделювання та візуалізації порядку подій, що відбуваються під час виконання конкретних ігрових сценаріїв, таких як взаємодія персонажа з іншими об'єктами або героями. Діаграма допомагає краще зрозуміти, які кроки має виконати гравець, у якому порядку вони реалізуються, а також які варіанти розвитку подій можливі залежно від рішень гравця.

Для побудови діаграми діяльностей у 3D action-shooter необхідно чітко визначити сценарій, що моделюється, після чого застосовуються стандартні символи та нотація для візуалізації послідовності дій і умов переходу. Важливо також врахувати взаємодію між гравцем та іншими персонажами, а також події, що можуть вплинути на подальший хід гри [14].

На першій діаграмі діяльностей зображено процес діалогу між гравцем і неігровим персонажем. Як показано на рисунку 3.4, гравець має вибір між двома варіантами діалогу: у випадку вибору звичайної розмови він отримує інформацію, а якщо обирає квестовий діалог, то після завершення розмови йому видається нове завдання для проходження.

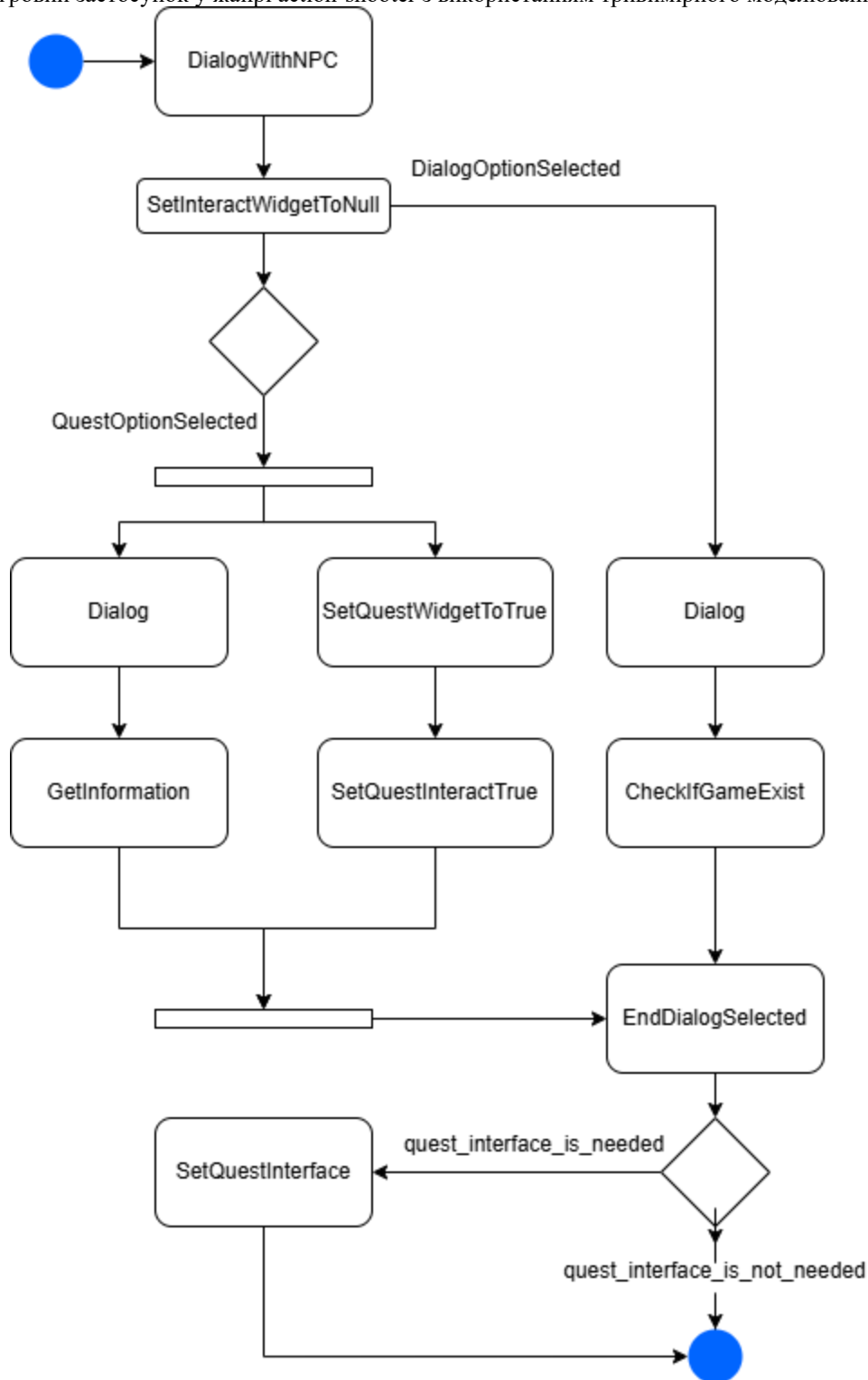


Рисунок 3.4 – Діаграма діяльності початку гри

Друга діаграма діяльностей відображає процес початку гри. На цьому етапі гравець має можливість обрати між початком нової гри або продовженням вже збереженої.

