

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет

імені Петра Могили

Факультет комп'ютерних наук

Кафедра інженерії програмного забезпечення

ЯШНИКОВ ДЕНИС ВЯЧЕСЛАВОВИЧ

УДК__

**Комп'ютерна 3D гра із використанням середовища візуального
створення скриптів**

**Автореферат кваліфікаційної роботи на здобуття
ступеня вищої освіти «Бакалавр»**

Спеціальність «Інженерія програмного забезпечення»

Освітня кваліфікація

«Бакалавр з інженерії програмного забезпечення»

Миколаїв – 2021

Кваліфікаційною роботою є рукопис.

Робота виконана в Чорноморському національному університеті імені Петра Могили Міністерства освіти і науки України на кафедрі інженерії програмного забезпечення

Керівник:

Старший викладач

Дворецька Світлана
Володимирівна

Рецензент:

Канд. техн. наук

Старший викладач

Кондратенко Галина
Володимирівна

Захист відбудеться «__»___ 2021 р. о___ год. на засіданні екзаменаційної комісії у Чорноморському національному університеті імені Петра Могили за адресою: вул. 68 Десантників, 10, Миколаїв, 54003.

З кваліфікаційною роботою можна ознайомитися в бібліотеці Чорноморського національного університету імені Петра Могили за адресою: вул. 68 Десантників, 10, Миколаїв, 54003.

Автореферат представлений «29»квітня 2021р.

Секретар
екзаменаційної
комісії,

аспірант

Кандиба І. О.

ЗАГАЛЬНА ХАРАКТЕРИСТИКА РОБОТИ

Актуальність роботи. У сучасному світі створення відеоігор є одним з найбільш великих сегментів індустрії розваг. По швидкості росту за останні п'ять років індустрія відеоігор істотно випередила кіноіндустрію.

Велика кількість компаній займаються створенням ігор. На сьогоднішній день Unreal Engine є одним з найпопулярніших ігрових рушіїв. Кожен рік на цьому рушії створюються і виходять сотні популярних ігор. Його нинішня версія успішно застосовується в іграх самих різних жанрів.

Unreal Engine має настільки потужні інструменти для роботи з графікою, що багато голлівудських кінокомпаній використовують цей ігровий рушії в кіно. Unreal Engine було використано у багатьох фільмах від Marvel та в останніх Зіркових війнах. На сьогодні Unreal Engine це ігровий рушії з максимально швидко зростаючим числом команд.

Мета роботи – покращення ігрового процесу за рахунок вибору та застосування найбільш оптимального ігрового рушія шляхом реалізації гри з використанням системи Blueprint в Unreal Engine 4.

Об'єктом дослідження є ігровий процес комп'ютерних 3D ігор та шляхи його реалізації у середовищі Unreal Engine.

Предметом дослідження є технологія Blueprint в Unreal Engine 4 у якості ігрового рушія для 3D комп'ютерної гри.

ОСНОВНИЙ ЗМІСТ РОБОТИ

У **вступі** обгрунтовано актуальність обраної теми. Досліджено індустрію розробки комп'ютерних ігор та сфери використання ігрового рушія Unreal Engine. Також сформульовано мету та задачі досліджень, визначено об'єкт та предмет дослідження.

У **першому** розділі було проведено аналіз предметної області та існуючих аналізів. Аналогами розроблюваної гри є Killing Floor та World War Z. Із недоліків було визначено недостатню свободу гравця в процесі гри та обмежений вплив на ігрові характеристики персонажа під час гри. Перевагою розроблюваної гри є створення системи покращення навиків головного персонажа. Можливість гравця вибирати самому тактику виживання під час гри. Гра буде розвивати:

- мислення;
- реакцію;
- механічну пам'ять;
- моторику рук.

Було проведено аналіз вибору рушія. На сьогоднішній день існує два потужних ігрових рушія – Unreal Engine 4 та Unity 3D. У обох рушіїв великі набори інструментів. Unity більше орієнтований на невеликі проєкти, зроблені в основному в 2D і для мобільних пристроїв. Unreal Engine 4 має потужні інструменти для роботи з графікою, пост обробкою сцени. В Unity ці інструменти платні, але навіть купивши, не вдасться досягти того ж результату, що в Unreal Engine 4. Також перевагою є те, що в Unreal Engine 4 не потрібно турбуватися про вибір версії, про те, що вона буде оновлюватися щотижня і швидко застаріє. Коли Unreal Engine 4 отримує патч перехід практично непомітний. У випадку з Unity потрібно подумати про те, в якій версії потрібно працювати, тому що не кожна отримає довгострокову підтримку від компанії. Крім того, перевагою Unreal Engine над Unity є простота у створенні візуальних ефектів.

Також великою перевагою Unreal Engine 4 над Unity є те, що розробник комп'ютерних ігор Epic Games придбав компанію Quixel, що створила найбільшу в світі бібліотеку фотограмметричних матеріалів Megascans. Megascans - бібліотека відсканованих ресурсів і потужних інструментів, які можна безкоштовно використовувати з Unreal Engine - в робочому процесі архітектурної візуалізації.

