

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Системотехніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

рівень вищої освіти другий (магістерський)

Розробка прототипу VR-гри на Unity
(тема)

Виконав:

студент II курсу, групи СПРМ-22-1

Бурим В.І.

(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки

(код і повна назва спеціальності)

Тип програми Освітньо-наукова

(освітньо-професійна або освітньо-наукова)

Освітня програма Системне проектування

(повна назва освітньої програми)

Керівник доц. Петрова Р. В.

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри системотехніки Гребеннік І.В.

(підпис)

(прізвище, ініціали)

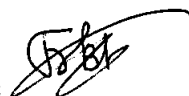
2024 р.

Я, як студент ХНУРЕ розумію і підтримую політику закладу із академічної доброчесності. Я не надавав і не одержував недозволену допомогу під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

Дата 15.06.24

Бурим В. І.

Підпис



Кваліфікаційна робота не містить відомостей заборонених до відкритого опублікування.

Кваліфікаційна робота виконана у відповідності до стандартів, що діють в Україні.

Попередній захист проведено.

Керівник кваліфікаційної роботи

доц. Петрова Р. В.

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____
Кафедра _____ Системотехніки _____
Рівень вищої освіти _____ другий(магістерський) _____
Спеціальність _____ 122 – «Комп'ютерні науки» _____
(код і повна назва)
Освітня програма _____ Системне проектування _____

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«__» _____ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові _____ Буриму Владиславу Ігоровичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи: Розробка прототипу VR-гри на Unity
затверджена наказом по університету від 01.04 2024 р. № 259СТ
2. Термін подання студентом роботи до екзаменаційної комісії 15.06.2024р
3. Вихідні дані до роботи: Дослідити засоби розробки ігор віртуальної та змішаної реальності. Розробити прототип VR-гри. Використовувати ОС Windows, середовище розробки Unity Engine та JetBrains Rider, мову програмування C#.
4. Перелік питань, що потрібно опрацювати в роботі 4.1 Вступ, 4.2 Аналіз предметної області, 4.2.1 Аналіз предметної області розроблювальної системи, 4.2.2 Аналіз реалізованих ігор-аналогів, 4.2.3 Постановка задачі та визначення сфери застосування розроблювальної системи, 4.3 Засоби реалізації mixed reality гри, 4.3.1 Огляд платформи Unity для розробки VR/AR ігор, 4.3.2 Аналіз технологій для створення імерсивного досвіду в прототипі, 4.3.3 Розробка системних вимог, 4.4 Створення концепції та вимог до гри, 4.4.1 Визначення концепції гри, 4.4.2 Визначення ігрового процесу та інтерактивних елементів, 4.4.3 Формулювання функціональних та нефункціональних вимог прототипу, 4.5 Програмна реалізація прототипу, 4.5.1 Використання зовнішніх SDK або API, 4.5.2 Реалізація основних ігрових механік, 4.5.3 Розробка інтерфейсу з урахуванням особливостей VR, 4.6 Висновки
5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) 5.1.1 Популярні mixed reality ігри на сайті Meta. 5.1.2 Популярні mixed reality ігри на сайті Side Quest. 5.2.1 Інтерфейс редактора Unity. 5.2.2 Unity Asset Store. 5.2.3 Пристрої Oculus. 5.2.4 Специфічні вимоги Meta Quest Link. 5.2.5 Відеокarti NVIDIA, що підтримують Meta Quest Link. 5.2.6 Відеокarti AMD, що підтримують Meta Quest Link. 5.4.1 Інструмент Package Manager. 5.4.2 Кімната в змішаній реальності на Meta Quest 2. 5.4.3 Об'єкти з яких з яких складається візуальне представлення кімнати. 5.4.4 Візуальне представлення кімнати в режимі

тестування. 5.4.5 Налаштування компоненту OVR Camera Rig. 5.4.6 Налаштування компоненту OVR Scene Manager. 5.4.7 Налаштування параметрів навігації. 5.4.8 Налаштування параметрів компоненту NavMesh Agent. 5.4.9 Згенерована навігаційна сітка кімнати. 5.4.10 Точки спавну ігрових об'єктів. 5.4.11 Міхато персонажі та анімації. 5.4.12 Модель звичайного ворога. 5.4.13 Модель головного персонажа. 5.4.14 Модель вежі. 5.4.15 Налаштування стартових характеристик ворога. 5.4.16 Налаштування стартових характеристик героя. 5.4.17 Налаштування стартових характеристик штурмової гвинтівки. 5.4.18 Налаштування стартових характеристик вежі. 5.4.19 Анімації персонажа. 5.4.20 Вікно налаштувань переходів між анімаціями у героя. 5.4.21 Вікно переходів між анімаціями у ворога. 5.4.22 Приклад налаштувань арени. 5.4.23 Налаштувань списку інтерфейсних вікон. 5.4.24 Інтерфейс головного меню. 5.4.25 Вікно налаштувань. 5.4.26 Меню паузи. 5.4.27 Поточне здоров'я головних об'єктів гри. 5.4.28 Вікно вибору непостійного покращення. 5.4.29 Вікно програшу.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування Розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
<i>Основна частина</i>	<i>доц. Петрова Р. В.</i>		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів	Примітка
1	Отримання завдання	13.01.24	Виконано
2	Аналіз предметної області та аналогів	13.01.2024-15.03.2024	Виконано
3	Вибір засобів для реалізації гри	15.03.2024-01.04.2024	Виконано
4	Розробка системних вимог	02.04.2024-08.04.2024	Виконано
5	Розробка концепції та вимог до гри	9.04.2024-12.04.2024	Виконано
6	Програмна реалізація прототипу гри	13.04.2024-06.05.2024	Виконано
7	Тестування застосунку	06.05.2024-09.05.2024	Виконано
8	Оформлення пояснювальної записки	10.05.2024-17.05.2024	Виконано
9	Оформлення програмної документації та додатків	18.05.2024-20.05.2024	Виконано
10	Представлення на рецензування	22.05.2024	Виконано
11	Представлення кваліфікаційної роботи до ЕК	15.06.2024	Виконано

Дата видачі завдання 13.01 2024р.

Студент  Бурим В. І.
(підпис)

Керівник роботи доц. Петрова Р. В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Кваліфікаційна робота містить 75 с., 37 рис., 1 додаток, 26 джерело інформації.

ІГРОВА МЕХАНІКА, ІМЕРСИВНІСТЬ, ІНТЕРАКТИВНІСТЬ, ПРОТОТИП VR-ГРИ, ШУТЕР З ВИДОМ ЗВЕРХУ, C#, MIXED REALITY, UNITY, VR

Об'єктом розробки є шутер з видом зверху у віртуальній реальності, що інтегрує елементи Mixed Reality, дозволяючи гравцю взаємодіяти з віртуальним середовищем у межах реального фізичного простору.

Предметом дослідження є використання технологій і інструментів Unity для створення прототипу VR-гри, яка розширює грані традиційних ігор, впроваджуючи можливості MR для створення унікального ігрового досвіду.

Мета дослідження – розробка прототипу, яка б поєднувала ігрову механіку класичних шутерів з перевагами імерсивності та інтерактивністю реального середовища, відкриваючи нові перспективи для розвитку ігрової індустрії.

Методи дослідження – аналіз предметної області та існуючих аналогів, визначення концепції, проектування та розробка прототипу гри за допомогою інструментів Unity та C#, та тестування створеного прототипу з метою валідації ігрових концепцій та механік.

Результатом роботи є розроблений прототип гри, що демонструє основні концепції геймплею, а також потенціал застосування технологій Mixed Reality у розробці ігор.

Практичне застосування розробки полягає у можливості використання отриманих результатів як основу для подальшої розробки повноцінної гри, що має потенціал стати новаторським продуктом на ринку VR-ігор

ABSTRACT

Master's Thesis: 75 pages, 37 figures, 1 appendix, 26 title.

C#, GAME MECHANICS, IMMERSIVITY, INTERACTIVITY, MIXED REALITY, TOP VIEW SHOOTER, UNITY, VR, VR GAME PROTOTYPE

The object of development is a top-down shooter in virtual reality that integrates elements of Mixed Reality, allowing the player to interact with the virtual environment within the bounds of the real physical space.

The subject of the research is the use of Unity technologies and tools to create a VR game prototype that expands the boundaries of traditional games by incorporating MR capabilities to create a unique gaming experience.

The goal of the research is to develop a prototype that combines the game mechanics of classic shooters with the advantages of immersion and interactivity of the real environment, opening new perspectives for the development of the gaming industry.

The research methods include analysis of the subject area and existing analogs, defining the concept, designing and developing the game prototype using Unity and C# tools, and testing the created prototype to validate game concepts and mechanics.

The result of the work is a developed game prototype that demonstrates the main gameplay concepts as well as the potential of applying Mixed Reality technologies in game development.

The practical application of the development lies in the possibility of using the obtained results as a basis for further development of a full-fledged game, which has the potential to become an innovative product in the VR game market

ЗМІСТ

ВСТУП.....	9
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	10
1.1 Аналіз предметної області розроблювальної системи	10
1.2 Аналіз реалізованих ігор-аналогів.....	11
1.3 Постановка задачі та визначення сфери застосування розроблювальної системи.....	14
2 ЗАСОБИ РЕАЛІЗАЦІЇ MIXED REALITY ГРИ	16
2.1 Огляд платформи Unity для розробки VR/AR ігор	16
2.2 Аналіз технологій для створення імерсивного досвіду в прототипі	19
2.3 Розробка системних вимог	21
3 СТВОРЕННЯ КОНЦЕПЦІЇ ТА ВИМОГ ДО ГРИ	28
3.1 Визначення концепції гри	28
3.2 Визначення ігрового процесу та інтерактивних елементів	30
3.3 Формулювання функціональних та нефункціональних вимог прототипу	32
4 ПРОГРАМНА РЕАЛІЗАЦІЯ ПРОТОТИПУ	35
4.1 Використання зовнішніх SDK або API	35
4.2 Реалізація основних ігрових механік	40
4.3 Розробка інтерфейсу з урахуванням особливостей VR	61
4.4 Тестування прототипу в Mixed Reality середовищі.....	69
ВИСНОВКИ.....	72
ПЕРЕЛІК ПОСИЛАНЬ	73

ПЕРЕЛІК УМОВНИХ ПОЗНАЧОК ТА СКОРОЧЕНЬ

Ігрова механіка – набір правил і способів, який реалізує певним чином деяку частину інтерактивної взаємодії гравця і гри

AI – Artificial Intelligence, штучний інтелект

API – Application Programming Interface, програмний інтерфейс застосунку

AR – Augmented Reality, доповнена реальність

HMD – Head-Mounted Display, шолом віртуальної реальності

MR – Mixed Reality, змішана реальність

SDK – Software Development Kit, набір для розробки програмного забезпечення

TDS – Top-Down Shooter, шутер з видом зверху

UI – User Interface, інтерфейс користувача

Unity – Unity Engine, багатоплатформовий інструмент для розробки відеоігор і застосунків, і рушій, на якому вони працюють

UX – User Experience, досвід взаємодії користувача

VR – Virtual reality, віртуальна реальність

ВСТУП

Віртуальна реальність за останні роки зробила революційний стрибок від науково-фантастичної концепції до доступної технології, що знайшла застосування у різноманітних сферах життя - від ігрової індустрії до освіти та медицини. Розвиток VR технологій відкрив нові можливості для розробників ігор, пропонуючи небачені досі імерсивні враження. В цьому контексті актуальним стає дослідження і розробка нових ігрових концепцій, які б використовували потенціал VR на повну.

Ця кваліфікаційна робота присвячена розробці прототипу VR-гри з використанням інструментарію Unity - одного з лідерів серед движків для створення ігор. Unity не лише забезпечує широкі можливості для реалізації ігрових проєктів, але й відкриває перед розробниками простір для експериментів з VR.

Метою дослідження є створення прототипу VR-гри, яка відкриває нові перспективи для ігрової механіки та інтерактивності в середовищі віртуальної реальності, а також оцінка можливостей Unity у контексті розробки VR-проєктів.

Об'єктом розробки виступає VR-гра, створена у Unity, а предметом дослідження є застосування специфічних технологій та інструментів цієї платформи для реалізації ідеї проєкту.

Задля досягнення визначеної мети, у роботі будуть використані методи дослідження, які охоплюють аналіз існуючих прикладів VR-ігор, розробку ігрового дизайну та механік, програмування за допомогою мови C# у середовищі Unity, та наступне тестування прототипу.

Практична значущість полягає у розширенні існуючих підходів до розробки VR-ігор, демонструючи потенціал застосування змішаної реальності для створення глибоких та захоплюючих ігрових вражень.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз предметної області розроблювальної системи

У наш час, коли цифрові інновації розвиваються стрімкими темпами, концепції віртуальної реальності та змішаної реальності набувають особливої актуальності та починають відігравати роль передових технологій. Ці технології значно розширюють можливості взаємодії користувачів з цифровим контентом, надаючи їм змогу не тільки спостерігати, але й активно брати участь у цифровому світі. Зокрема, використання віртуальної реальності надає користувачам унікальну можливість поринути в повністю вигадані, детально пророблені віртуальні світи, які є відокремленими від реального світу. Водночас, змішана реальність пропонує ще один напрямок взаємодії, дозволяючи злиття елементів реального світу з віртуальними об'єктами, що у свою чергу створює абсолютно новий, невиданий досвід для користувачів. Такі передові технології відкривають безпрецедентні перспективи для геймдизайнерів та розробників програмного забезпечення, особливо у сфері ігрової індустрії, де створення імерсивних віртуальних середовищ, де забезпечення глибокої взаємодії користувачів з ігровим контентом мають вирішальне значення.

Коли мова йде про застосування технологій віртуальної реальності [10] в рамках ігрової індустрії [6], особливо важливим стає усвідомлення того, що переважна більшість розробок на сьогодні зосереджена на створенні підвищеної імерсивності шляхом імплементації перспективи першої особи. Проте, цей конкретний проект прагне розширити традиційні межі використання VR, пропонуючи зовсім інший підхід - розробку шутера з видом зверху. Такий нетрадиційний для VR метод розгляду ігрового процесу уможливорює інтеграцію елементів змішаної реальності, роблячи інтеракцію з ігровим світом ще більш

насиченою та змістовною. Застосування технологій MR в такому контексті не просто відкриває нові горизонти для ігрової інтерактивності, але й створює унікальні умови, де гравці мають змогу вступати у фізичну взаємодію з компонентами як віртуального, так і фізичного світів, надаючи глибоке та багатогранне враження від гри.

Однією з найважливіших переваг, яку надають технології віртуальної та змішаної реальності, є здатність до створення надзвичайно імерсивних ігрових вражень, що кардинально відрізняються від того, до чого звикли користувачі традиційних відеоігор. Такі ігри прагнуть надати можливість відчувати та дослідити унікальні сенсорні досвіди та забезпечити небувалий рівень залученості. Ця особливість відкриває перед ігровими розробниками ряд нових викликів, а також надає їм широкі можливості для творчості у створенні ігрового контенту, який би в повній мірі розкривав потенціал VR та MR технологій. Для досягнення цієї мети вимагається особливий підхід як до розробки ігрових механік, так і до концепції сюжетів, щоб забезпечити гравцям захоплюючий ігровий досвід.

1.2 Аналіз реалізованих ігор-аналогів

Детальний аналіз [1] ігор, які вже були реалізовані в рамках сфер віртуальної та змішаної реальності, підкреслив один цікавий аспект: на ринку існує лише обмежена кількість ігор із видом зверху, що активно використовують елементи змішаної реальності. Цей факт не зменшує інтерес до такого типу ігор, а, навпаки, вказує на значний потенціал для інновацій [21], який ще не був повністю реалізований у даній області. Більшість сучасних ігор у VR зосереджуються на створенні поглибленого імерсивного досвіду за допомогою першої особи, тоді як досвід, який пропонується MR, часто обмежується лише доповненням реального середовища віртуальними об'єктами без залучення гравця у складний інтерактивний процес. Проте, на основі аналізу ігор можна зробити висновок, що

використання технологій MR відкриває широкі можливості для створення взаємопов'язаних та динамічних ігрових середовищ, які могли б значно розширити глибину геймплею. Особливо це стосується можливості інтеграції активних фізичних взаємодій гравців із ігровим простором, що відкриває нові горизонти для розвитку ігрового дизайну та взаємодії в іграх.

Досліджуючи ринок mixed reality ігор на таких сайтах як Meta [2] (рисунок 1.1) та Side Quest [7] (рисунок 1.2), стало очевидно, що інтеграція виду зверху у віртуальних реальності іграх відкриває перед гравцями надзвичайні можливості для проведення стратегічного аналізу та управління ігровим процесом, надаючи їм цілковитий контроль над ігровою ареною. Ця перспектива не лише спонукає до розвитку новаторських ігрових механік і способів взаємодії з ігровим світом, але також активізує креативне мислення серед гравців, відкриваючи перед ними широкий спектр стратегій для досягнення успіху у грі. Однак, разом з новими можливостями виникають і певні виклики, зокрема, потреба в інтеграції якісних анімацій персонажа та ефективної взаємодії як з реальним, так і з віртуальним просторами. Це вимагає від розробників глибокого занурення у нюанси розробки інтерфейсу та оптимізації користувацького досвіду, щоб забезпечити залученість гравців та інтуїтивне розуміння геймплею, що є ключовими елементами для створення успішного ігрового процесу.

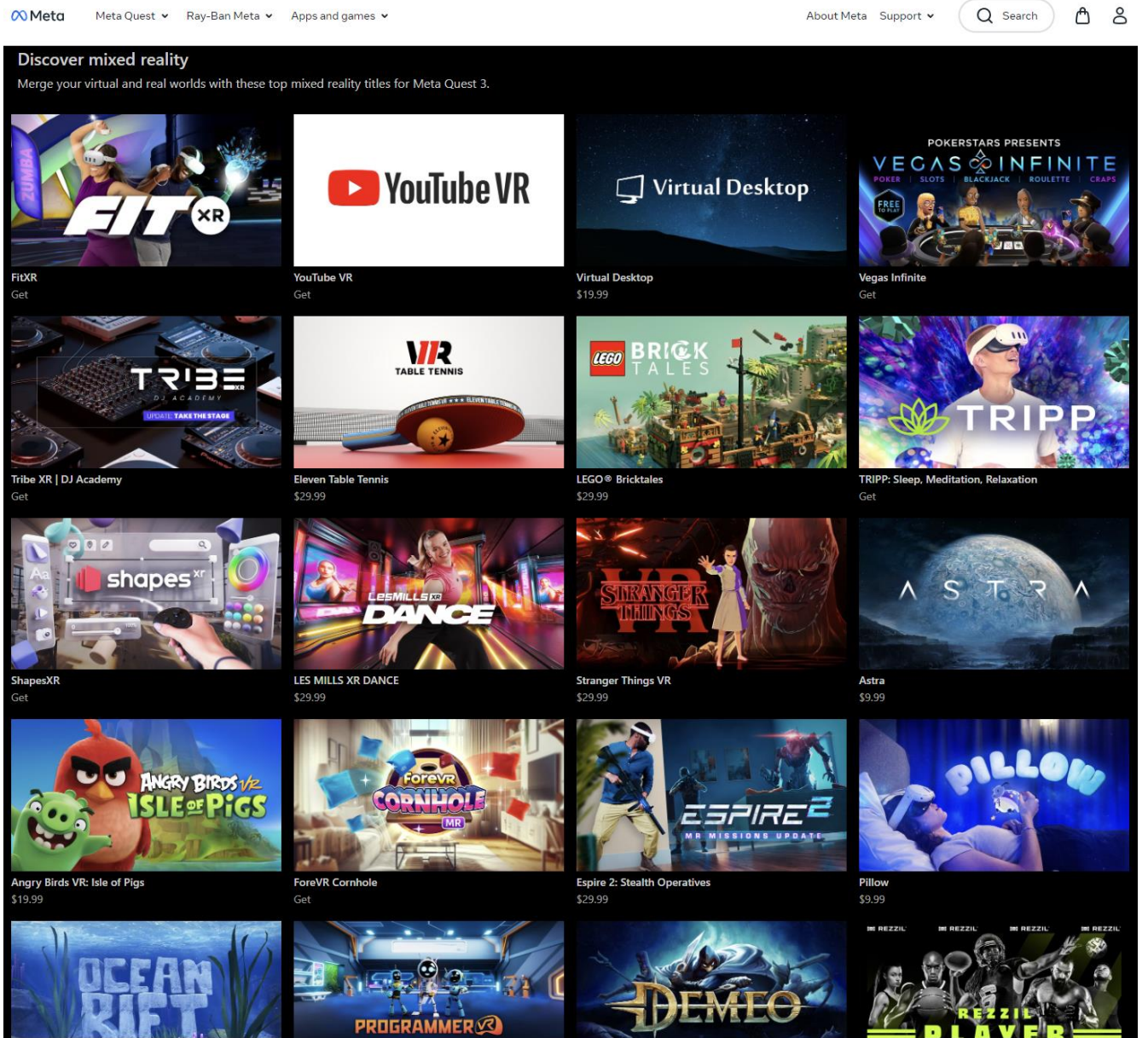


Рисунок 1.1 – Популярні mixed reality ігри на сайті Meta

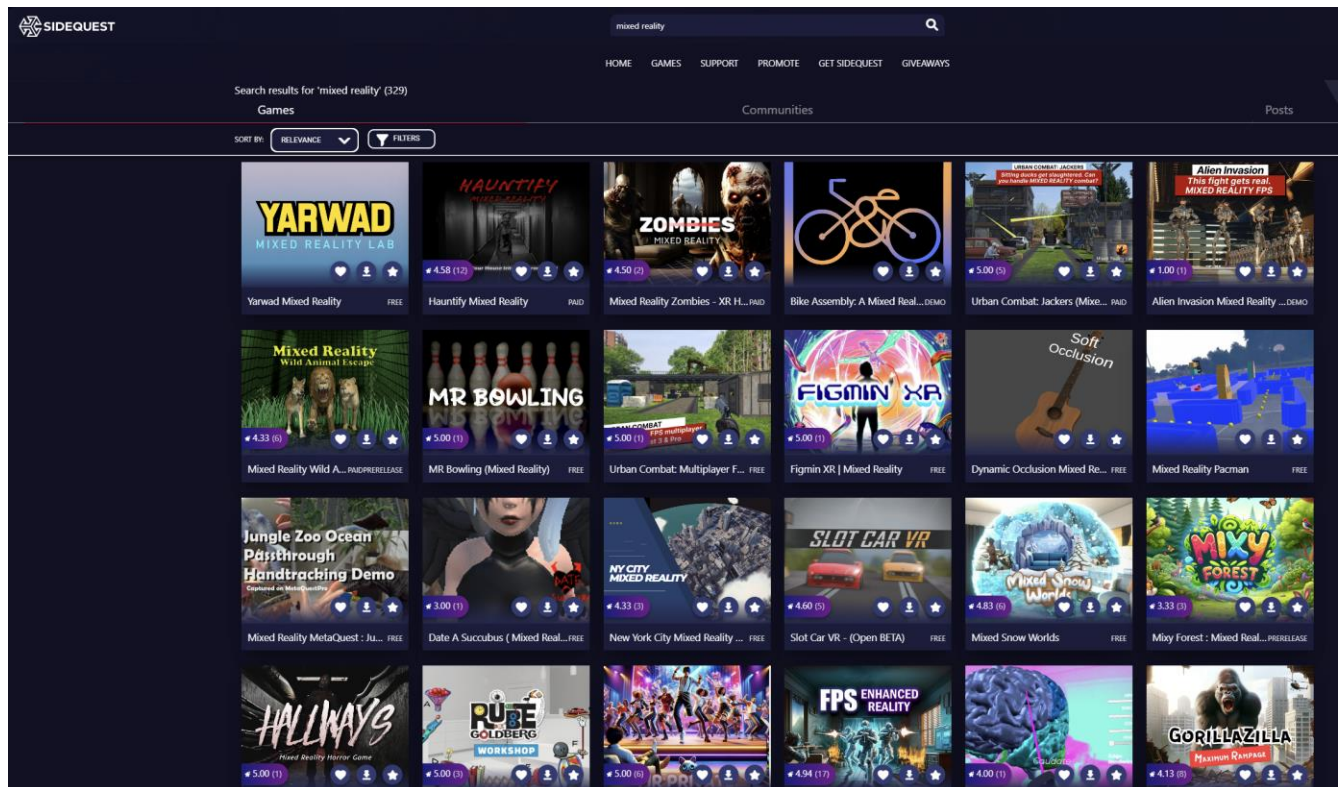


Рисунок 1.2 – Популярні mixed reality ігри на сайті Side Quest

1.3 Постановка задачі та визначення сфери застосування розроблювальної системи

Мета дослідження полягає в розробці прототипу шутера з видом зверху для VR-середовища з інтеграцією елементів змішаної реальності (MR), що забезпечить високий рівень взаємодії та занурення в ігровий процес.