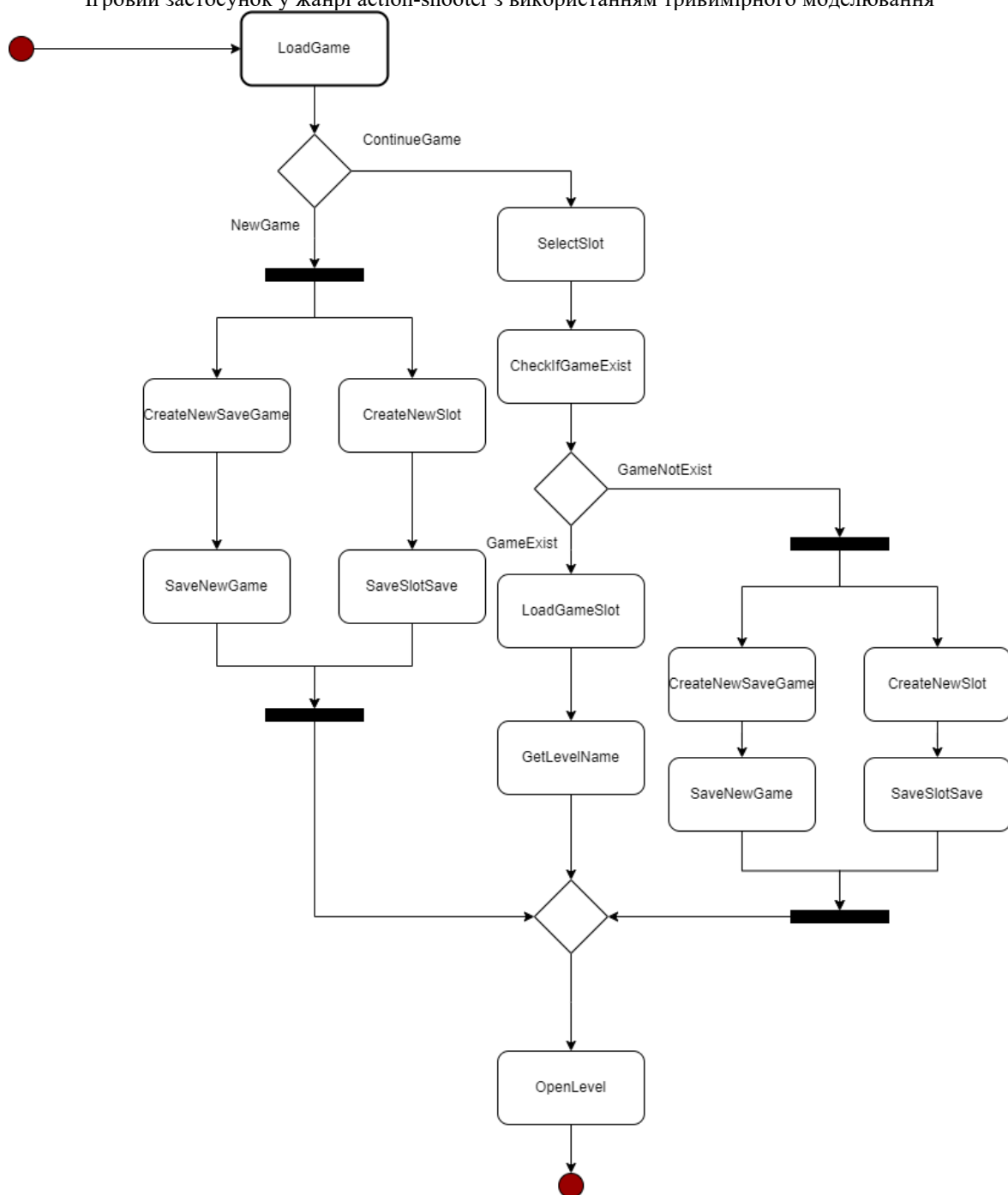


Рисунок 3.5 – Діаграма діяльності початку гри

Якщо обирається нова гра, запускається початковий рівень, після чого автоматично здійснюється збереження прогресу. У разі вибору продовження гри, гравцеві пропонується обрати відповідний слот збереження, після чого відбувається завантаження потрібного рівня та збереження поточного стану гри.

3.5 Побудова діаграми взаємодії

Діаграми взаємодії відображають обмін діями та повідомленнями між різними об'єктами або системними компонентами в межах ігрового застосунку в жанрі action-shooter. Вони дозволяють моделювати та аналізувати різноманітні сценарії взаємодії в ігровому процесі, зокрема комунікацію між гравцем і системою, а також між різними елементами гри.

Існують кілька типів діаграм взаємодії, серед яких найпоширенішими є діаграми послідовності та діаграми співпраці. Діаграми послідовності демонструють порядок надсилання та отримання повідомлень між об'єктами, відображаючи часову послідовність подій. Діаграми співпраці акцентують увагу на структурі взаємодій і залежностях між об'єктами [9].

Використання діаграм взаємодії в action-shooter грі допомагає краще розуміти, як має функціонувати система, і виявляти потенційні проблеми у комунікації між її складовими. Це важливий інструмент під час розробки та оптимізації архітектури гри.

На діаграмі взаємодії, що була побудована, представлено процес взаємодії користувача з головним меню ігрового застосунку.

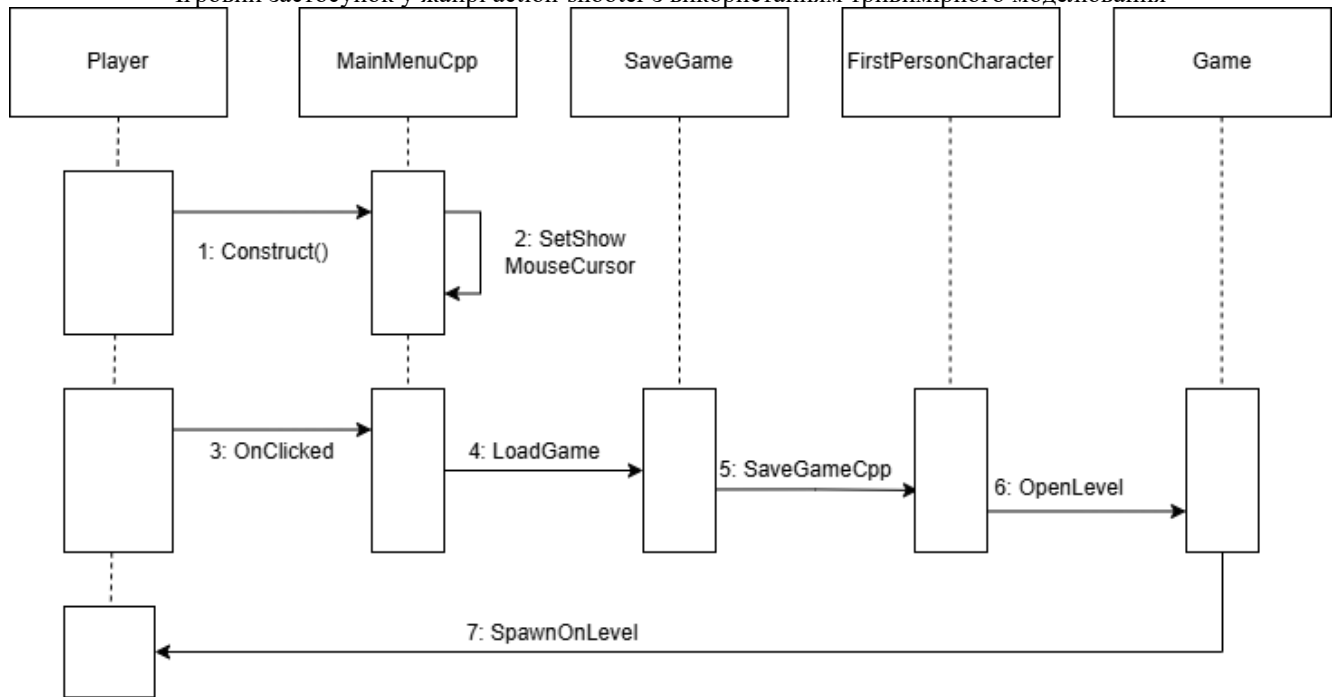


Рисунок 3.6 – Діаграма взаємодії

Діаграма демонструє послідовність завантаження гри: гравець викликає меню, де відображається курсор миші. Після натискання завантажується збережена гра, передаються дані персонажу, відкривається рівень, і гравець з'являється в потрібному місці.

Висновки до розділу 3

У третьому розділі було досліджено процес створення різних типів UML-діаграм, зокрема діаграм станів, варіантів використання, взаємодії та діяльності, адаптованих до специфіки ігрового застосунку в жанрі action-shooter з використанням тривимірного моделювання. Було розроблено власні UML-діаграми, що відображають ключові сценарії взаємодії користувача з грою, а також описано варіанти використання у формі короткого, оглядового та детального сценаріїв. На основі проведеного аналізу підготовлено підґрунтя для початку розробки ігрового застосунку у вибраному жанрі.

4 ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Тестування action-shooter гри з 3D-графікою для ПК – це комплексний процес, спрямований на забезпечення високої якості та стабільності гри перед її релізом. Основна увага приділяється продуктивності, геймплейним механікам, графічній складовій та загальному користувацькому досвіду.

