

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ПОЛІСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій, обліку та фінансів

Кафедра комп'ютерних технологій
і моделювання систем

Кваліфікаційна робота
на правах рукопису

Гаврилюк Артур Вадимович

УДК 005.382.7:681.8

КВАЛІФІКАЦІЙНА РОБОТА

Розробка гри в жанрі ACTION-RPG

122 «Комп'ютерні науки»

Подається на здобуття освітнього ступеня бакалавр

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

(підпис, ініціали та прізвище здобувача вищої освіти)

Керівник роботи
Тимонін Юрій Олександрович,
доцент кафедри КТіМС

Житомир – 2024

Висновок кафедри _____

за результатами попереднього захисту: _____

Протокол засідання кафедри _____

№ _____ від «_____» _____ 20____ р.

Завідувач кафедри _____

(науковий ступінь, вчене звання) (підпис) (прізвище, ім'я, по батькові)

«_____» _____ 20____ р.

Результати захисту кваліфікаційної роботи

Здобувач вищої освіти _____ захистив (ла)

(прізвище ,ім'я, по батькові)

кваліфікаційну роботу з оцінкою:

сума балів за 100-бальною шкалою _____

за шкалою ECTS _____

за національною шкалою _____

Секретар ЕК

(науковий ступінь, вчене звання) (підпис) (прізвище, ім'я, по батькові)

АНОТАЦІЯ

Гаврилюк А. В. Розробка гри в жанрі ACTION-RPG – кваліфікаційна робота на правах рукопису.

Кваліфікаційна робота на здобуття освітнього ступеня бакалавра за спеціальністю 122 – Комп’ютерні науки. – Поліський національний університет, Житомир, 2024.

Розробка ігор є динамічною сферою, яка постійно пропонує нові технології.

Мета кваліфікаційної роботи – полягає в розробці концепції та дизайну, створенні повноцінної гри. У вступі обґрунтовується актуальність теми роботи, формується її мета та завдання, визначається наукова новизна та практична значимість.

Ключові слова: Unreal Engine 5, RPG, гра, тактика, екшн, відеогра.

SUMMARY

Havryliuk A.V. Game developing in ACTION-RPG genre – Qualification work as a manuscript.

Qualification work for the degree of Bachelor in the specialty 122 – Computer Science. – Polissia National University, Zhytomyr, 2024.

Game development is a dynamic field that constantly offers new technologies.

The purpose of the qualification work is to develop a concept and design, to create a full-fledged game. In the introduction, the relevance of the topic of the work is substantiated, its purpose and tasks are formed, scientific novelty and practical significance are determined.

Keywords: Unreal Engine 5, RPG, game, tactics, action, videogame.

Зміст

ВСТУП.....	6
РОЗДІЛ 1. ТЕОРЕТИЧНІ ЗАСАДИ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	9
1.1. Аналіз інформаційних потреб і визначення предметної області для розробки action-гри	9
1.2. Вибір середовища програмування для action-гри.....	11
1.3. Моделювання бізнес-процесів предметної області	13
Висновки до першого розділу	14
РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ГРИ	16
2.1 Моделювання інформаційної системи action-гри.....	16
2.2. Проектування інтерфейсу інформаційної системи action-гри.....	20
2.3 Процес реалізації гри в середовищі Unreal Engine	21
Висновки до другого розділу	23
РОЗДІЛ 3. ТЕСТУВАННЯ ГРИ ТА АНАЛІЗ РЕЗУЛЬТАТІВ	25
3.1. Інструкція користувачу інформаційної системи.....	25
3.2. Тестування гри та аналіз результатів	26
Висновки до третього розділу.....	29
ЗАГАЛЬНІ ВИСНОВКИ	30
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	32
ДОДАТКИ.....	35

ВСТУП

Актуальність теми дослідження. Сьогодні Action-RPG ігри, що пропонують елементи битв та глибоку систему прокачки персонажів, здобувають все більшу популярність, адже вони дають можливість гравцям не лише битися, але й розвивати своїх героїв, роблячи їх сильнішими та могутнішими. Цей жанр пропонує динамічний геймплей, цікаву історію, завдяки чому завойовує все більший сегмент ігрової індустрії, про що свідчить успіх таких ігор, як Elden Ring, The Witcher 3: Wild Hunt, God of War (2018) та Horizon Zero Dawn.

Стрімкий розвитку технологій уможливив створення більш складних та красивих Action-RPG ігор. Сучасні графічні двигуни, фізичні симулятори та штучний інтелект дозволяють розробникам реалізовувати динамічний геймплей з деталізованими світами та захоплюючі історії.

Нині українська геймдев-індустрія переживає бурхливий ріст, що зумовлює зростання попиту на ігри, створені в Україні. Цей феномен пояснюється тим, що чимало українських розробників мають світове визнання та створюють ігри високої якості. У свою чергу геймери все частіше цікавляться іграми, які розкривають унікальні аспекти української ідентичності. Action-RPG ігри можуть розповідати про будь-які теми та світи, від фентезі та наукової фантастики до альтернативної історії та реального світу. Це робить жанр дуже гнучким та дозволяє розробникам реалізовувати свої творчі ідеї.

Таким чином, розробка Action-RPG гри є актуальною, цікавою та має великий потенціал для реалізації успішного проекту.

Мета кваліфікаційної роботи полягає в розробці концепції та дизайну, створенні повноцінної гри, дослідженні аспектів розробки Action-RPG ігор, аналізі існуючих ігор в цьому жанрі, порівнянні різних підходів до розробки, вивченні специфіки розробки Action-RPG ігор в Україні.

Завдання кваліфікаційної роботи: аналіз ринку та визначення цільової аудиторії гри; моделювання бізнес-процесів предметної області та інформаційної системи action-гри; проектування інтерфейсу; розробка концепції та прототипу гри; тестування розробленої гри.

Предметом дослідження є процес розробки гри в жанрі Action-RPG. **Об'єктом дослідження** є гра в жанрі Action-RPG, яка буде розроблена в рамках курсової роботи.

Для дослідження теми було використано наступні **методи**: *теоретичні* (аналітичний, системний та порівняльний аналіз допоміг сформулювати загальні уявлення про актуальність обраної теми; виділити ключові аспекти для розробки власної гри в жанрі Action-RPG на основі дослідження аналогів); *емпіричні* (тестування – допомогло виявити проблеми з геймплеєм); *моделювання* (використано для створення математичної моделі для дослідження балансу гри та визначення складності; опису бізнес-процесів предметної області та інформаційної системи Action-гри) та *експерименту* (дослідження різних механік геймплею, для визначення найефективнішої для жанру Action-RPG).

Практичне значення отриманих результатів. Результати дослідження можуть бути використані іншими розробниками ігор як джерело інформації про процес розробки Action-RPG ігор.

Наукова новизна полягає у розробленій Action-RPG грі, яка може бути конкурентноздатною відносно інших представників жанру. **Унікальністю** кваліфікаційної роботи є розробка Action-RPG гри з новаторськими механіками геймплею, що полягають у бойовій системі, яка буде змінюватись в залежності від складності та типу супротивників.

Роботу апробовано на двох Всеукраїнських науково-практичних конференціях та опубліковано тези "Інноваційні тенденції у розробці комп'ютерних ігор" та "Дизайн ландшафтів для комп'ютерних ігор" у збірниках конференцій:

1. Гаврилюк А. Інноваційні тенденції у розробці ігор // Безпека, технології, інновації: нові горизонти : збірник праць учасників міжфакультетської науково-практичної інтернет-конференції здобувачів вищої освіти і молодих вчених, 15 листопада 2023 р. – Житомир : Поліський національний університет, 2023. – 69 с. – С. 59–60.

2. Гаврилюк А., Ковальчук М. Дизайн ландшафтів для комп'ютерних ігор // Дизайн, візуальне мистецтво та творчість: сучасні тенденції та технології : матеріали II Міжнародної науково-практичної конференції, 12 грудня 2023 р. – Запоріжжя : Запорізький національний університет, 2023. Том 2. 150 с. – С. 63–65.

Структура та обсяг роботи. Кваліфікаційна робота складається з вступу, трьох розділів, висновку та списку використаних джерел. Загальний обсяг роботи становить 31 сторінку.

РОЗДІЛ 1. ТЕОРЕТИЧНІ ЗАСАДИ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Аналіз інформаційних потреб і визначення предметної області для розробки action-гри

Для кращого розуміння теми дослідження і визначення предметної області для розробки Action-RPG проаналізуємо наступні аналоги (The Witcher, Dark Souls 3, Far Cry 6, Baldur's Gate 3, Skyrim, Mass Effect), які визнані кращими в цьому жанрі за останній час [1],[2],[3],[4],[5],[6] за наступними критеріями: *жанр* (відкритий світ; ігри з сюжетним акцентом, шутери з лутом та кооперативним режимом; Roguelike-ігри; веселі та легкі ігри); *цільова аудиторія*; *унікальні особливості* (складний сюжет, політичні інтриги, моральний вибір, нелінійне проходження, система розвитку персонажа, епічні битви тощо). Кожна з проаналізованих ігор має свої унікальні механіки, сеттинг, сюжет та атмосферу.

Важливо також проаналізувати існуючі Action-RPG ігри, щоб зрозуміти, які функції та елементи геймплею їм найбільше подобаються цільовій аудиторії.

Action-RPG гра має фентезі тематику, а саме елементи магії, альтернативну історію.

Сюжет розповідає про трагічну боротьбу вимираючого людства проти відьм. Головний герой є людиною, персонаж створюваний (раса, стать, клас персонажа), антагонізмом є головна відьма.

Світ схожий на середньовіччя з елементами магічних предметів, мечі, луки, щити, броня, магічні посохи, рівнини та маленькі села-поселення, де живуть люди.

Геймплей має основні Action-RPG-механіки, такі, як розвинуту бойову систему, дерево навичок, навички розвиваються, використовуючи досвід. Вороги теж можуть мати рівень і від цього залежить їх кількість здоров'я та шкоду йому в бою, відносно рівня гравця. Гра має відкритий світ, який можна досліджувати вільно з самого початку гри, паралельно беручи та виконуючи квести

Одним із важливих кроків у теоретичному аналізі інформаційних потреб є визначення цільової аудиторії для Action-RPG гри. Цільова аудиторія – це група

людей, на яку буде орієнтована гра. Її характеристики, такі як вік, стать, інтереси, рівень досвіду в іграх, слід враховувати при розробці гри.

При аналізі цільової аудиторії було виявлено, що вона є досить широкою і різноманітною. Проводилось опитування в соц. мережах, за допомогою якого можна виділити декілька основних груп:

- Чоловіки віком 18-35 років. Це найбільша група гравців. Їх зазвичай цікавлять динамічні та захоплюючі ігри з акцентом на бойову систему та прокачування персонажа
- Жінки віком 18-35 років. Ця група гравців швидко стає популярною. Їх може цікавити як динаміка, так і сюжет
- Підлітки 12-17 років. Ця група цікавиться іграми із більш простим геймплеєм та яскравою графікою
- Дорослі люди старше 35 років. Має менше вільного часу, тому дають перевагу іграм з більш коротким ігровим процесом

Отже, у межах кваліфікаційної роботи будемо розробляти динамічну Action-гру з камерою від третього лиця з RPG елементами та бойову систему, яка є основною унікальністю кваліфікаційної роботи та буде змінюватись в залежності від складності та типу супротивників. Мета гри полягатиме у проходженні сюжету з обов'язковим та необов'язковим дослідженням окремих елементів ігрового світу, розвиток тактичних, ментальних, психологічних, навичок.

Проведене дослідження проблемної області і аналіз аналогів дозволяє розробити бойову систему, систему прокачки персонажа, варіанти дослідження відкритого ігрового світу та інтерфейс користувача, графіку, дизайн та анімації які будуть описані у наступних підрозділах кваліфікаційної роботи.

1.2. Вибір середовища програмування для action-гри

Середовище програмування (IDE) – це програмний пакет, який допомагає програмістам писати, редагувати, налагоджувати та тестувати код[7]. IDE зазвичай включають в себе такі функції:

- Редагування коду – IDE забезпечують базові функції редагування тексту, такі як автозаповнення, підсвічування синтаксису та виділення помилок.

- Налагодження – IDE містять потужні інструменти для налагодження коду, такі як точки зупинки, дебаггер та перегляд стека.

- Рефакторинг – IDE забезпечують інструменти для рефакторингу коду, такі як перейменування, вилучення та винесення [8].

- Тестування – IDE містять інструменти для тестування коду, такі як одиничні тести, інтегровані тести та тести інтерфейсу користувача.

Вибір середовища програмування є важливим рішенням для кожного програміста. Існує безліч різних IDE, які пропонують різні функції та можливості.

При виборі середовища програмування слід враховувати такі фактори:

- мова програмування – перш за все, слід вибрати IDE, який підтримує мову програмування, з якою ви збираєтеся працювати;

- платформа – IDE також можуть бути доступні для різних платформ, таких як Windows, macOS та Linux;

- функціональність – IDE пропонують різні набори функцій, деякі IDE є простими і легкими у використанні, тоді як інші є більш потужними і багатофункціональними;

- вартість - деякі IDE є безкоштовними, тоді як інші платні.

Існують два основних типи середовищ програмування:

- інтегровані середовища розробки (IDE) – IDE є найбільш популярним типом середовищ програмування. Вони пропонують широкий спектр функцій, які допомагають програмістам писати, редагувати, налагоджувати та тестувати код;

– редактори коду – редактори коду є більш простими і легкими у використанні, ніж IDE. Вони зазвичай не включають в себе такі функції, як налагодження та рефакторинг;[9]

Серед популярних середовищ програмування для різних мов програмування можна виділити такі:

- на C++: JetBrains Rider, Visual Studio 2022, Eclipse for C++;
- на Java: IntelliJ IDEA, Eclipse, Visual Studio Code;
- на Python: PyCharm, Visual Studio Code, Sublime Text;
- на JavaScript: Visual Studio Code, Atom, Sublime Text.

Для вибору середовища здійснили аналіз існуючих середовищ та текстових редакторів, таких як: JetBrains Rider, Visual Studio 2022, Eclipse for C++, Visual Studio Code є найпопулярнішими варіантами [10],[11],[12],[13].

В таб. 1.1 додатку А, було наведено порівняння різних IDE для розробки JetBrains Rider – це потужний і багатофункціональний IDE, який підтримує широкий спектр мов програмування і платформ. Він має розширені можливості налагодження, автодоповнення, рефакторингу та тестування.

Visual Studio 2022 – це ще один потужний IDE, який підтримує широкий спектр мов програмування і платформ. Він має багато спільних функцій з JetBrains Rider, але не такий розширений.

Eclipse for C++ – це безкоштовний IDE, який підтримує C++ і деякі інші мови програмування. Він має базовий набір функцій, але не такий потужний, як інші IDE в цьому порівнянні.

Visual Studio Code – це легкий і гнучкий IDE

Для написання коду у UE5 ми будемо використовувати Microsoft Visual Studio 2022 та вбудовані інструменти Unreal Engine 5, Blueprints. Серед безкоштовних Visual Studio 2022 є найпотужнішим IDE для написання важких програм, їх тестування та рефакторингу.[14]

1.3. Моделювання бізнес-процесів предметної області

Враховуючи, що ігри жанру action можуть мати різні варіації і піджанри, у курсовій роботі буде розроблено гру у тематиці open-world action RPG [15],[16],[17].

Однією з головних переваг даного жанру є варіативність проходження та стратегії бойової системи, тому бізнес-процеси предметної області будуть базуватися на виборі тактики ближнього бою, вона буде унікальною гейм-дизайнерською стратегією у грі.

Інформаційна система тактики ближнього бою наведена на рисунку 1.1 у додатку Б.

Система приймає дані про наявні ресурси та тактики ближнього бою, які доступні користувачеві та на виході ми отримуємо перемогу у бою. Вона отримує вхідні дані у вигляді інформації про тактики, закони, методи ведення бою, ресурси, загрози та середовище. Ці дані використовуються для аналізу ситуації та розробки плану ведення бою. План включає в себе вибір відповідних тактик. Коли бойові дії починаються, команди повинні бути координовані та виконувати тактики, щоб перемогти.