Було проведено аналіз освітлення в Unreal Engine 4. Основним джерелом світла в Unreal Engine є Directional Light. Directional Light імітує світло, яке спускається з джерела, що нескінченно далеке. Це означає, що всі тіні, які відбиває це світло, будуть паралельними, що робить це ідеальним вибором для імітації сонячного світла. Інструмент Atmospheric Fog дає уявлення про розсіяне світло через атмосферу планети. Це робить рівні на відкритому повітрі більш реалістичними. Для створення туману. Для додавання реалізму та оптимізації використовується інструмент Exponential Height Fog шляхом приховування незавантажених частин карти.

У другому розділі було проведено моделювання об'єкту та предмету дослідження. Було створено діаграму використання (рис. 1).

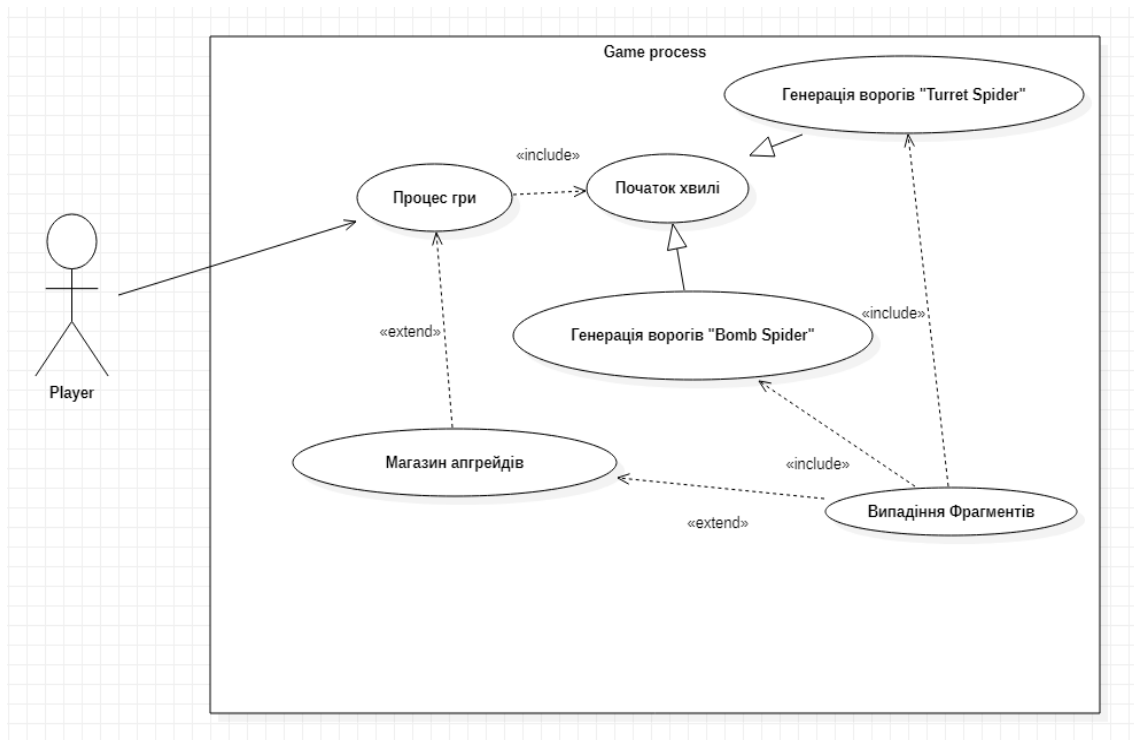


Рисунок 1 – Діаграма варіантів використання

Діаграма варіантів використання є вихідним концептуальним поданням системи в процесі її проектування і розробки.

Було створено діаграму класів (рис. 2). Діаграма класів – це статичне представлення структури моделі. Відображає статичні (декларативні) елементи, Такі як: класи, типи даних, їх зміст та їх відношення.