Першочерговою задачею є перевірка продуктивності та оптимізації. Гра повинна стабільно працювати на різних конфігураціях ПК – від бюджетних до потужних систем. Тестувальники аналізують навантаження на процесор, відеокарту та оперативну пам'ять, відстежують падіння частоти кадрів у інтенсивних сценах та перевіряють адаптацію гри до різних графічних налаштувань.

Важливим аспектом є тестування геймплейних механік. Для action-shooter критичною є чітка робота систем стрільби, переміщення та взаємодії з навколишнім середовищем. Перевіряється баланс зброї, фізика руху персонажа, робота штучного інтелекту ворогів та коректність ігрової фізики.

Графічна складова потребує ретельної перевірки. Тестувальники оцінюють якість текстур, освітлення, тіней та спецефектів, відстежують можливі візуальні артефакти та переконуються у коректній роботі анімацій. Особлива увага приділяється інтерфейсу – він повинен бути інтуїтивним та не заважати геймплею.

Керування та інтерфейс – ключові елементи комфортної гри. Тестування включає перевірку відчуттів від керування на клавіатурі, миші та геймпаді, зручності навігації в меню та налаштуваннях. Для мультиплеєрних режимів додатково аналізується стабільність підключення та баланс у мережевій грі.

Окремо перевіряється сумісність з різними версіями операційних систем та апаратними конфігураціями. Гра повинна коректно працювати на Windows 10/11, а також, за можливості, на Linux через Proton. Тестувальники перевіряють роботу з різними графічними API та на моніторах з нестандартними співвідношеннями сторін.

Фінальним етапом є оцінка загального враження від гри. Тестувальники

Ігровий застосунок у жанрі action-shooter з використанням тривимірного моделювання аналізують, наскільки захоплюючим є ігровий процес, чи виникають моменти розчарування через технічні недоліки, і чи залишається бажання продовжувати гру. Це дозволяє виявити та усунути проблеми, які можуть вплинути на репутацію проекту після релізу.

Результати тестування допомагають розробникам доопрацювати гру, усунути критичні помилки та створити продукт, який задовольнить очікування гравців. Кожен виявлений недолік, усунений до релізу, значно підвищує шанси гри на успіх у конкурентному середовищі ігрової індустрії.

4.1 Створення трьохвимірних моделей

Процес створення тривимірних моделей є базовим етапом розробки ігрового середовища та об'єктів у шутері. Саме на цьому етапі формуються всі ключові елементи, з якими гравець безпосередньо взаємодіє: персонажі, зброя, предмети, інтер'єри та ландшафти. Від якості побудови моделей залежить як візуальна привабливість гри, так і коректна робота колізій, анімацій і фізичних симуляцій.

Створення моделей розпочинається зі створення базової геометрії — мішення, згладжування країв, додавання деталей, після чого застосовуються текстури та шейдери. У прикладі нижче показано процес побудови моделі зброї, яка надалі буде використана у грі як один з основних елементів озброєння персонажа.

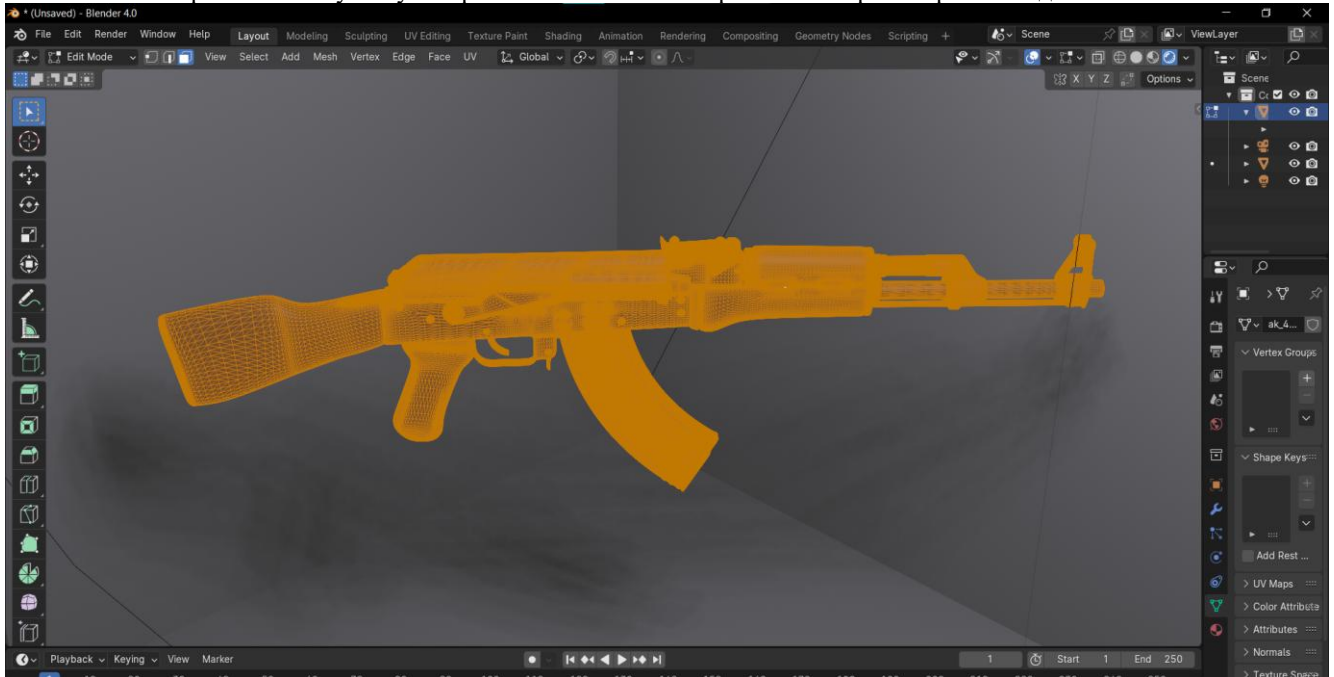


Рисунок 4.1 – Створення трьохвимірної моделі зброї

На зображенні продемонстровано одну з початкових стадій моделювання — формування корпусу зброї, деталізація компонентів (приклад, затвор, ствол) та підготовка до текстурювання. Важливим етапом є оптимізація полігональної сітки для досягнення балансу між візуальною якістю та продуктивністю у грі. Після завершення створення геометрії модель експортується у формат, сумісний із рушієм Unreal Engine, де відбувається подальше налаштування — додавання колізій, анімацій, матеріалів та інтеграція у геймплей.

4.2 Тестування ігрової логіки

Ігрова логіка визначає поведінку основних механік шутера — прицілювання, обчислення відстані, реакція ворогів на шкоду. На цьому етапі тестування важливо перевірити, наскільки коректно працює механіка у типових ситуаціях бою, як система враховує взаємодію гравця з довкіллям і як саме відбувається обробка подій, пов'язаних з атакою та отриманням шкоди.