Завдання дослідження включають:

- Аналіз сучасних технологій віртуальної та змішаної реальності для виявлення їхніх можливостей і обмежень.
- Дослідження існуючих ігрових продуктів у жанрі шутер з видом зверху для визначення кращих практик ігрового дизайну.
- Розробка концепції гри, яка поєднує елементи VR і MR для створення новаторського ігрового досвіду.

г) Створення прототипу гри на базі рушія Unity з використанням його інструментів для VR і MR.

г) Оптимізація ігрового процесу для забезпечення плавності і реалістичності взаємодії у VR.

д) Проведення тестування прототипу для оцінки його ефективності та зручності використання.

Об'єкт дослідження – ігрова індустрія, зокрема розробка VR і MR ігор, що сприяє покращенню взаємодії та занурення гравців в ігровий процес.

Методи дослідження:

а) Аналітичний метод - для вивчення наукової літератури, статей та досліджень з теми VR і MR технологій.

б) Компаративний метод - для порівняння різних підходів до розробки VR ігор та їхніх інтеграцій з MR.

в) Експериментальний метод - для перевірки розробленого прототипу в умовах реального використання.

г) Метод моделювання - для створення прототипу гри на базі Unity.

г) Статистичний метод - для аналізу результатів тестувань.

д) Метод експертної оцінки - для оцінки якості розробленого прототипу та виявлення можливих напрямків його покращення.

Розроблений прототип шутера з видом зверху, який інтегрує VR і MR елементи, що може бути впроваджений у ігрову індустрію.

2 ЗАСОБИ РЕАЛІЗАЦІЇ MIXED REALITY ГРИ

2.1 Огляд платформи Unity для розробки VR/AR ігор

Unity Engine [3] встановлює стандарти в індустрії розробки ігор, вирізняючись своєю універсальністю та спроможністю підтримувати великий діапазон сучасних технологій, зокрема технології віртуальної та доповненої реальності. Завдяки своїм вдосконаленим можливостям як платформи для створення ігор з елементами змішаної реальності, Unity надає розробникам доступ до величезного набору інструментів та функцій для реалізації їх найамбітніших інноваційних ідей та творчих концепцій.

Unity відзначається вбудованою підтримкою важливих платформ віртуальної та доповненої реальності, включаючи Oculus і SteamVR для VR та ARKit від Apple, ARCore від Google для AR. Ця особливість забезпечує, що проекти, розроблені на Unity, можуть без зусиль адаптуватися та оптимізуватися для використання на різноманітному обладнанні, що робить їх доступними для широкого кола користувачів і значно розширює аудиторію проектів.

Рушій забезпечує розробників широким спектром інструментів для творення ігрових світів, пропонуючи від інтуїтивно зрозумілого перетягування об'єктів у редакторі (рисунок 2.1) до розробки складних анімаційних секвенцій та написання скриптів. Редактор сцен в Unity дозволяє розробникам в деталях візуалізувати та налаштовувати кожен компонент ігрового світу в реальному часі, забезпечуючи високу ступінь контролю над процесом творення. Додатково, вдосконалена система частинок та потужний фізичний движок дозволяють втілити в життя реалістичні ефекти та динамічні взаємодії всередині ігрового світу, збагачуючи геймплей глибокими та переконливими деталями.

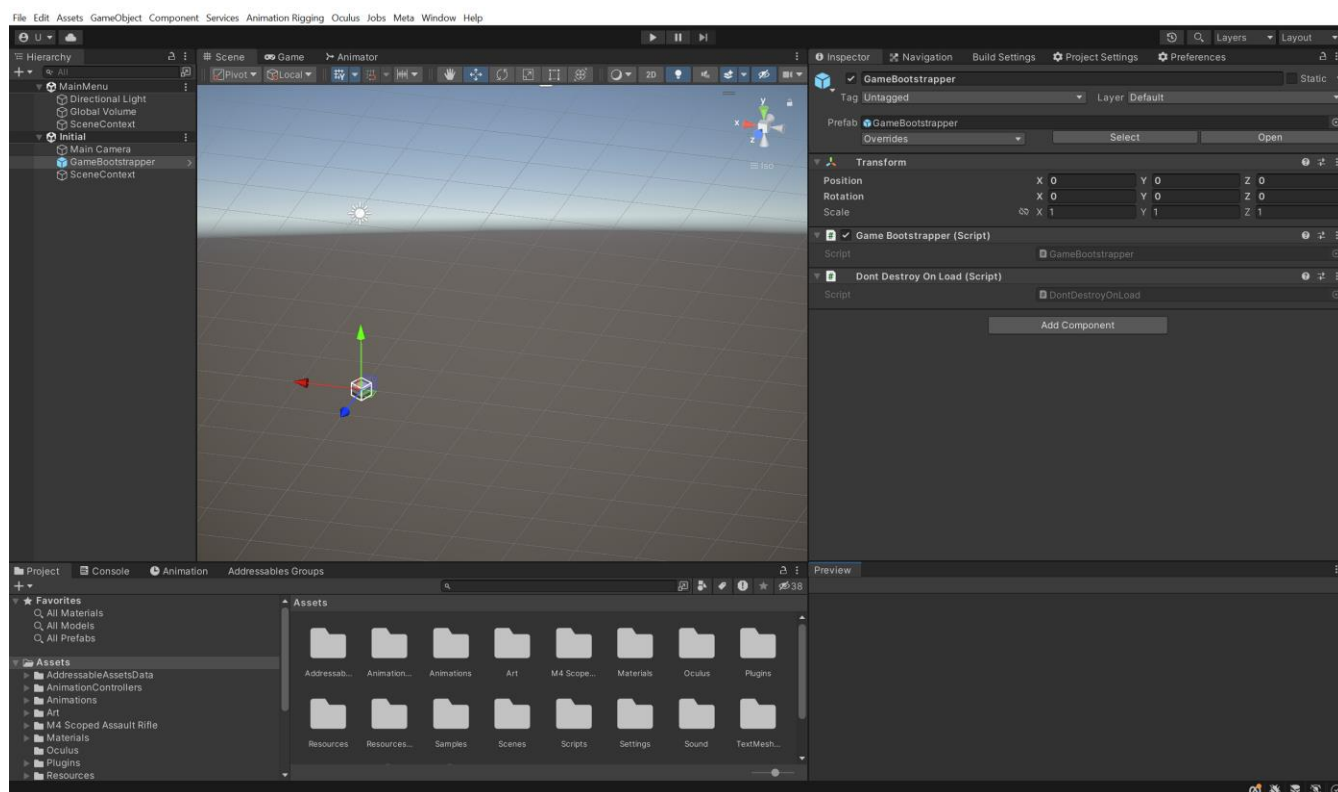


Рисунок 2.1 – Інтерфейс редактора Unity

В Unity програмування здійснюється через використання мови C# [18], що надає розробникам величезний арсенал можливостей для створення складних ігрових механізмів і логіки. Це охоплює все від розробки систем управління персонажами до створення складного штучного інтелекту для супротивників. Unity надає програмістам можливість гнучко налаштовувати і кастомізувати поведінку ігрових елементів, дозволяючи їм втілювати унікальні ідеї та концепції, які забезпечують гравцям неповторний і захопливий ігровий досвід.

Unity значно спрощує процес тестування та налагодження ігор для віртуальної та доповненої реальності, надаючи розробникам змогу швидко застосовувати зміни та миттєво оцінювати результати у безпосередньому VR- або AR-середовищі. Такий підхід не тільки забезпечує вищу якість фінального продукту, але й сприяє глибокій оптимізації проектів під різноманітні типи

обладнання, відкриваючи шлях до ефективного вдосконалення ігрового досвіду для різних платформ та пристроїв.

Unity Asset Store (рисунок 2.2) слугує незамінним ресурсом для прискорення процесу розробки, надаючи розробникам доступ до тисяч готових до використання активів. Це охоплює широкий спектр ресурсів, таких як 3D моделі, анімації, звукові ефекти, скрипти, і навіть цілісні шаблони для ігор, які дозволяють розробникам економити час та зусилля, які інакше були б витрачені на створення цих елементів з нуля. Використання таких ресурсів може суттєво підвищити ефективність робочого процесу, дозволяючи командам зосередитись на інноваціях та унікальних аспектах своїх ігрових проєктів, забезпечуючи при цьому високу якість та професійність фінального продукту.

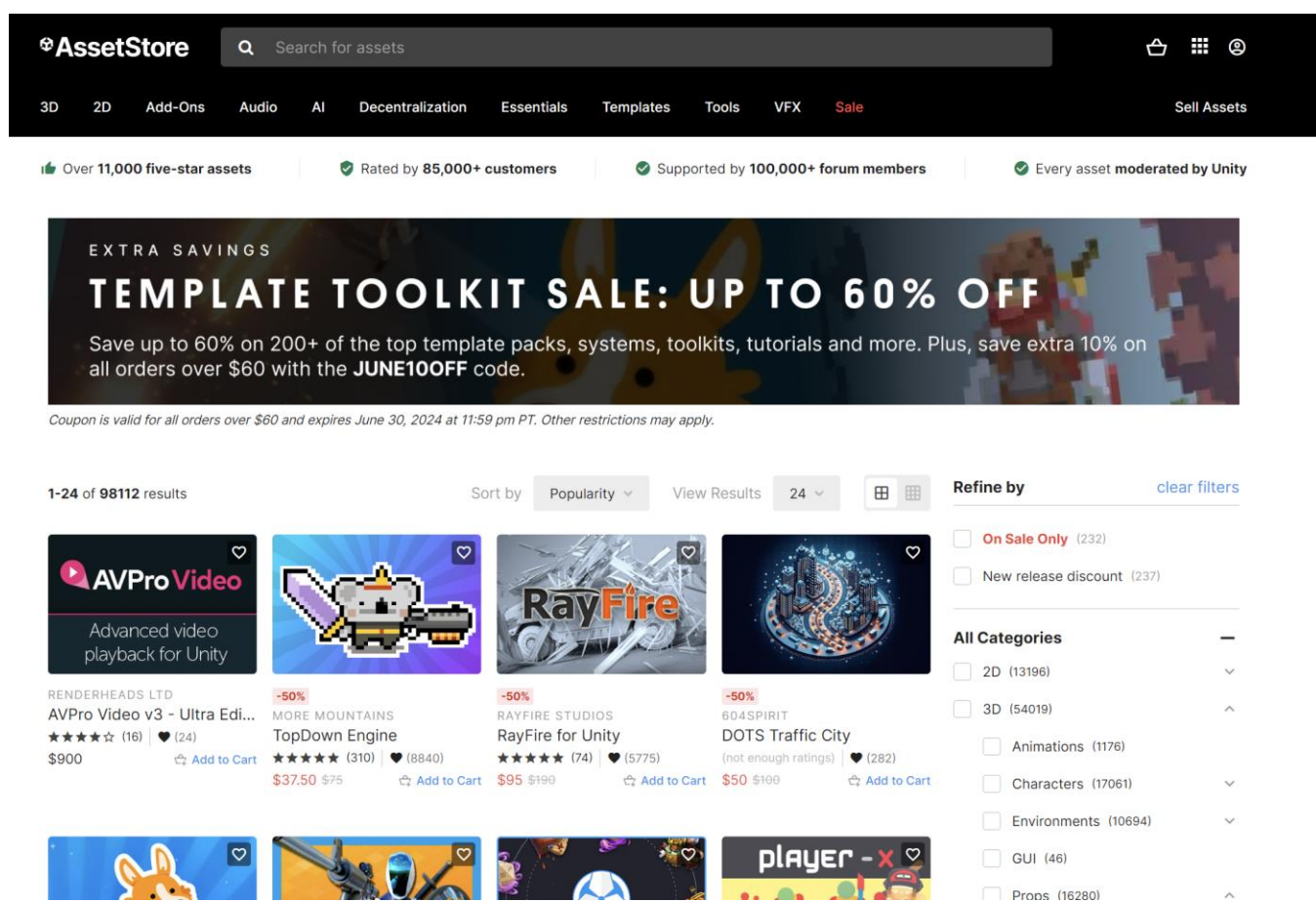


Рисунок 2.2 – Unity Asset Store

Unity вирізняється своїм великим та активним співтовариством розробників, яке створює неперевершену екосистему, що включає обширний арсенал навчальних матеріалів [11], готові до використання активи та форуми [13] для обміну знаннями та досвідом [14]. Така підтримка сприяє не тільки швидкому навчанню та вдосконаленню професійних навичок розробників на всіх рівнях досвіду, але й надає безпосередній доступ до готових рішень і технічних знахідок, які можуть бути легко використані або адаптовані для власних ігрових проєктів. Це, у свою чергу, стимулює інновації та співпрацю серед розробників, відкриваючи нові можливості для креативного та технічного зростання у сфері розробки ігор.

Враховуючи всі ці переваги, Unity стає ідеальним рішенням для реалізації проєкту. Його гнучкість та комплексна підтримка обладнання для віртуальної та доповненої реальності, а також потужні візуальні та програмні інструменти відкривають безмежні можливості для створення передових інноваційних ігрових досвідів, особливо у сфері змішаної реальності. Unity не лише сприяє втіленню творчих задумів у життя, але й гарантує високу якість і оптимізацію кінцевого продукту, роблячи його вибором номер один для розробників, що прагнуть перетворити свої ідеї в реальність.

2.2 Аналіз технологій для створення імерсивного досвіду в прототипі

Розробка прототипу гри з використанням технологій Mixed Reality передбачає комбінування різних передових технологій для створення глибоко імерсивного ігрового досвіду, що надзвичайно залучає гравців. Основні технології, що становлять фундамент імерсивного досвіду, охоплюють трекінг руху, реалістичне відтворення звуку, візуальні ефекти та оптимізацію інтерфейсу користувача.

Відмінність MR-ігор від класичних відеоігор полягає у необхідності реалізації високоточного трекінгу рухів гравця, включно з позиціонуванням голови,

рук, а іноді і цілого тіла. Цей аспект критично важливий для створення враження повного занурення, адже він дозволяє користувачам взаємодіяти з віртуальним середовищем так, нібито вони дійсно перебувають усередині нього. Сучасні технології трекінгу, як-от оптичний та інерційний, грають ключову роль у цьому процесі, забезпечуючи детальне відображення рухів гравця у віртуальному світі, що значно підсилює реалістичність взаємодії.

Звуковий дизайн є ще одним невід'ємним елементом занурення в ігри змішаної реальності. Застосування просторового аудіо дозволяє створювати ілюзію реальних звуків, що відтворюються з певних напрямків і на різних відстанях, імітуючи їх поширення у фізичному світі. Це досягається за допомогою складних алгоритмів просторового аудіо, які підсилюють відчуття присутності користувача в іншому світі, збагачуючи віртуальне середовище глибиною та реалістичністю перцепції звукового оточення.

Для досягнення реалістичної візуалізації у MR-іграх, використовуються передові технології графічного рендерингу. Ключові елементи такого процесу включають динамічне освітлення, яке надає глибину та реалізм сценам, високоякісні текстури, що детально відображають реальні поверхні, та реалістичну симуляцію матеріалів, яка змушує віртуальні об'єкти виглядати невідірзаними від справжніх. Однак, при цьому розробники стикаються з необхідністю знайти оптимальний баланс між візуальною привабливістю та продуктивністю системи, щоб гарантувати безперебійну гру без втрати якості навіть на пристроях із обмеженими ресурсами. Вирішення цього завдання дозволяє створити захоплюючі ігрові світи, що забезпечують глибоке занурення та високу продуктивність, адаптовану до широкого спектра апаратних конфігурацій.

В іграх, що використовують технології змішаної реальності, інтерфейс користувача відіграє критично важливу роль, оскільки має бути не тільки інтуїтивно зрозумілим, але й спроможним ефективно адаптуватися до унікальних умов та особливостей взаємодії, які притаманні віртуальному середовищу. Розробка

користувацького інтерфейсу, який буде легко читатися та використовуватися під час гри, вимагає від розробників використання специфічних дизайнерських рішень. Це може включати створення інтерфейсу, що гармонійно інтегрується у віртуальний світ, забезпечуючи легкий доступ до ігрових елементів управління без необхідності відволікатися від ігрового процесу. Ключовими аспектами ефективного UI у таких іграх є чіткість візуального представлення, логічність розташування елементів керування, та здатність інтерфейсу швидко реагувати на дії гравця, забезпечуючи таким чином зручність та ефективність управління.

Ефективне використання технологій, що лежать в основі ігор змішаної реальності, ставить перед розробниками завдання не тільки технічного характеру, але й вимагає глибокого розуміння потреб та вподобань кінцевих користувачів. Правильна інтеграція та налаштування кожного з цих компонентів є вирішальною для створення високоякісного імерсивного MR-досвіду.

2.3 Розробка системних вимог

Вибір Oculus Quest 2, Oculus Quest Pro, та Oculus Quest 3 (рисунок 2.3) як цільових платформ для розробки гри обумовлений декількома важливими факторами. Насамперед, ці пристрої є передовими представниками автономних систем віртуальної реальності, що надають розробникам широкі можливості для створення ігор. Вони забезпечують високу роздільну здатність зображення, що є критично важливим для забезпечення реалістичної візуалізації та глибокого занурення в ігровий світ. Точний трекінг рухів дозволяє гравцям вільно переміщатися та взаємодіяти з віртуальним середовищем, що підвищує імерсивність досвіду. Крім того, використання автономних VR-систем виключає потребу в зовнішніх сенсорах чи додатковому обладнанні, такому як потужний персональний комп'ютер, роблячи ігровий досвід більш доступним та зручним для широкого кола користувачів. Таким чином, вибір цих моделей Oculus Quest як

основних платформ для розробки не тільки сприяє створенню якісних ігрових проєктів, але й відкриває потенціал досягнення великої аудиторії завдяки легкості та зручності користування цими пристроями.



Рисунок 2.3 – Пристрої Oculus

Вибір моделей Oculus Quest як основних платформ для розробки гри також обґрунтований їх широкою популярністю та поширеністю серед користувачів віртуальної реальності. Ця серія пристроїв зарекомендувала себе як одна з лідерів на ринку VR завдяки своїй доступності, високій якості зображення, і зручності використання. Велика база активних користувачів цих пристроїв відкриває перед розробниками значний ринок, забезпечуючи доступ до широкої аудиторії, зацікавленої в нових ігрових досвідах.

Для розробки гри, оптимізованої для платформ Oculus Quest 2, Pro та 3, необхідно встановити вичерпні системні вимоги для розробників.