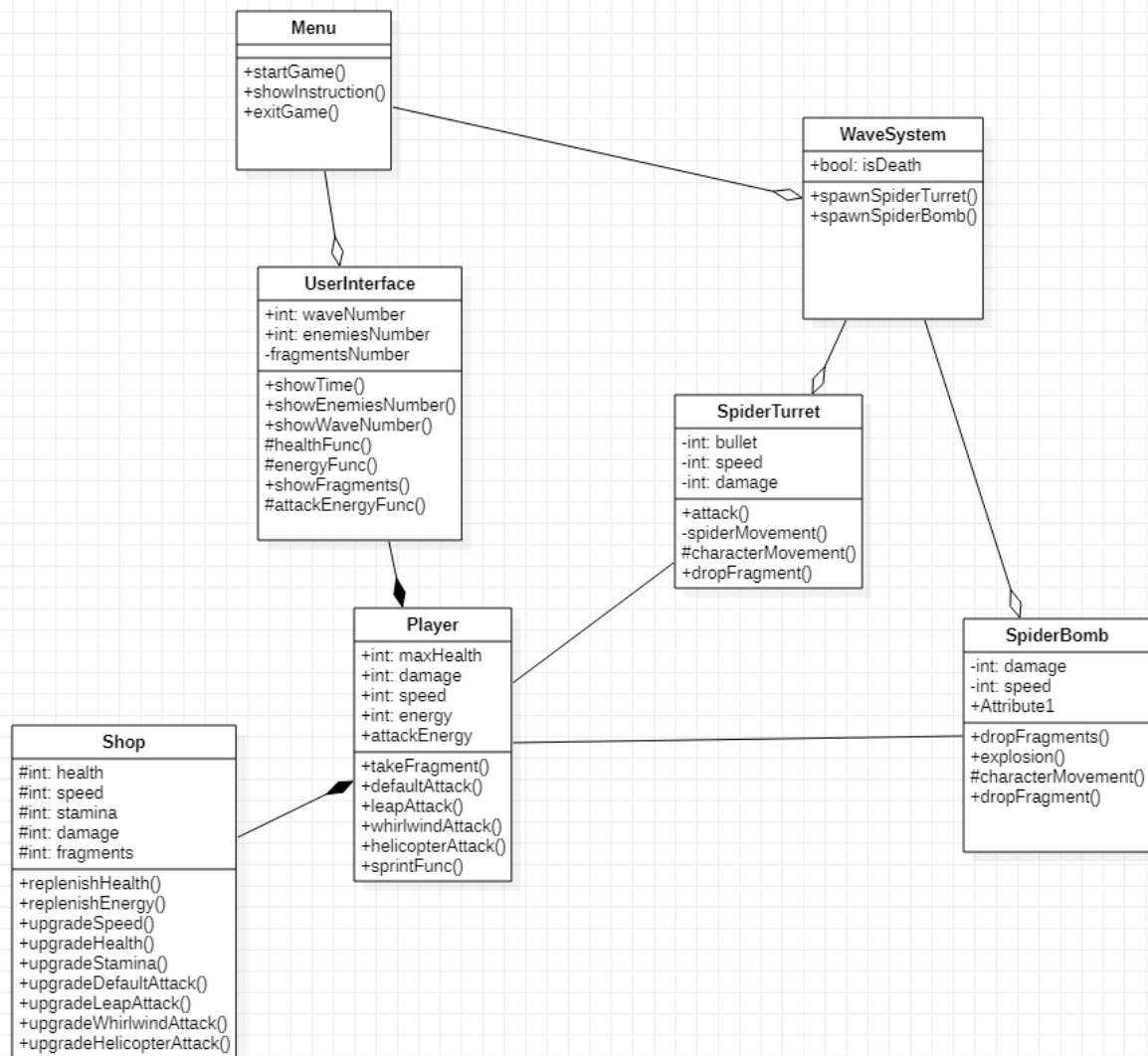


Рисунок 2 – Діаграма класів

Було створено діаграми діяльності (рис. 3). Діаграма діяльності – це UML-діаграма, на якій показані дії, специфікація виконуваної поведінки у вигляді координованого послідовного і паралельного виконання підлеглих елементів - вкладених видів діяльності і окремих дій, з'єднаних між собою потоками, які йдуть від виходів одного вузла до входів іншого.