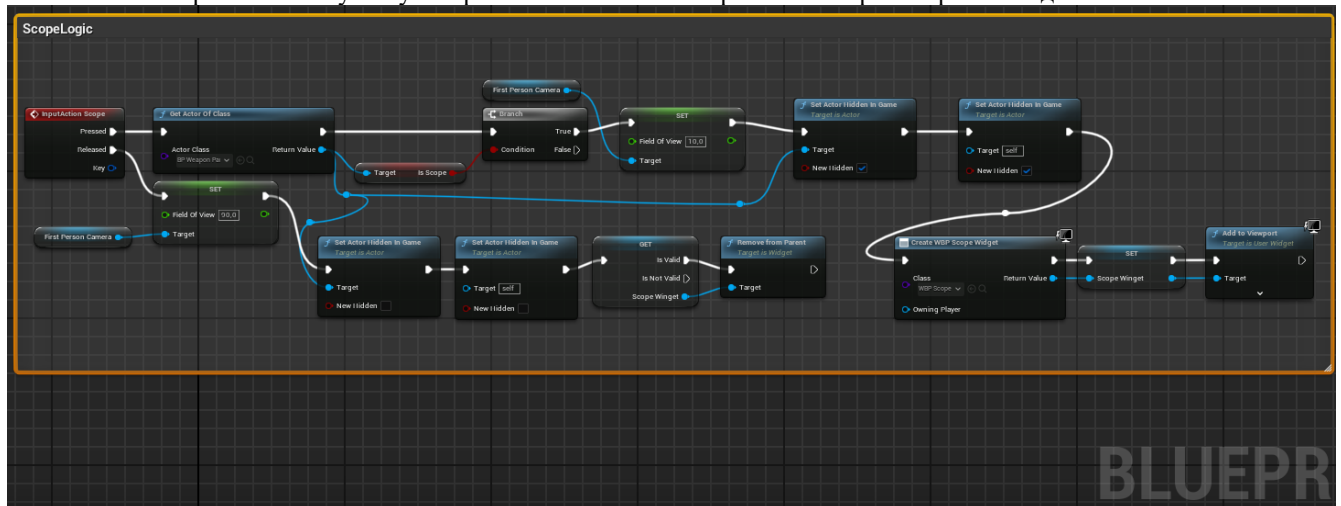


Рисунок 4.1 – Логіка прицілювання з прицілом

Зображення демонструє реалізацію системи прицілювання. Приціл динамічно змінює положення залежно від руху гравця та положення камери, забезпечуючи точність і зручність стрільби. Логіка передбачає перевірку на перешкоди, коригування напрямку стрільби і плавність переходів між станами прицілювання.

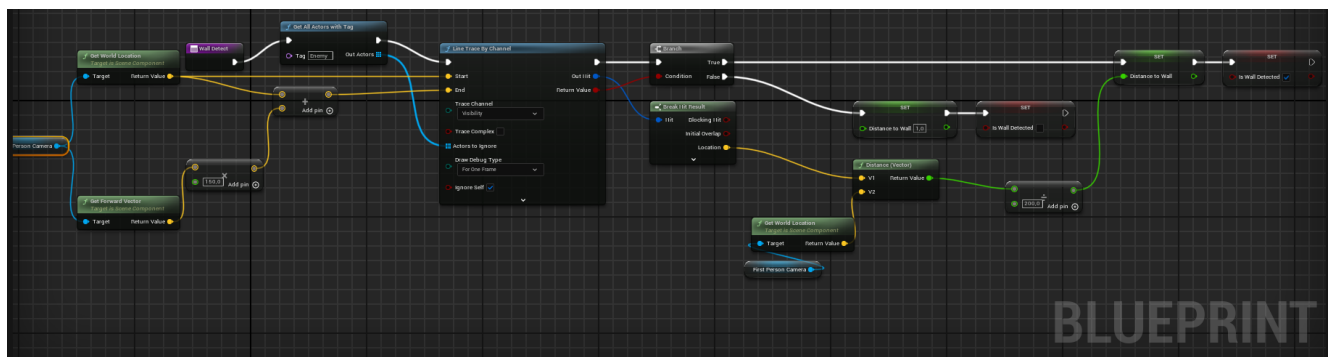


Рисунок 4.2 – Логіка вирахування відстані до стіни

Цей скріншот відображає механізм трасування променя до найближчої перешкоди перед гравцем. Така логіка необхідна для обмеження дальності атаки, активації певних анімацій або для тригерів подій, наприклад, коли гравець наближається до об'єкта.

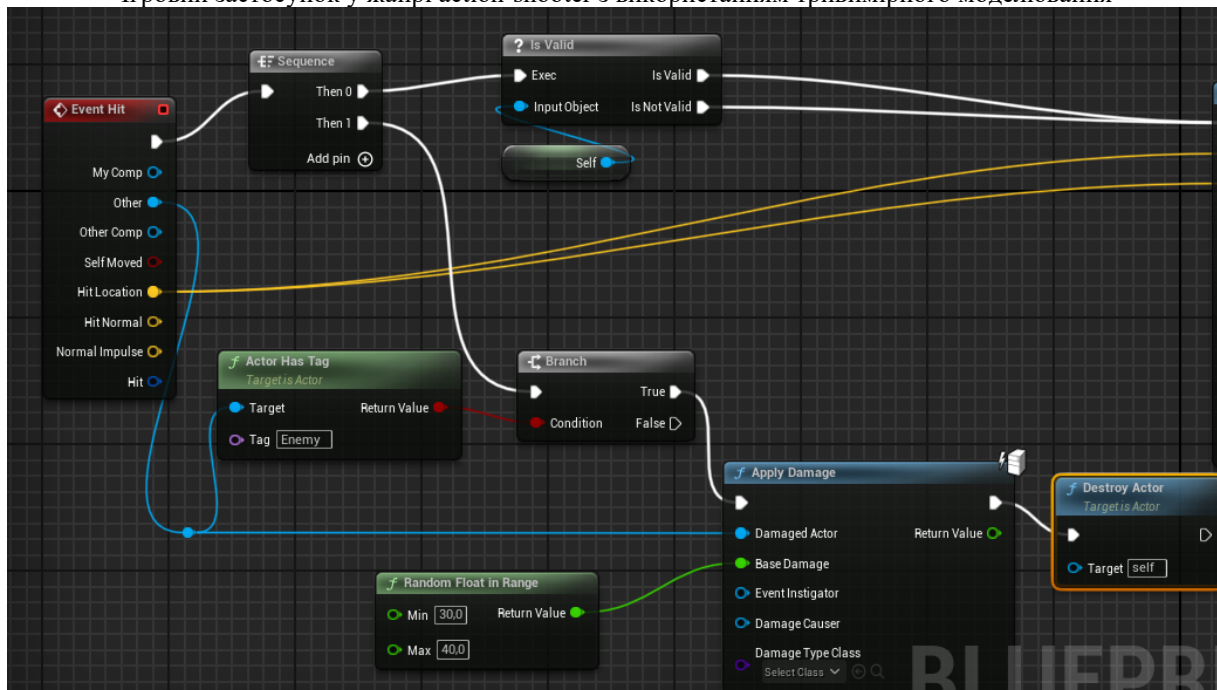


Рисунок 4.3 – Логіка отримання дамагу ворогом

Ілюстрація відображає механізм розрахунку шкоди, що враховує зону влучання (наприклад, голова, тіло чи кінцівки), тип зброї та броню ворога. Такий підхід дозволяє створити більш реалістичну бойову систему. Завдяки перевірці цих аспектів забезпечується стабільна логіка ведення бою та взаємодії зі світом.