Діаграма має три рівні деталізації. На першому рівні (A0) представлена загальна схема системи. На другому рівні (A1) деталізуються входи, виходи та процеси системи. На третьому рівні (A2) деталізуються процеси системи ще більше.

У контексті діаграми, "виконання бойових тактик" означає планування, підготовку та реалізацію дій, необхідних для досягнення бойових цілей.

Декомпозицію моделі IDEF0 зображено на рисунку 1.2 у додатку Б.

Декомпозиція описує, як саме обирається тактика для перемоги суперника в ближньому бою. Спочатку ми аналізуємо загрозу, для того, щоб розрахувати, чи існують взагалі шанси на перемогу в бою. Далі ми аналізуємо середовище, це потрібно для того, щоб вибрати правильну позицію на полі бою для отримання переваги над суперником. Далі ми використовуємо ресурси, які наявні на полі бою задля отримання додаткової зброї, продумування пасток та використання

середовища безпосередньо для нанесення удару по супернику. Далі, обрахувавши ці фактори ми обираємо виграшну тактику та починаємо її реалізовувати.

Задля кращого уявлення про роботу ІС «Тактика ближнього бою» було сформовано IDEF3 модель, наведену на рисунку 1.3 у додатку Б.

Модель «Тактика ближнього бою» описує тактичні вибори, які впливають на планування та реалізацію тактики ближнього бою, у залежності від присутності команди чи зброї на полі бою. Спочатку, аналізуючи тактику бою та загрозу, звертається увага на те, чи є у когось з бійців холодна чи стрілецька зброя. Зважаючи на це, здійснюється рішення планування тактики залежно від наявності такої.

Далі, якщо ближній бій командний, тобто, з одної і/або з іншої сторони наявно два чи більше бійців, то здійснюється координування командних дій, задля ефективнішої хімії між бійцями на полі бою. Інакше можна, спланувавши тактику повністю та реалізовувати її безпосередньо в бою. У першому розділі проведено аналіз інформаційних потреб та визначено предметну область дослідження для створення комп'ютерної гри в жанрі екшн. Геймдизайн є ключовим аспектом створення будь-якої комп'ютерної гри, і для розробки успішної гри в жанрі екшн необхідно створити захоплюючий сценарій, цікавий геймплей та привабливих персонажів.[18]

Висновки до першого розділу

На основі аналізу існуючих ігрових рушіїв, оптимальним вибором є Unreal Engine 5, оскільки має ряд переваг, які роблять його ідеальним для розробки високоякісних комп'ютерних ігор, включаючи: потужний двигун рендерингу, який дозволяє створювати вражаючі; візуальні ефекти та фотореалістичні графічні зображення; підтримка розробки для різних платформ; безкоштовність для використання для початкового рівня; велика та активна спільнота, яка надає підтримку та допомогу; широка документація та навчальні матеріали, які

полегшують вивчення двигуна; багато вбудованих готових компонентів та бібліотек, які спрощують розробку та прискорюють час виробництва; система Blueprints, яка дозволяє розробникам без програмувального досвіду створювати складні системи та геймплей за допомогою візуального програмування; потужні інструменти для реалізації штучного інтелекту та навчання машин.

Було розроблено інформаційну систему по темі «Тактика ближнього бою», як одне з основних гейм-дизайнерських рішень у грі. Тактика є важливою складовою open-world ігор, яка використовується задля реіграбельності проекту та залучення нових гравців.

РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ГРИ

2.1 Моделювання інформаційної системи action-гри

Інформаційні системи action-ігор є складними системами, які включають широкий спектр компонентів, таких як графічний та фізичний движок, штучний інтелект, мережеве програмне забезпечення та багато іншого. Моделювання є важливим завданням, оскільки дозволяє краще зрозуміти структуру та поведінку системи, а також виявити потенційні проблеми. Для розуміння роботи ІС гри у жанрі action було розроблено UML-діаграму станів, наведену у рис. 2.1 додатку В.

Діаграма станів, представляє просту модель поведінки комп'ютерної гри в жанрі екшн. Гравець керує персонажем, який може перебувати в одному з п'яти станів: стартовий екран (стан, в якому гравець може вибрати рівень для гри); головне меню (це стан, в якому гравець може вибрати опції гри, такі як рівень складності або звук); геймплей (це стан, в якому гравець контролює персонажа, щоб переміщатися по рівню, боротися з ворогами та виконувати завдання). пауза (це стан, в якому гравець може призупинити гру, щоб переглянути карту або змінити налаштування); кінець гри (це стан, в якому гравець переміг у грі або програв.)

Діаграма показує, як гравець може переміщатися між цими станами. Наприклад, гравець може перейти від стартового екрану до головного меню, натиснувши кнопку "Меню". Гравець може перейти від головного меню до геймплея, вибравши рівень для гри.

Діаграма також показує, які події можуть призвести до переходу між станами. Наприклад, гравець перейде до геймплея, коли натисне кнопку "Відтворити". Гравець перейде до паузи (головне меню), коли натисне паузу.

Гравець перейде до кінця гри, коли переможе або програє.

Для розуміння як працює головне меню інформаційної системи, та які є можливі дії, було розроблено діаграму активності, зображену на рис. 2.2, додатку В.

Ця діаграма активностей представляє поведінку головного меню

комп'ютерної гри. Головне меню – це місце, де гравець може почати гру, вибрати опції гри або вийти з гри.

Діаграма складається з чотирьох станів:

– Початковий стан - це стан, в якому діаграма починає роботу. У цьому стані меню відображається на екрані.

– Стан вибору рівня - це стан, в якому гравець може вибрати рівень для гри. У цьому стані гравець може переглядати список рівнів і вибрати один з них.

– Стан вибору опцій - це стан, в якому гравець може вибрати опції гри. У цьому стані гравець може змінити рівень складності, звук або інші параметри гри.

– Стан виходу - це стан, в якому гравець виходить з гри. У цьому стані гра завершується.

Діаграма також показує, як гравець може переміщатися між цими станами. Наприклад, гравець може перейти від початкового стану до стану вибору рівня, натиснувши кнопку "Відтворити". Гравець може перейти від стану вибору рівня до стану вибору опцій, натиснувши кнопку "Опції". Гравець може перейти від стану вибору опцій до початкового стану, натиснувши кнопку "Назад". Гравець може перейти від будь-якого стану до стану виходу, натиснувши кнопку "Вихід".

Коли діаграма починає роботу, вона переходить у початковий стан. У цьому стані меню відображається на екрані. Гравець може вибрати рівень для гри, натиснувши кнопку "Відтворити". Це призведе до переходу до стану вибору рівня.

У стані вибору рівня гравець може переглядати список рівнів і вибрати один з них. Коли гравець вибирає рівень, діаграма переходить до стану гри.

У стані гри гравець може грати в гру. Коли гравець хоче повернутися до головного меню, він може натиснути кнопку "Меню". Це призведе до переходу до стану вибору опцій.

У стані вибору опцій гравець може змінити рівень складності, звук або інші параметри гри. Коли гравець вносить зміни, діаграма зберігає ці зміни. Коли

гравець хоче повернутися до головного меню, він може натиснути кнопку "Назад". Це призведе до переходу до початкового стану.

Коли гравець хоче вийти з гри, він може натиснути кнопку "Вихід". Це призведе до переходу до стану виходу. У цьому стані гра завершується.

На Рис. 2.3, додатку В представлено UML-діаграму класів інформаційної системи

UObject є базовим класом для всіх об'єктів у Unreal Engine. Він забезпечує такі можливості, як зберігання, відстеження та ідентифікація об'єктів. Наприклад, UObject використовується для зберігання змінних, які використовуються для управління поведінкою об'єкта.

AActor є базовим класом для всіх акторів, які можуть існувати у світі Unreal Engine. Він забезпечує такі можливості, як колізія, анімація та вхідні події. Наприклад, AActor використовується для визначення того, чи може актор переміщуватися через інший актор.

APawn є базовим класом для всіх гравців та штучного інтелекту у світі Unreal Engine. Він забезпечує додаткові можливості для павука, такі як управління входом, переміщення та анімація. Наприклад, APawn використовується для визначення того, як гравець або штучний інтелект контролює рух павука.

ACharacter є базовим класом для всіх персонажів у світі Unreal Engine. Він забезпечує додаткові можливості для персонажа, такі як рух, стрибки та анімація. Наприклад, ACharacter використовується для визначення того, як персонаж рухається, стрибає та використовує зброю.

UGameModeBase є базовим класом для всіх режимів гри. Він забезпечує базові можливості для режиму гри, такі як запуск гри та завершення матчу. Наприклад, UGameModeBase використовується для визначення того, коли починається і закінчується гра

Для наглядності, чим займається розробник, а чим користувач (гравець) – створено діаграму прецедентів, зображену на рис. 2.4 додатку В.

У цій ІС є два ключових актора: гравець та розробник. Гравець, «відповідальний» за ігровий досвід, має можливість проходження захоплюючих

квестів, що додає глибину його взаємодії з ігровим світом. Він також може досліджувати різні аспекти гри, досліджуючи широкий світ гри.

З іншого боку, розробник виступає як творча сила, що займається створенням контенту. Його функції включають створення різноманітних сюжетних ліній, додавання нових аспектів гри та розробку механік та написання коду гри.

Щоб відобразити взаємодію користувача з інформаційною системою було побудовано діаграму послідовності, зображено на рис. 2.5, додаток В.

Надана діаграма послідовностей описує процес запуску та гри в комп'ютерну гру. Діаграма складається з чотирьох основних елементів:

Користувач – це людина, яка грає в гру.

Інтерфейс – це графічний інтерфейс користувача, який дозволяє гравцеві взаємодіяти з грою.

Ігровий рушій – це програмний компонент, який відповідає за відтворення ігрового світу та обробку подій.

Модуль збереження – це програмний компонент, який відповідає за збереження та завантаження ігрового прогресу.

Діаграма показує, як ці елементи взаємодіють один з одним, коли гравець запускає гру та починає грати.

Процес запуску гри починається з того, що користувач запускає гру. Інтерфейс отримує запит від користувача і передає його ігровому рушію. Ігровий рушій запускає гру і завантажує ігровий світ. Після того, як гра запущена, гравець може взаємодіяти з грою за допомогою інтерфейсу. Інтерфейс отримує вхідні дані від гравця і передає їх ігровому рушію. Ігровий рушій обробляє вхідні дані і змінює стан ігрового світу відповідно.

Ігровий рушій відповідає за відтворення ігрового світу. Він генерує кадри зображення, відтворює звуки та обробляє події, такі як рух гравця або взаємодія з об'єктами в ігровому світі. Модуль збереження відповідає за збереження та завантаження ігрового прогресу. Він дозволяє гравцеві зберігати свій прогрес у грі і відновлювати його, якщо гравець закриє гру або запустить її повторно.

Конкретні дії, які виконує кожен елемент у діаграмі, визначаються конкретними вимогами гри. Наприклад, для гри в жанрі action гравець може використовувати інтерфейс, щоб керувати рухом свого персонажа, стріляти з зброї та використовувати предмети. Ігровий рушій буде відповідати за відтворення графіки та звуку, обробку руху персонажа гравця та обробку влучень у цілі. Модуль збереження буде відповідати за збереження положення персонажа гравця, запасу боєприпасів та інших ігрових даних.

Наведена діаграма є лише загальним описом процесу запуску та гри в комп'ютерну гру. Конкретні деталі можуть відрізнятись залежно від конкретних вимог гри.

2.2. Проектування інтерфейсу інформаційної системи action-гри

Для належної функціональності інформаційної системи необхідно створити користувацький інтерфейс. Гра складається з інтерфейсів з якими користувач взаємодіє. Для розробки інтерфейсів у Unreal Engine, використовується Widget Blueprint, його функціонал забезпечує розробку UI інформаційної системи на Unreal Engine 5 [19]. Додаючи елементи у віджет, можливо зробити унікальний інтерфейс, наприклад HUD чи інтерфейс головного меню, тощо, приклади інтерфейсу інформаційної системи наведено у додатку Г на рисунках 2.6, 2.7 та 2.8. [20] На рис. 2.6 дод. Г зображено головне меню ІС.

Користувач має можливість обрати одну з запропонованих дій для подальшої взаємодії з грою, або для виходу з неї.

З гри можливо вийти у меню паузи, яке зображено на рис. 2.7. дод. Г

Меню містить кілька кнопок, які дозволяють користувачеві:

Продовжити гру – ця кнопка дозволяє користувачеві повернутися до гри з того місця, де він зупинився.

Опції – ця кнопка відкриває меню налаштувань, де користувач може змінити різні параметри гри, наприклад, графіку, звук та елементи керування.

Вихід – ця кнопка дозволяє користувачеві вийти з гри та повернутися до головного меню.

На рис. 2.8 дод. Г зображено HUD, який показує кількість життів, досвіду, кількість життів ворогів та босів, підказки до елементів геймплею.

Елементи HUD:

Життя – показує кількість життів, які залишилися у гравця. Він може бути представлений у вигляді сер сердець або іншого візуального індикатора. На зображенні видно, що у гравця залишилося 3 життя.

Досвід – демонструє поточний рівень досвіду гравця. На рис. 2.8 дод. Г видно, що у гравця 75/100 очок досвіду.

Вороги – показує кількість ворогів, які залишилися в поточному рівні або завданні. Він може бути представлений у вигляді числового значення або візуального індикатора. На зображенні видно, що у гравця 1 ворог.

Боси – вказує на кількість босів, які залишилися в грі.

Підказки – дозволяє скористатися підказками та інструкціями щодо елементів геймплею. На рис. 2.8 дод. Г продемонстровано підказку "Натисніть E, щоб взаємодіяти".

Таким чином, HUD надає гравцю важливу інформацію, яка допомагає йому приймати рішення та просуватися в грі. Наприклад, знаючи кількість життів, що залишилися, гравець може вирішити, чи йти на ризик або відступити. Знаючи кількість ворогів, що залишилися, гравець може спланувати свою стратегію.

2.3 Процес реалізації гри в середовищі Unreal Engine

Після визначення концепцій та ідеї гри, потрібно розробити Core-механіки, такі як рухи персонажа та анімації, Blueprint рухів персонажа зображений на рис. 2.9 дод. Д.

Анімації персонажа розроблені за допомогою Animation Blueprint, який включає в себе State Machine (стейт-машину), за допомогою стейт машини

змінюються стани та умови, за яких персонаж відіграє ту чи іншу анімацію[21],[22]. Головну стейт-машину зображено на рис. 2.10 дода. Д. Анімації та моделі були попередньо імпортовані з інтернет-джерел, деякі були розроблені самостійно. На початковому етапі розробки був розроблений рух, присід персонажу та наклон відносно повороту. Скріншоти анімацій наведено у додатку Д, рис. 2.11, 2.12. Деякі механіки, такі як перестрибування перешкод було розроблено за допомогою механіки Raycasts (рейкасти)[23]. Рейкасти дозволяють створювати деякий абстрактний луч, який «бачить» об'єкти на сцені.

Наступним корком було створення системи «швидкої нейтралізації», яка дозволяє вирубити ціль на деякий час, підійшовши до неї ззаду, для цього була імпортована анімація і звуки, створена окрема кнопка за допомогою Input Action, Input Action дозволяє призначати кнопки на дії, прописані в коді[24]. Blueprint та підказка зображені на рис. 2.13, 2.14 дод. Д. Механіка зображена на рис. 2.15.

Після створення рухів та анімацій, було створено систему статистики персонажів та HUD. Було розроблено шкали здоров'я, витривалості та досвіду і рівнів. Для цього було розроблено 8 функцій, для прибавки здоров'я, досвіду та стаміни, та для відбавки їх, також окремо була розроблена функція зміни рівнів персонажа. Функції зображені на рис. 2.16 дод. Д. Додатково було розроблено HUD на екрані за допомогою віджетів, більш детально описаних у розділі 2.2. HUD зображено на рис. 2.17 дод. Д. Разом з тим було розроблено систему нейтралізації персонажу, коли здоров'я нульове, зображено на рис. 2.18 дод. Д.