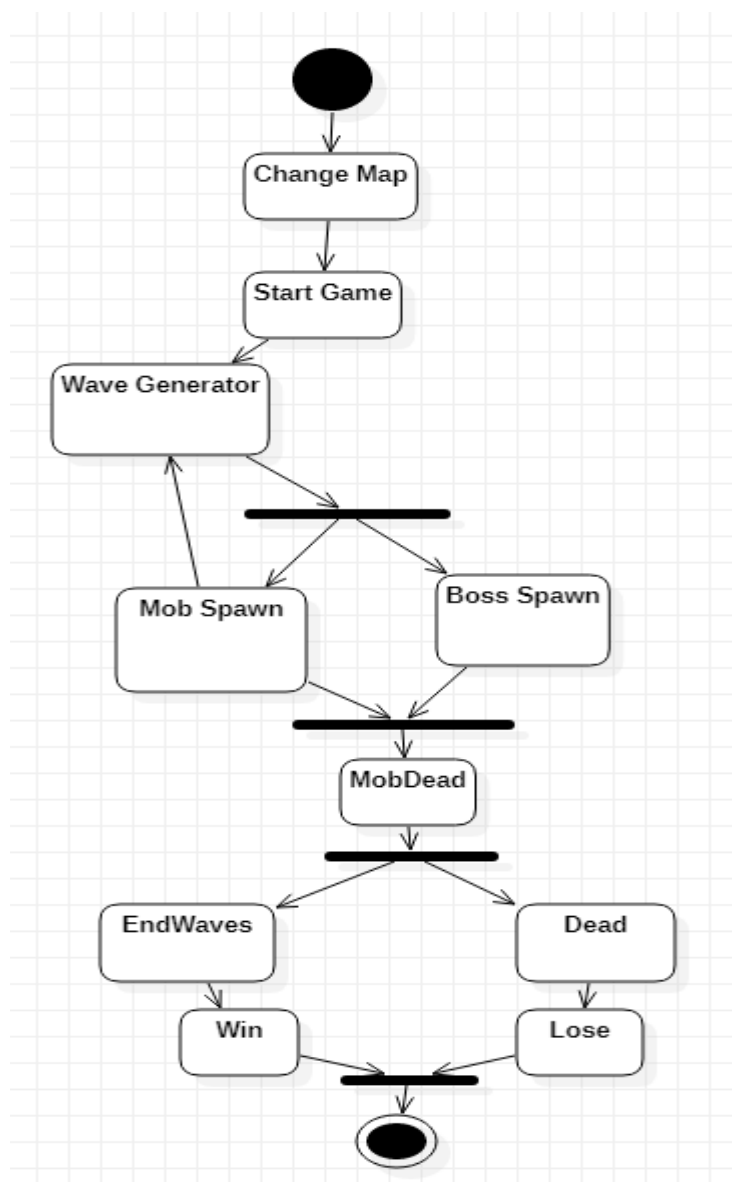


Рисунок 3 – Діаграма діяльності

Було ознайомлено с середовищем Unreal Engine 4. Unreal Engine 4 створено на мові програмування c++, тому скриптинг відбувається на c++ або на мові візуального програмування blueprint (BP) – де логіка будується шляхом послідовного зв'язку вузлів. Велика система інструментів і редакторів дозволяє організовувати ресурси і маніпулювати ними, щоб створити ігровий процес для гри. Компоненти UE4 включають в себе середовища: звукове, фізичне, графічне, введення і структуру ігрового процесу, а також онлайн-модуль. Звукове середовище відповідає за наявність музики і звуків у грі. Об'єкти стикаються і приходять в рух відповідно до законів руху Ньютона. Графічне середовище відповідає за виведення на

дисплей. Система введення перетворює натискання клавіш і кнопок гравцем в дії, що виконуються ігровим персонажем. Штучний інтелект в Unreal Engine складається з складних наборів правил, які допомагають об'єктам приймати рішення і виконувати свої задумані функції / поведінку. Unreal Engine 4 має можливість створювати ігри для багатьох платформ. При створенні гри, її можна переносити на різні платформи, такі як Web, iOS і Android. Unreal Engine має ряд редакторів, які допомагають у створенні гри. За замовчуванням Unreal Editor є редактором запуску Unreal Engine. Його можна розглядати як головний редактор, що забезпечує доступ до інших підсистем, таких як підсистеми Material і Blueprint.

Було розроблено макети для гри. Інтерфейс для гравця містить у собі інформацію про кількість часу у грі, кількість фрагментів, номер хвили ворогів, кількість ворогів.

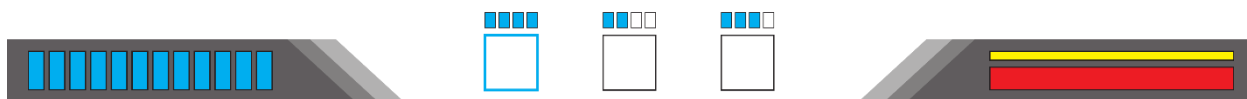


Рисунок 4 – Макет для інтерфейсу гравця

Було створено макет для магазину, який складається з таких покращень: здоров'я, енергія, швидкість, сила пошкодження, 3 спеціальні атаки, відновлення показників здоров'я та енергії. Всі показники можна скинути, для цього існує кнопка RESET.

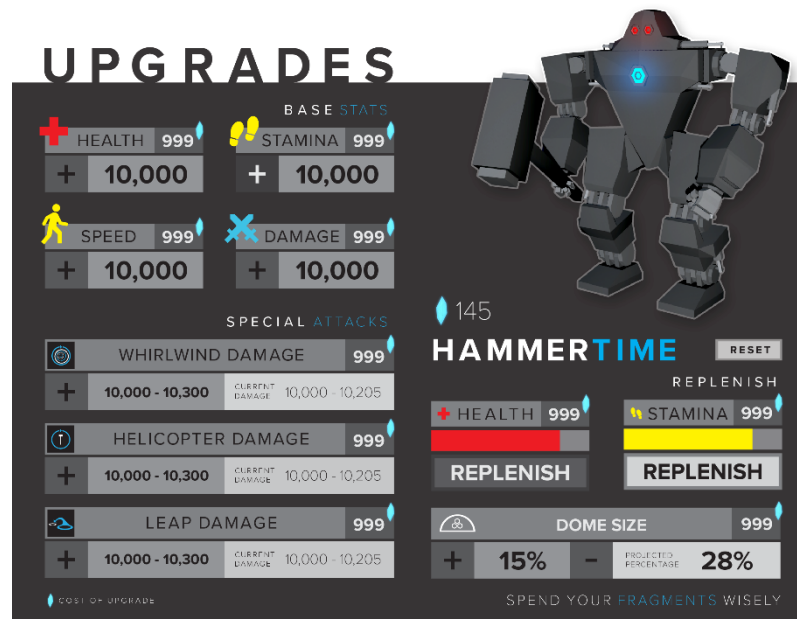


Рисунок 5 – Макет для вибору покращення
У третьому розділі було розроблено гру.