4.3 Тестування фізики та колізій

Фізика гри відповідає за реалізм ігрового середовища, взаємодію об'єктів між собою, а також вплив дій гравця на ці об'єкти. У цьому розділі тестується коректність роботи колізій, реакції тіл на зовнішні впливи, рух снарядів і предметів після пострілу.

Кафедра інженерії програмного забезпечення
Ігровий застосунок у жанрі action-shooter з використанням тривимірного моделювання

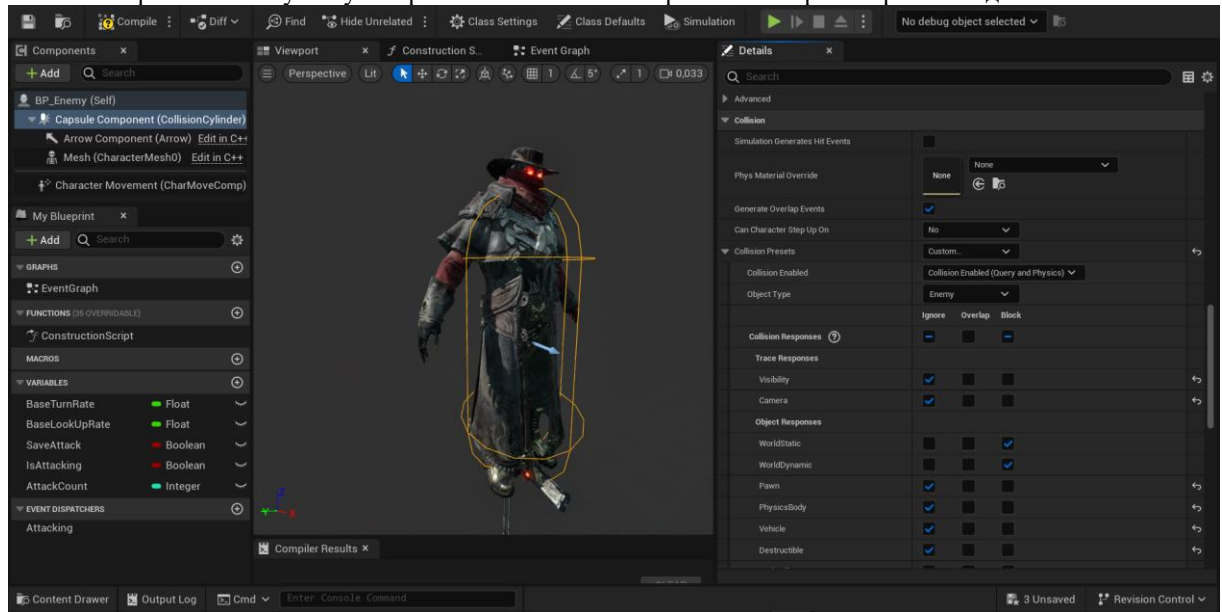


Рисунок 4.4 – Логіка отримання дамагу ворогом

Даний скріншот показує, як обробляється подія зіткнення снаряда з тілом ворога. Тестується точність позиціонування колізійних об'єктів та правильність завдання шкоди при контакті.

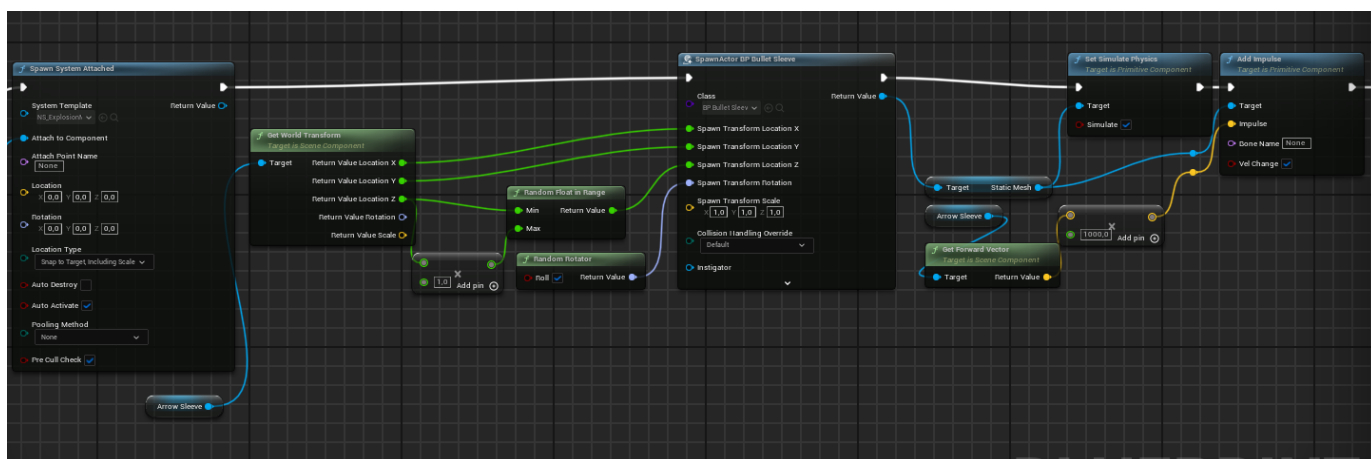


Рисунок 4.5 – Виліт гільз з затвора

Зображено процес генерації гільз, що вилітають після кожного пострілу. Реалізація включає симуляцію сили, кута відхилення, взаємодії з поверхнями та відтворення відповідних ефектів.