Для платформ Oculus Quest 2, Pro та 3, необхідно підключатися через Meta Quest Link. І розробникам необхідно враховувати системні вимоги (рисунок 2.4), визначені Meta [26], які включають мінімальні та рекомендовані характеристики ПК, такі як процесор, графічна карта, оперативна пам'ять та порти.

Windows PC requirements

The following table lists the current Windows PC minimum requirements for Link.

Component	Minimum Specifications
Processor	Intel i5-4590 / AMD Ryzen 5 1500X or greater
Graphics Card	See GPU tables below
Memory	8 GB+ RAM
Operating System	Windows 10, Windows 11
USB Ports	1x USB port

The following table lists the current Windows PC recommended requirements for Link.

Component	Recommended Specifications
Processor	Intel i7 / AMD Ryzen 7
Graphics Card	Nvidia RTX 20 Series* / AMD Radeon RX 6000 Series
Memory	16 GB DDR4 RAM
Operating System	Windows 10, Windows 11
USB Ports	1x USB-C port

Note: NVIDIA GeForce RTX 2050 is not supported

Рисунок 2.4 – Специфічні вимоги Meta Quest Link

Також треба звернути увагу, що не всі відеокарти (рисунок 2.5 та 2.6) підтримують Meta Quest Link і коректно працюють.

Supported GPUs for Link

NVIDIA GPU	Supported	Not Currently Supported
NVIDIA Titan Z		X
NVIDIA Titan X	X	
NVIDIA GeForce GTX 970	X	
NVIDIA GeForce GTX 1060 Desktop, 3GB		X
NVIDIA GeForce GTX 1060 Desktop, 6GB	X	
NVIDIA GeForce GTX 1060M		X
NVIDIA GeForce GTX 1070(all)	X	
NVIDIA GeForce GTX 1080(all)	X	
NVIDIA GeForce GTX 1650		X
NVIDIA GeForce GTX 1650 Super	X	
NVIDIA GeForce GTX 1660	X	
NVIDIA GeForce GTX 1660 TI	X	
NVIDIA GeForce RTX 20-series (all except 2050)	X	
NVIDIA GeForce RTX 30-series*	X	
NVIDIA GeForce RTX 40-series	X	

*Please Note: NVIDIA 3050 (laptop) and 3050ti GPUs are not recommended for use with Link.

Рисунок 2.5 – Відеокарти NVIDIA, що підтримують Meta Quest Link

AMD GPU	Supported	Not Currently Supported
AMD 200 Series		X
AMD 300 Series		X
AMD 400 Series	X	
AMD 500 Series	X	
AMD 5000 Series	X	
AMD 6000 Series**	X	
AMD Vega Series	X	

****Please note: the Radeon RX 6500 is not recommended for use with Link.**

Рисунок 2.6 – Відеокарти AMD, що підтримують Meta Quest Link

За цими даними можна виділити такі системні вимоги для розробників:

- а) Операційна система: Розробка вимагає від розробників використання ПК з операційною системою, сумісною з останньою версією Oculus Integration SDK / Meta XR All-in-one SDK. Рекомендовані операційні системи включають Windows 10 або новішу;
- б) Процесор та графічний процесор: Потужний процесор і графічний процесор, сумісні з VR-вимогами, необхідні для розробки та тестування гри. Рекомендується використання процесора Intel i5-4590 / AMD Ryzen 5

1500X або краще та графічного процесора, еквівалентного NVIDIA GTX 1060 / AMD Radeon RX 480 та краще;

в) Оперативна пам'ять: Мінімум 8 ГБ RAM, проте 16 ГБ або більше рекомендується для оптимальної продуктивності;

г) Вільний простір на диску: Мінімум 256 ГБ вільного місця на SSD для ефективної розробки та швидкого запуску проектів.

При розробці ігрового прототипу, оптимізованого для Oculus Quest 2, Oculus Quest Pro, та Oculus Quest 3, встановлення чітких системних вимог є важливим не тільки для розробників, але й для кінцевих користувачів. Ці вимоги повинні охоплювати всі аспекти, які впливають на ігровий процес, і забезпечувати, щоб гравці мали відповідне обладнання для найкращого ігрового досвіду.

Системні вимоги для користувачів:

а) VR-гарнітура: Очевидно, користувачі повинні мати одну з підтримуваних VR-гарнітур: Oculus Quest 2, Oculus Quest Pro, або Oculus Quest 3. Ці пристрої забезпечують стандартизовану платформу для ігрового досвіду;

б) Місце для гри: Для комфортного користування та безпеки під час гри у VR, користувачам рекомендується мати достатньо вільного простору. Мінімально рекомендований простір становить 2x2 метри, щоб забезпечити безпечну взаємодію з ігровим середовищем без ризику для фізичного здоров'я;

в) Оновлення програмного забезпечення: Користувачі повинні встановити останнє оновлення програмного забезпечення для своїх Oculus Quest пристроїв, щоб забезпечити сумісність з іграми та оптимальну продуктивність;

г) Інтернет-з'єднання: Хоча багато ігор та досвідів на Oculus Quest можуть бути доступні офлайн, стабільне інтернет-з'єднання необхідне для завантаження ігор та отримання оновлень;

г) Вимоги до заряду: Користувачі повинні забезпечити, що їхні гарнітури повністю заряджені перед використанням, щоб мінімізувати перерви у грі через необхідність перезарядки.

Встановлення цих системних вимог забезпечить стабільний розвиток проекту та допоможе уникнути потенційних технічних проблем, забезпечуючи безперервний і якісний ігровий досвід на Oculus Quest платформах для користувачів.

3 СТВОРЕННЯ КОНЦЕПЦІЇ ТА ВИМОГ ДО ГРИ

3.1 Визначення концепції гри

У центрі концепції нашої гри знаходиться захоплююча ідея: гравець грає за робота-захисника, чиє основне завдання полягає в захисті вежі від постійних атак ворожих сил [5]. Ця концепція комбінує елементи жанрів тавер дефенс та шутера з видом зверху, пропонуючи унікальний ігровий досвід. Гравцям пропонується не лише активно застосовувати зброю проти наближаючихся ворогів, але й залучатися до стратегічного планування оборони, вибудовування тактик для вдалого захисту. Такий підхід забезпечує глибину геймплею, змушуючи гравців мислити наперед і адаптуватися до змінюваних умов бою, тим самим підвищуючи загальну захопленість та залученість у процес гри.

У серці динаміки нашої гри лежить взаємодія між двома ключовими фігурами: роботом-захисником та оборонною вежею [8]. Кожен з цих елементів володіє своїми унікальними характеристиками. Гравець, керуючи роботом, має забезпечити ефективний захист, розумно розподіляючи свої ресурси між міцністю вежі та своїми захисними здібностями як робота-захисника. Унікальною особливістю гри є те, що у випадку знищення робота він може бути відроджений, проте під час процесу відродження вежа залишається беззахисною перед наступом ворогів. Це додає додаткову напругу та підкреслює важливість балансу між агресивними діями проти ворогів та стратегічним управлінням ресурсами для захисту ключових об'єктів. Така механіка змушує гравців не тільки активно атакувати, але й приділяти увагу захисту та відновленню своїх сил.

Вороги, що намагаються заволодіти вежею та перемогти робота-захисника, відрізняються між собою різноманітністю характеристик та бойових стратегій. Деякі противники виявляють особливу агресивність щодо робота, намагаючись

Його знешкодити та вивести з ладу, тоді як інші цілеспрямовано рухаються до вежі, аби нанести їй максимальну шкоду. Цей спектр тактик і викликів ставить перед гравцем завдання, що вимагають не лише блискавичної реакції та оперативності в прийнятті рішень, але й вміння глибоко аналізувати ситуацію на полі бою, розробляти витончені стратегії оборони та швидко адаптуватися до кожного нового виклику, який кидає противник [5]. Від гравця вимагається не тільки майстерність управління своїм персонажем і захисту вежі, але й здатність передбачати рухи ворога, ефективно розподіляти ресурси та вибудовувати тактику, яка дозволить вистояти під натиском різноманітних загроз.

Після завершення кожної хвили ворогів гравець стикається з вибором трьох можливих покращень [19]. Ці удосконалення можуть кардинально вплинути на динаміку гри, надаючи можливість збільшення швидкості руху або атаки, збільшення шкоди по ворогам, скорочення часу перезарядки зброї, підвищення рівня здоров'я тощо. Такий механізм вибору дозволяє гравцям індивідуалізувати свій ігровий процес, розробляючи власну тактику та стратегію захисту відповідно до унікальних викликів, які постають перед ними.

Після кожної завершеної гри гравці отримують монети, які можуть бути використані на покращення постійних характеристик робота-захисника або вежі [17]. Ці удосконалення охоплюють широкий спектр аспектів, від посилення оборонних здібностей до збільшення атакуючого потенціалу, що дозволяє гравцям кастомізувати свій ігровий досвід відповідно до власних стратегій та стилів гри.

Особливим моментом у грі є досягнення певної кількості хвиль ворогів, що веде до зіткнення з фінальним босом [9]. Цей бос має унікальний набір механік та викликів, що ставить перед гравцями виклик не тільки вижити, але й змушує їх максимально використати стратегічний потенціал набутих покращень. Його унікальний набір механік та викликів вимагає від гравців передбачливості та глибокого планування вибору покращень протягом усієї гри. Перемога над цим

потужним ворогом відкриє доступ до нових арен зі свіжими викликами та більш складними противниками.

Такий підхід забезпечує постійне відчуття прогресу та вдосконалення, водночас підтримуючи інтерес та захопленість гравців через непередбачуваність і різноманітність ігрового процесу. Гравці мають можливість експериментувати з різними стратегіями та тактиками, адаптуючись до нових викликів та ворогів, що робить кожну гру унікальною та захоплюючою.

Ця гра забезпечує поєднання стратегічного планування та динамічної бойової дії, пропонуючи гравцям не лише боротьбу за виживання, але й можливість випробувати різні стратегії та тактики оборони. Ця гра пропонує нестандартний підхід до взаємодії з віртуальним світом, де кожна нова хвиля ворогів є не просто черговим викликом, а можливістю для гравця проявити креативність, аналітичне мислення та здатність швидко адаптуватися до еволюціонуючого середовища бою.

3.2 Визначення ігрового процесу та інтерактивних елементів

Ігровий процес побудовано на основі повторюваних циклів хвиль ворогів, які неуханно наступають на вежу та робота-захисника [20]. Кожна така хвиля є не просто черговим натиском, а унікальним викликом, що включає в себе різні типи ворогів із власними особливостями і тактикою [9]. Ця динаміка змушує гравця залишатися в постійній готовності, реагувати на мінливі умови бою, адаптуватися до нових загроз, та ефективно застосовувати наявний арсенал і розроблені стратегії для захисту.

Вежа стоїть у центрі ігрового процесу як найважливіший елемент, що вимагає непохитного захисту з боку гравця. Вона обладнана показниками здоров'я, і її руйнування прирівнюється до програшу в грі, підкреслюючи високу ставку кожного раунду оборони. Ворожі сили, що прагнуть знищити цей ключовий об'єкт,

використовують різноманітні тактики атаки, додаючи грі стратегічної глибини та тактичної різноманітності.

У грі гравець виступає у ролі робота-захисника, оснащеним зброєю для захисту не лише оборонної вежі, але й самого себе від атак ворогів. Цей робот здатен зазнавати пошкоджень від ворожих ударів, що додає ігровому процесу елемент ризику та необхідності обережності під час бою. У випадку знищення, робот не втрачає своєї бойової спроможності назавжди, а має змогу відродитися після певного проміжку часу.

Система покращень є ключовою складовою геймплею, пропонуючи гравцям значний стратегічний вибір після завершення кожної хвили ворогів. З трьох доступних опцій покращень, гравець може обрати ту, що найкраще відповідає його поточним потребам та стратегії гри.

За захист від хвиль ворогів гравці нагороджуються монетами, які можна використовувати після гри для постійних покращень робота або вежі. Ці покращення можуть стосуватися як робота-захисника, так і оборонної вежі, надаючи гравцям можливість удосконалити їхні оборонні або атакуючі характеристики. Це вносить важливий елемент стратегічного планування в ігровий процес, дозволяючи гравцям ретельно обирати, в які характеристики варто інвестувати зароблені ресурси прямо зараз.

Після проходження певної кількості хвиль відкриває доступ до боїв з босами, кожен з яких вимагає унікальної стратегії та тактики для перемоги. Перемога над босом нагороджує гравця монетами та відкриває нову арену для подальших свіжі виклики та випробування, що забезпечує відновлення інтересу до гри та спонукає до подальшого розвитку в грі.

У грі будуть компаньйони – це можуть бути милі тваринки або високотехнологічні дрони, що надають підтримку роботу-захиснику в його місії по обороні вежі. Ці супутники додають грі додатковий шар різноманітності та глибини.

Ігровий процес та інтерактивні елементи були ретельно розроблені, щоб забезпечити гравцям глибокий та змістовний ігровий досвід, який ставить перед ними складні завдання. Гравці зіткнуться з необхідністю постійно вирішувати тактичні та стратегічні проблеми, розвиваючи свої навички планування та адаптації до динамічно змінних умов на полі бою.

3.3 Формулювання функціональних та нефункціональних вимог прототипу

У ході проектування було чітко окреслено ключові функціональні та нефункціональні вимоги, що формують основу для ігрового досвіду, буде наданий гравцям на етапі прототипу.

Функціональні вимоги визначають спектр дій і можливостей, які має забезпечувати програмне забезпечення, детально описуючи ігровий процес та інтеракції з гравцем. Водночас, нефункціональні вимоги стосуються якості та характеристик системи, таких як продуктивність, надійність та сумісність, забезпечуючи високий рівень занурення та комфорту під час гри.

Для реалізації прототипу були виявлені такі вимоги:

а) Функціональні вимоги:

- 1) Взаємодія з VR-меню: Прототип повинен забезпечувати інтуїтивно зрозуміле VR-меню для легкого доступу до гри, характеристик гравця тощо;
- 2) Гравці повинні мати можливість вмикати та вимикати музичний супровід, редагувати гучність звуків, адаптуючи ігровий досвід під свої вподобання;
- 3) Гравці мають змогу покращувати або відновлювати непостійні характеристики свого персонажа та інших ігрових об'єктів;

4) Гравець повинен мати можливість проходити різні рівні гри. Це включає в себе структуру рівнів, яка дозволяє гравцю поступово підніматися в складності, відкривати нові виклики та досягати нових етапів гри. Кожен рівень має бути унікальним, з власними труднощами та ворогами, що забезпечують різноманітність ігрового процесу;

5) Керування персонажем з використанням VR-контролерів. Це включає інтуїтивно зрозуміле управління рухами персонажа, стрільбою та взаємодією з оточуючими об'єктами. Управління повинно бути точним та відповідати реальним рухам гравця, забезпечуючи високу ступінь занурення у віртуальний світ;

6) Персонажі гри повинні мати необхідні анімації;

7) Можливість ставити гру на паузу та перезапускати рівень. Гравець повинен мати доступ до цих функцій у будь-який момент гри, що дозволить йому контролювати процес гри, зупинити її у разі необхідності та повертатися до гри пізніше;

8) Реалізація інтуїтивно зрозумілої та відчутної бойової системи;

9) Необхідність слідкувати за характеристиками свого персонажа та інших ігрових об'єктів для успішної оборони та боротьби з ворогами.

б) Нефункціональні вимоги:

1) Проект має бути написаний за допомогою рушія Unity Engine та мови C# для VR-платформ;

2) Додаток має автоматично зберігати необхідні дані до реєстру, а також автоматично завантажувати їх під час запуску програми;

3) Прототип гри повинен коректно функціонувати на заданих системними вимогами приладах;

Обидва набори вимог взаємодіють, створюючи фундамент для розробки ігрового досвіду, що задовольняє як ігрові цілі, так і забезпечує високу якість взаємодії з користувачем.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ ПРОТОТИПУ

4.1 Використання зовнішніх SDK або API

Для реалізації прототипу гри я використовував різноманітні SDK та пакети, що значно полегшили процес розробки та додали нові функціональні можливості до гри. Основне завантаження різних пакетів та SDK в Unity здійснюється за допомогою Package Manager (рисунок 4.1). Package Manager є зручним інструментом для управління залежностями проекту, що дозволяє легко знаходити, завантажувати та оновлювати необхідні пакети безпосередньо в середовищі Unity. Це забезпечує доступ до останніх версій інструментів і бібліотек, спрощуючи інтеграцію нових можливостей у проект.

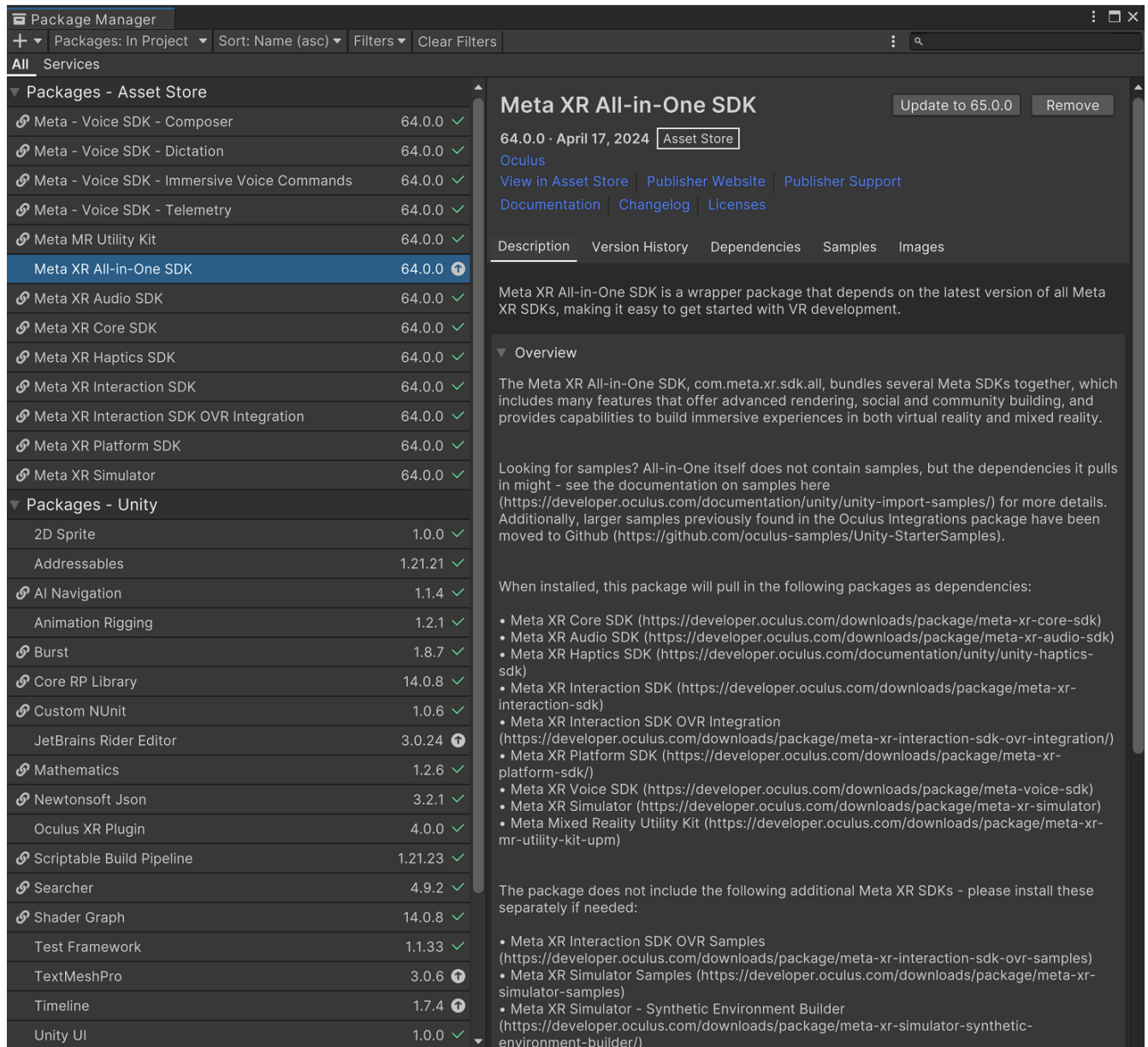


Рисунок 4.1 – Інструмент Package Manager

Нижче наводиться детальний опис використаних SDK та пакетів:

а) Oculus Integration SDK / Meta XR All-in-one SDK:

На початковому етапі розробки для реалізації віртуальної реальності я використовував Oculus Integration SDK. Цей SDK надає широкий спектр інструментів для роботи з VR-технологіями, включаючи підтримку трекінгу рухів, рендеринг віртуального середовища, взаємодію з VR-

контролерами та багато іншого. Пізніше Oculus Integration SDK був замінений на Meta XR All-in-one SDK [12], що є більш сучасним та універсальним рішенням для розробки VR-додатків. Meta XR All-in-one SDK включає в себе всі необхідні компоненти для створення VR-додатків, забезпечуючи підтримку пристроїв Meta (Oculus) та надаючи більш розширені можливості для інтеграції MR-елементів. Цей SDK значно спростив процес розробки, дозволивши зосередитися на створенні контенту та ігрових механік, замість вирішення технічних проблем з інтеграцією обладнання.

б) Zenject:

Zenject — це потужна бібліотека для впровадження залежностей у Unity, яка значно полегшила організацію коду та управління залежностями між компонентами гри. Використання Zenject дозволило створити більш чисту та зрозумілу архітектуру проекту, що полегшує подальше розширення та підтримку коду. За допомогою Zenject я зміг легко впровадити залежності між різними ігровими об'єктами та компонентами, зменшуючи кількість коду, що відповідає за створення та управління цими залежностями вручну. Це особливо корисно у великих проектах, де кількість взаємодій між об'єктами може бути значною.