Далі було розроблено систему бою мечем, включаючи комбо та поодинокі удари, зображено на рис. 2.19 дод. Д. Демонстрацію blueprint системи атаки розміщено на рис. 2.20 додатку Д. Разом з тим було додано систему здоров'я ворогам та ефекти крові, захват цілі і пережат, зображений на рис. 2.21 дод. Д.

Система інвентаря розроблена за допомогою віджетів та структур, які дозволили поділити речі на підтипи – мечі, броня, далекобійна зброя, щит, демонстрація розміщена на рис. 2.22 та рис. 2.23 додатку Д.

Наступним корком, була розроблена системи штучного інтелекту. Штучний інтелект в Unreal Engine 5 створюється за допомогою дерева задач (рис.

2.24 дод. Д), кожна з яких наповнена своїм функціоналом, розроблені такі задачі як патрулювання території, можливість бою штучним інтелектом, переслідування та зачистки. Класи задач продемонстровані на рис. 2.25, 2.26, 2.27 та 2.28 дод. Д.

Система паркура представляє собою систему рейкастинга, вона визначає точку, де закінчується та починається перешкода під кутом та за допомогою визначеної кнопки переходить у режим паркура. Демонстрація зображена на рис. 2.29 та 2.30 додатку Д.

Система квестів розроблена за допомогою комбінації віджетів (UI елементів) та використовуючи програмування. Квести були поділені на різні підтипи – головні квести, побічні, з предметами та ключами. Blueprint системи квестів та демонстрація розміщені на рис. 2.31 та 2.32 дод. Д.

Система броні та далекобійної зброї, були імпортовані моделі броні, луку і стріл, налаштований приціл та функціонал стрільби по ворогам. Blueprint та демонстрація стрільби зображена на рис. 2.33, рис. 2.34 додатку Д.

Система відкритого світу була розроблена після відточення головних ігрових механік. Створена вона була за допомогою ландшафту, вбудованої технології Unreal Engine 5 та використання карти висот, яка допомогла надати ландшафту реалістичний вигляд. Разом з цим була створена система дня і ночі та декілька основних локацій. Демонстрація редактору ландшафту UE5 та готового ландшафту зображена на рис. 2.35 та 2.36 додатку Д.

У кінці розробки було створено головне меню та меню паузи, систему збереження та загрузки. Головне меню та меню паузи зображено на рис. 2.37 та 2.38 дод. Д. У процесі тестування гри, було додано ще декілька механік, таких як скін персонажа, різні ефекти, нові анімації, локації та ін.

Висновки до другого розділу

У другому розділі була змодельована робота інформаційної системи, використовуючи різні види UML-діаграм (зокрема, діаграми стану, активності,

класів, послідовності та прецедентів). За допомогою моделювання роботи ІС можемо зрозуміти, як працює інформаційна система, описати бізнес-процеси, проаналізувати та впровадити його, охарактеризовано інтерфейс інформаційної системи action гри на движку Unreal Engine, що є способом взаємодії користувача з програмним забезпеченням. Він складається з елементів, таких як кнопки, текстові поля, меню та значки. Інтерфейс користувача визначає, як користувачі можуть взаємодіяти з ним. При розробці інтерфейсу користувача важливо враховувати потреби користувачів.

Було описано процес реалізації гри у рушію, розробка рухів, анімацій, бойової системи, інтерфейсів користувача у грі, відкритий світ, квести, штучний інтелект

РОЗДІЛ 3. ТЕСТУВАННЯ ГРИ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

3.1. Інструкція користувачу інформаційної системи

Інструкція для користувача є важливою частиною будь-якої ігрової програми. Екшн-ігри часто мають складні правила та механіки, які важко зрозуміти без інструкцій. Інструкція може допомогти гравцям швидко та легко ознайомитися з основами гри, а також дізнатися про більш складні функції та можливості.

Екшн-ігри часто мають високий темп і вимагають від гравців швидко діяти. Саме тому, інструкція може допомогти гравцям швидко ознайомитися з основами гри, зрозуміти правила, дізнатися про додаткові функції та можливості гри.

Сучасні екшн-ігри, як правило, вимагають потужної комп'ютерної системи для свого запуску. Це пов'язано з тим, що вони часто використовують складні графіку та ефекти, які потребують значних ресурсів.

Вимоги до системи

Для його запуску на мінімальних настройках графіки необхідні такі компоненти:

- Процесор: Intel Core i5 9400N або його еквівалент Ryzen 5 1600.
- Відеокарта: Nvidia GeForce RTX 2050.
- Оперативна пам'ять: 12 Гб.
- Накопичувач: 120 Гб (жорсткий диск або SSD).

Для оптимального відображення графіки та отримання максимальної кількості кадрів в секунду рекомендується використовувати наступні компоненти:

- Процесор: Intel Core i5 10600K(F) або його еквівалент Ryzen 5 3600X.
- Відеокарта: Nvidia GeForce RTX 3060.
- Оперативна пам'ять: 16 Гб.
- Накопичувач: 120 Гб SSD.

Запуск гри

Для запуску гри необхідно виконати наступні кроки:

- Завантажте та встановіть гру.
- Знайдіть файл гри з розширенням .exe.
- Подвійним клацанням миші запустіть файл.
- Дочекайтеся завантаження гри.
- Управління.

Стандартне управління в екшн-іграх – для пересування використовуються клавіші WASD, для стрибка — Space, для взаємодії — E, для перемикання зброї/екіпіровки — Enter і для виходу в меню паузи — I. Система бою складається з комбо та поодиноких ударів, для удару використовується ЛКМ, для вистрілу з луку теж використовується ЛКМ, коли лук в руках. Для присіду використовується Left Ctrl, ПКМ використовується для тихої нейтралізації цілі, коли персонаж знаходиться позаду неї, швидкий біг — Left Shift, захват цілі Tab, пережат X, кидати камінь T, лізти по стіні V.

Для гравців, які вперше знайомляться з екшн-іграми, доступний базовий тьюторіал, що ґрунтується на основній механіці гри та управлінні персонажа. Він допоможе легко влитися в ігровий світ та його особливості.

3.2. Тестування гри та аналіз результатів

Тестування відеоігор є критично важливим етапом розробки, який гарантує випуск високоякісного, приємного та безпомилкового продукту. Цей комплексний процес спрямований на виявлення та усунення дефектів, щоби покращити загальний ігровий досвід. Завдяки ретельному тестуванню розробники можуть забезпечити якість та виправити помилки гри до релізу, покращити ігровий процес, щоби зробити гру більш захопливою, знизити витрати ресурсів після релізу та підвищити репутацію.

Функціональне тестування [25] — один із видів тестування, спрямованого на перевірку відповідностей функціональних вимог ПЗ його реальним характеристикам. Основним завданням функціонального тестування є підтвердження того, що програмний продукт, який розробляється, володіє усім

необхідним замовнику функціоналом. Функціональне тестування продемонстровано на рис. 3.1 дод. Е. У верхній частині вікна гри зображено HUD, який містить інформацію про стан гравця, його прогрес, кількість ворогів, босів та інші дані. У центрі вікна гри зображено ігрове поле, де відбувається дія. На ньому видно гравця, ворогів, бонуси, перешкоди та інші елементи геймплею. У нижній частині вікна гри видно панель керування, де розташовані кнопки для пересування, стрільби, використання предметів та виконання інших дій.

На зображенні видно, що більшість функцій гри пройшли тестування успішно. Однак виявлено кілька помилок, які потребують виправлення. Наприклад, у описі помилки "Неможливо використовувати предмет X" зазначено, що гравець не може використовувати певний предмет у грі.

Функціональне тестування цієї гри може включало наступні кроки:

1) *Тестування геймплею* – було перевірено: чи може персонаж гравця вільно пересуватися по ігровому світу; чи може персонаж гравця атакувати ворогів різними способами; чи можуть вороги атакувати персонажа гравця; чи можна збирати предмети та використовувати їх бонуси; чи може персонаж гравця втрачати життя та гинути; чи можна завершити рівень або гру.

2) *Тестування інтерфейсу* – включало перевірку наступної інформації: чи відображає HUD всю необхідну інформацію; чи можна використовувати меню для налаштування параметрів гри; чи можна завантажити або розпочати нову гру.

3) *Тестування технічних характеристик* – полягало у перевірці того, чи працює графіка без лагів або артефактів; чи чіткий звук і не спотворюється в процесі гри.

На основі результатів функціонального тестування можемо зробити висновок про те, чи гра відповідає всім установленим функціональним вимогам.

Тестування продуктивності [26] — це комплекс типів тестування, метою якого є визначення працездатності, стабільності, споживання ресурсів та інших атрибутів якості додатка в умовах різних сценаріїв використання та навантажень. Тестування продуктивності дозволяє знаходити можливі вразливості та недоліки

в системі з метою запобігання їх негативному впливу на роботу програми в умовах використання. У нашому випадку, тестування продуктивності було проведено за допомогою виміру кадрів у різних ігрових ситуаціях, результати тестів представлені у дод. Е. Це дало змогу оцінити, наскільки плавно та без затримок працює гра на різних комп'ютерах.

Аналіз результатів показав, що FPS в цілому знаходиться на комфортному рівні. Більшість часу гра працює плавно, без затримок. FPS може трохи знижуватися в деяких ситуаціях – в бойових сценах, або в сценах з великою кількістю об'єктів на екрані. FPS залежить від роздільності та налаштувань графіки. Чим вища роздільність та якість графіки, тим нижчий FPS.

Гра загалом добре оптимізована для різних конфігурацій комп'ютерів. Навіть на комп'ютерах з не найпотужнішим апаратним забезпеченням гра працювала без суттєвих проблем.

Тестування сумісності [27] — тестування програмного забезпечення, призначене щоб побачити, наскільки сумісне програмне забезпечення з певним середовищем — операційною системою, платформою чи обладнанням. Гра розроблена тільки під ПК платформи з можливістю підключення іншого маніпулятора (геймпад). Гра була протестована на декількох популярних ОС, таких як Windows, macOS та Linux, і не було виявлено жодних проблем з сумісністю. Вона не перевантажує процесор, пам'ять або інші ресурси системи, навіть на слабких комп'ютерах. Гра без проблем розпізнавала та працювала з геймпадами різних виробників. Однак, тестування сумісності було проведено на обмеженому наборі ПК-платформ та периферійних пристроїв. Можливо, що на деяких інших платформах або пристроях гра працюватиме не так добре.

У цілому, можна зробити висновок, що гра стабільно працює, оптимізована для різних конфігурацій комп'ютерів, не має суттєвих проблем з функціоналом, підключенням геймпаду та сумісністю..

Висновки до третього розділу

У третьому розділі описано інструкцію з управління грою для користувача. Вона призначена для чіткого розуміння користувачем світу гри та ігрових механік, які можна використовувати у подальшому. Описано методи тестування, які застосовувались для фінального етапу розробки, такі як функціональне тестування, задля розуміння відповідностей гри реальним вимогам; тестування продуктивності, яке включало в себе тестування ресурсовитрат гри у різних ситуаціях; тестування сумісності для розуміння принципу взаємодії розробленої гри та різних платформ.

ЗАГАЛЬНІ ВИСНОВКИ

У межах виконання кваліфікаційної бакалаврської роботи були виконані всі поставлені завдання, спрямовані на розробку Action-RPG гри.

Здійснено аналіз ринку та визначено цільову аудиторію гри. Спираючись на опитування проведене серед різних шарів суспільства, аналіз інформаційної системи та складових, які вона в себе включає, такі як: конкурентний аналіз, доречність використання того чи іншого функціоналу чи ігрових особливостей; було розроблено концепцію гри, ігровий світ та сюжет.

Було проведено моделювання бізнес-процесів предметної області та інформаційної системи action-гри. Було розроблено бізнес моделі IDEF0, IDEF3, що сприяють глибшому розумінню процесів у системі. Також було описано UML діаграми активностей, прецедентів, послідовності, станів та класів задля тієї ж цілі.

Було проаналізовано та спроектовано інтерфейси користувача гри та розроблено інструкцію користувача для покращення ігрового досвіду та швидкого інтегрування у ігровий світ. Інтерфейс користувача був розроблений з акцентом на простоту та зручність використання. Навігація інтуїтивно зрозуміла, а всі елементи інтерфейсу чітко позначені та легко доступні. Розробка чіткого та лаконічного меню, надала гравцям швидкий доступ до всіх необхідних функцій. Створення інформативного HUD (Heads-Up Display) дозволило надавати гравцям важливу інформацію про їхній стан, прогрес та оточення.

Завдяки використанню Unreal Engine 5 вдалося створити готову гру зі всіма необхідними механіками та функціоналом. Це включає: ігрову карту з різноманітними локаціями та завданнями; систему персонажів з унікальними здібностями та характеристиками; рівень штучного інтелекту, який забезпечує складних та цікавих ворогів; інвентарну систему та систему крафту; звукове оформлення та музику, які доповнюють атмосферу гри.

Проведено тестування розробленої гри: функціональне тестування (дозволило переконатися, що всі функції гри працюють без помилок та відповідають поставленим вимогам); тестування продуктивності (забезпечило

плавний геймплей, на різних комп'ютерах, без затримок); тестування сумісності (гарантувало, правильну роботу гри на різних платформах та геймпадами різних виробників).

Розроблена Action-RPG гра повністю відповідає поставленим вимогам та задачам. Вона має зручний інтерфейс користувача, візуально цікавий та динамічний світ, продуманий геймплей та безліч функцій, які роблять її цікавою та захоплюючою для гравців.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. The Witcher Universe | Action-Adventure RPGs: веб-сайт. URL: <https://www.thewitcher.com> (Дата звернення: 17.06.2024)
2. Dark Souls Wiki - Fandom: веб-сайт. URL: https://darksouls.fandom.com/wiki/Dark_Souls_Wiki (Дата звернення: 17.06.2024)
3. Far Cry 6 for Xbox One, PS4: веб-сайт. URL: <https://www.ubisoft.com/en-us/game/far-cry/far-cry-6> (Дата звернення: 17.06.2024)
4. Baldur's Gate 3 у Steam: веб-сайт. URL: https://store.steampowered.com/app/1086940/Baldurs_Gate_3/?l=ukrainian (Дата звернення: 17.06.2024)
5. The Elder Scrolls V: Skyrim Special Edition: веб-сайт. URL: https://store.steampowered.com/app/489830/The_Elder_Scrolls_V_Skyrim_Special_Edition/?l=ukrainian (Дата звернення: 17.06.2024)
6. Mass Effect™ Legendary Edition: веб-сайт. URL: <https://www.ea.com/games/mass-effect/mass-effect-legendary-edition> (Дата звернення: 17.06.2024)
7. Що таке IDE та SDK?: веб-сайт. URL: <https://apix-drive.com/ua/blog/useful/ide-i-sdk-strashnye-termyny-prostymi-slovami> (Дата звернення: 17.06.2024)
8. Що таке рефакторинг коду і навіщо він потрібний: веб-сайт. URL: <https://brainlab.com.ua/uk/blog-uk/shho-take-refactoryng-kodu-i-navishho-vin-potribnyi> (Дата звернення: 17.06.2024)
9. 10 кращих редакторів коду для програмістів: веб-сайт. URL: <https://mate.academy/blog/front-end-and-js/top-10-text-editors/> (Дата звернення: 17.06.2024)
10. Rider: The Cross-Platform .NET IDE from JetBrains: веб-сайт. URL: <https://www.jetbrains.com/rider/> (Дата звернення: 17.06.2024)
11. Visual Studio 2022 IDE - Programming Tool for Software Developers: веб-сайт. URL: <https://visualstudio.microsoft.com/vs/> (Дата звернення: 17.06.2024)