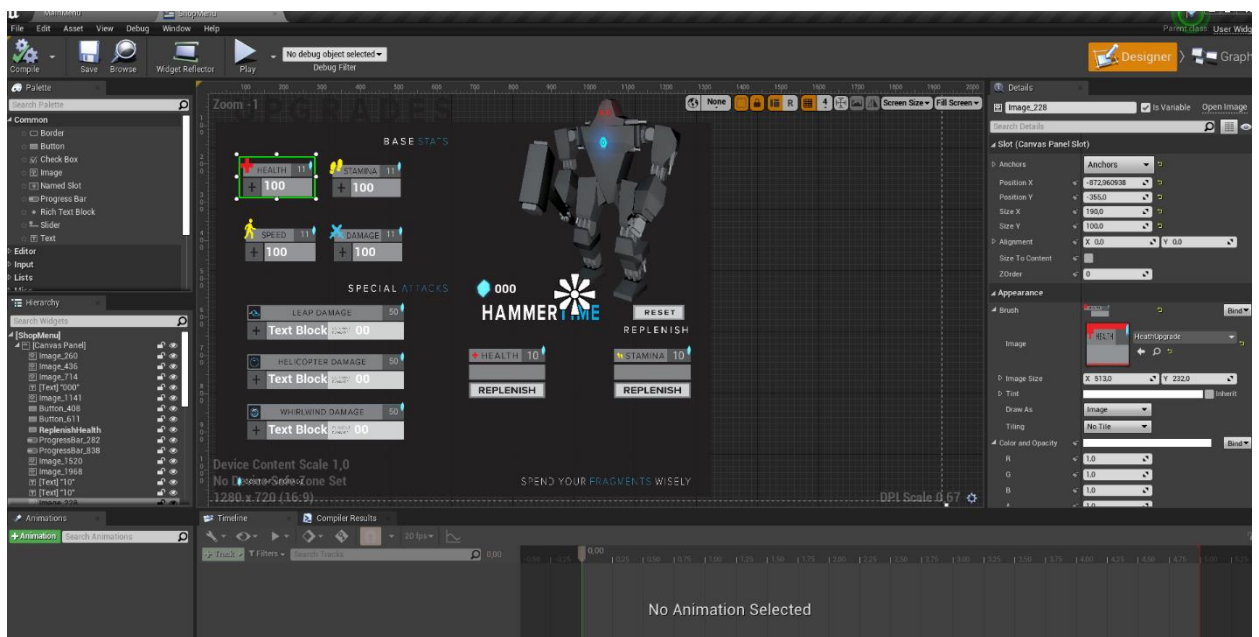


Рисунок 6 – Створення інтерфейсу магазину

Кожен елемент має логіку, у кожного покращення є динамічна ціна (рис. 7). Після натиснення на «+» та достатньої кількості фрагментів відбувається покращення характеристики, вартість наступної збільшується в два рази. Викликається об'єкт персонажа і від нього змінюється Health Upgrade Cost, що за замовченням дорівнює 5. Якщо достатньо фрагментів, у

персонажа збільшується вибрана характеристика, віднімаються фрагменти та змінюється вартість наступного покращення в два рази.

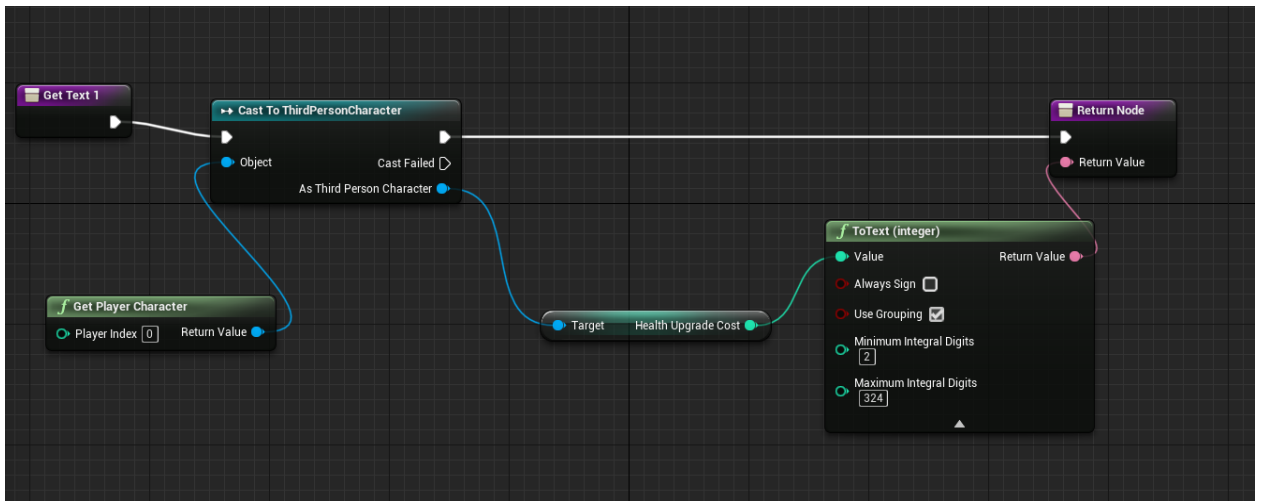


Рисунок 7 – Динамічна ціна вартості покращення

Було розроблено штучний інтелект. Для налаштування кута для периферичного зору та радіусу, наскільки далеко буде бачити штучний інтелект було використано компонент PawnSensing (рис. 8). Після цього було створено event – On See Pawn (рис. 9).