в) 2D Sprite [22]:

2D Sprite package є важливим інструментом для створення та управління 2D-спрайтами в грі. Цей пакет надає широкий спектр можливостей для роботи з 2D-графікою, включаючи створення спрайтів, анімацій, управління їхніми властивостями та інтеграцію з іншими компонентами гри. Використання 2D Sprite package дозволило створити детальні та високоякісні візуальні елементи, які додають глибину та динаміку ігровому процесу. Зокрема, спрайти використовувалися для створення UI елементів.

г) Addressables [23]:

Addressables package є потужним інструментом для керування ресурсами в Unity. Він дозволяє завантажувати ресурси за адресами, що значно спрощує процес організації та оптимізації використання пам'яті у грі. Використання Addressables дозволило розподілити ресурси на різні групи, які можуть бути завантажені та вивантажені в залежності від потреб гри, що підвищило продуктивність та зменшило час завантаження рівнів. Крім того, цей пакет надає можливості для роботи з динамічним контентом, що дозволяє легко оновлювати гру, додаючи нові елементи без необхідності випускати нові версії програми.

г) AI Navigation [24]:

AI Navigation є ключовим компонентом для реалізації штучного інтелекту в грі, зокрема для навігації персонажів. Цей SDK надає інструменти для створення навігаційних сіток (NavMesh), шляхових точок та інших елементів, що допомагають персонажам пересуватися по ігровому світу. Використання AI Navigation дозволило створити реалістичну систему пересування ворогів, забезпечуючи їхню здатність знаходити оптимальні маршрути до цілей, уникати перешкод та реагувати на зміни у середовищі.

д) Animation Rigging [25]:

Animation Rigging є важливим пакетом для створення та керування анімаціями персонажів у Unity. Він дозволяє створювати складні анімаційні сценарії, включаючи інтерактивні та процедурні анімації, що додає реалізму рухам персонажів. Використання Animation Rigging дозволило створити та відредагувати реалістичні анімації для персонажів та ворогів, що включають базові рухи (ходьба, біг, стрибки) та бойові дії. Крім того, цей пакет дозволив легко інтегрувати анімації з іншими компонентами гри, забезпечуючи плавність та злагодженість ігрового процесу.

Також в ході розробки використовувалися декілька API, які забезпечували доступ до різних сервісів та функціональних можливостей. Хоча основний фокус був на використанні SDK та пакетів, API також відіграли важливу роль у забезпеченні інтеграції з зовнішніми системами та розширенні можливостей гри. Використовувалися:

а) Unity API:

Основне API, яке використовувалося у проекті, це API Unity. Воно надає доступ до всіх базових функцій та компонентів Unity, включаючи управління сценами, фізикою, анімацією, введенням користувача та багато іншого. Unity API було критично важливим для реалізації всіх аспектів гри, забезпечуючи інструменти для створення та керування ігровими об'єктами, взаємодії з користувачем та обробки ігрових подій. Завдяки Unity API вийшло легко інтегрувати різноманітні компоненти гри та забезпечити їхню злагоджену роботу.

б) Meta API:

Для взаємодії з пристроями Oculus (Meta) використовувалося Meta API. Це API надає доступ до функцій трекінгу рухів, рендерингу на VR-гарнітурах, керування контролерами та інших специфічних функцій для пристроїв Meta. Використання Meta API дозволило реалізувати всі необхідні функції для роботи гри на VR-пристроях, забезпечуючи високу якість трекінгу та взаємодії з користувачем. Це API також надало можливості для інтеграції з іншими компонентами гри, забезпечуючи злагоджену роботу всієї системи.

в) NavMesh API:

NavMesh API використовувалося для реалізації навігації персонажів у грі. Це API надає функції для створення та оновлення навігаційних сіток, а також для управління рухом персонажів по цих сітках. Використання NavMesh API дозволило створити реалістичну систему навігації, що забезпечує персонажам можливість знаходити оптимальні маршрути до

цілей, уникати перешкод та реагувати на зміни у середовищі. Це значно покращило ігровий процес, додаючи елемент стратегії та виклику для гравців.

Таким чином, використання зовнішніх SDK та API дозволило значно прискорити процес розробки та додати до гри багато функціональних можливостей, що забезпечують високий рівень інтерактивності та занурення у віртуальний світ.

4.2 Реалізація основних ігрових механік

Meta API надає широкий спектр інструментів і функцій для створення VR-додатків, і вивчення його можливостей стало ключовим кроком у реалізації прототипу. На початковому етапі основна увага була приділена вивченню принципів роботи з Meta API [4], що дозволило інтегрувати віртуальну взаємодію з реальною кімнатою (рисунок 4.2), у якій знаходиться гравець. Це включало створення візуального представлення кімнати (рисунок 4.4) у вигляді різнокольорових об'єктів (рисунок 4.3), що допомогло при тестуванні та налагодженні гри. Хочу зазначити що на Meta Quest 2 підтримується показ оточення лише в чорно-білих тонах, коли як Meta Quest 3 та Pro підтримується оточення в кольорі.



Рисунок 4.2 – Кімната в змішаній реальності на Meta Quest 2

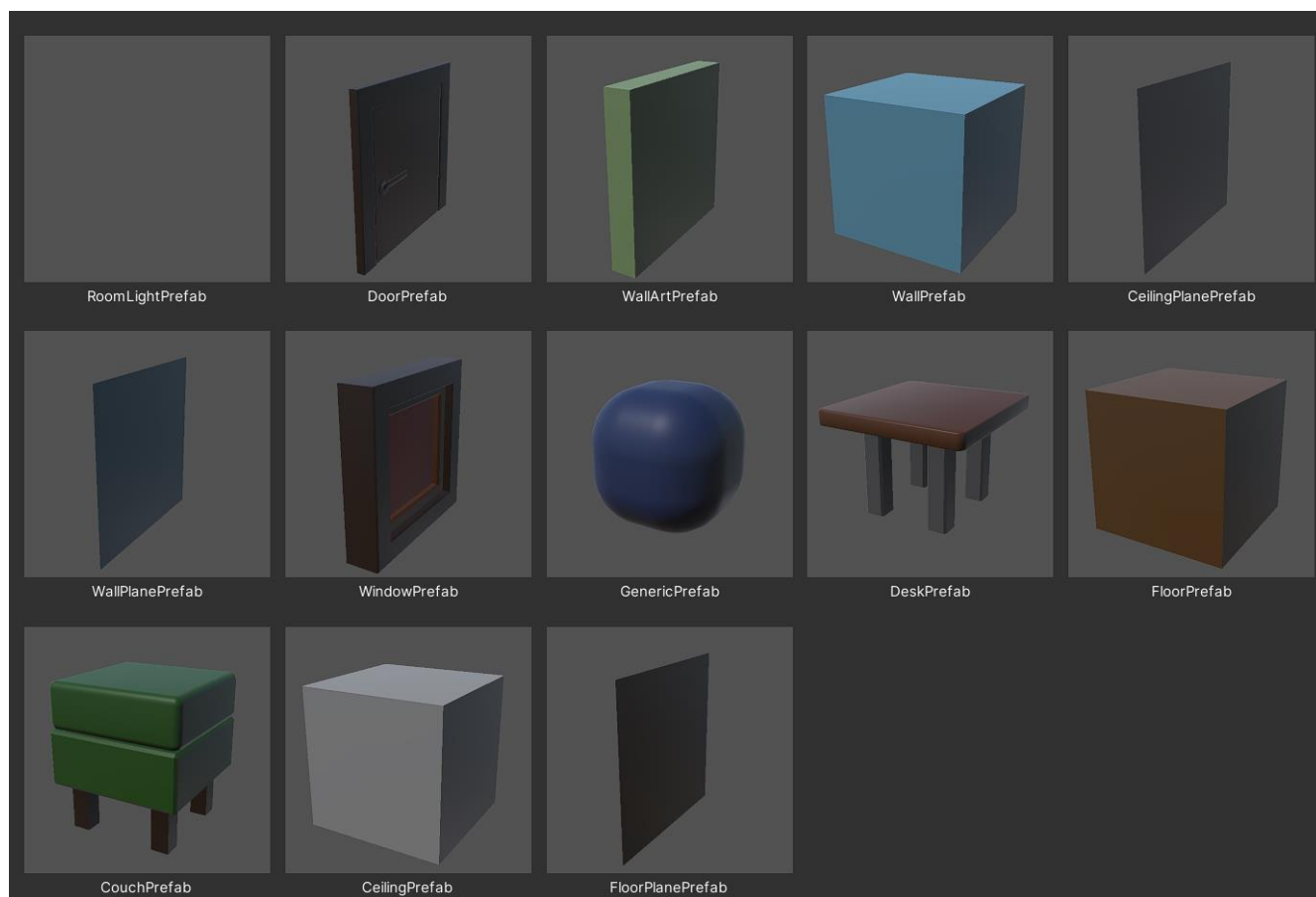


Рисунок 4.3 – Об’єкти з яких складається візуальне представлення кімнати



Рисунок 4.4 – Візуальне представлення кімнати в режимі тестування

Для реалізації віртуальної взаємодії з реальним оточенням використовувався компонент OVR Camera Rig (рисунок 4.5). OVR Camera Rig надає можливість відслідковувати положення та рухи гравця, забезпечуючи максимальну точність і реалістичність віртуального досвіду.

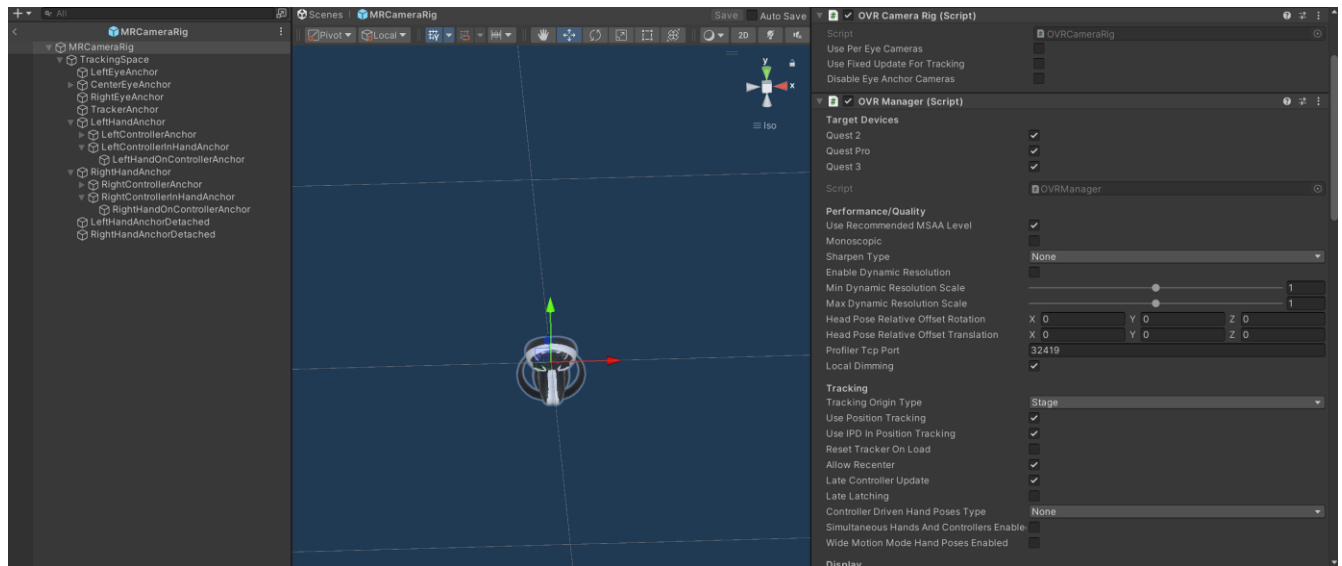


Рисунок 4.5 – Налаштування компоненту OVR Camera Rig

Крім того, завантаженням якорів кімнати займається компонент OVR Scene Manager (рисунок 4.6). OVR Scene Manager відповідає за зберігання та обробку даних про реальну кімнату, дозволяючи створювати віртуальні копії реального простору, з яким гравець взаємодіє. Це включає розміщення об'єктів у віртуальній кімнаті, що відповідають реальним об'єктам, і забезпечує точну інтеграцію між віртуальним і реальним світом.

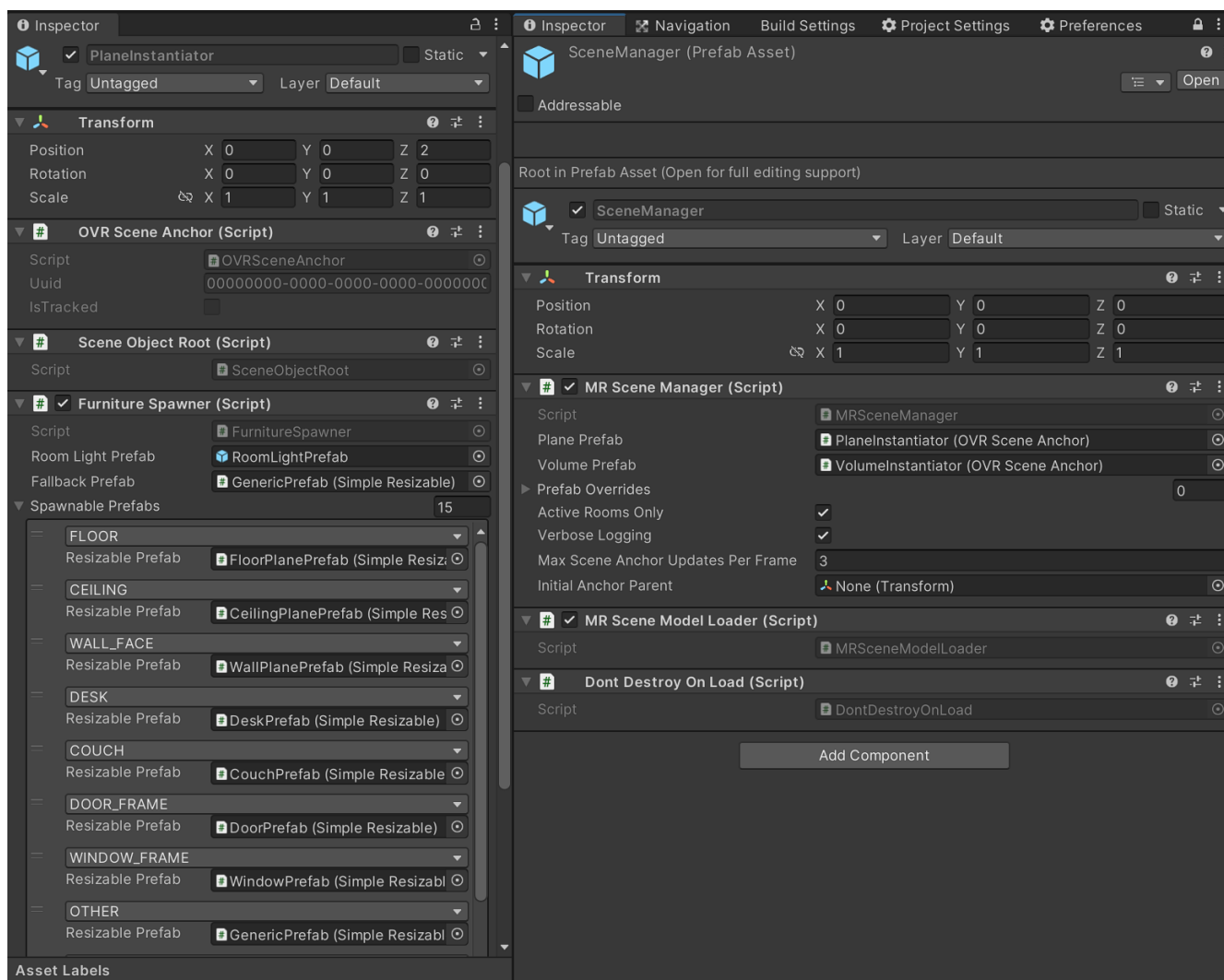


Рисунок 4.6 – Налаштування компоненту OVR Scene Manager

Наступним значущим кроком у нашій роботі була генерація NavMesh Surface прямо під час гри. NavMesh Surface є компонентом Unity, що використовується для створення навігаційної сітки (NavMesh) у 3D-просторі. Ця сітка дозволяє персонажам гри, наприклад, ворогам, пересуватися по складному ігровому середовищу, знаходити оптимальні маршрути та уникати перешкод. Зазвичай NavMesh налаштовується під конкретний створений рівень, але у нашому випадку необхідно було створювати її під час гри. Оскільки кожна кімната у кожного користувача може бути налаштована по-різному, важливо було забезпечити динамічну генерацію навігаційних сіток, що дозволяють ворогам знаходити

оптимальні маршрути для руху. Для цього використовувався NavMesh API, який надає інструменти для створення та оновлення навігаційних сіток у реальному часі.

Це дозволило адаптувати ігрове середовище до змін у конфігурації кімнати користувача та забезпечити безперебійну роботу алгоритмів навігації. Використовуючи NavMesh Surface, ми налаштували параметри генерації навігаційних сіток через вікно Navigation у Unity.

У вікні Navigation (рисунок 4.7) були налаштовані такі параметри, як агентська висота, радіус та крок, що забезпечило правильну обробку різноманітних поверхонь і перешкод у грі. Крім того, були встановлені настройки для визначення навігаційних областей, таких як прохідні та непрохідні поверхні.

Крім цього, для керування персонажами, що рухаються по NavMesh, використовувався компонент NavMesh Agent (рисунок 4.8). NavMesh Agent надає можливість контролювати рух персонажа, дозволяючи йому плавно пересуватися по навігаційній сітці, уникати перешкод та досягати заданих цілей. У вікні налаштувань NavMesh Agent були налаштовані такі параметри, як швидкість руху, радіус, висота агента, а також обробка перешкод. Це забезпечило природний і реалістичний рух ворогів у грі.

На рисунку 4.9 блакитним кольором зображено навігаційну сітку, якою можуть рухатися вороги і персонаж.