12. Eclipse IDE for C/C++ Developers | Eclipse Packages: веб-сайт. URL: <https://www.eclipse.org/downloads/packages/release/kepler/sr2/eclipse-ide-cc-developers> (Дата звернення: 17.06.2024)
13. Visual Studio Code - Code Editing. Redefined веб-сайт. URL: <https://code.visualstudio.com/> (Дата звернення: 17.06.2024)
14. Blueprints VS C++: веб-сайт. URL: <https://dev.epicgames.com/community/learning/tutorials/waOX/unreal-engine-blueprints-vs-c> (Дата звернення: 17.06.2024)
15. What is an Action Game?: веб-сайт. URL: <https://www.computerhope.com/jargon/a/action-game.htm> (Дата звернення: 17.06.2024)
16. An Introduction to Open World Games: веб-сайт. URL: <https://www.gameopedia.com/introduction-to-open-world-games/> (Дата звернення: 17.06.2024)
17. RPG / CRPG — це що таке, суть, жанри, види і приклади РПГ: веб-сайт. URL: <https://termin.in.ua/rpg-i-crpg/> (Дата звернення: 17.06.2024)
18. What is Game Design: веб-сайт. URL: <https://www.igi-global.com/dictionary/unifying-instructional-game-design/11822> (Дата звернення: 17.06.2024)
19. Learn more about widget blueprints in UEFN: веб-сайт. URL: <https://dev.epicgames.com/community/learning/tutorials/JpeE/fortnite-learn-more-about-widget-blueprints-in-uefn> (Дата звернення: 17.06.2024)
20. Що таке HUD / UI у відеоіграх?: веб-сайт. URL: <https://tseivo.com/b/jargoniist/t/pn8oobb8bz> (Дата звернення: 17.06.2024)
21. Animation Blueprint Editor: веб-сайт. URL: https://dev.epicgames.com/documentation/en-us/unreal-engine/animation-blueprint-editor-in-unreal-engine?application_version=5.4 (Дата звернення: 17.06.2024)
22. State machine - MDN Web Docs Glossary: веб-сайт. URL: https://developer.mozilla.org/en-US/docs/Glossary/State_machine (Дата звернення: 17.06.2024)

23. Ray Casting: веб-сайт. URL: <https://developer.playcanvas.com/user-manual/physics/ray-casting/> (Дата звернення: 17.06.2024)
24. Enhanced Input In Unreal Engine | Unreal Engine 5.4: веб-сайт. URL: <https://dev.epicgames.com/documentation/en-us/unreal-engine/enhanced-input-in-unreal-engine> (Дата звернення: 17.06.2024)
25. Ф Довженко, І., Ковальчук, М., Яворський, О., & Гінайло, С. (2022). Авторський задум і його втілення засобами motion-дизайну. Актуальні проблеми сучасного дизайну, 232-234. Київський національний університет технологій та дизайну. <https://er.knutd.edu.ua/handle/123456789/17821>
26. Функціональне тестування: веб-сайт. URL: <https://qalight.ua/baza-znaniy/funktsionalne-testuvannya/> (Дата звернення: 17.06.2024)
27. Тестування продуктивності: веб-сайт. URL: <https://qalight.ua/baza-znaniy/testuvannya-produktivnosti/> (Дата звернення: 17.06.2024)
28. Види тестування та відмінності між ними. URL: [https://qagroup.com.ua/publications/vydy-testuvannya-ta-vidminnosti-mizh-nymy/#:~:text=Compatibility%20Testing%20\(Тестування%20сумісності\)%20—,операційною%20системою%20С%20платформою%20чи%20обладнанням.](https://qagroup.com.ua/publications/vydy-testuvannya-ta-vidminnosti-mizh-nymy/#:~:text=Compatibility%20Testing%20(Тестування%20сумісності)%20—,операційною%20системою%20С%20платформою%20чи%20обладнанням.) (Дата звернення: 17.06.2024)
29. Маєвський, О. М., & Ковальчук, М. І. (2023). Сучасні технології підготовки мультимедійних творчих продуктів. У Prospective And Priority Directions Of Scientific Research In Technical And Agricultural Sciences (с. 191-214). Boston: International Science Group. – Primedia eLaunch.
30. Jane McGonigal. «Reality is Broken: Why Games Make Us Better and How They Can Change the World.»: посібник, Penguin Books, 2020, 416 с.
31. Jesse Schell. «The Art of Game Design: A Book of Lenses»: посібник, CRC Press, 2021, 520 с.

ДОДАТКИ

ДОДАТОК А

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Характеристика	JetBrains Rider	Visual Studio 2022	Eclipse for C++	Visual Studio Code
Ціна	Платна	Безкоштовна	Безкоштовна	Безкоштовна
Платформи	Windows, macOS, Linux	Windows, macOS, Linux	Windows, macOS	Windows, macOS, Linux, ChromeOS
Підтримувані мови	C++, C#, Java, JavaScript, Python, PHP, Go, Kotlin, Rust, Swift, TypeScript, Dart, Scala, Clojure, F#, XAML, MSBuild, Unreal Engine, Unity, Docker	C++, C#, F#, Visual Basic, ASP.NET, JavaScript, TypeScript, HTML, CSS, Python, SQL, XML, XSD, JSON, XAML, MSBuild, Unity, Unreal Engine	C++, C, Java, JavaScript, Python, PHP, R, Ruby, Ada, Fortran, COBOL, HTML, CSS, XML, JSON, Makefiles, Ant, Maven, Gradle	C++, C#, Java, JavaScript, Python, PHP, Go, Ruby, Rust, Swift, TypeScript, Dart, Scala, Kotlin, Haskell, Clojure, Erlang, Elixir, Elm, Prolog, R, Julia, Rust, Lua, Crystal, Nim, D, F#
Інструменти для налагодження	Розширений налагоджувач, дебаггер, точки зупинки, перегляд стека, аналізатори пам'яті, відладка JIT	Розширений налагоджувач, дебаггер, точки зупинки, перегляд стека, аналізатори пам'яті, відладка JIT	Розширений налагоджувач, дебаггер, точки зупинки, перегляд стека, аналізатори пам'яті, відладка JIT	Розширений налагоджувач, дебаггер, точки зупинки, перегляд стека, аналізатори пам'яті, відладка JIT
Інструменти для автодоповнення	Редагування IntelliJ IDEA, ReSharper, Code Completion, Code Snippets	IntelliSense, Code Completion, Code Snippets	Code Completion, Code Snippets	IntelliSense, Code Completion, Code Snippets
Інструменти для рефакторингу	ReSharper, Code Refactoring, Rename, Extract, Introduce Variable, Introduce Constant, Extract Method, Inline Method, Extract Class, Extract Class,	Refactoring Tools, Rename, Extract, Introduce Variable, Introduce Constant, Extract Method, Inline Method, Extract Class, Rename Field, Rename	Refactoring Tools, Rename, Extract, Introduce Variable, Introduce Constant, Extract Method, Inline Method, Extract Class, Rename Field, Rename	Refactoring Tools, Rename, Extract, Introduce Variable, Introduce Constant, Extract Method, Inline Method, Extract Class, Rename Field, Rename

	Rename Field, Rename Parameter, Rename Variable	Parameter, Rename Variable	Parameter, Rename Variable	Parameter, Rename Variable
Інструменти для тестування	Unit Testing, Integration Testing, UI Testing, Continuous Integration	Unit Testing, Integration Testing, UI Testing, Continuous Integration	Unit Testing, Integration Testing, UI Testing, Continuous Integration	Unit Testing, Integration Testing, UI Testing, Continuous Integration
Інші інструменти	Підтримка Git, GitHub, Visual Studio Team Services, Docker, Kubernetes, Unreal Engine, Unity	Підтримка Git, GitHub, Visual Studio Team Services, Docker, Kubernetes	Підтримка Git, GitHub, Docker, Kubernetes	Підтримка Git, GitHub, Docker, Kubernetes
Сервіси	Marketplace, JetBrains Academy, JetBrains Hub	Visual Studio Marketplace	Eclipse Marketplace	Marketplace, Extensions

Таб. 1.1 – порівняльна характеристика IDE

ДОДАТОК Б

МОДЕЛЮВАННЯ БІЗНЕС-ПРОЦЕСІВ ПРЕДМЕТНОЇ ОБЛАСТІ

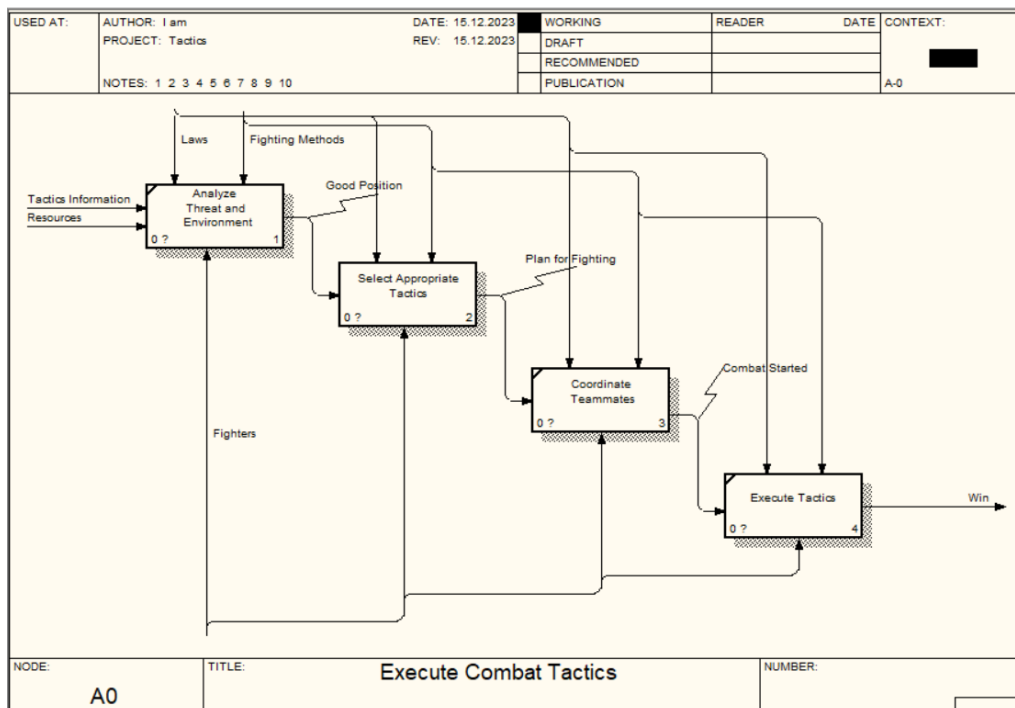


Рис. 1.1 – IDEF0 «Тактика ближнього бою»

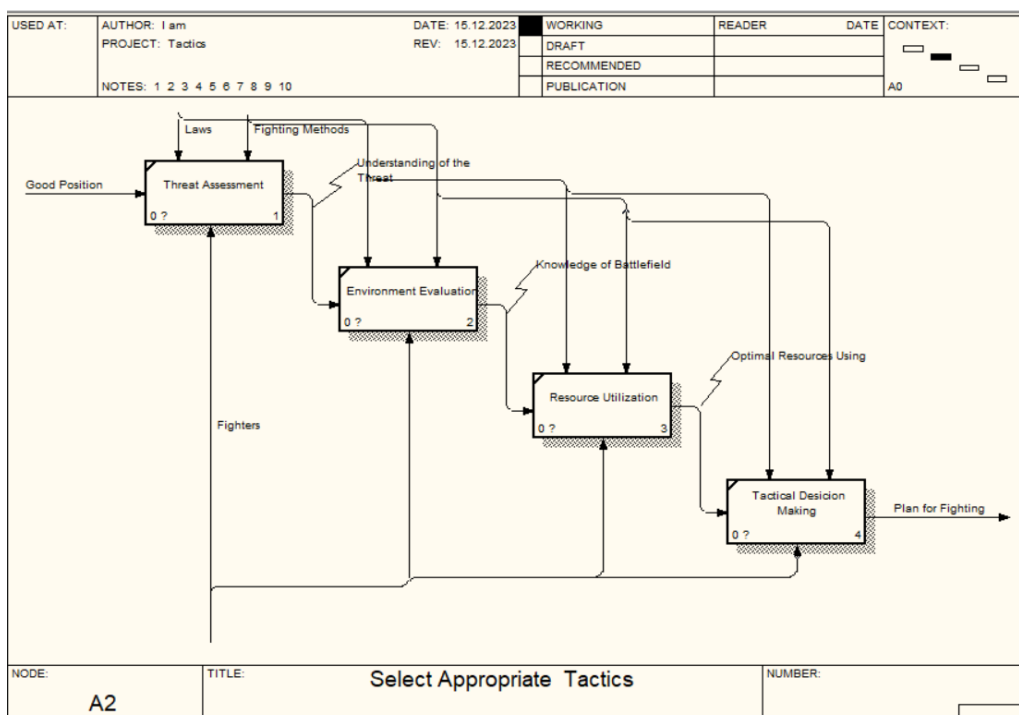


Рис. 1.2 – декомпозиція IDEF0 «Тактика ближнього бою»

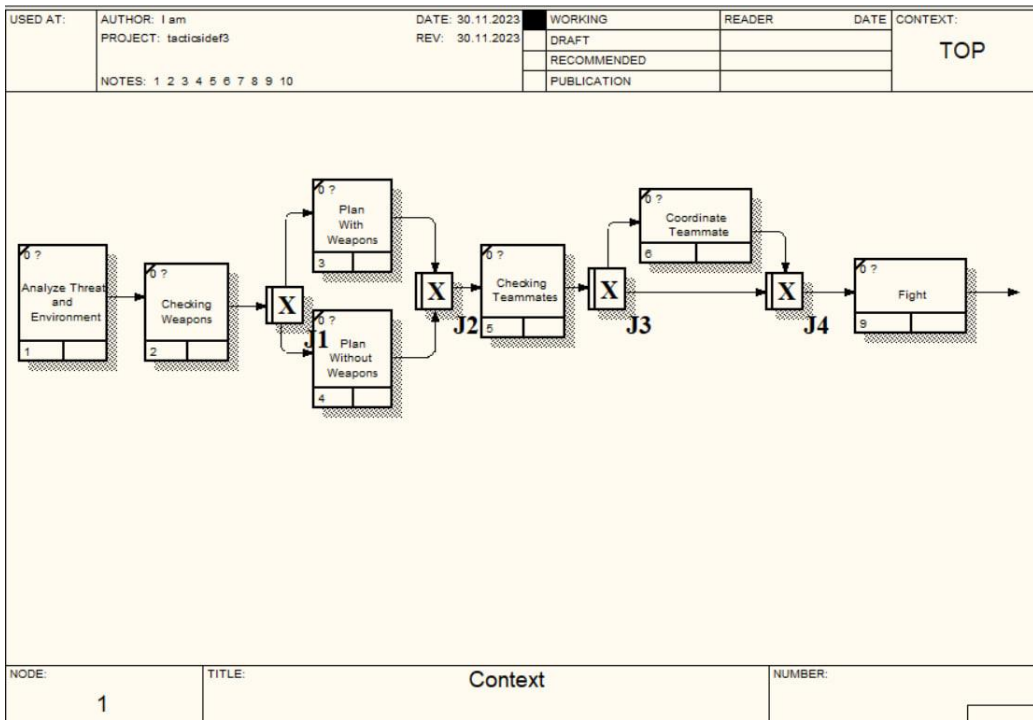


Рис. 1.3 – IDEF3 «Тактика ближнього бою»