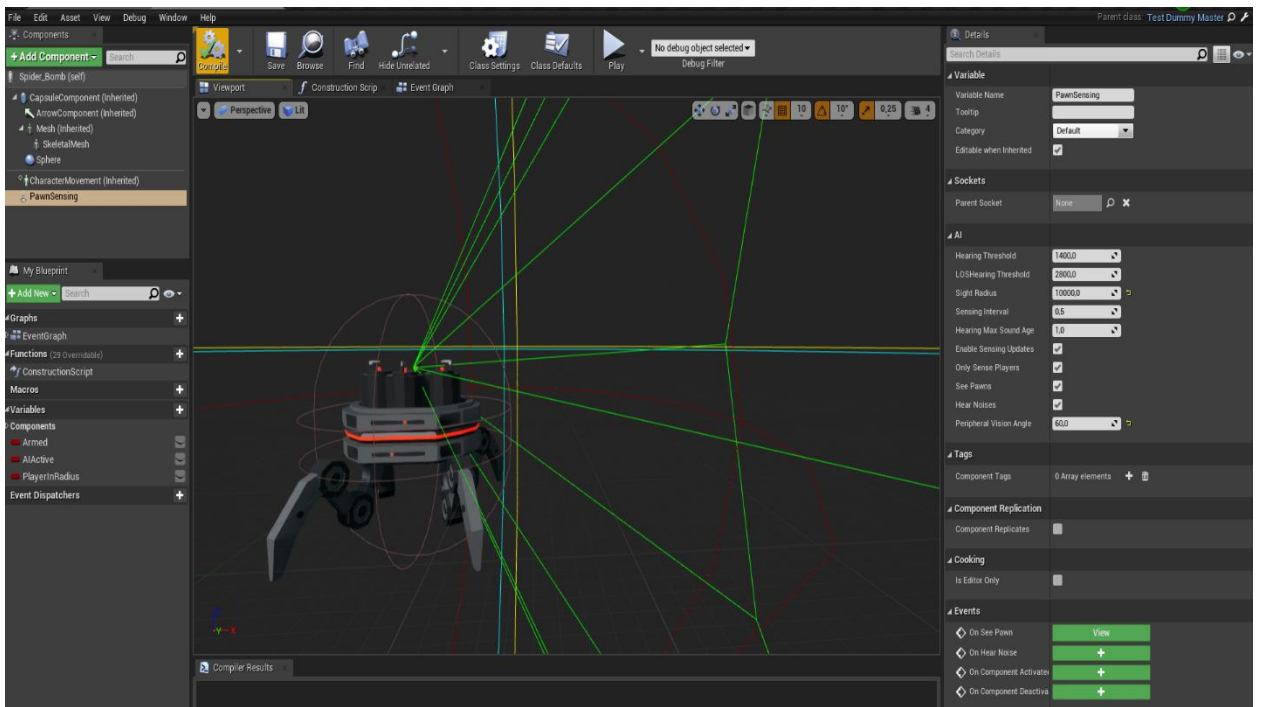


Рисунок 8 – Компонент PawnSensing

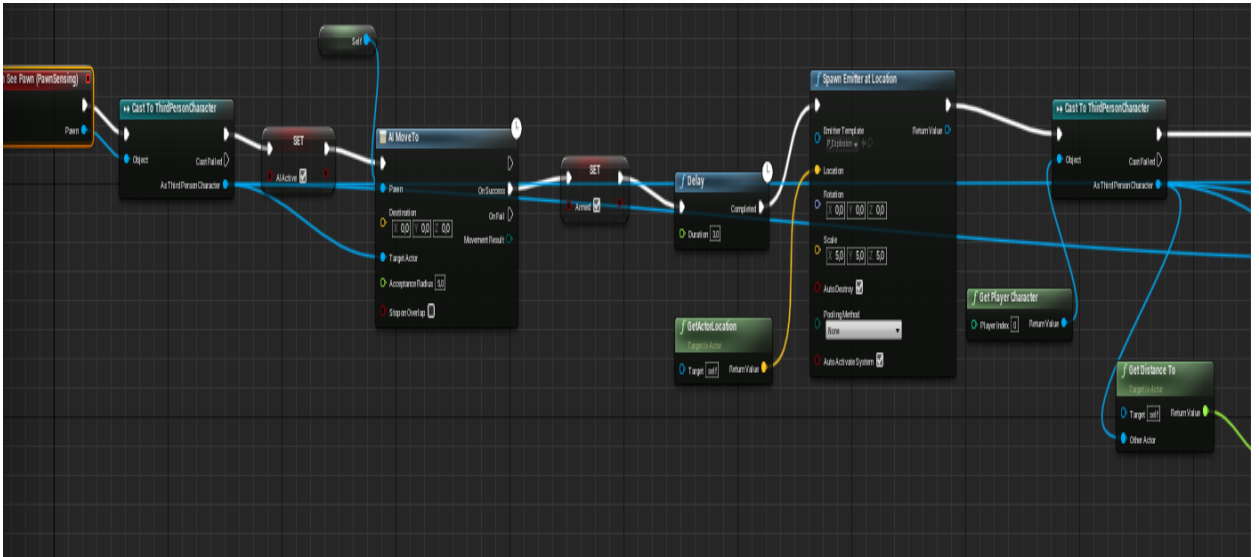


Рисунок 9 – Створення On See Pawn

Було створено карту для гри. Для цього спочатку було зроблено ландшафт за допомогою інструменту Landscape. Для ландшафту було використано авто-матеріал, який дозволяє налаштовувати перехід з низовини на гори, від текстури трави до текстури каменю (рис. 10). За допомогою інструменту Foliage на сцену було додано траву та дерева. Основним джерелом світла на сцені є Directional Light. Для обробки світла було використано ефект Post Process Volume. Для створення туману використано інструмент Exponential Height Fog.