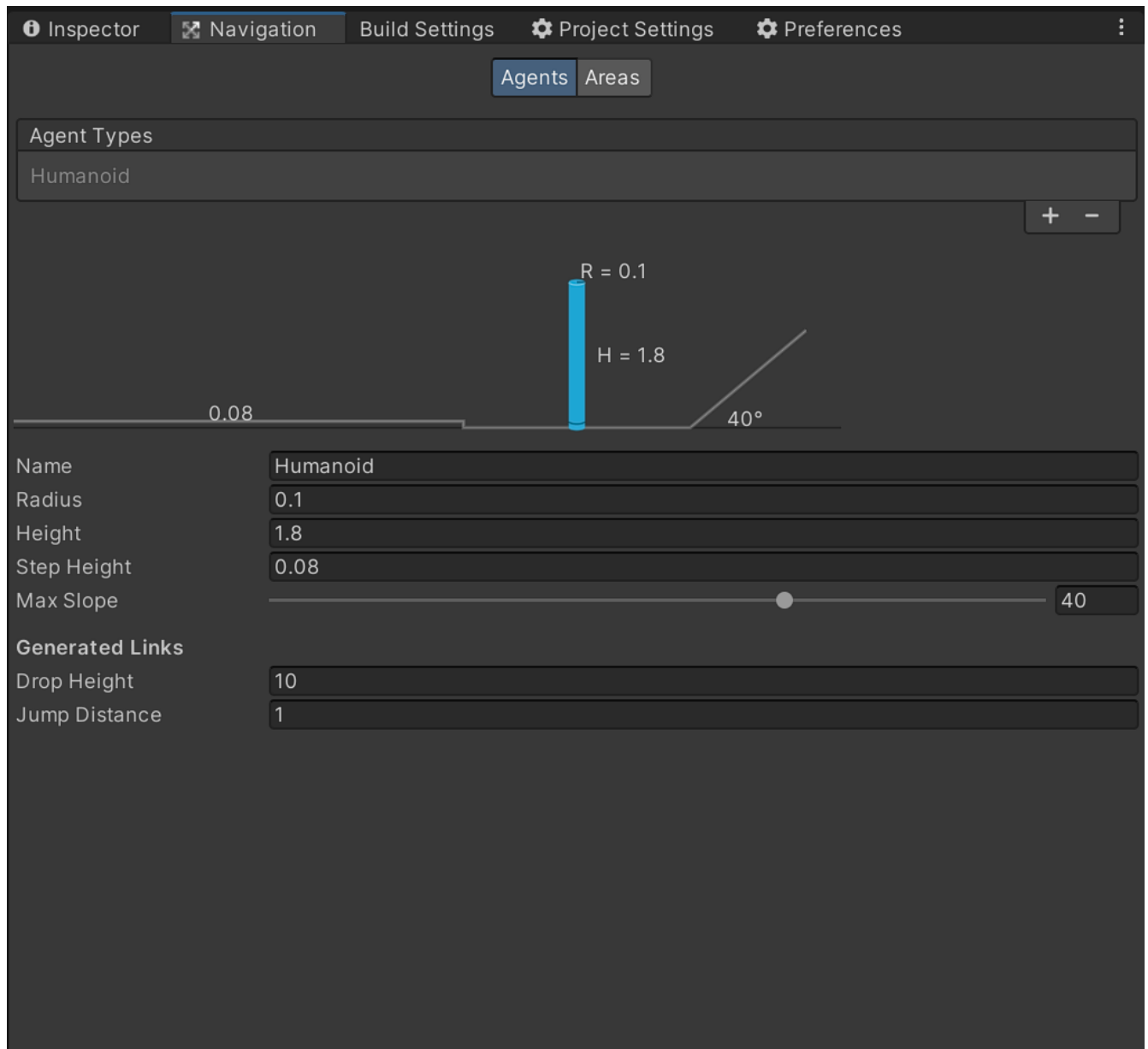


Рисунок 4.7 – Налаштування параметрів навігації

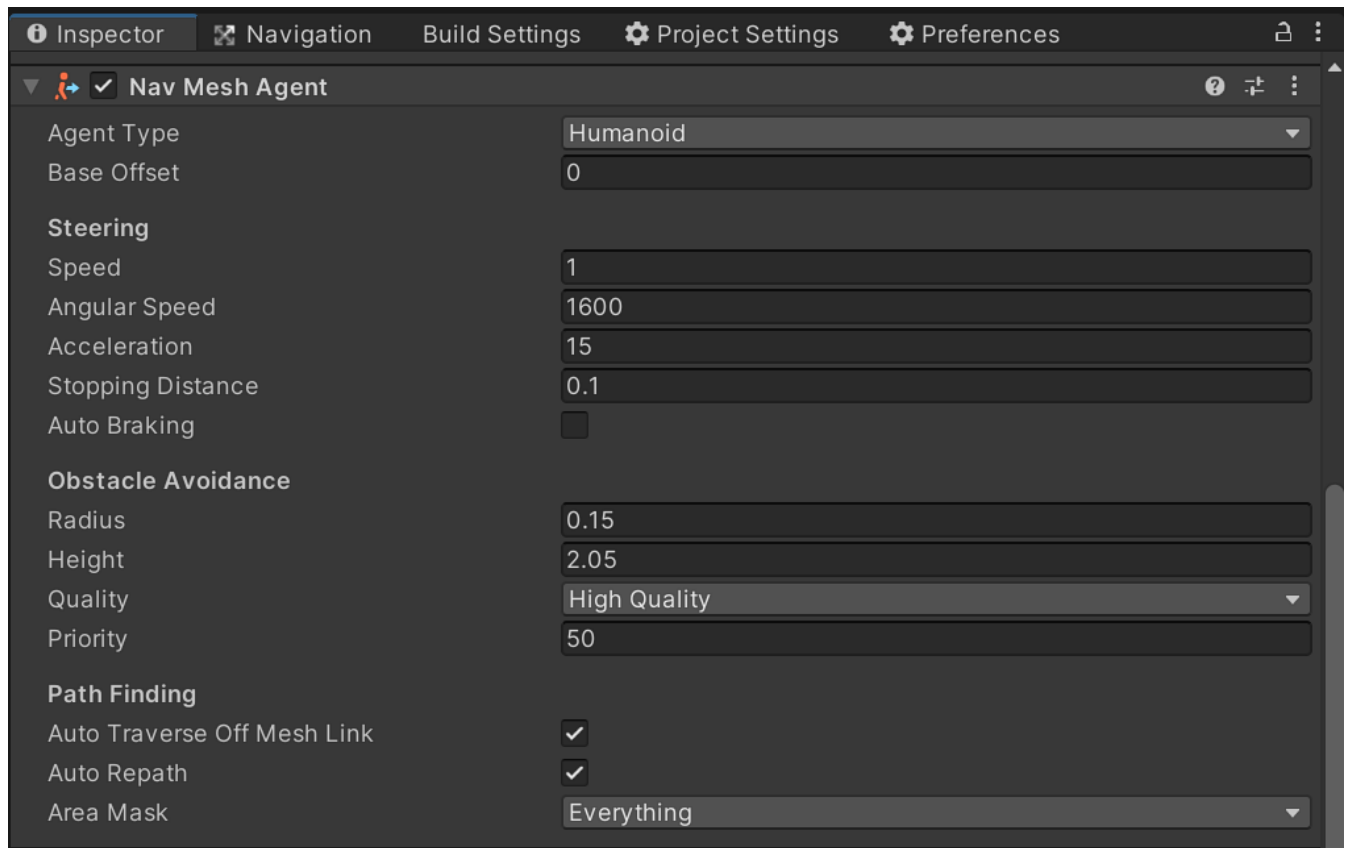


Рисунок 4.8 – Налаштування параметрів компоненту NavMesh Agent

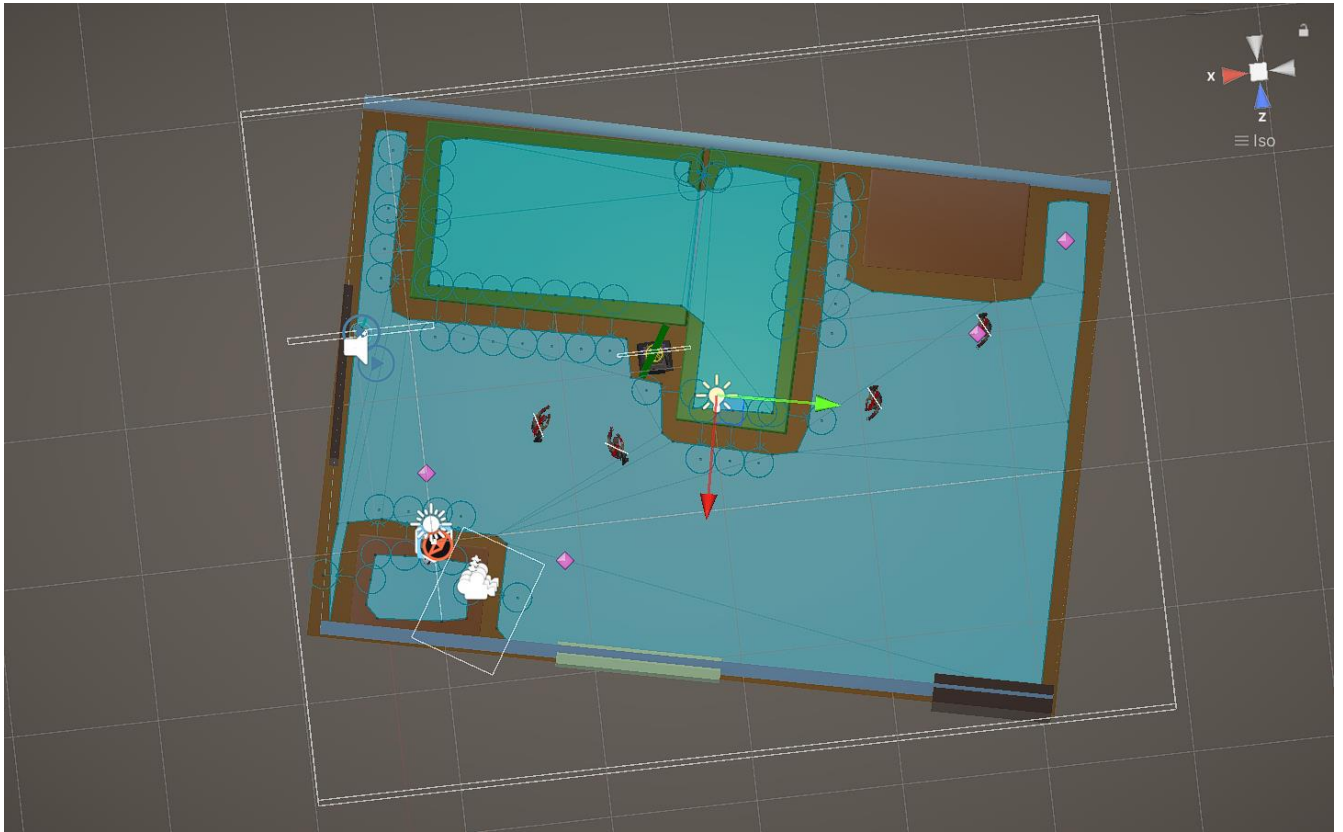


Рисунок 4.9 – Згенерована навігаційна сітка кімнати

Додатково було реалізувано генерацію точок створення для веж, ворогів та персонажа гри. Це дозволило створювати динамічні сценарії гри, де вороги та вежі з'являються у випадкових або попередньо визначених місцях, забезпечуючи різноманітність і непередбачуваність геймплею. Динамічне розташування об'єктів допомогло уникнути одноманітності та зробило кожну сесію гри унікальною. На рисунку 4.10 синім та зеленим написом зображено згенеровану точку створення героя та вежі. А рожевими ромбами позначено точки створення ворогів.

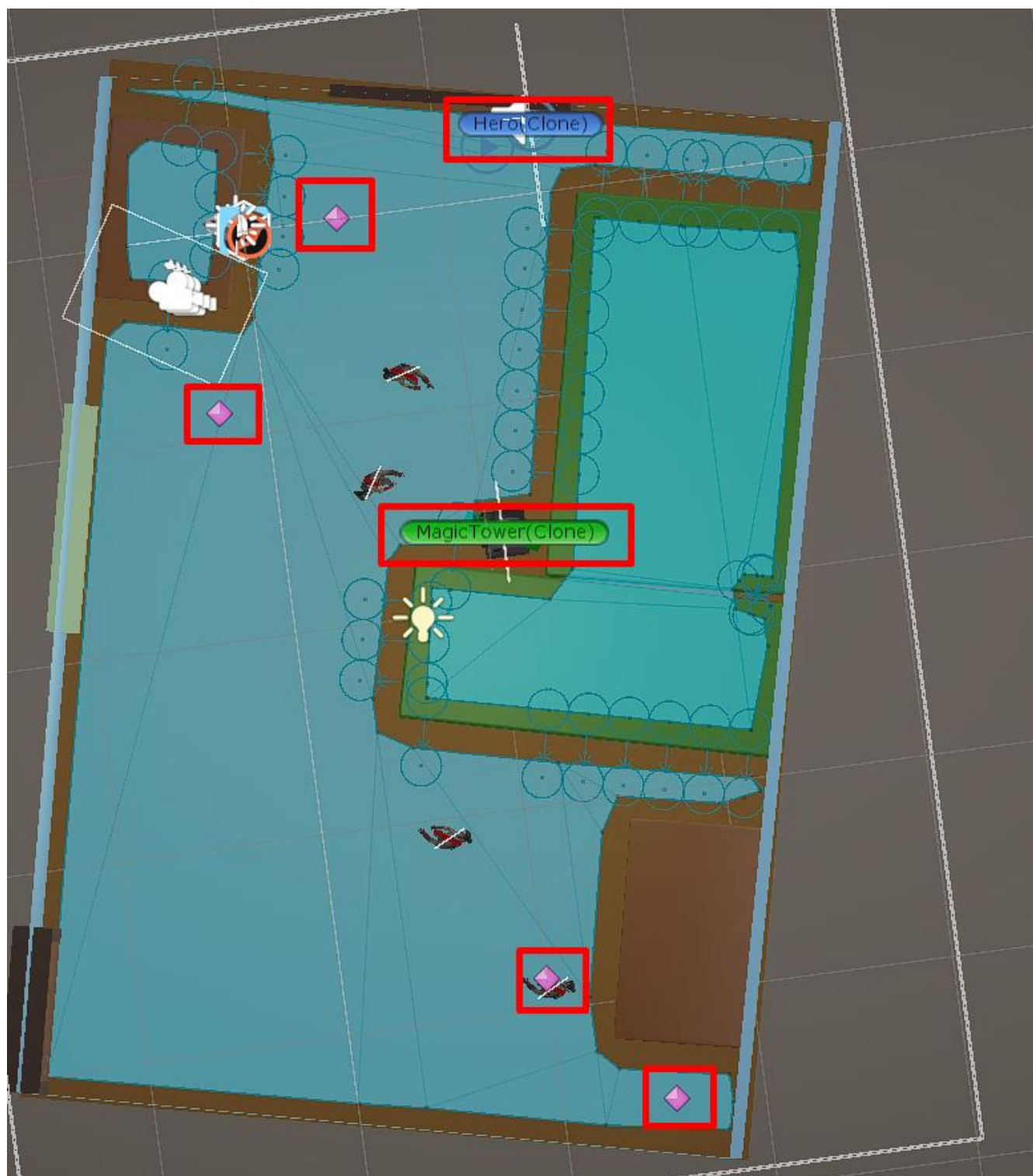


Рисунок 4.10 – Точки спавну ігрових об'єктів

Для розробки прототипу змішаної реальності гри, персонажі та анімації були отримані з ресурсу Міхато (рисунок 4.11). Міхато [16] є одним з провідних веб-сайтів, що надає безкоштовні 3D-моделі персонажів і анімації для розробників ігор. Використання цього ресурсу значно полегшило процес розробки, дозволяючи зосередитися на основних аспектах гри, таких як механіка ігрового процесу та інтеграція з VR-технологіями.

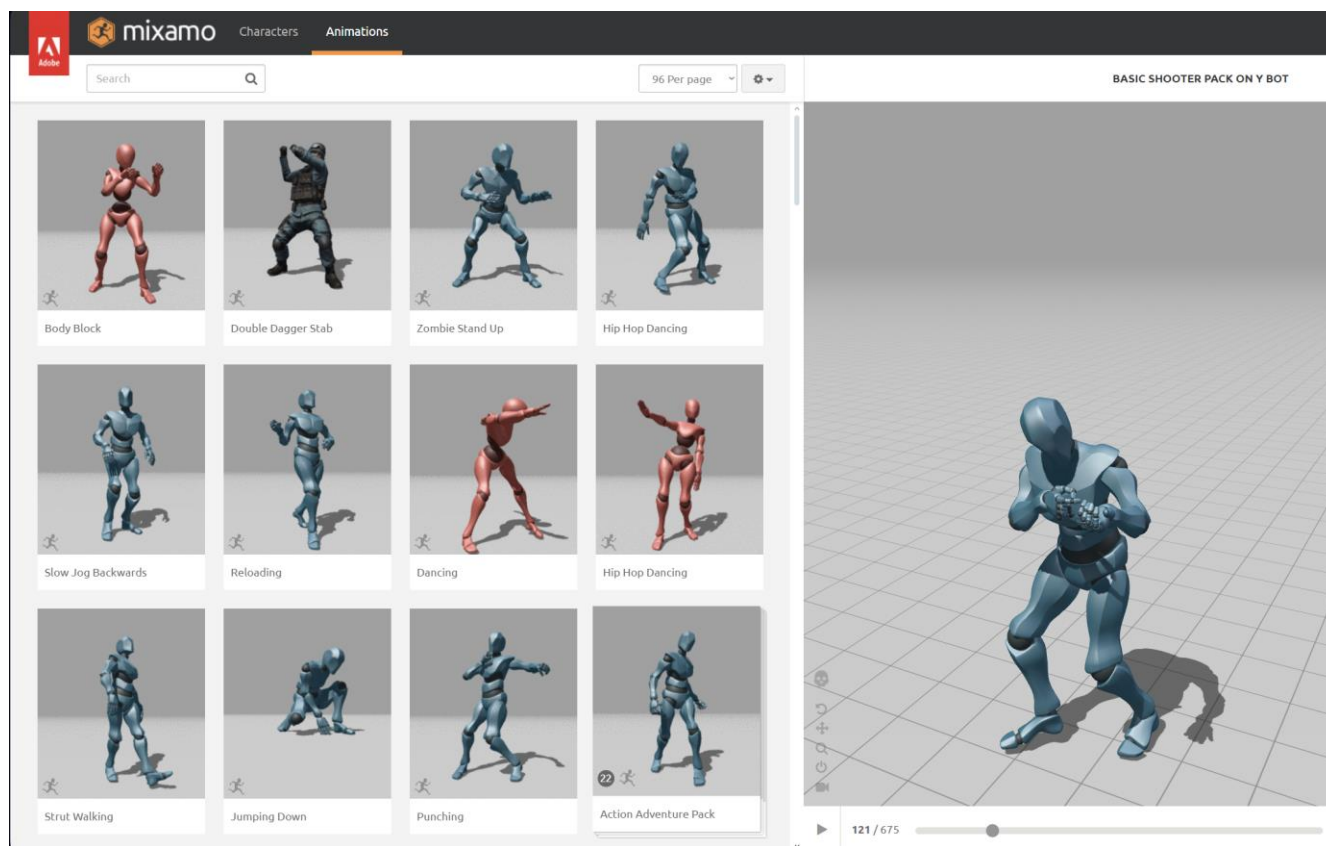


Рисунок 4.11 – Міхато персонажі та анімації

Після завантаження модельок персонажів (рисунок 4.12 та 4.13) з Міхато, вони були ретельно відредаговані відповідно до вимог прототипу. Редагування включало в себе налаштування скелетних структур та анімацій, щоб забезпечити відповідність специфічним потребам гри. Наприклад, були створені і адаптовані

різні анімації для стрільби, ходьби, бігу та взаємодії з об'єктами, що значно підвищило реалістичність ігрового процесу.

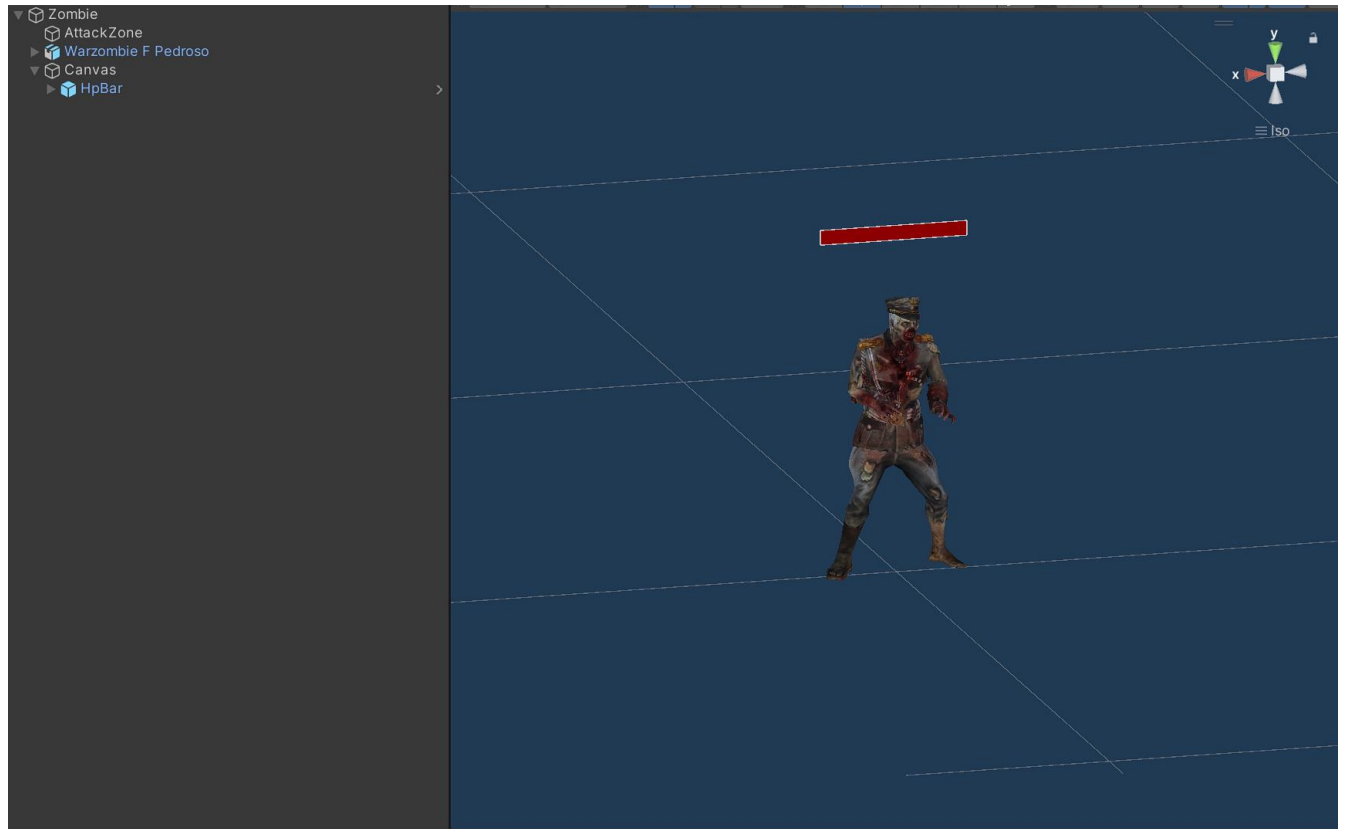


Рисунок 4.12 – Модель звичайного ворога

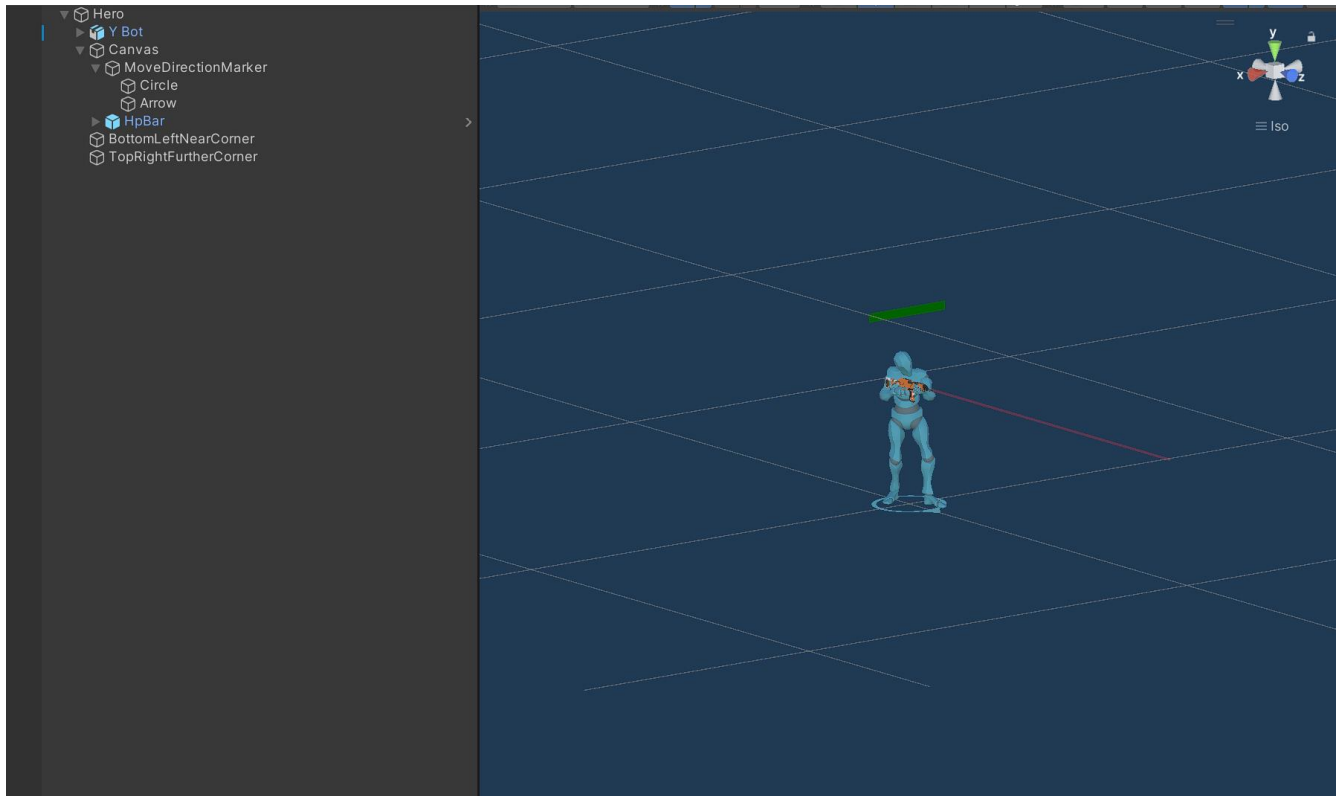


Рисунок 4.13 – Модель головного персонажа

Модель вежі (рисунок 4.14) була створена шляхом поєднання різних моделей з асетів, які були взяті з Unity Asset Store [15]. Поєднання різних моделей та їхня адаптація під специфічні вимоги прототипу допомогли створити унікальну модель вежі, яка органічно вписується в загальний дизайн гри. Це рішення також забезпечило високу якість та деталізацію вежі, що підвищило загальний рівень візуальної привабливості гри.