ДОДАТОК В

МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ АСТІОН-ГРИ



Рис. 2.1 – UML діаграма станів

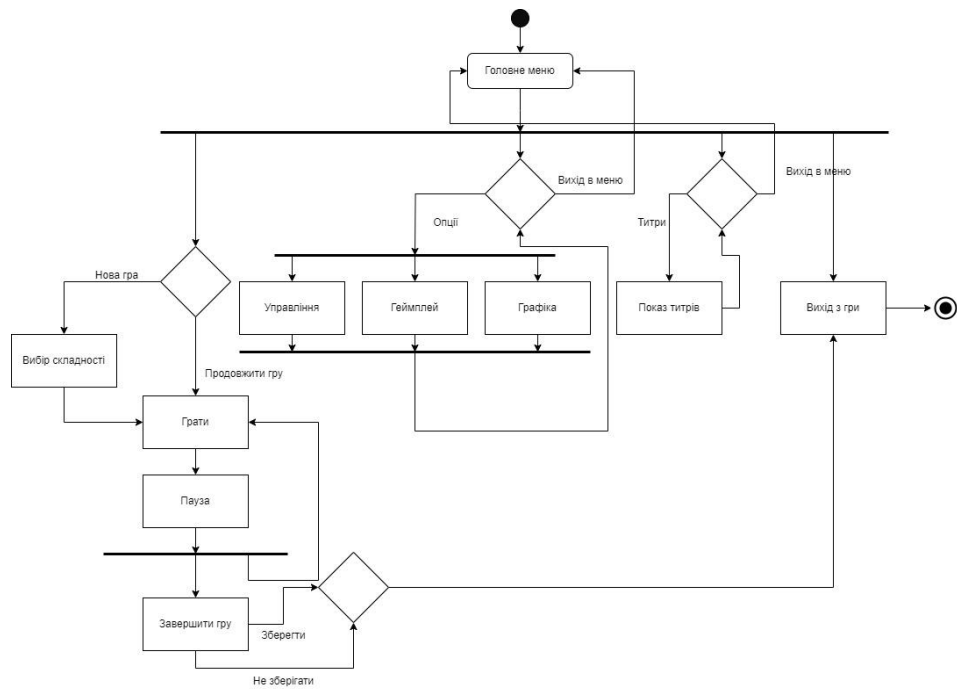


Рис. 2.2 – UML діаграма активностей

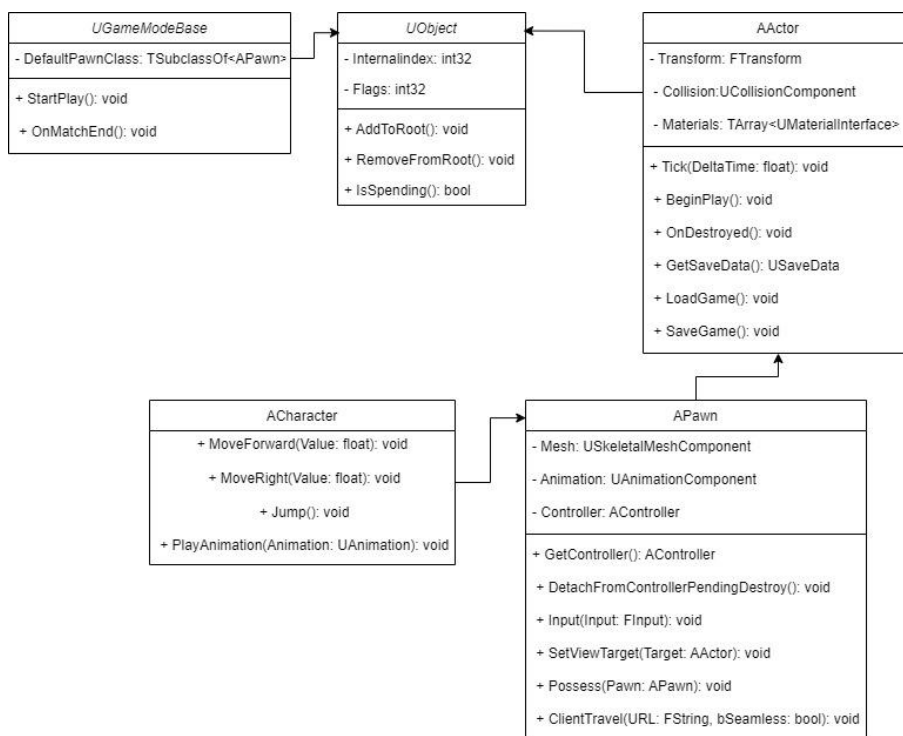


Рис. 2.3 – UML діаграма класів

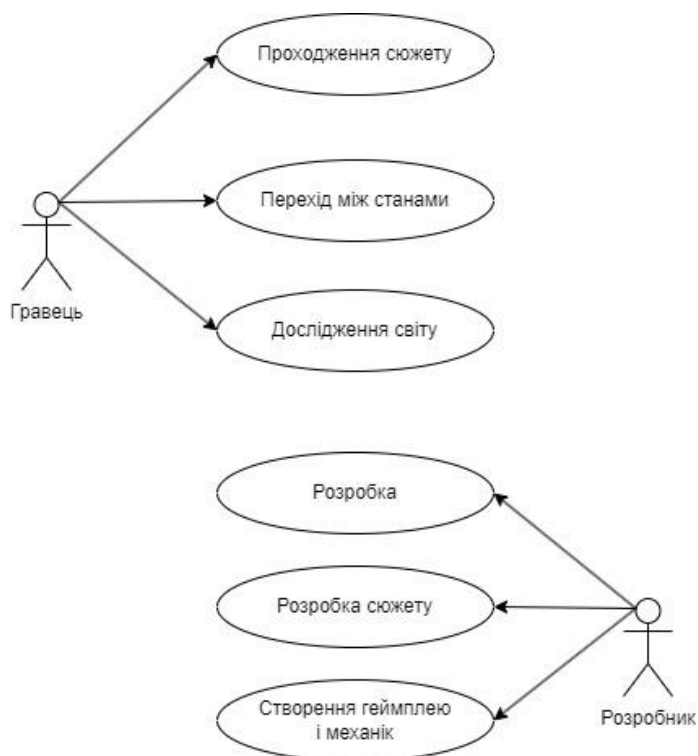


Рис. 2.4 – UML діаграма прецендентів

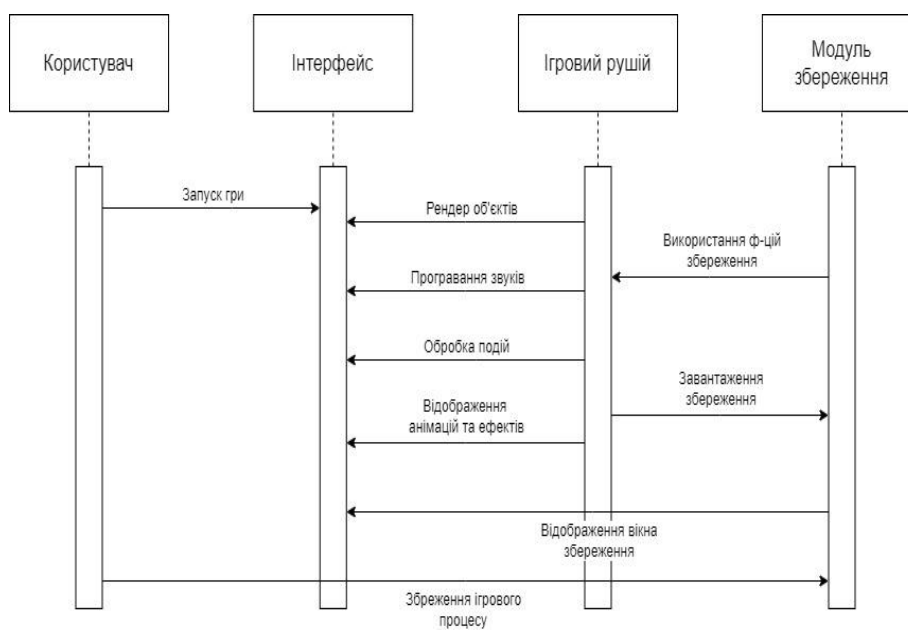


Рис. 2.5 – UML діаграма послідовностей

ДОДАТОК Г

ПРОЕКТУВАННЯ ІНТЕРФЕЙСУ ІНФОРМАЦІЙНОЇ СИСТЕМИ АСТІОН-ГРИ

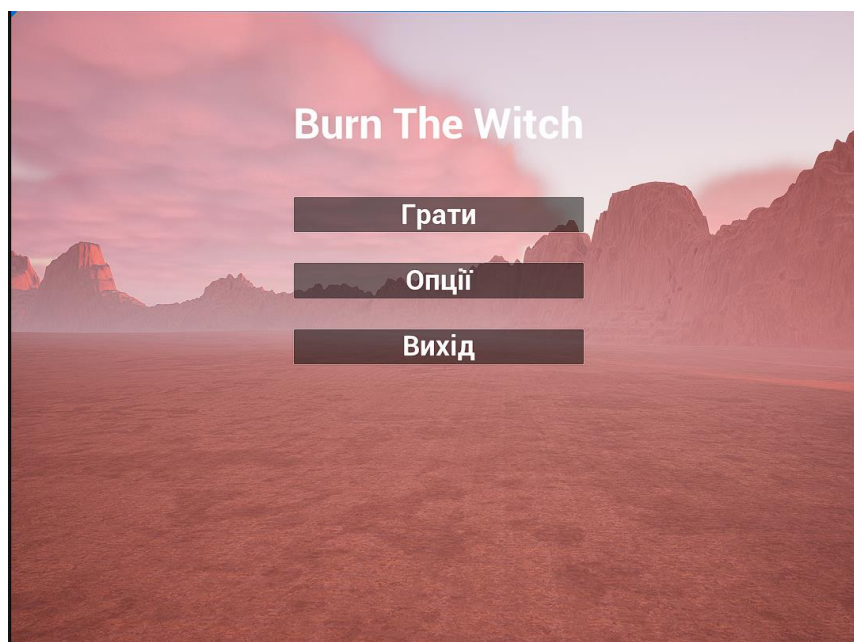


Рис. 2.6 – головне меню

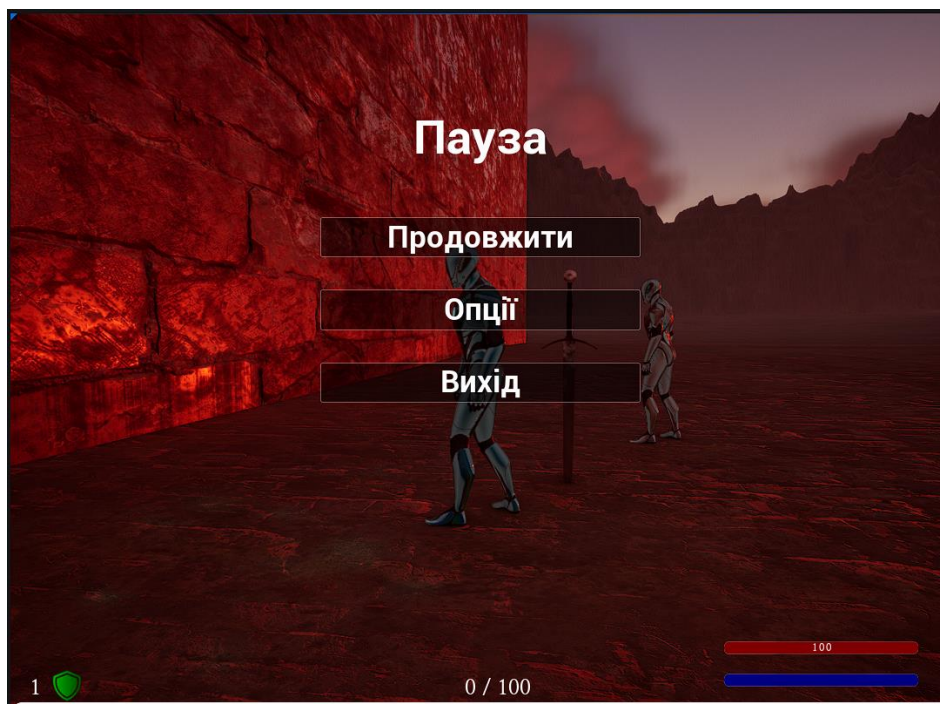


Рис. 2.7 – меню паузи

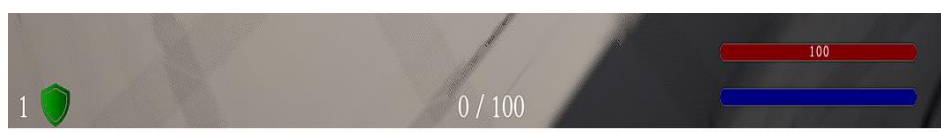


Рис. 2.8 – HUD інформаційної системи

ДОДАТОК Д

Процес реалізації гри в середовищі Unreal Engine

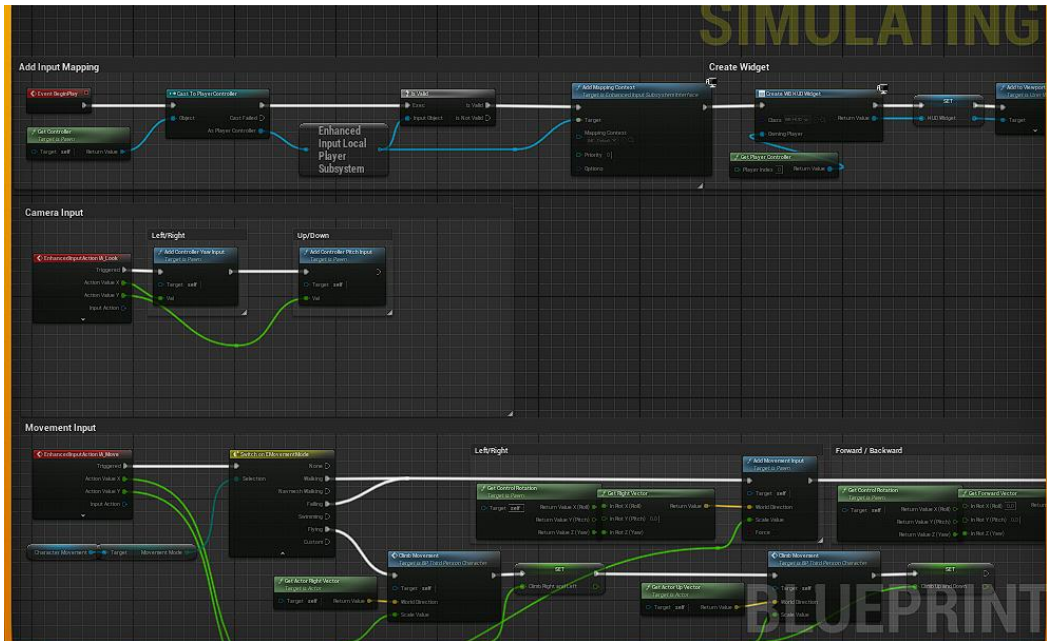