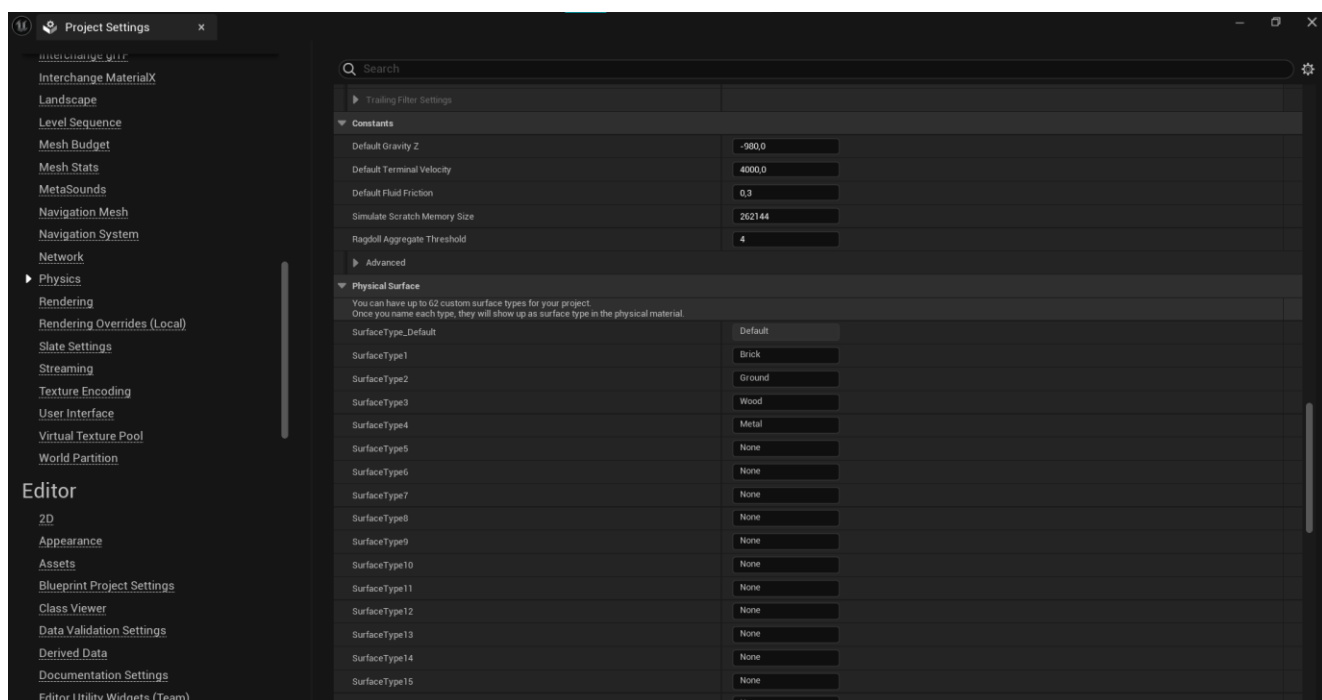


Рисунок 4.6 – Налаштування фізичних властивостей поверхонь

Ілюстрація демонструє, як у рушії налаштовуються фізичні матеріали, що визначають поведінку об'єктів при зіткненнях з різними типами поверхонь (метал, бетон, ґрунт тощо). Тестування фізичних процесів необхідне для уникнення

"проломів" у колізіях, некоректної взаємодії тіл або нереалістичної поведінки об'єктів.

4.3 Тестування камер і візуальних ефектів

Цей етап тестування зосереджено на налаштуванні камери та візуальних ефектів, які формують естетику ігрового процесу. Коректна робота камери значно впливає на сприйняття бою та загальну атмосферу гри.

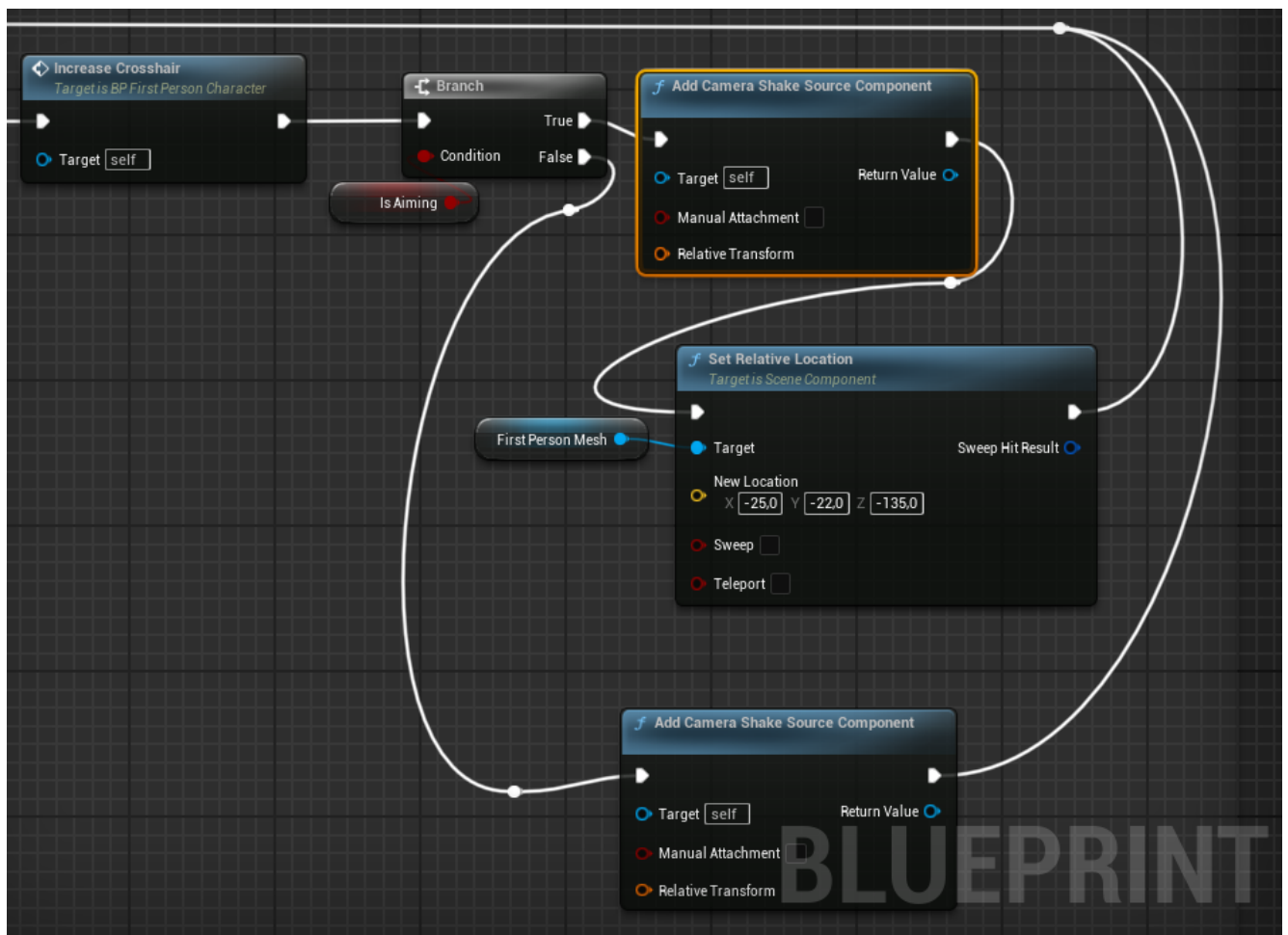


Рисунок 4.7 – Налаштування тряски камери при прицілюванні та без

Скріншот демонструє систему динамічної тряски камери, яка активується при стрільбі або бігу. Це дозволяє підсилити ефект присутності і зробити гру більш напруженою.