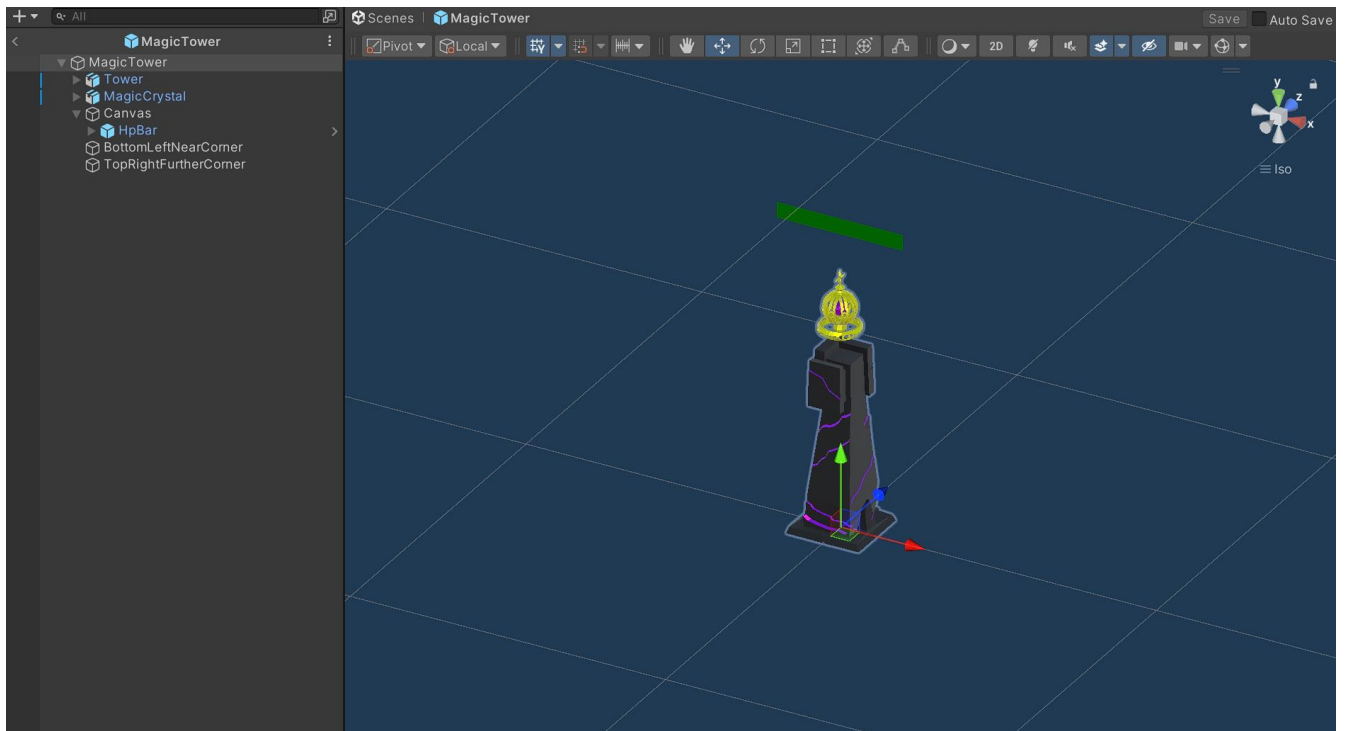


Рисунок 4.14 – Модель вежі

За допомогою Scriptable Objects було створено налаштовувані параметри для героя, ворогів, зброї та веж. Використання Scriptable Objects у Unity дозволило легко керувати різними аспектами гри, такими як характеристики персонажів, параметри зброї та властивості веж, без необхідності змінювати код. Цей підхід спростив процес налаштування та тестування, дозволяючи швидко вносити зміни та коригувати баланс гри.

Наприклад, можна було легко налаштувати здоров'я, швидкість руху та силу атаки героя і ворогів, а також параметри зброї, такі як скорострільність, влучність та кількість патронів. Завдяки цьому гра стала більш гнучкою та настроюваною, що сприяло покращенню геймплею та загальної якості прототипу.

На рисунках 4.15, 4.16, 4.17 та 4.18 зображено приклад налаштування основних об'єктів прототипу, таких як герой, вороги, зброя та вежі.

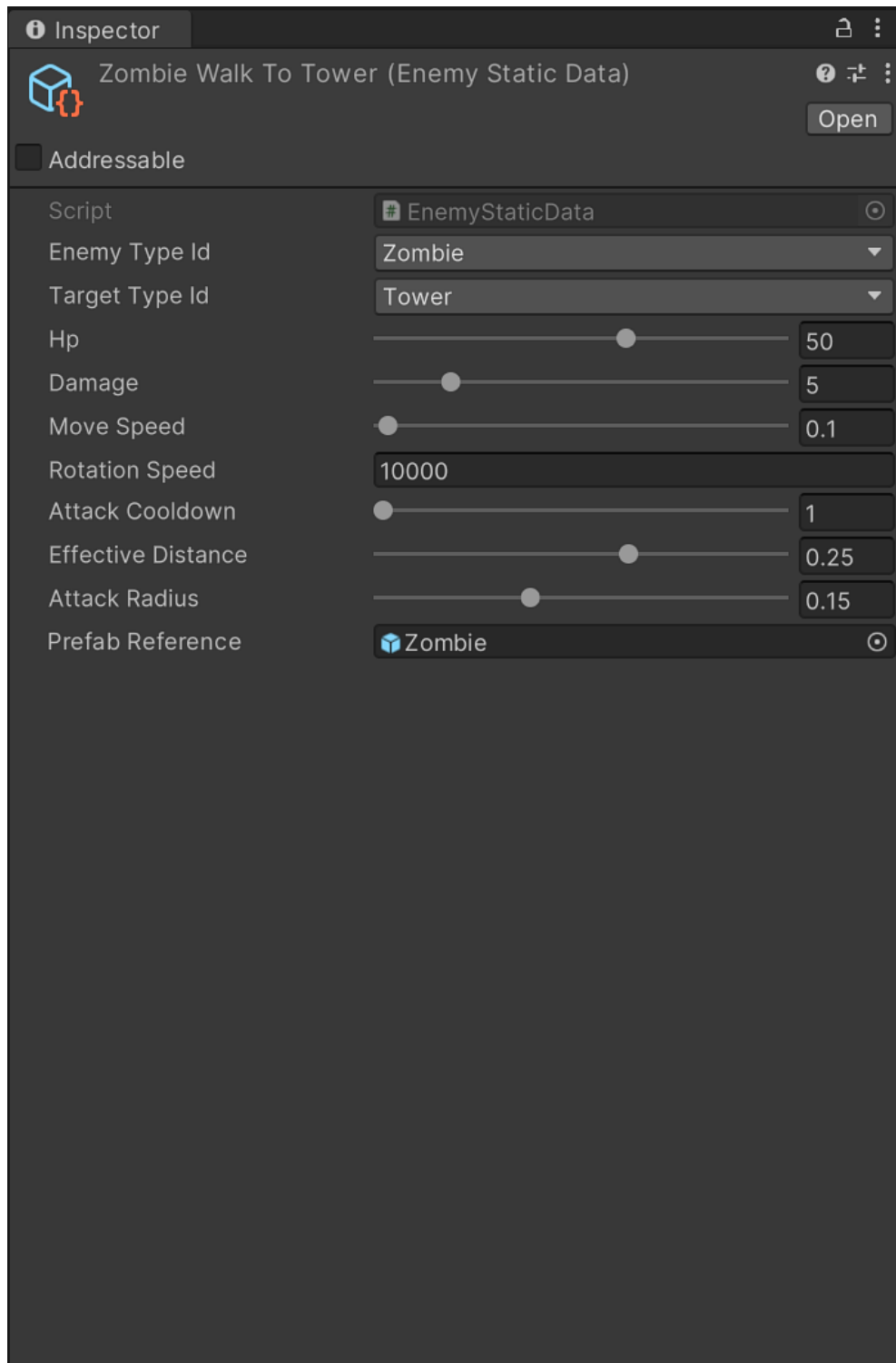


Рисунок 4.15 – Налаштування стартових характеристик ворога

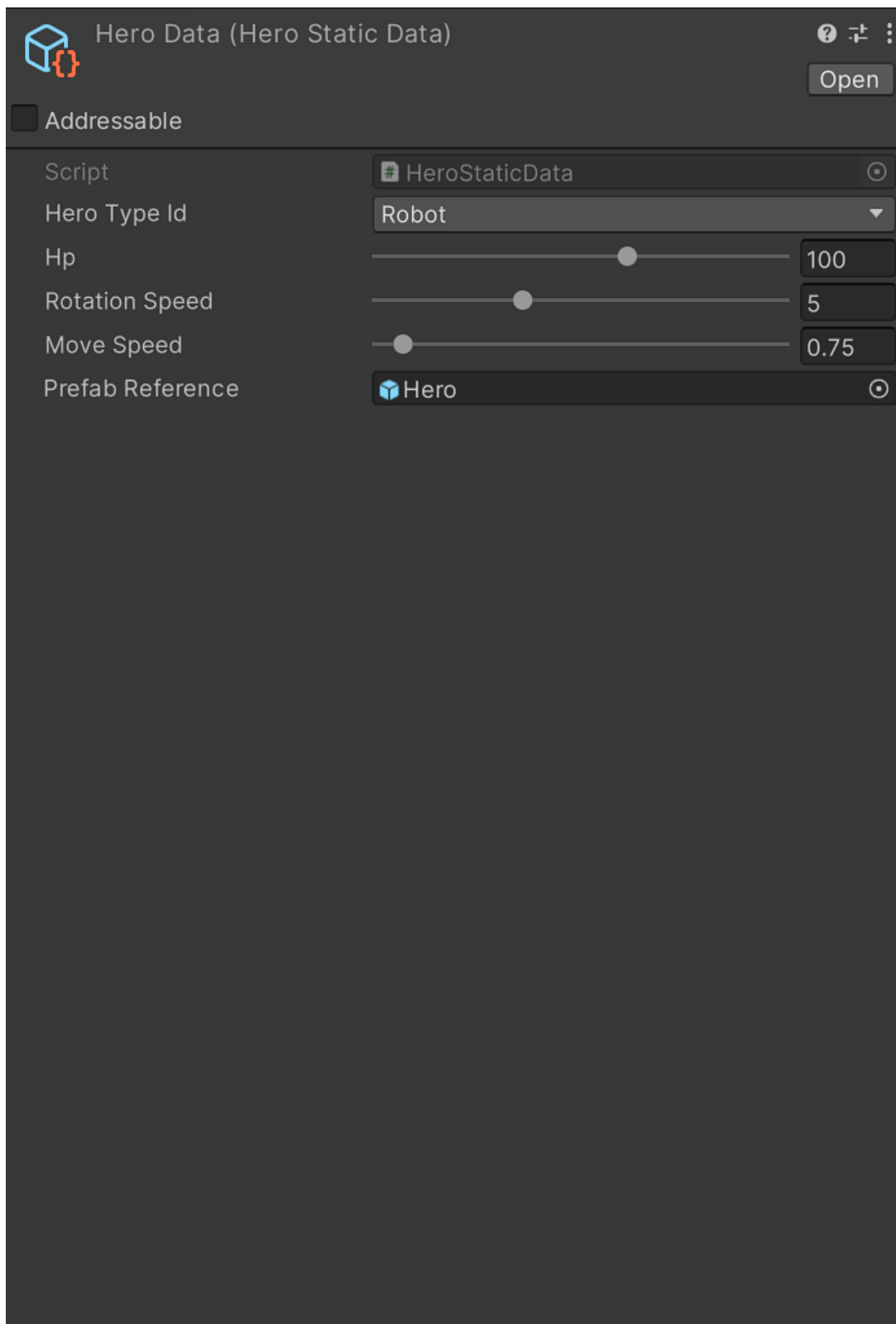


Рисунок 4.16 – Налаштування стартових характеристик героя

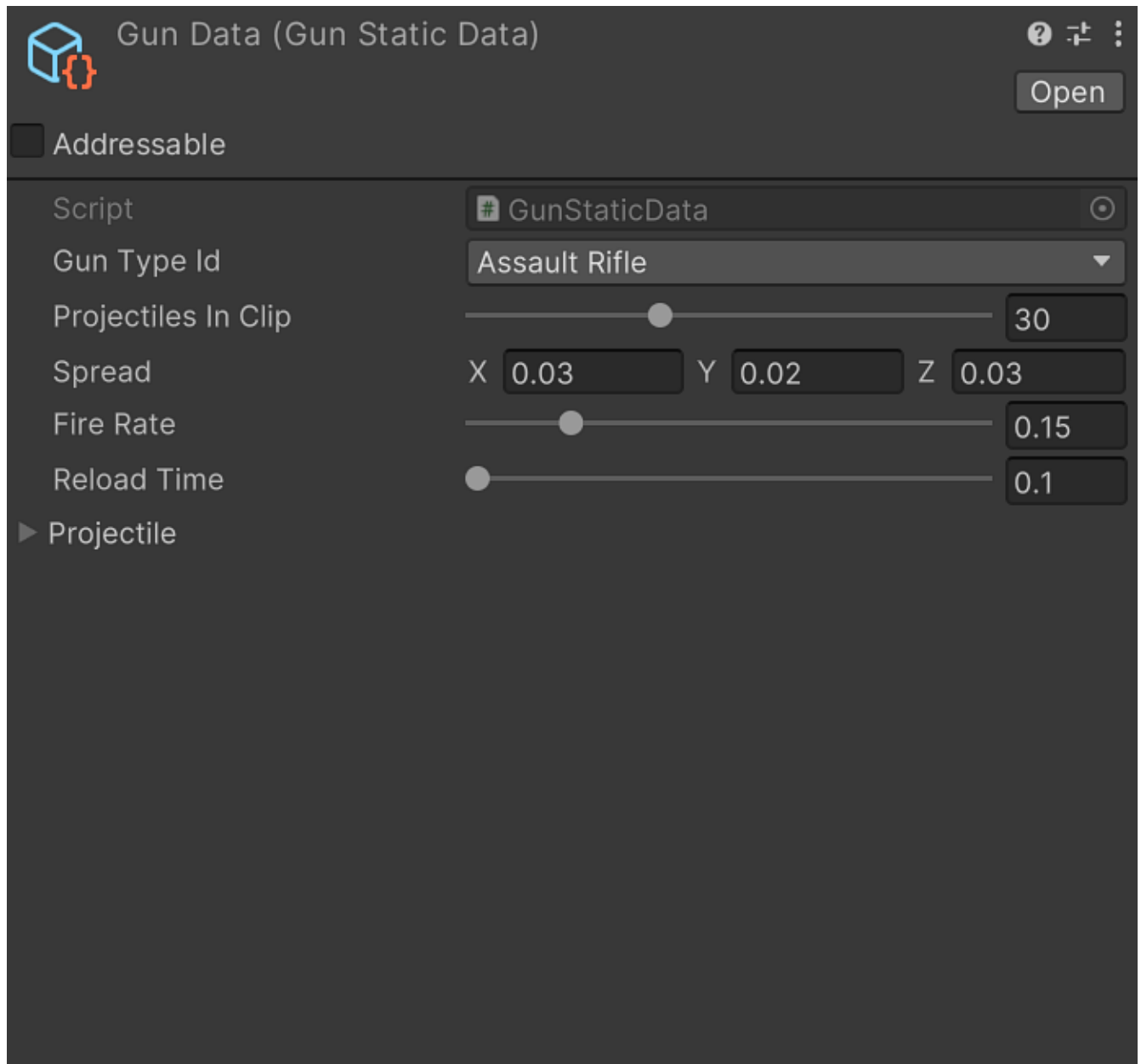


Рисунок 4.17 – Налаштування стартових характеристик штурмової гвинтівки

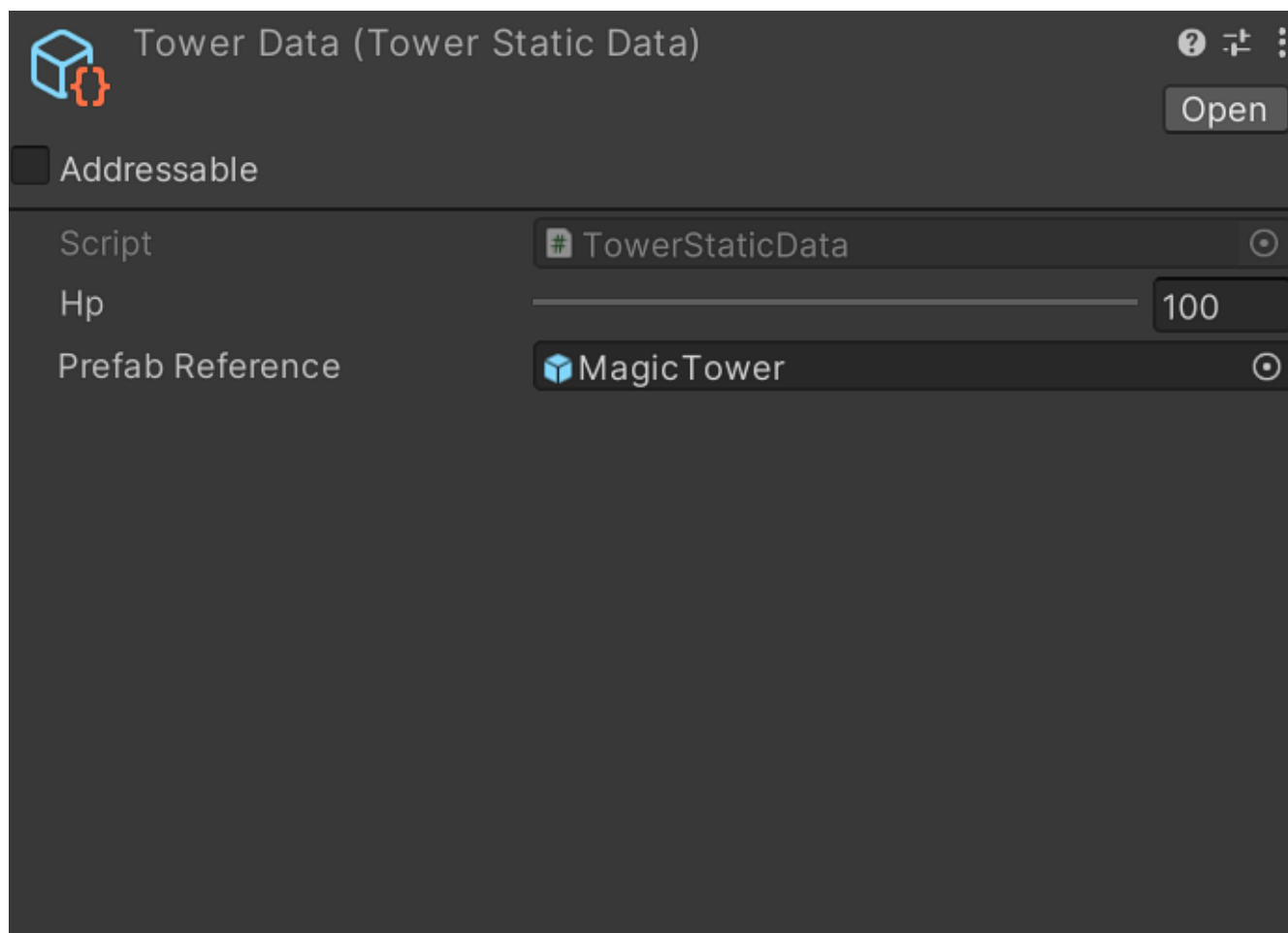


Рисунок 4.18 – Налаштування стартових характеристик вежі

Були також реалізовані всі необхідні анімації (рисунок 4.19) для стрільби та руху персонажа за допомогою контролерів Oculus. Ці анімації значно підвищили рівень реалістичності та динамічності ігрового процесу. Кожен рух персонажа був ретельно проанімований, що дозволило забезпечити максимальну реалістичність гри у власній кімнаті користувача. Особливу увагу було приділено дрібним деталям, що зробило рухи природнішими та інтерактивнішими.