Рис. 2.9 – Blueprint клас рухів

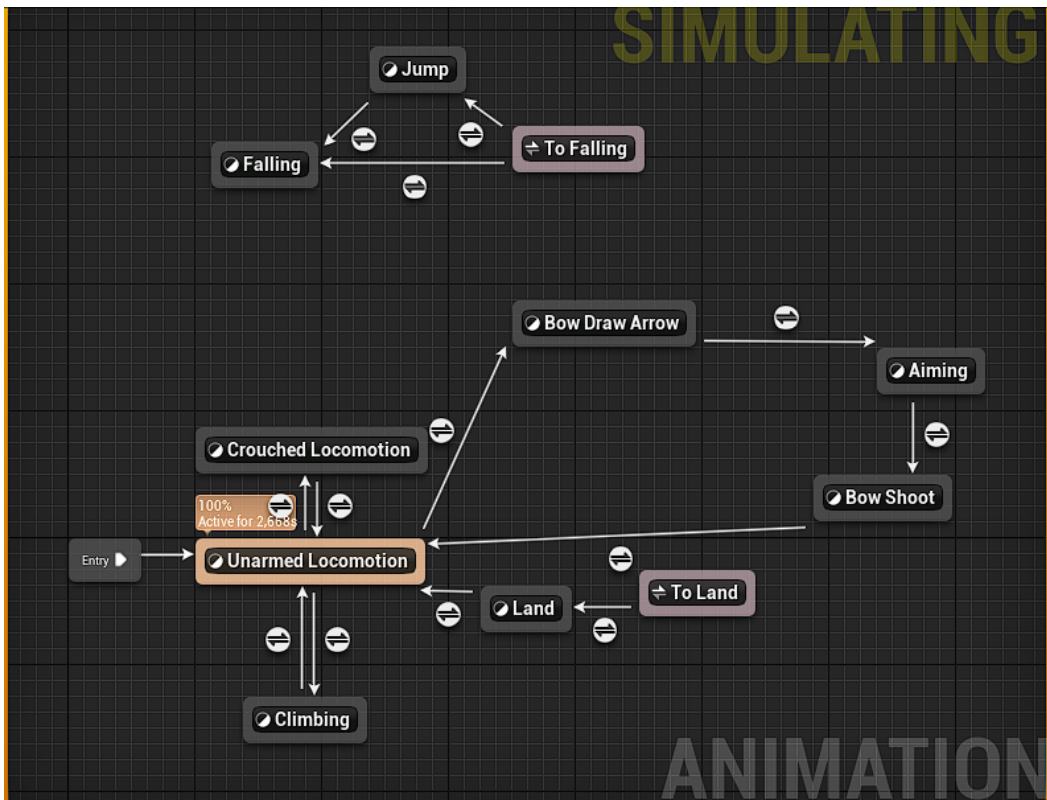


Рис. 2.10 – стейт машина



Рис. 2.11 – анімація бігу



Рис. 2.12 – анімація присіду

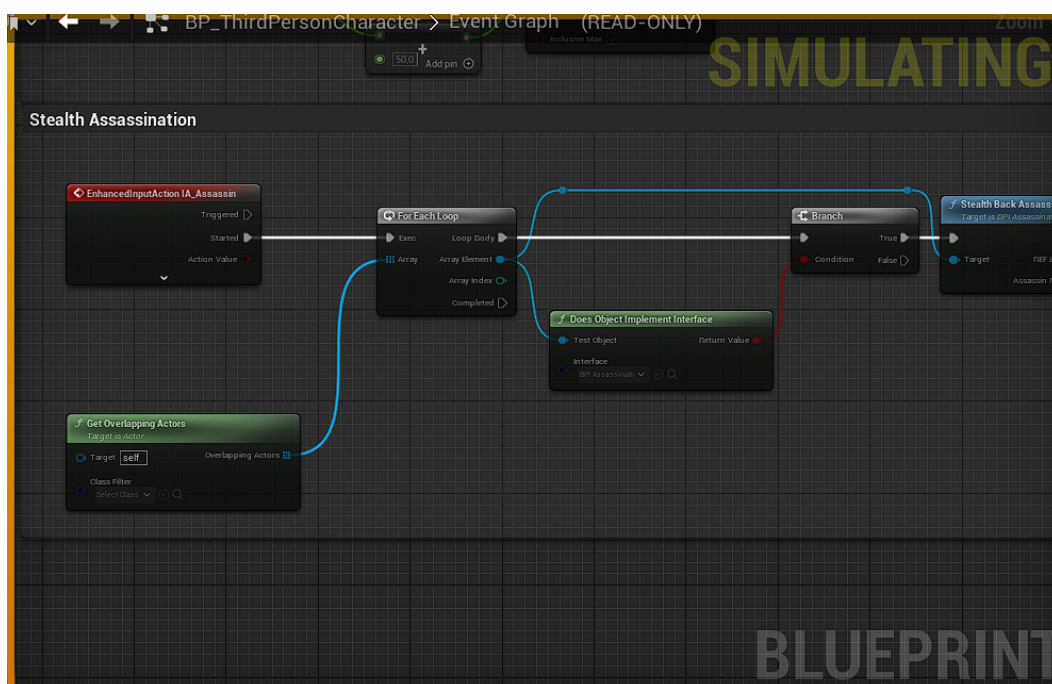


Рис. 2.13 – Blueprint із кодом швидкої нейтралізації

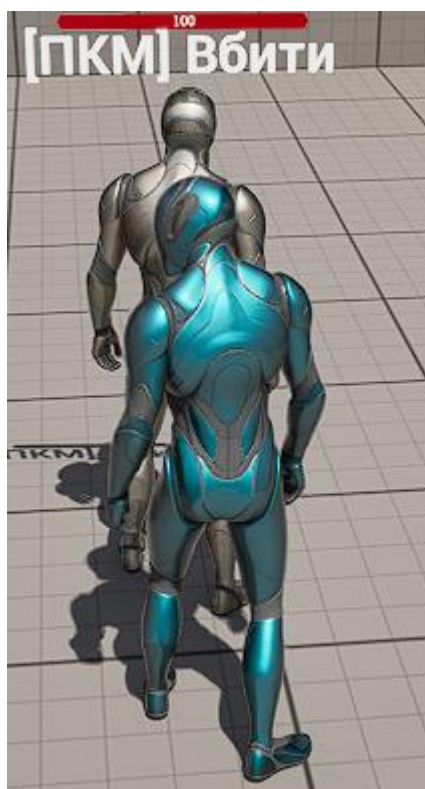


Рис. 2.14 – демонстрація віджету

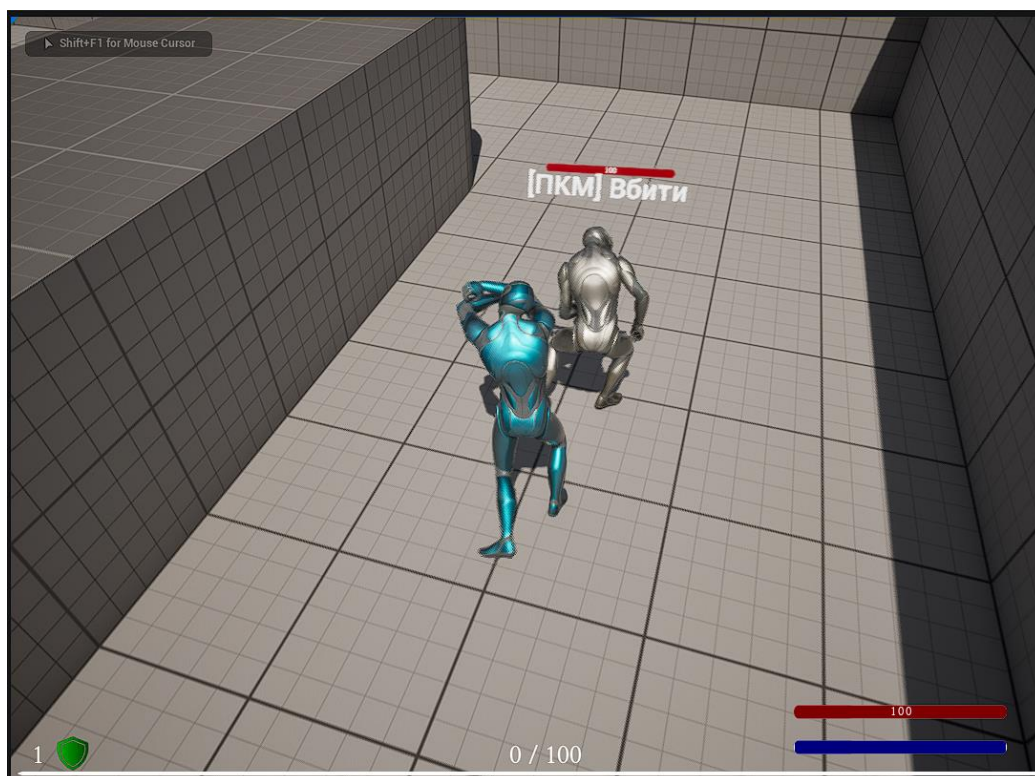


Рис. 2.15 – демонстрація анімації

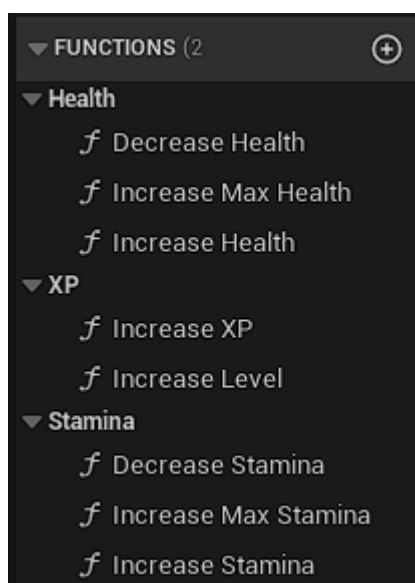


Рис. 2.16 – функції статистики персонажа

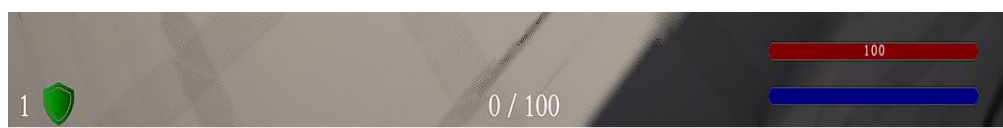


Рис. 2.17 – HUD статистики персонажа

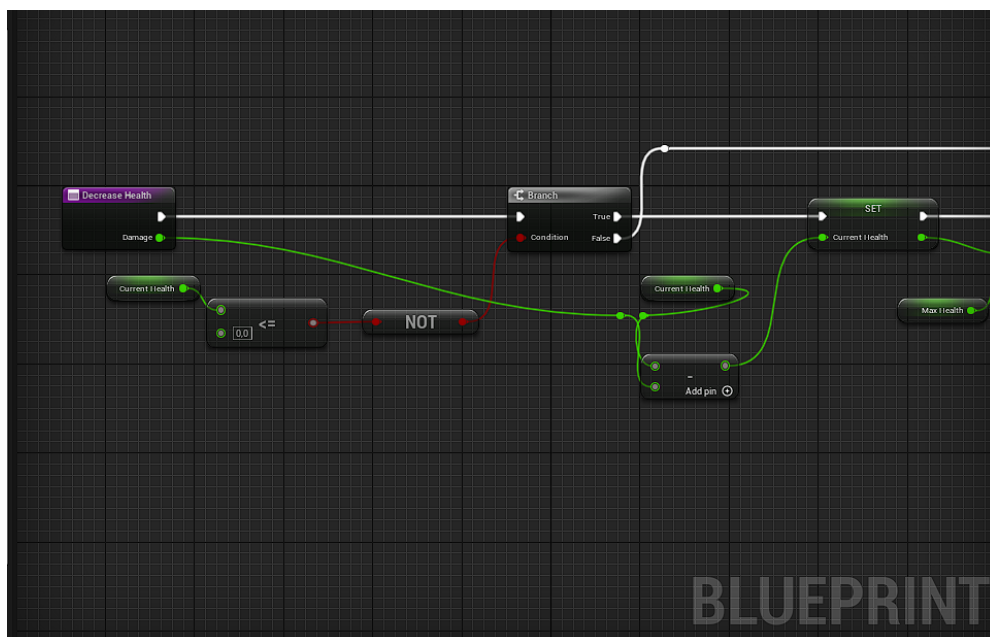


Рис. 2.18 – Blueprint клас зниження здоров'я

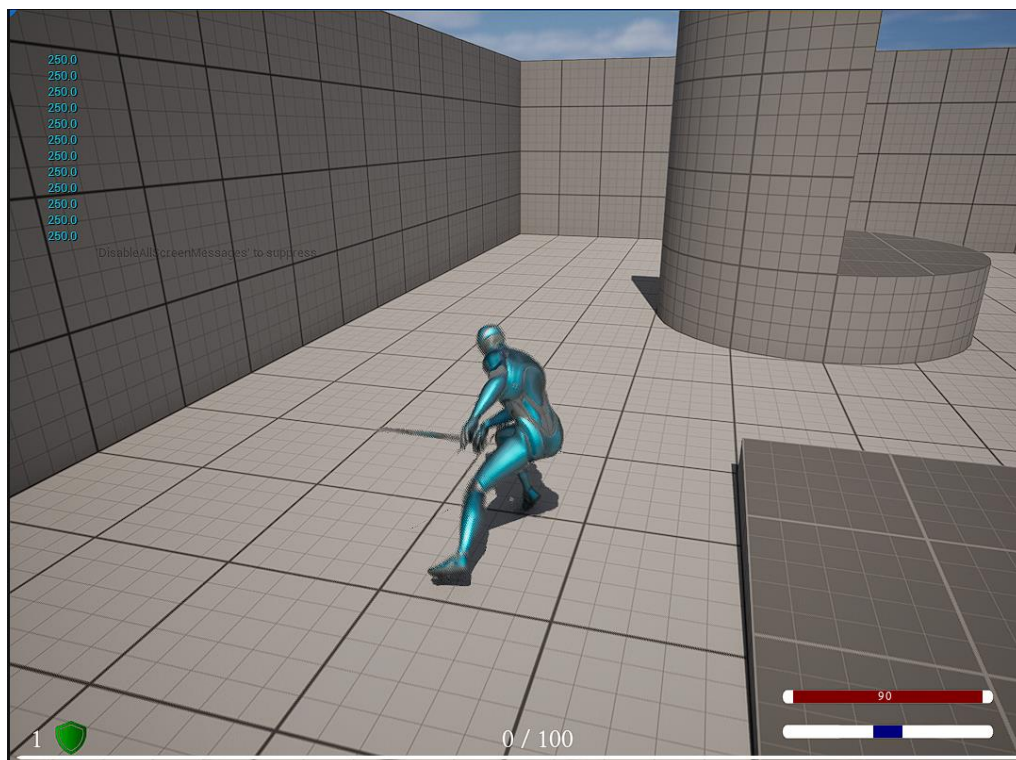


Рис. 2.19 – демонстрація анімації атаки

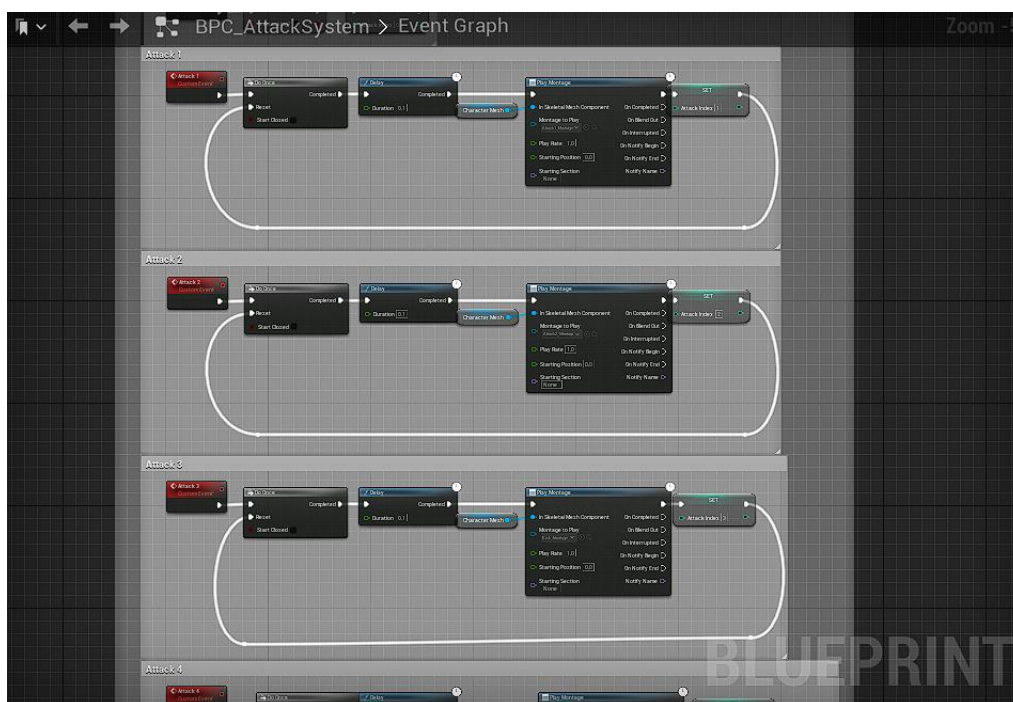


Рис. 2.20 – Blueprint клас «Система атаки»