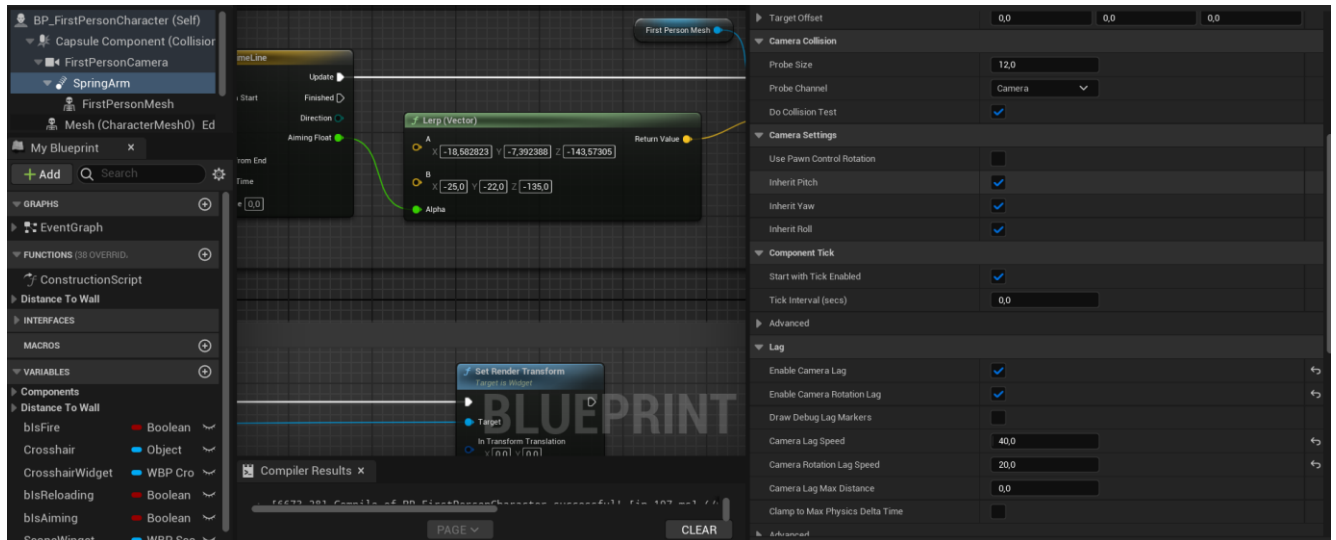


Рисунок 4.8 – Створення імітації Body Cam

Ілюстрація показує налаштування камери у стилі бодикамери – перспективи, що повторює реальні поліцейські чи військові записи. Такий підхід додає імерсивності.

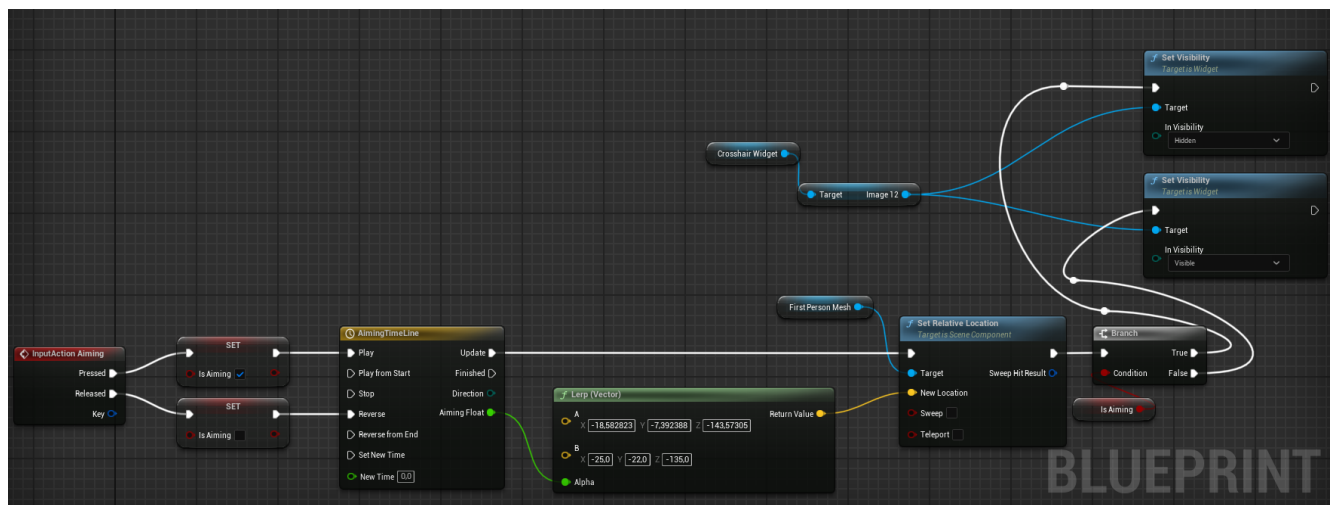


Рисунок 4.9 – Схема прицілювання

Зображено схему побудови траєкторії прицілу з урахуванням об'єктів на лінії пострілу. Система дозволяє уникнути стрільби "крізь" стіни та синхронізує приціл з візуальною точкою попадання. Коректна робота візуальних ефектів напряду впливає на геймплей та занурення у гру, тому потребує детального тестування.

4.4 Тестування звуку

Звуковий супровід — важлива складова, що додає атмосферності грі. У цьому розділі перевіряється активація звуків у відповідь на події в грі, затримки, відстань звучання та просторове розташування.

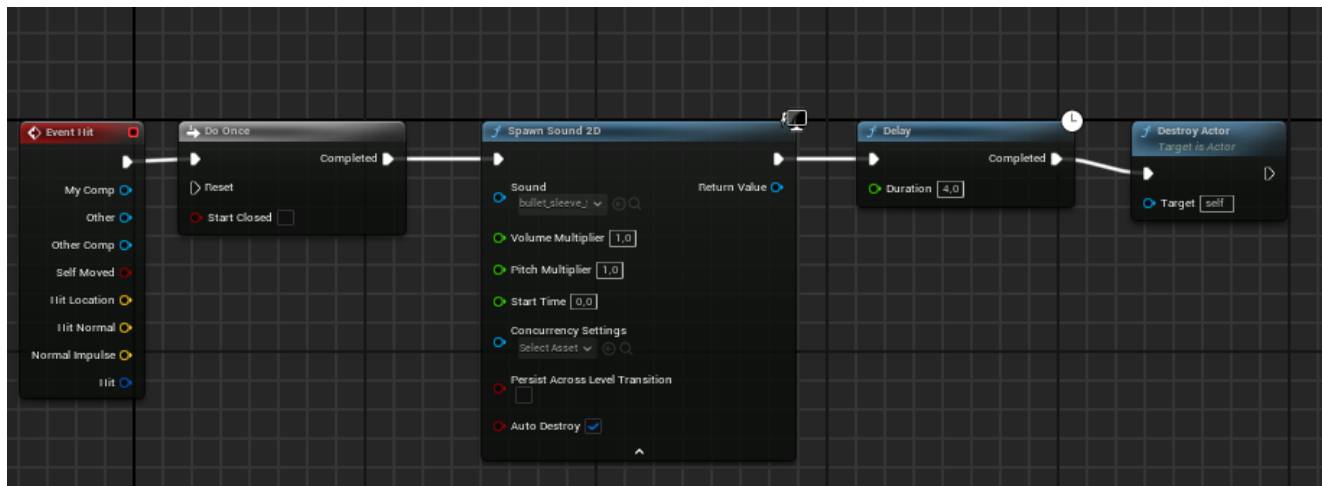


Рисунок 4.10 – Звук падіння та знищення гільз

Скріншот ілюструє реалізацію тригерів для звукових ефектів, що активуються під час падіння гільз на різні поверхні, враховуючи тип матеріалу та кількість об'єктів. Перевірка таких ефектів дозволяє уникнути зайвих звукових повторів, накладень або затримок, які можуть погіршити досвід гравця.

4.5 Функціональне тестування інтерфейсу

Інтерфейс користувача — основний засіб взаємодії гравця з ігровими механіками. Тут перевіряється відображення важливої інформації, коректність змін станів, інтуїтивність навігації.



Рисунок 4.10 – Загальний вигляд інтерфейсу

Зображення показує структуру HUD (індикатори здоров'я, боєприпасів, цілей тощо), а також взаємодію з меню. Тестується адаптивність до різних роздільностей, зручність розташування та реакція на вхідні дані. Цей етап критично важливий для забезпечення позитивного користувацького досвіду.