Рух персонажа був реалізований за допомогою технології Blend Tree (рисунок 4.20), яка дозволяє створювати плавні переходи між анімаціями. Це забезпечило більш точне і реалістичне управління персонажем, що важливо для занурення гравця в ігровий світ. Таким чином, поєднання якісної анімації та інноваційної технології контролю створило глибокий та захоплюючий ігровий досвід.

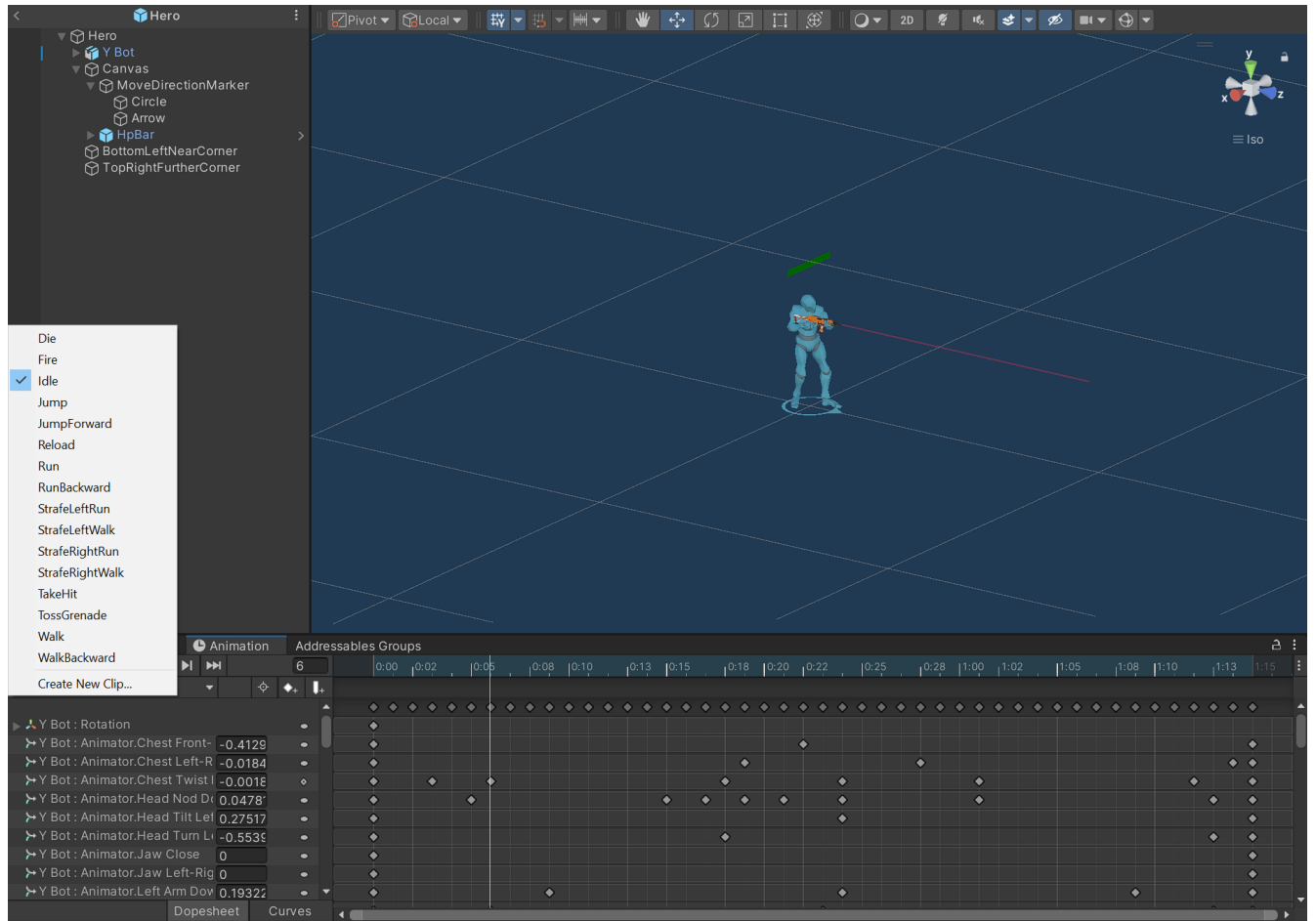


Рисунок 4.19 – Анімації персонажа

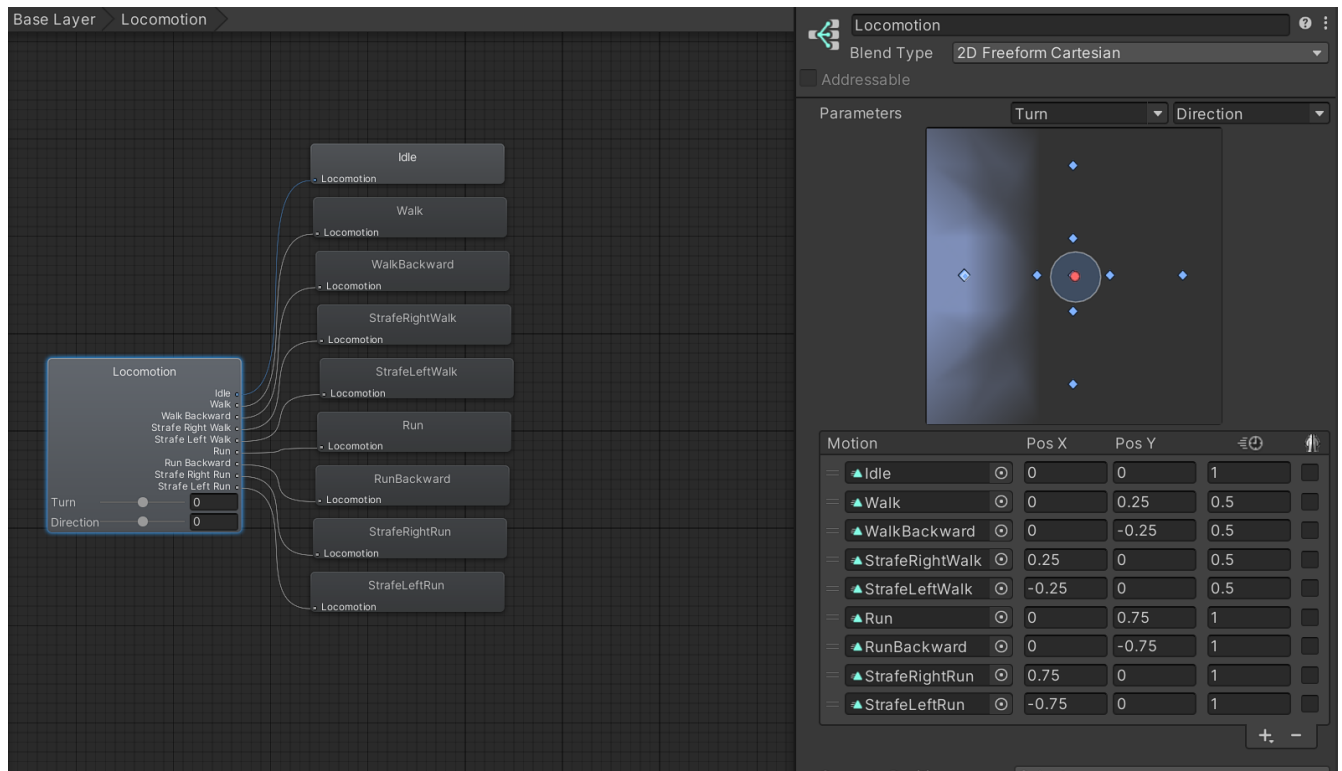


Рисунок 4.20 – Вікно налаштувань переходів між анімаціями у героя

Вороги у грі рухаються по згенерованому NavMesh Surface і мають можливість атакувати за допомогою ближнього бою. Реалізація ворогів включала анімації (рисунок 4.21) для руху та атаки, що дозволило створити реалістичну і загрозливу поведінку ворогів. Завдяки цьому гравці відчували справжню небезпеку і необхідність бути постійно на варті.

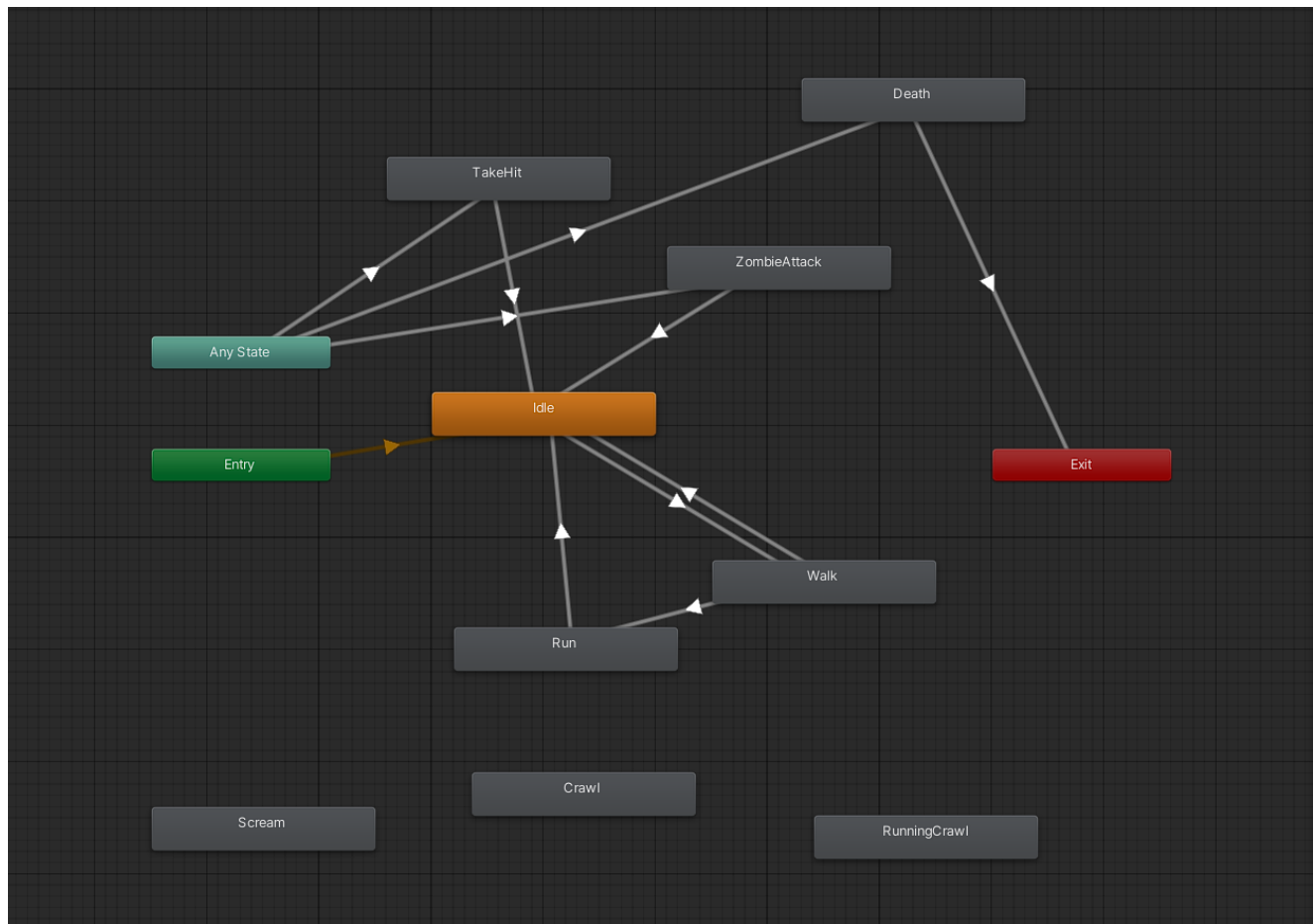


Рисунок 4.21 – Вікно переходів між анімаціями у ворога

Одним з ключових елементів гри стала система хвиль, що передбачала завантаження налаштувань (рисунок 4.22) конкретної арени. У кожній арені визначена певна кількість хвиль ворогів, і кожна наступна хвиля повинна бути складнішою за попередню. Гравець повинен витримати всі хвилі ворогів, щоб пройти арену. Це додало до гри елемент стратегії та планування, оскільки гравець мав враховувати сили та слабкості ворогів, розміщення своєї вежі, об'єктів власної кімнати, точок спавна ворогів та власні ресурси для успішної оборони.

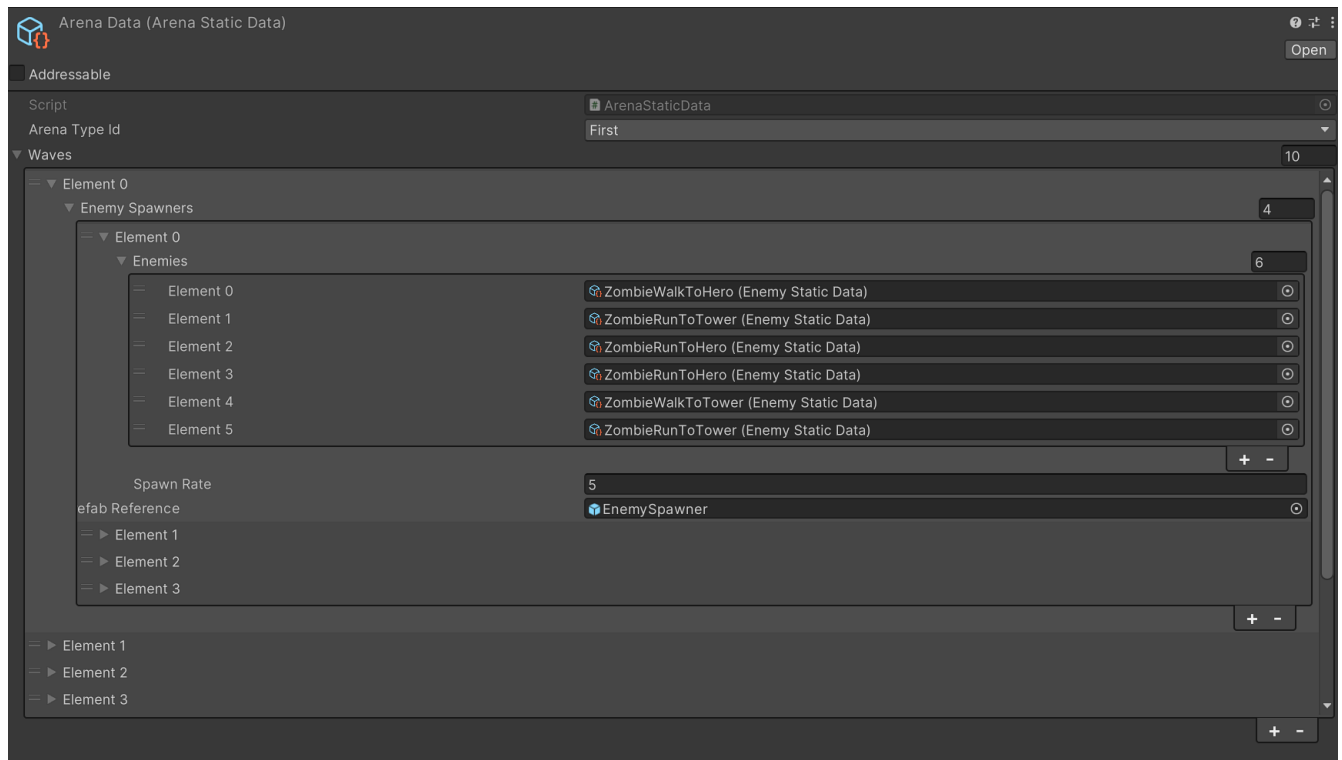


Рисунок 4.22 – Приклад налаштувань арени

Загалом, реалізація основних ігрових механік включала дослідження та використання Meta API для інтеграції MR, динамічну генерацію навігаційних сіток, створення систем спавну, реалізацію руху та анімацій для персонажа і ворогів, а також впровадження системи арен та хвиль для підвищення виклику та інтересу до гри. Усі ці елементи були ключовими для створення захоплюючого та інтерактивного ігрового досвіду прототипа. Використання сучасних технологій та динамічних механік дозволило надати цікавості та задоволення від ігрового процесу.

4.3 Розробка інтерфейсу з урахуванням особливостей VR

На мою думку, найвдалішим рішенням для відображення інтерфейсу у VR є показувати його або перед обличчям гравця, або безпосередньо на полі бою. Це дозволяє зберегти максимальну зануреність і забезпечити зручність використання

інтерфейсу. Інтерфейс повинен бути максимально інтуїтивно зрозумілий та простий. На даному етапі, візуальна частина інтерфейсу не була детально пророблена, акцент був спрямований на функціонал.

Для зручної зміни та завантаження інтерфейсних вікон у прототипі було створено Scriptable Object (рисунок 4.23), що містить всі вікна, якими можна керувати за допомогою ID. Це дозволило швидко й ефективно змінювати інтерфейсні елементи в залежності від ситуації у грі, забезпечуючи гнучкість та адаптивність інтерфейсу.

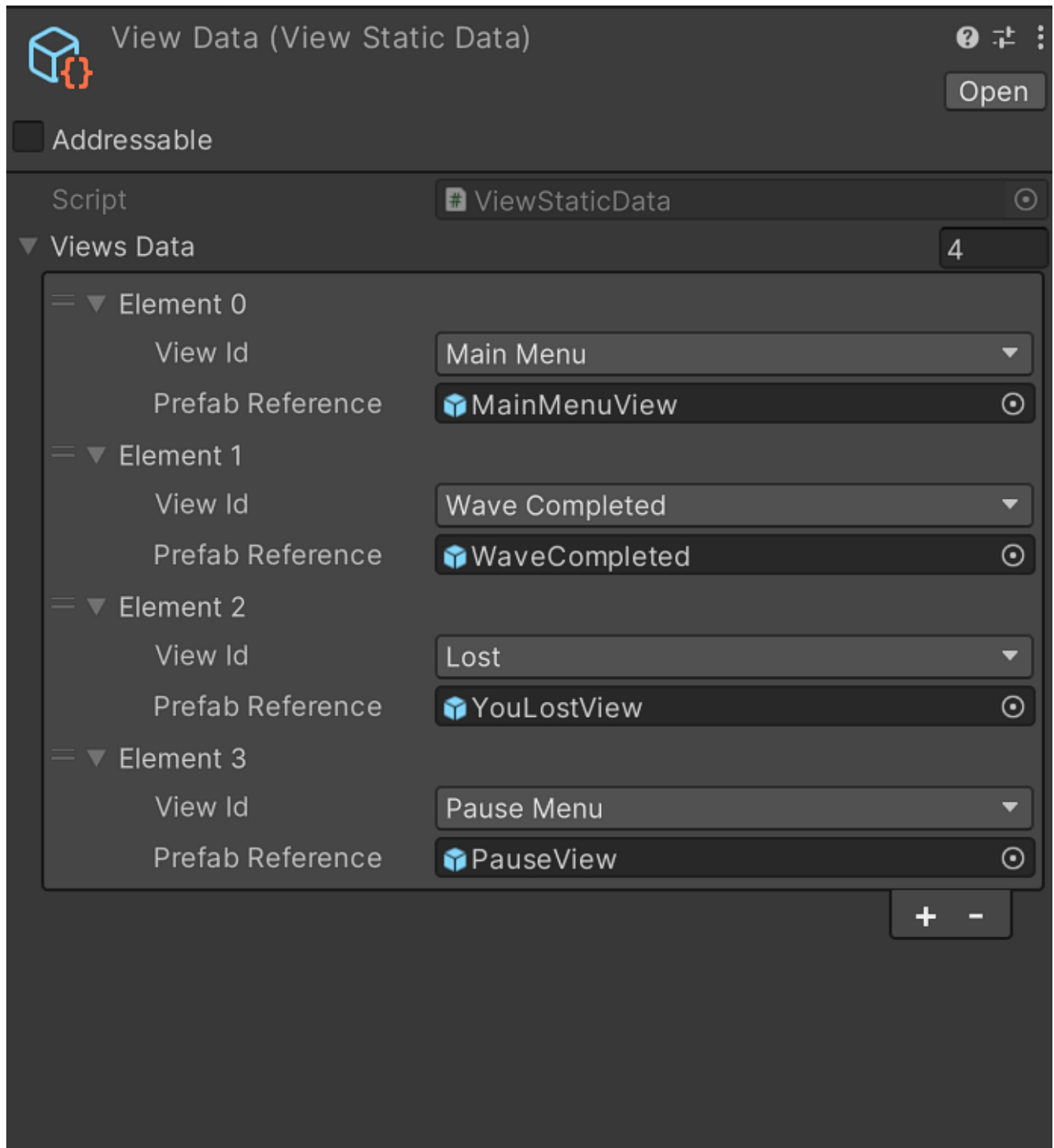


Рисунок 4.23 – Налаштувань списку інтерфейсних вікон

Запустивши гру користувач потрапляє у головне меню (рисунок 4.24), яке включає наступні елементи:

- а) Кнопка Play – ця кнопка стартує гру, переводячи гравця до поточної арени;

б) Кнопка Settings – ця кнопка відкриває вікно налаштувань (рисунок 4.25), де гравець може налаштувати:

- 1) Music On/Off: Дозволяє вмикати або вимикати музичний супровід;
- 2) Sound Volume: Налаштування гучності звуків у діапазоні від 0 до 100. Ці опції дозволяють гравцям адаптувати аудіо настройки відповідно до своїх вподобань, що робить ігровий процес комфортнішим.