Рис. 2.21 – анімація перекачу

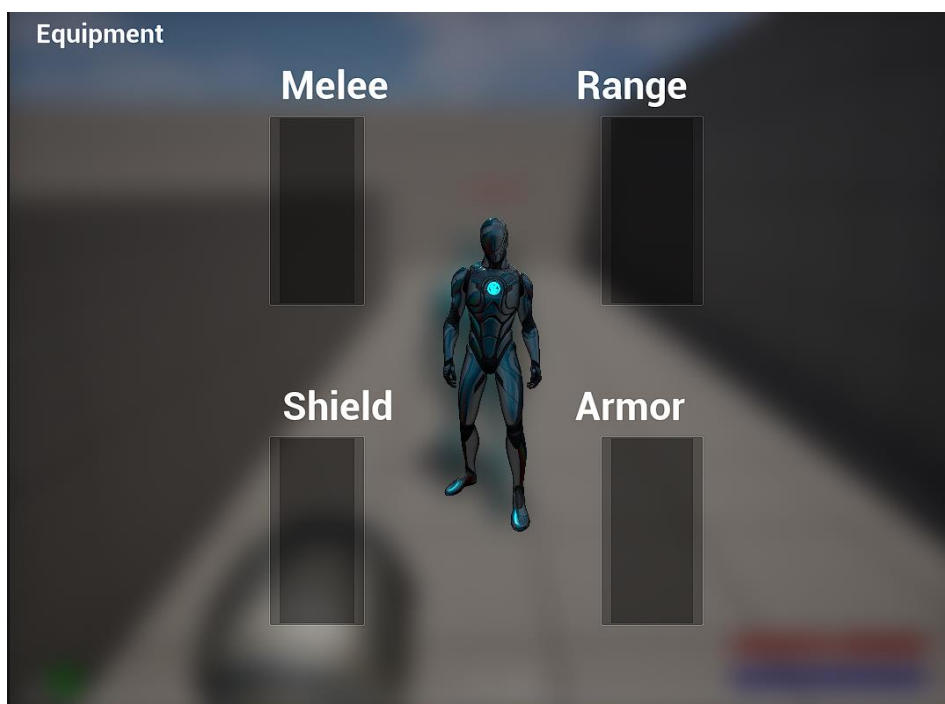


Рис. 2.22 – інвентар



Рис. 2.23 – система відображення гравця зі зброєю в інвентарі

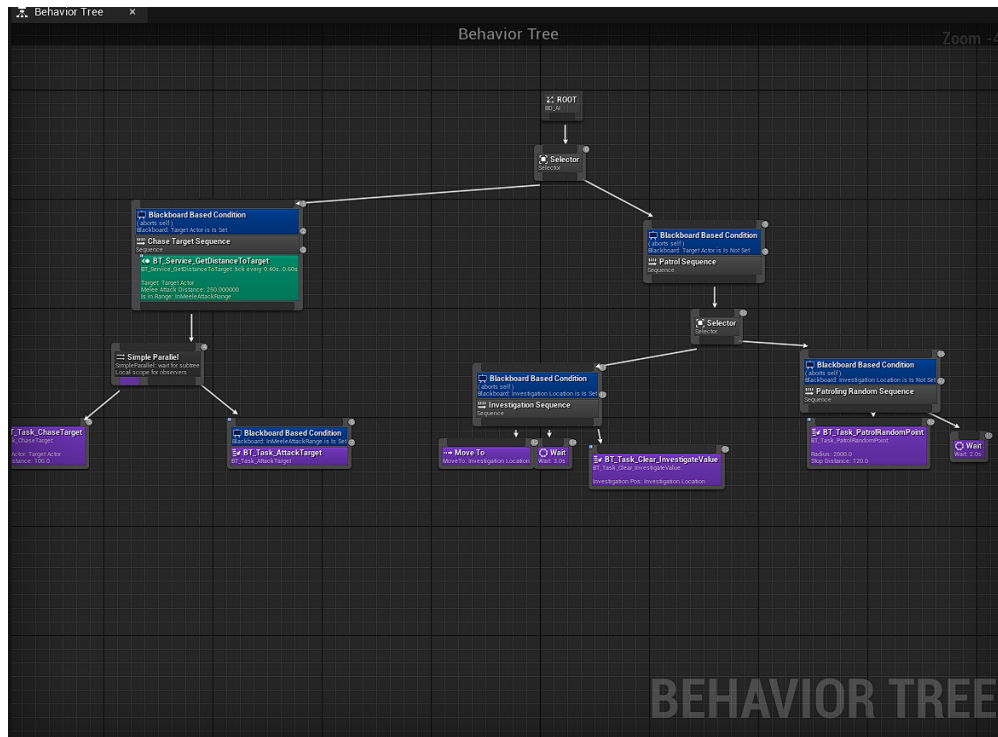


Рис. 2.24 – дерево задач III

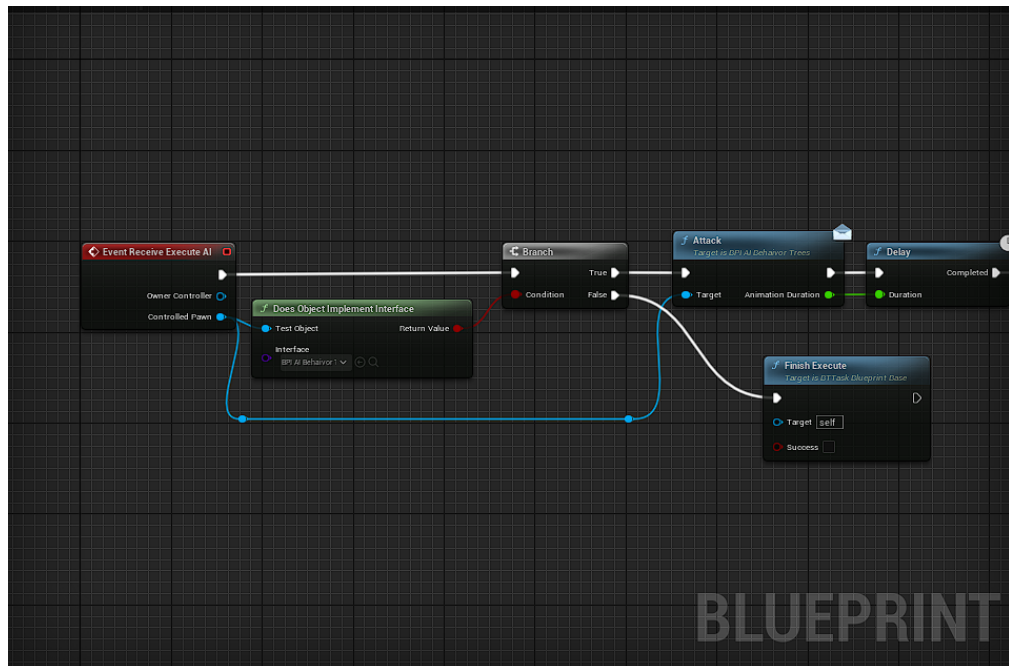


Рис. 2.25 – Blueprint задачі «Атака»

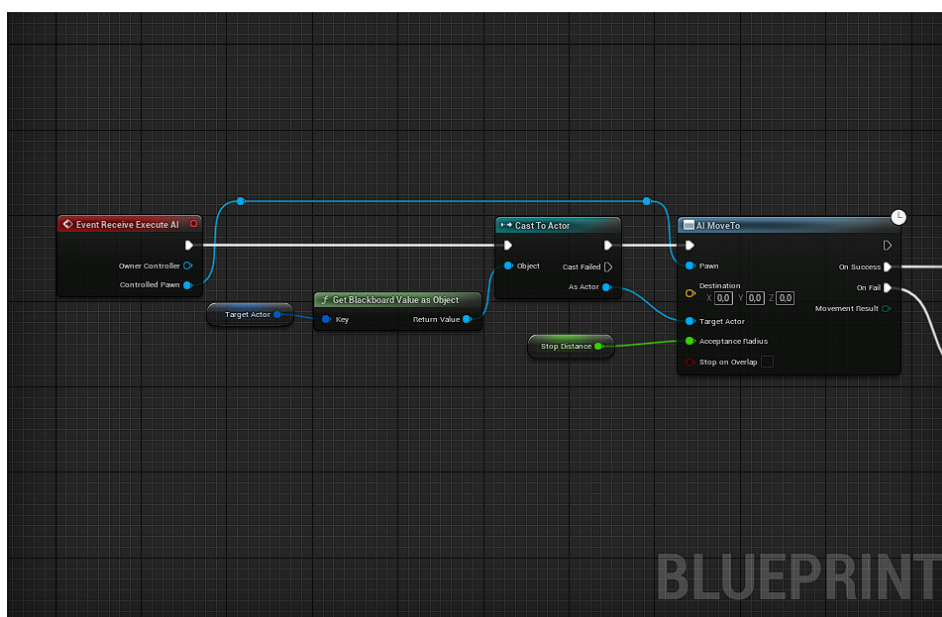


Рис. 2.26 – Blueprint задачі «Рух в рандомні точки»

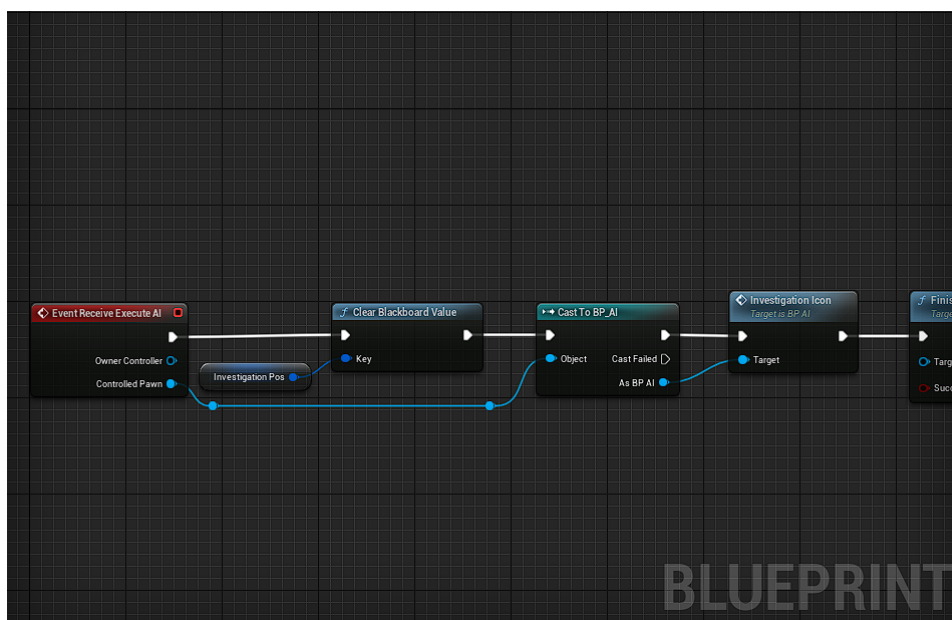


Рис. 2.27 – Blueprint очистки від інших задач для примінення нової

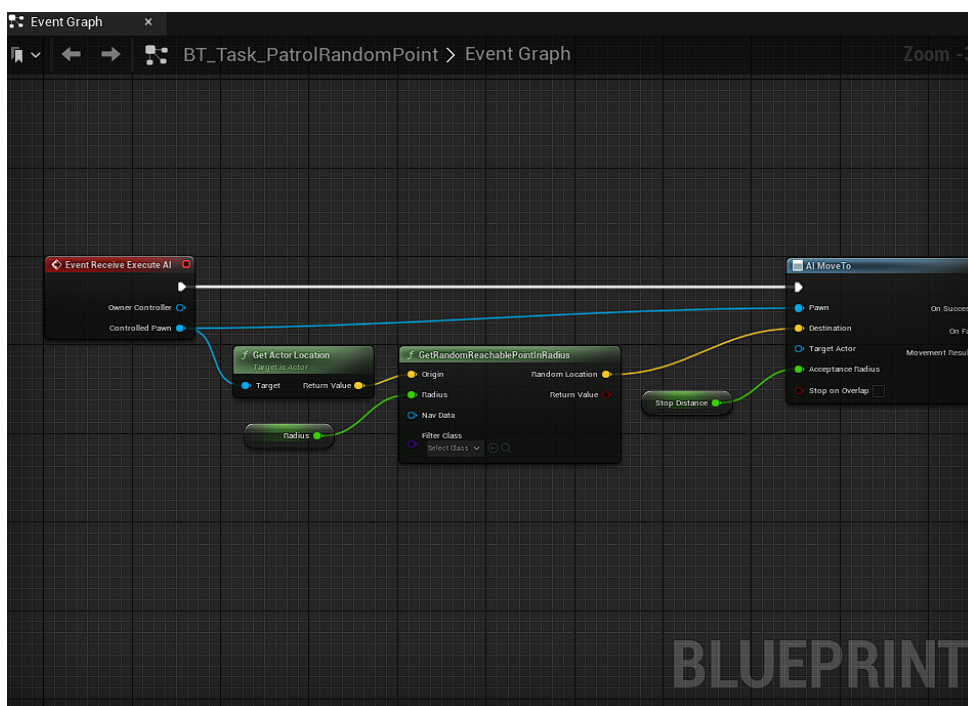


Рис. 2.28 – Blueprint задачі «Патрулювання»