4.6 Установка, запуск, мінімальні вимоги

Установка, запуск, мінімальні вимоги і склад застосунку "Симулювання польоту дрону з використанням Unreal Engine 5":

Установка: застосунок не вимагає встановлення, після компіляції він вже встановлений на ПК.

Запуск: подвійний клік на ярлику запуску застосунку.

Мінімальні вимоги:

- вимоги до технічного забезпечення - ОС: Windows 10;

- процесор - Intel Core i7-7700 і Ryzen 5 1400;
- оперативна пам'ять - 16 GB ОП;
- відеокарта - NVIDIA GeForce GTX 1070 і Radeon RX 580;
- DirectX - версії 10.0.

Впевнитися, що комп'ютер відповідає мінімальним вимогам перед запуском застосунку.

Висновки до розділу 4

У четвертому розділі було детально досліджено процес тестування програмного забезпечення для action-shooter гри з тривимірною графікою, розробленої на основі Unreal Engine. Особливу увагу приділено перевірці ключових компонентів: ігрової логіки, фізики та колізій, роботи камери, візуальних і звукових ефектів, а також користувацького інтерфейсу.

У межах підрозділів проаналізовано роботу основних бойових механік: система прицілювання, логіка нанесення шкоди, обробка колізій, а також реалістична симуляція фізичних процесів — таких як виліт гільз та взаємодія об'єктів із середовищем. Проведено тестування параметрів камери та візуальних ефектів, які формують естетику геймплею, включаючи імітацію бодікамери та налаштування тряски. Також проаналізовано реакції на звукові події та функціональність елементів інтерфейсу, що є ключовими для зручності взаємодії користувача з грою.

На основі проведених тестувань були виявлені і усунені недоліки, що впливали на цілісність ігрового досвіду, стабільність роботи застосунку та його сумісність із різними конфігураціями обладнання. Сформовано чітке уявлення про подальші кроки з оптимізації та фінального доопрацювання гри перед релізом.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи бакалавра було розроблено ігровий застосунок у жанрі action-shooter з використанням сучасних технологій тривимірного моделювання, що сприяє формуванню просторового мислення, реакції та координації, а також розвитку навичок програмування та проектування ігрових систем.

Для досягнення поставленої мети було вирішено наступні завдання:

- проаналізовано сучасний ринок комп'ютерних ігор;
- досліджено особливості жанру action-shooter;
- сформовано основні функціональні та нефункціональні вимоги до застосунку;
- створено 3D-моделі персонажів, зброї та ігрового середовища;
- розроблено анімації та інтерактивні елементи;
- розроблено елементи інтерфейсу користувача;
- виконано моделювання та проектування архітектури застосунку за допомогою UML-діаграм;
- реалізовано логіку гри та бойову систему на базі Unreal Engine 5;
- проведено тестування та оптимізацію прототипу.

Під час роботи було здійснено аналіз популярних ігор у відповідному жанрі для виявлення ключових механік, які мають найбільший вплив на залучення гравця. Було обґрунтовано вибір технологій та інструментів, зокрема використання рушія Unreal Engine 5 та 3D-редактора Blender, які забезпечують високу якість графіки та широкі можливості для інтерактивності.

У рамках моделювання застосунок були побудовані діаграми сценаріїв використання, станів, діяльностей і взаємодій, що дозволило системно описати поведінку гри та взаємозв'язки між її компонентами. Це дозволило на ранніх етапах виявити потенційні недоліки в логіці гри та усунути їх до початку реалізації.

Розробка включала створення ігрового рівня-прототипу, який демонструє основні елементи геймплею: навігацію по ігровому середовищу, взаємодію з об'єктами, використання зброї та бойові сценарії. Завдяки цьому користувач отримує базове уявлення про механіки гри та її потенціал.

Проведене тестування підтвердило працездатність ігрового прототипу. Були використані методи функціонального тестування, тестування продуктивності та ручне тестування, що дозволило виявити та усунути низку недоліків у логіці гри та продуктивності.

Отримані результати підтверджують доцільність використання обраних інструментів та методів для реалізації проєкту. Подальший розвиток може передбачати розширення контенту, створення багаторівневої системи, реалізацію мережевого режиму та вдосконалення візуальної складової гри.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Bhattacharjee, S. Shooting Game Using Unreal Engine [Електронний ресурс] // ResearchGate. URL: https://www.researchgate.net/publication/369667301_Shooting_Game_Using_Unreal_Engine (дата звернення: 09.06.2025).
2. C++ Interfaces Pt 2/3 – Interactable Class UE4 / Unreal Engine 4 C++ // YouTube. URL: <https://www.youtube.com/watch?v=18yoUybI0TU> (дата звернення: 10.04.2025).
3. Discuss the UE4 official integration // Quixel Help Center. URL: <https://help.quixel.com/hc/en-us/community/topics/360000090277-UE4> (дата звернення: 18.04.2025).
4. Introduction to C++ Programming in UE4 // Unreal Engine Documentation. URL: <https://docs.unrealengine.com/4.27/en-US/ProgrammingAndScripting/ProgrammingWithCPP/IntroductionToCPP/> (дата звернення: 05.03.2025).
5. Kaiser, R. C++ mit Visual Studio 2019. Berlin, Heidelberg : Springer Berlin Heidelberg, 2021. URL: <https://doi.org/10.1007/978-3-662-59476-6> (дата звернення: 05.04.2025).
6. Lee, J., Doran, J. P., Misra, N. Unreal Engine: Game Development from A to Z. Packt Publishing, 2016. 837 с.
7. Level Editor Modes // Unreal Engine Documentation. URL: <https://docs.unrealengine.com/4.27/en-US/BuildingWorlds/LevelEditor/Modes/> (дата звернення: 05.04.2025).
8. Player Input and Pawns // Unreal Engine Documentation. URL: <https://docs.unrealengine.com/5.0/en-US/quick-start-guide-to-player-input-in-unreal-engine-cpp/> (дата звернення: 05.03.2025).

9. Programming Quick Start // Unreal Engine Documentation. URL: <https://docs.unrealengine.com/4.27/en-US/ProgrammingAndScripting/ProgrammingWithCPP/CPPProgrammingQuickStart/> (дата звернення: 10.03.2025).
10. Schell, J. The Art of Game Design. Carnegie Mellon University, 2008. 518 с.
11. Sherif, W. Learning C++ by Creating Games with UE4. Packt Publishing, 2015. 342 с.
12. Tools and Editors // Unreal Engine Documentation. URL: <https://docs.unrealengine.com/4.27/en-US/Basics/ToolsAndEditors/> (дата звернення: 05.04.2025).
13. UE4 Dialogue System Part 1 // GitHub Blog. URL: https://jinyuliao.github.io/blog/html/2017/12/15/ue4_dialogue_system_part1.html (дата звернення: 06.04.2025).
14. Unreal Engine 5 Multiplayer C++ Inventory and Shop #1: "Setup Character and Interactions" // YouTube. URL: <https://youtu.be/gl2sRHit2KM> (дата звернення: 06.04.2025).
15. Unreal Engine Import Export Tutorial – Blender to UE4 // YouTube. URL: <https://www.youtube.com/watch?v=yPVPyfcgG2g> (дата звернення: 18.04.2025).