в) Кнопка Character – ця кнопка відкриває вікно персонажів, де можна покращити їх постійні характеристики за ігрову валюту. Це дозволяє створювати більш персоналізований ігровий досвід та стимулює гравців до подальшого розвитку персонажа;

г) Кнопка Quit – ця кнопка дозволяє вийти з гри.



Рисунок 4.24 – Інтерфейс головного меню

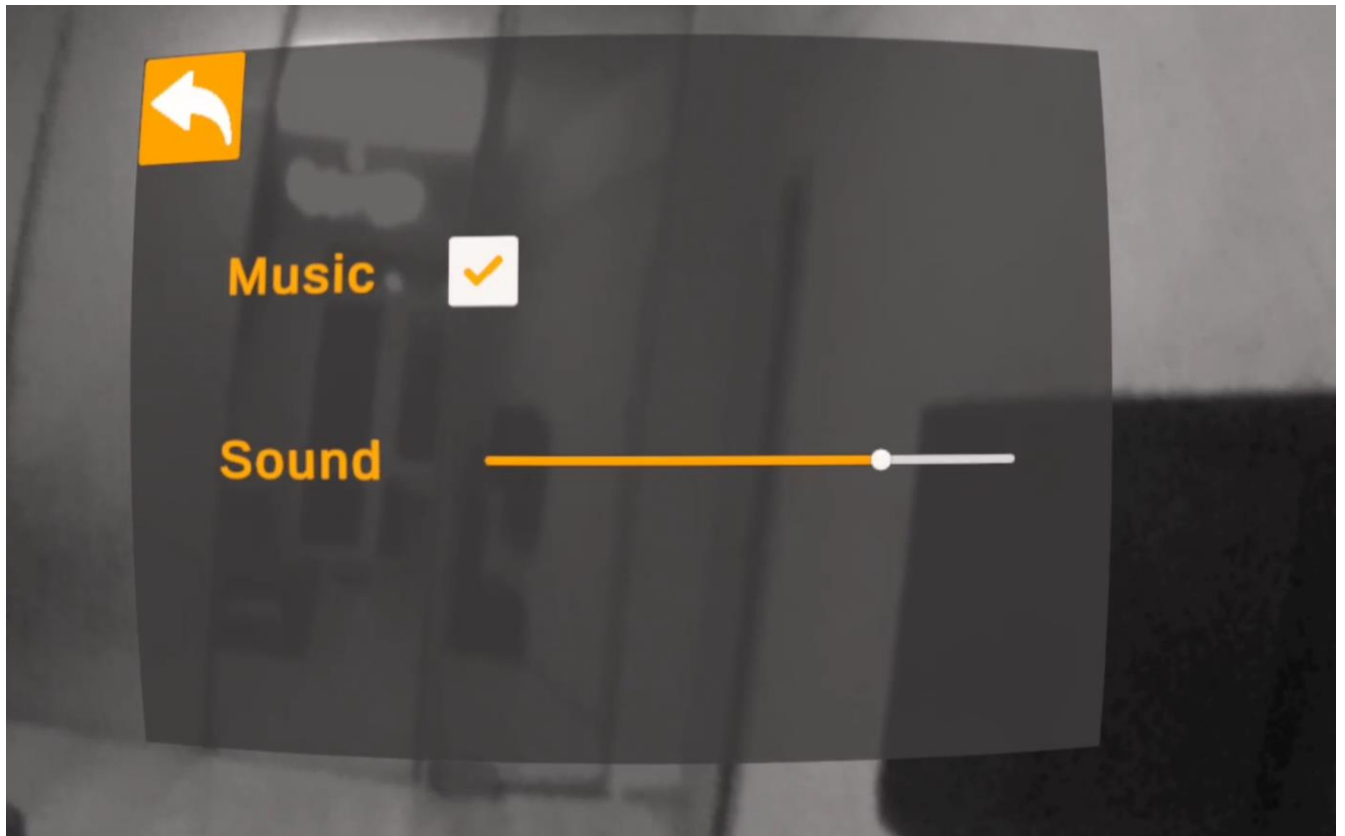


Рисунок 4.25 – Вікно налаштувань

Окрім основного меню, було реалізовано меню паузи (рисунок 4.26), в яке можна потрапити в будь-який момент під час гри, що включає наступні елементи:

а) Кнопка **Resume** – ця кнопка повертає назад в гру, та дозволяє продовжити гру з моменту коли вона була зупинена;

б) Кнопка **Restart** – ця кнопка дозволяє почати проходження арени заново, що дозволяє гравцям виправити помилки та спробувати нові стратегії;

в) Кнопка **Menu** – ця кнопка дозволяє повернутися в головне меню.

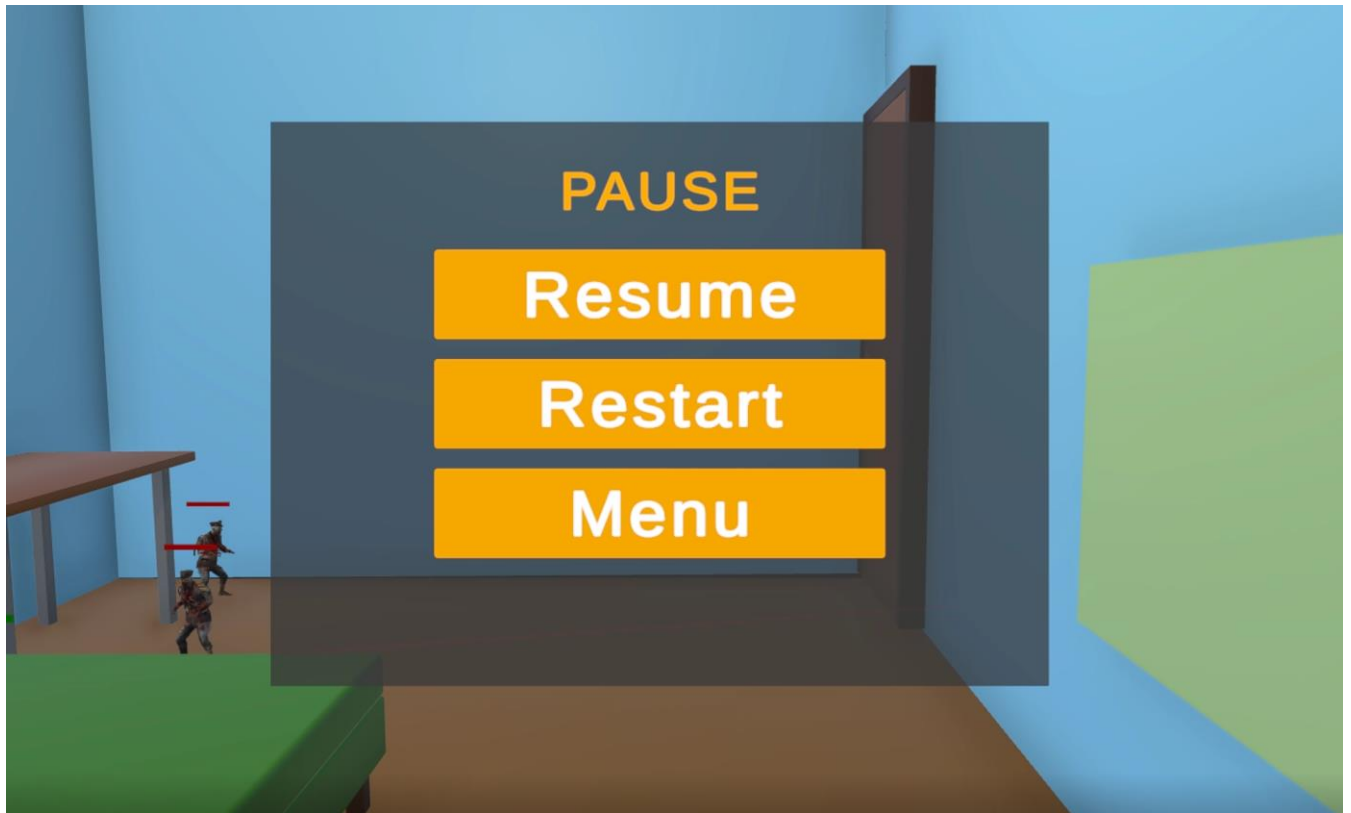


Рисунок 4.26 – Меню паузи

Під час гри гравець може бачити своє поточне здоров'я, ворогів та вежі (рисунок 4.27), щоб правильно розрахувати свої сили при обороні.

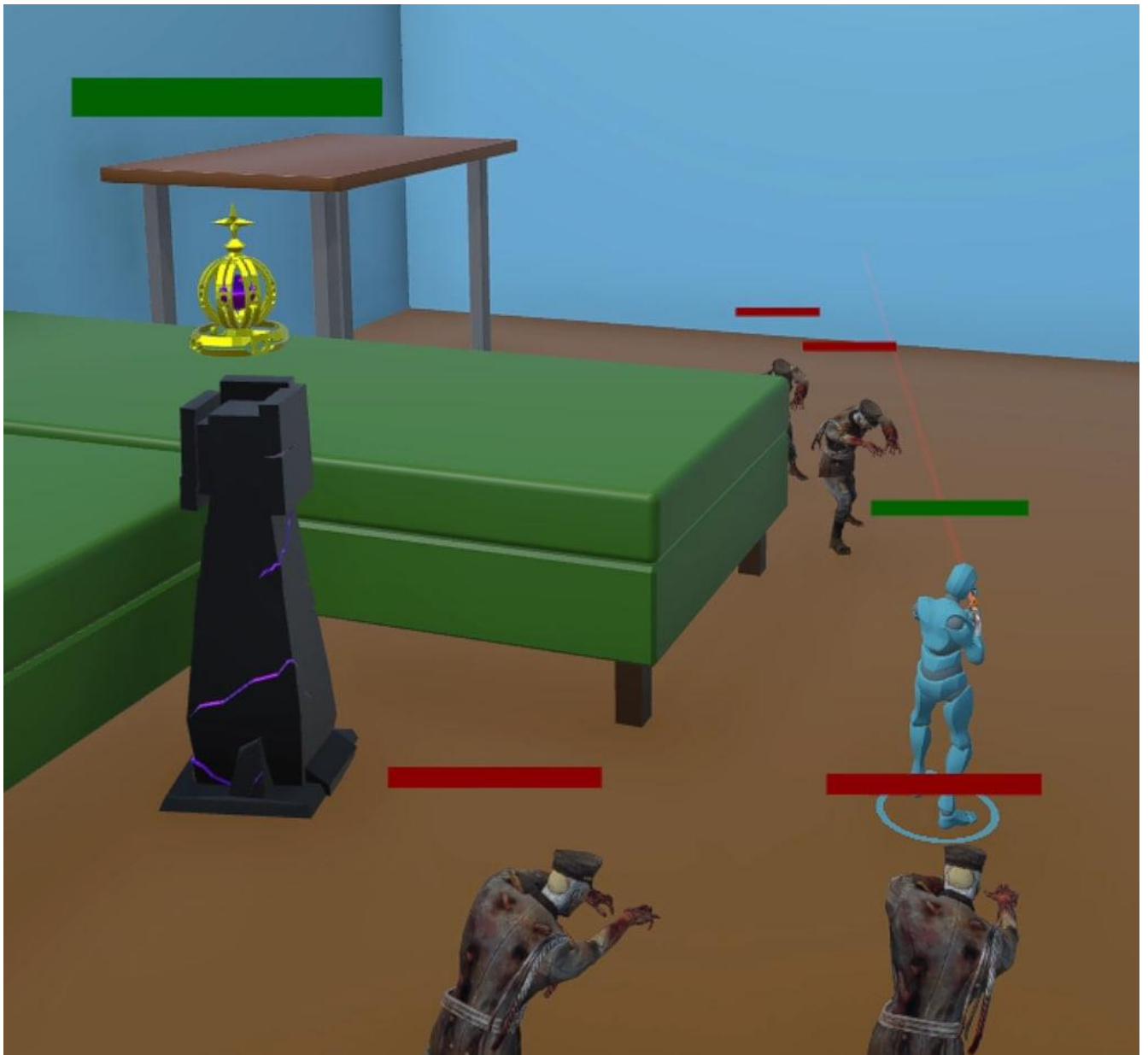


Рисунок 4.27 – Поточне здоров'я головних об'єктів гри

Також після кожної хвили ворогів гравець має вибір з трьох можливих непостійних покращень (рисунок 4.28). Ці покращення діють тільки під час проходження конкретної арени, що додає елемент стратегії та робить кожну гру унікальною.



Рисунок 4.28 – Вікно вибору непостійного покращення

У разі програшу гравець бачить вікно програшу (рисунок 4.29), з якого можна почати проходження арени заново. Це забезпечує зручність та швидкий доступ до повторного проходження рівня, що робить гру менш фруструючою та більш захоплюючою.



Рисунок 4.29 – Вікно програшу

Таким чином, інтерфейс гри було розроблено з урахуванням особливостей віртуальної реальності, забезпечуючи легкість у використанні та максимальне занурення у гру.

4.4 Тестування прототипу в Mixed Reality середовищі

Тестування прототипу в Mixed Reality (MR) середовищі є важливим етапом для перевірки та вдосконалення ключових аспектів гри. Основна мета тестування полягала у виявленні можливих проблем та оцінці ефективності реалізованих ігрових механік. У процесі тестування були виконані наступні кроки:

а) Підготовка середовища для тестування:

- 1) Обладнання: Для тестування використовувалася гарнітура Meta (Oculus) Quest 2 та відповідні контролери;
- 2) Простір: Було підготовлено фізичний простір для забезпечення безпеки та комфортних умов для тестування, враховуючи можливість активного руху гравця.

б) Перевірка навігаційних сіток (NavMesh):

- 1) Було перевірено, як добре генерується NavMesh Surface у реальному часі в залежності від різних конфігурацій кімнат;
- 2) Тестування включало перевірку правильності руху ворогів по згенерованих навігаційних сітках, зокрема їх здатності уникати перешкод та знаходити оптимальні маршрути до цілей.

в) Тестування інтерактивності та управління:

- 1) Було перевірено, наскільки інтуїтивним є управління персонажем за допомогою VR-контролерів;
- 2) Оцінювалася точність та реактивність рухів персонажа, а також взаємодія з віртуальними об'єктами.

г) Аналіз користувацького інтерфейсу:

- 1) Тестування проводилося для оцінки зручності використання VR-меню та інших інтерфейсних елементів, розташованих перед обличчям гравця або на полі бою;
- 2) Оцінювалася видимість, доступність та зрозумілість елементів інтерфейсу, включаючи основне меню, налаштування, вікно паузи та вікно програшу.

г) Оцінка ігрового процесу:

- 1) Тестування проводилося для оцінки динаміки та реалістичності бойових дій, включаючи стрільбу та анімації персонажа;
- 2) Було перевірено систему хвиль ворогів, включаючи поступове збільшення складності та генерацію спавн точок для ворогів і веж.

Виявлені недоліки та пропозиції щодо покращення:

- а) Занурення у гру: Було відзначено, що рівень занурення можна підвищити, додавши більше звукових та візуальних ефектів. Наприклад, звукові ефекти для взаємодії з об'єктами та оточенням можуть значно покращити сприйняття гри;
- б) Оптимізація продуктивності: Варто провести додаткову оптимізацію гри для забезпечення стабільної роботи на різних пристроях та конфігураціях. Це включає зниження затримок та підвищення швидкості завантаження рівнів.
- в) Навігаційні сітки: Було виявлено, що при деяких специфічних налаштуваннях кімнати, можуть зустрічатися неправильні генерації навігаційних сіток.
- г) Інтерактивність: Підвищити рівень інтерактивності ігрового середовища. Наприклад, додати можливість взаємодії з більшою кількістю об'єктів, що дозволить гравцям глибше зануритися у віртуальний світ.

Таким чином, тестування прототипу в Mixed Reality середовищі дозволило виявити важливі аспекти, які потребують вдосконалення. Зібраний зворотний зв'язок та проведений аналіз допоможуть у подальшій розробці, щоб створити більш занурюючий та захоплюючий ігровий досвід.

ВИСНОВКИ

В ході виконання кваліфікаційної роботи було успішно вирішено ряд поставлених задач, що охоплювали широкий спектр діяльності від аналітичної роботи до креативного процесу розробки.

Було проведено глибокий аналіз предметної області, включаючи детальний огляд існуючих ігор-аналогів та визначення ключових задач і сфер застосування для розробки MR-гри. Це дозволило не тільки визначити актуальні тренди і вектори розвитку галузі, але й виявити потенційні можливості для інновацій у цій сфері.

Були оглянуті сучасні технології і платформи для розробки VR-ігор, з особливим акцентом на можливості Unity як інструменту для створення імерсивних ігрових досвідів.

Дослідження сучасних технологій та платформ для розробки VR-ігор з акцентом на можливості Unity як інструменту для створення імерсивних ігрових досвідів відкрило нові перспективи для реалізації проекту. Unity Engine було визначено як оптимальне середовище для розробки, завдяки його гнучкості, масштабованості та великому співтовариству розробників, що надає цінні ресурси та підтримку.

Проведене дослідження не тільки підтвердило реальність та виконуваність запропонованої ігрової ідеї в рамках наявних технологій, але й відкрило широкі перспективи для подальшого розвитку і поліпшення проекту. Отримані знання та досвід стануть фундаментом для уникнення потенційних помилок у подальшій програмній реалізації прототипу, зробить розробку якіснішою та швидшою.

Робота пройшла апробацію у рамках IV міжнародної науково-практичної конференції «PERSPECTIVES OF CONTEMPORARY SCIENCE: THEORY AND PRACTICE», – Львів, 26-28 травня 2024 р.

ПЕРЕЛІК ПОСИЛАНЬ

1. Методичні вказівки до організації виконання та захисту кваліфікаційної роботи на здобуття другого (магістерського) рівня вищої освіти спеціальності 122 Комп'ютерні науки, освітньо-наукова програма «Системне проектування» / Упорядники: І.В. Гребеннік, Н.І. Калита, І.М. Рябченко, З.А Імангулова, О.Б. Колесник, М.Ю. Вишняк. Харків: ХНУРЕ, 2021. 54 с.
2. Ігри в магазині Meta Oculus в жанрі Mixed reality – 2024 – URL: <https://www.meta.com/experiences/view/6557290314361866/> (дата звернення: 12.02.2024)
3. Unity User Manual – 2024 – URL: <https://docs.unity3d.com/Manual/index.html> (дата звернення: 23.03.2024)
4. Oculus Developer Guide – 2024 – URL: <https://developer.oculus.com/documentation/> (дата звернення: 19.04.2024)
5. Schell, Jesse. "The Art of Game Design: A Book of Lenses." CRC Press, 2019. 600 p.
6. Road to VR – Virtual Reality News – 2024 – URL: <https://www.roadtovr.com> (дата звернення: 17.02.2024)
7. SideQuest: Oculus Quest Games – 2024 – URL: <https://sidequestvr.com> (дата звернення: 16.02.2024)
8. Adams, Ernest. "Fundamentals of Game Design." Pearson Education, 2014. 533 p.
9. Moore, Michael E. "Basics of Game Design." A K Peters/CRC Press, 2011. 392 p.
10. Game Developers Conference (GDC) – 2024 – URL: <https://gdconf.com> (дата звернення: 21.02.2024)
11. Unity Learn – 2024 – URL: <https://learn.unity.com> (дата звернення: 16.03.2024)
12. Oculus Developer Forums – 2024 – URL: <https://forums.oculusvr.com/developer> (дата звернення: 18.04.2024)
13. Unity Forums – 2024 – URL: <https://forum.unity.com> (дата звернення: 19.03.2024)

14. Stack Overflow - Where Developers Learn, Share, & Build Careers – 2024 – URL: <https://stackoverflow.com/> (дата звернення: 21.03.2024)
15. Unity Asset Store – 2024 – URL: <https://assetstore.unity.com> (дата звернення: 20.04.2024)
16. Mixamo - Animate 3D characters for games, film, and more. – 2024 – URL: <https://www.mixamo.com/#/> (дата звернення: 21.04.2024)
17. Gibson, Jeremy "Introduction to Game Design, Prototyping, and Development: From Concept to Playable Game with Unity and C#." Pearson Education, 2014. 944 p.
18. C# documentation – 2024 – URL: <https://learn.microsoft.com/en-us/dotnet/csharp/> (дата звернення: 18.03.2024)
19. Tidwell, Jen. "Designing Interfaces: Patterns for Effective Interaction Design." O'Reilly Media, 2021. 599 p.
20. Norman, Don. "The Design of Everyday Things." Basic Books, 2013. 368 p.
21. O. Gutsa, D. Yelchaninov, D. Yanushkevich I. Tolkunov L. Ivanov, R. Petrova, A. Morozova. Conceptualization of Intelligent Control System for Humanitarian Demining Robotic Complexes Based on Verbal Methods. // Науково-практичний журнал НАН України «Science and Innovation», 20(2), 82—95. <https://doi.org/10.15407/scine20.03.082> (дата звернення: 07.02.2024)
22. 2D Sprite – 2024 – URL: <https://docs.unity3d.com/Packages/com.unity.2d.sprite@1.0/manual/index.html> (дата звернення: 19.04.2024)
23. Addressables package – 2024 – URL: <https://docs.unity3d.com/Packages/com.unity.addressables@1.21/manual/index.html> (дата звернення: 21.04.2024)
24. AI Navigation – 2024 – URL: <https://docs.unity3d.com/Packages/com.unity.ai.navigation@1.1/manual/index.html> (дата звернення: 25.04.2024)

25. Animation Rigging – 2024 – URL:
<https://docs.unity3d.com/Packages/com.unity.animation.rigging@1.2/manual/index.html>
(дата звернення: 17.04.2024)

26. Requirements to use Meta Quest Link – 2024 – URL:
<https://www.meta.com/help/quest/articles/headsets-and-accessories/oculus-link/requirements-quest-link/> (дата звернення: 05.04.2024)