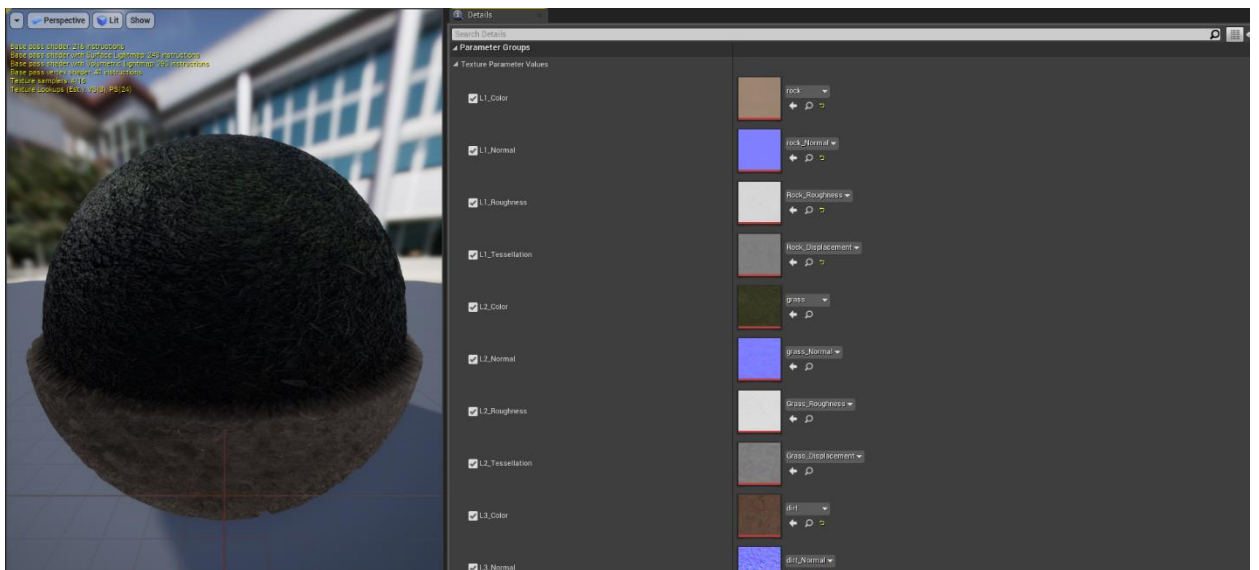


Рисунок 10 – Авто-матеріал ландшафту

Після того як було закінчено карту, було додано базу – місце появи героя. Коли герой підходить до бази відкриваються автоматично двері, куди

не можуть потрапити вороги, а також на карту додано генератори ворогів. Після генерації, вороги рандомно переміщуються в визначених межах, коли герой потрапляє у радіус зору штучного інтелекту, ворог прив'язується до гравця.

У **четвертому** розділі було проведено функціональне тестування та тестування продуктивності.

Було зроблено тести основних механік гри: купівля покращень в магазині, поповнення здоров'я, підбор фрагментів, систему прискорення, видимість штучного інтелекту. Після проведених тестів не було виявлено помилок, всі механіки працюють вірно.

Було проведено тестування продуктивності. Для цього в Unreal Engine 4 на сцені було обрано декілька спеціальних режимів. Режим Shader Complexity використовує кольоровий спектр, щоб вказати, наскільки оптимізована сцена. Зелений через червоний представляють лінійне відношення «дуже мало» до «багато», тоді як рожевий і білий означають великий стрибок до «дуже багато» пікселів (рис. 11).

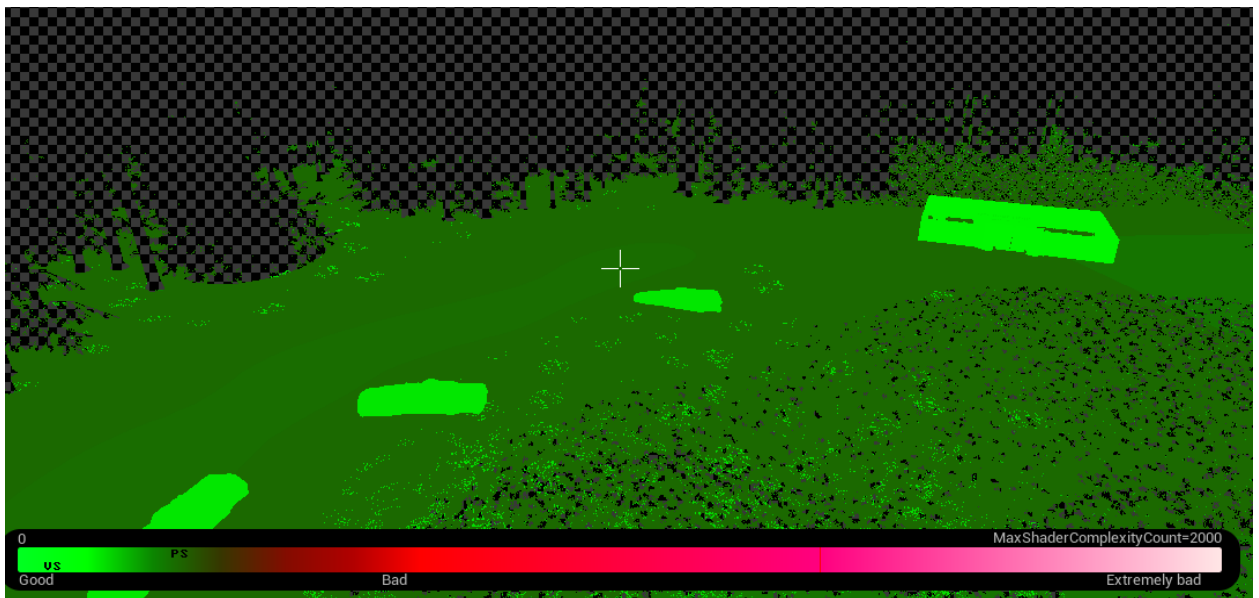


Рисунок 11 – Режим Shader Complexity

Режим Lightmap Destiny відображає щільність світлової карти об'єктів, які нанесені на текстуру. Сцена має рівномірну щільність, що дозволяє отримати постійне освітлення світлової карти.