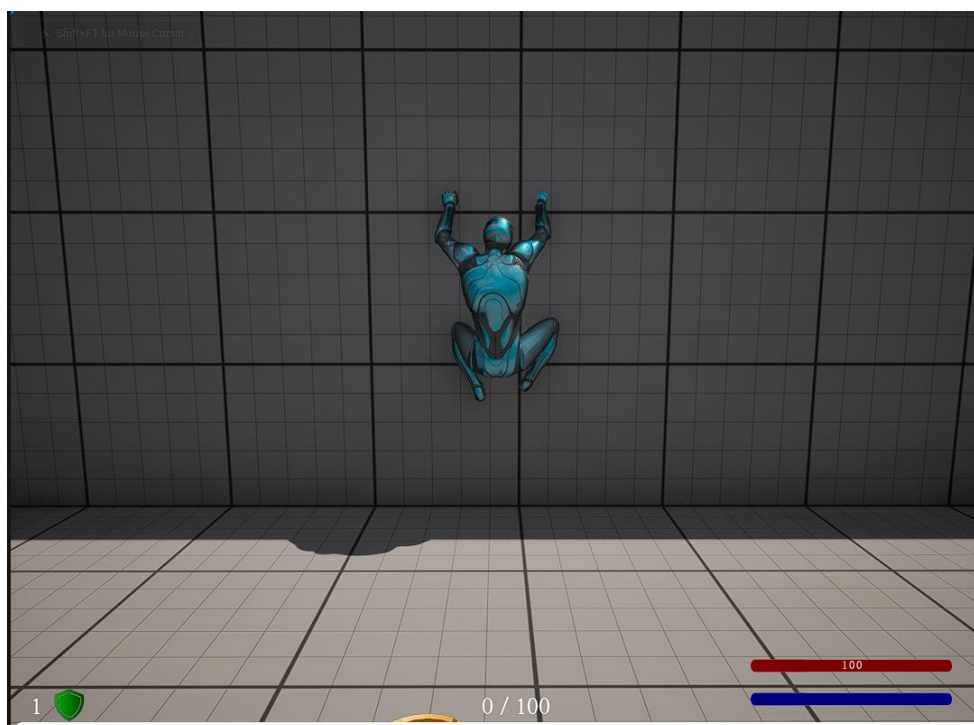


Рис. 2.29 – анімація та демонстрація паркуру

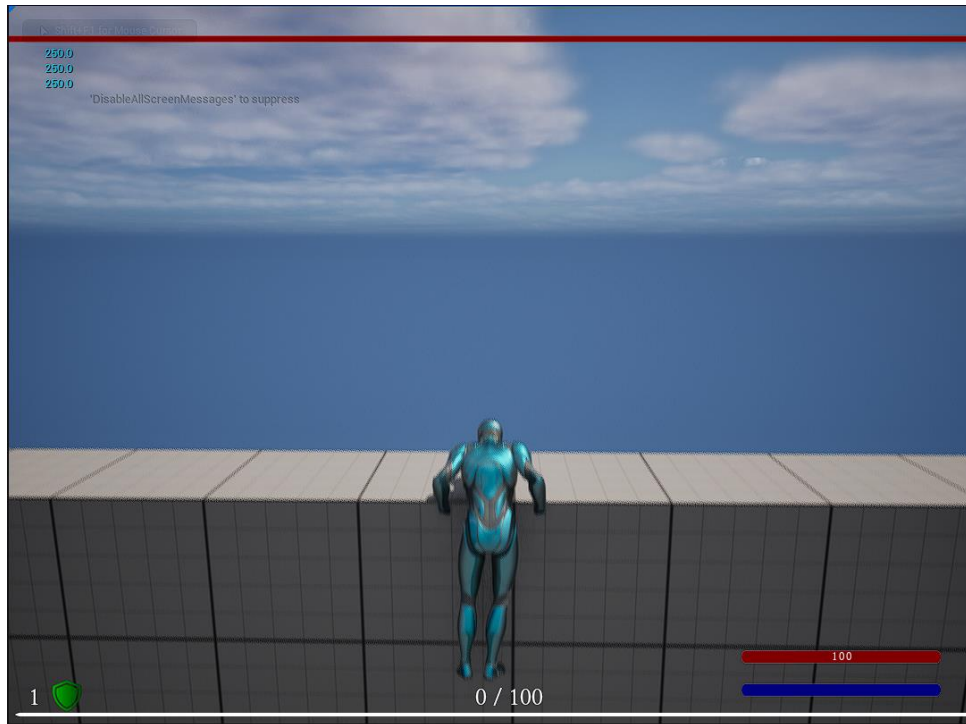


Рис. 2.30 – анімація завершення вилазки на стіну

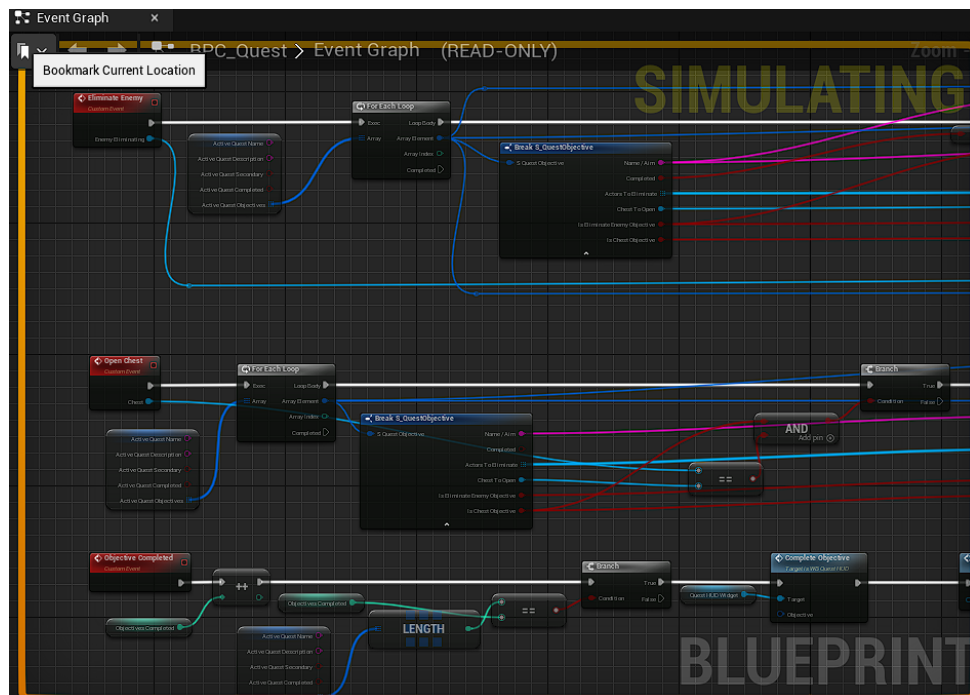


Рис. 2.31 – Blueprint клас системи квестів

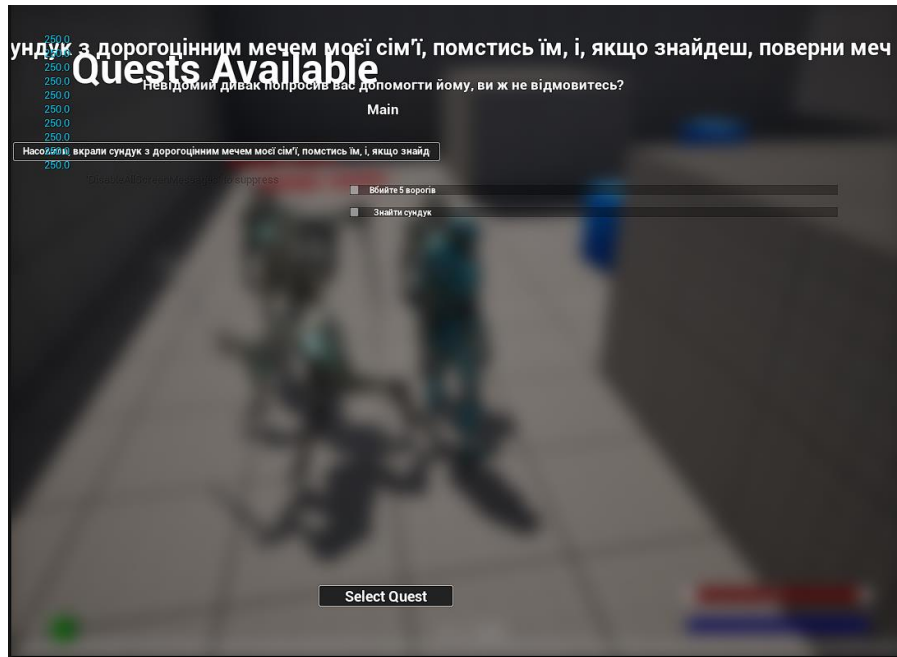


Рис. 2.32 – демонстрація квест-системи у грі

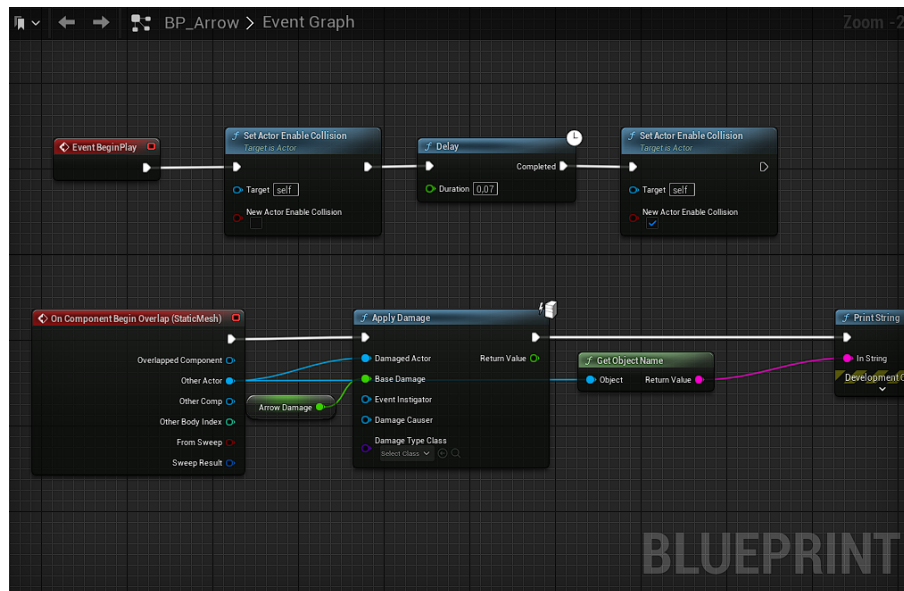


Рис. 2.33 – код нанесення урону луком

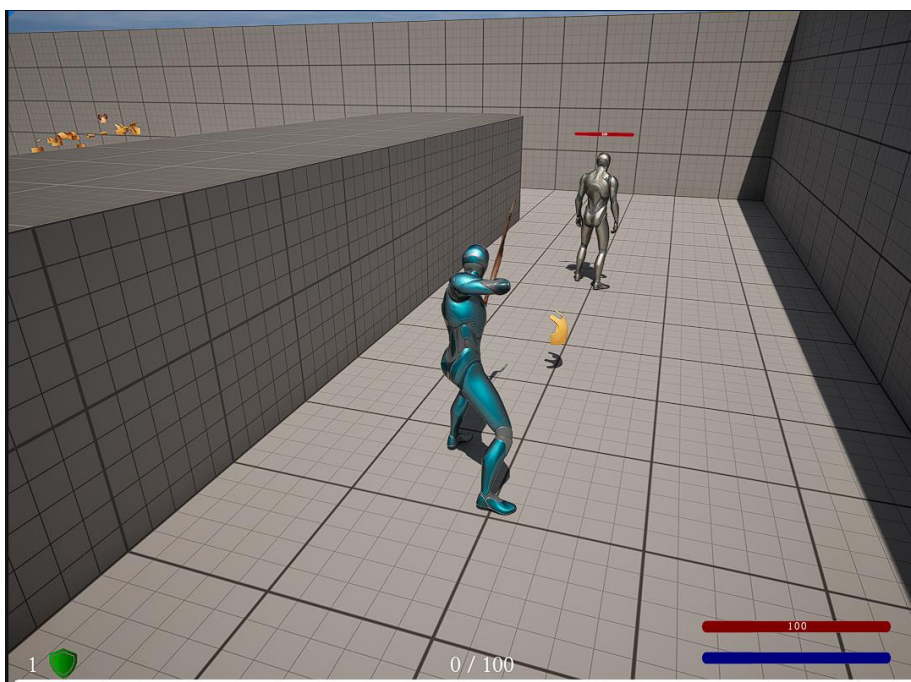


Рис. 2.34 – анімація прицілювання з луку

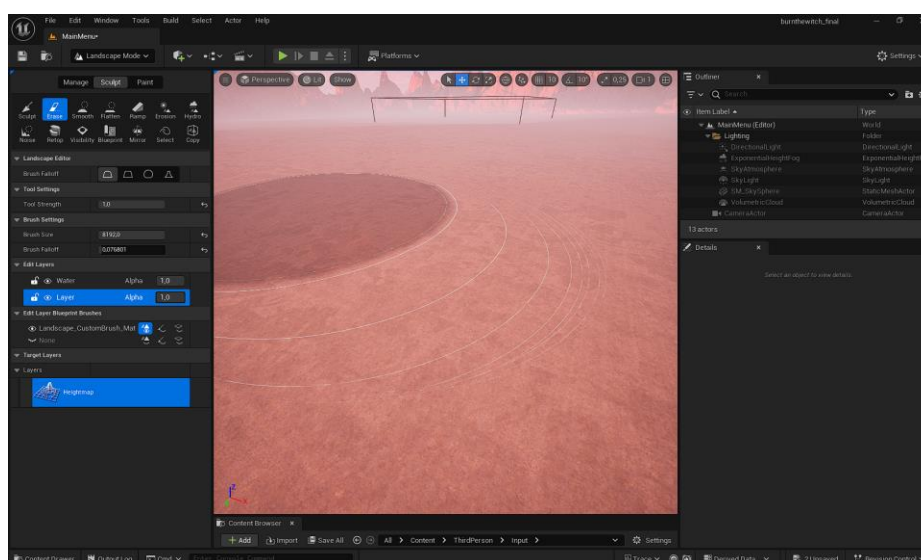


Рис. 2.35 – демонстрація редактора ландшафту

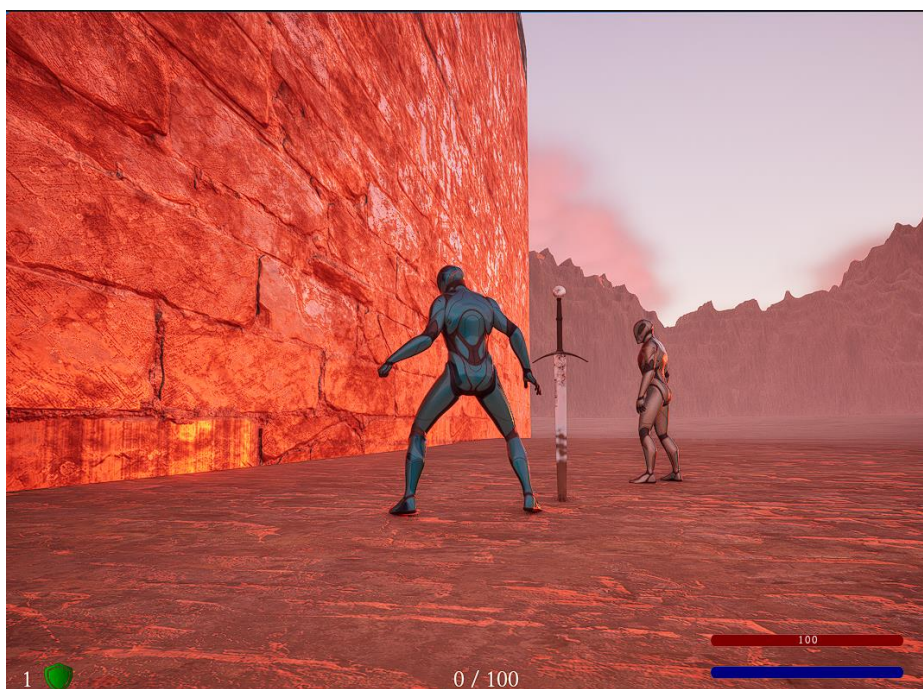


Рис. 2.36 – демонстрація тестового готового ландшафту



Рис. 2.37 – демонстрація головного меню



Рис. 2.38 – демонстрація меню паузи

ДОДАТОК Е

ТЕСТУВАННЯ ГРИ ТА АНАЛІЗ РЕЗУЛЬТАТІВ



Рис. 3.1 – демонстрація кадрів в секунду у спокійній фазі

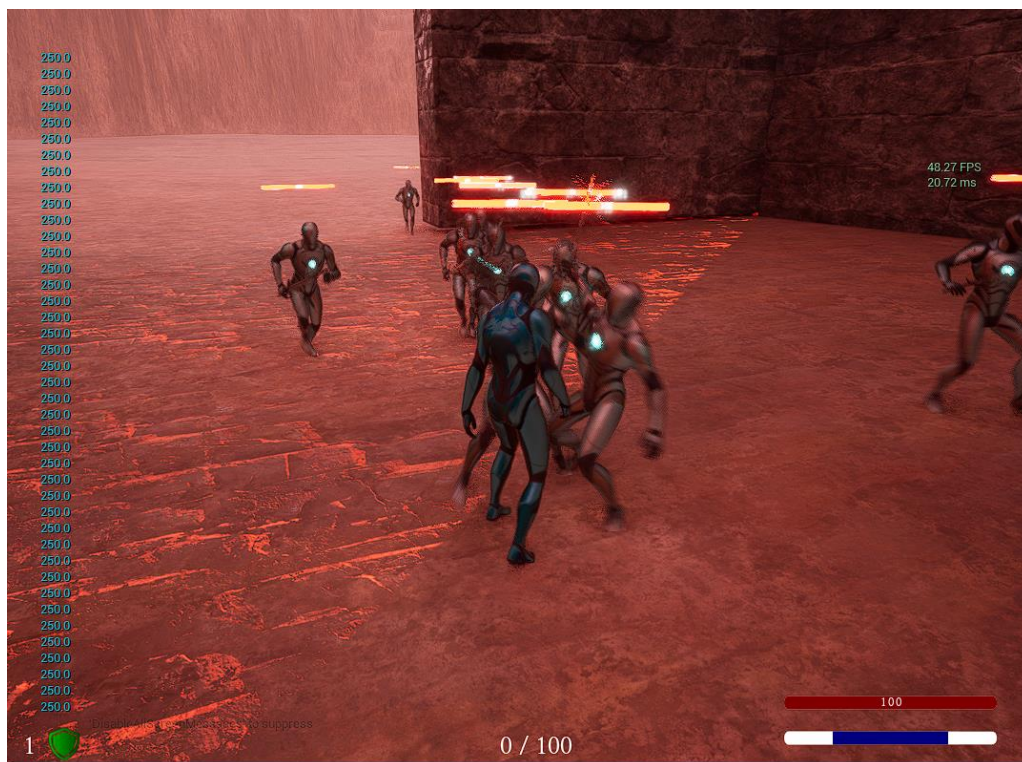


Рис. 3.2 – демонстрація кадрів в секунду при наявності більшої кількості об'єктів на карті



Рис. 3.3 – демонстрація кількості кадрів про добавленій високополігональній моделі