У графічного процесора є багато блоків, що працюють паралельно, різні блоки прив'язані к різним частинам кадру. GPU Visualiser показує розбивку одного кадру на графічному процесорі. На рисунку 12 показано один кадр, який зайняв 15 мс для візуалізації.

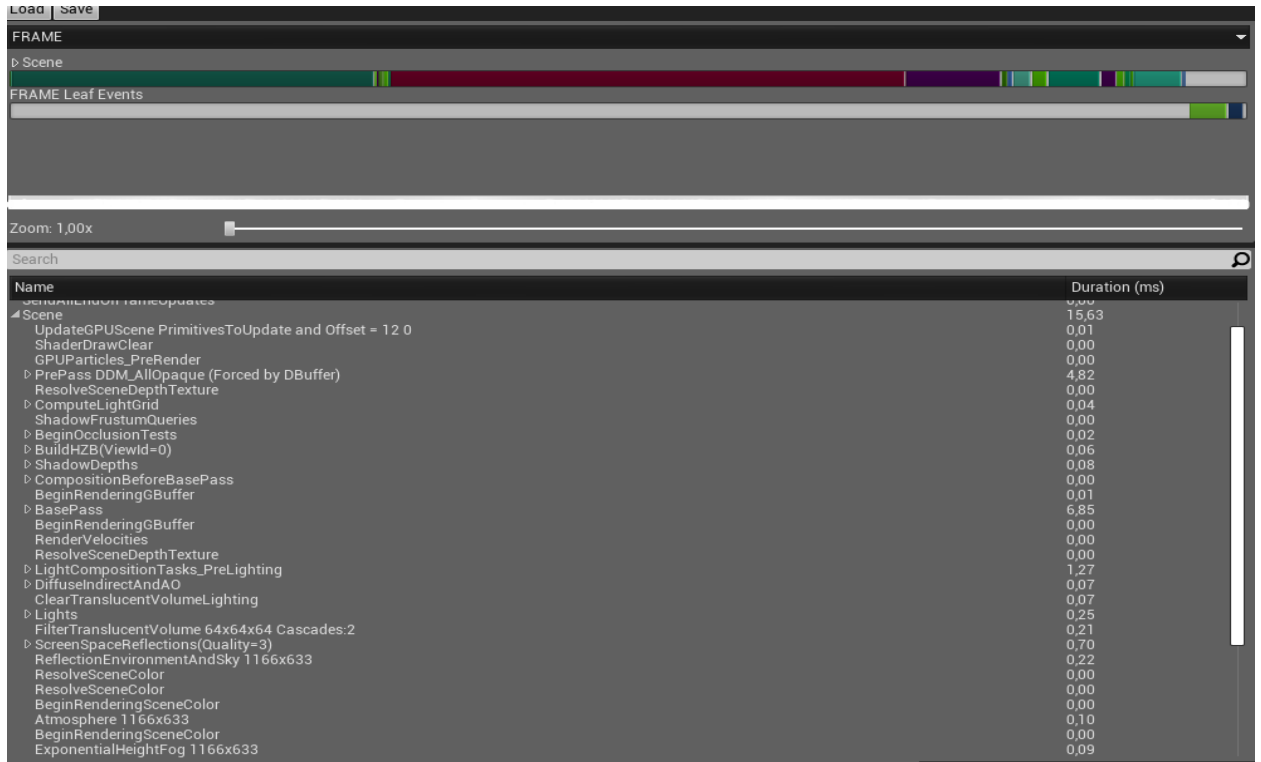


Рисунок 12 – Інструмент GPU Visualiser

ЗАГАЛЬНІ ВИСНОВКИ

Дипломна робота пов'язана з використанням технології розробки комп'ютерної 3D гри в жанрі виживання на мові візуального програмування Blueprint на Unreal Engine 4. Розроблена гра дозволяє вдосконалити існуючі ігри в жанрі виживання шляхом збільшення свободи гравця та створення унікальних механік гри.

Проведено аналіз предметної області та виявлено аналоги гри. В аналогах виявлено функції ігрового процесу, основні переваги та недоліки.

Досліджено та порівняно ігрові рушії Unity3D та Unreal Engine 4. Для комп'ютерних ігор краще підходить Unreal Engine за рахунок своїх можливостей створення графіки.

Змодельовано та спроектовано гру, що включає в себе побудову діаграми варіантів використання та діаграми класів. Також діаграми станів та послідовності. Це допомогло визначити базові функції системи та зв'язки у грі.

Розроблено авто-матеріал, виконано роботу з ландшафтом, використано інструмент Foliage. Розглянуто інструменти для роботи з освітленням: Directional Light, Atmospheric Fog, SkyLight. Для пост-обробки було використано інструмент PostProcess Volume.

Для створення гри було використано мову візуального програмування Blueprint. Розроблено елементи штучного інтелекту.

Після розробки гри було проведено функціональне тестування та тестування продуктивності. Після проходження функціонального тестування не виявлено помилок, гра працює коректно. Тестування продуктивності показало достатню кількість FPS для комфортної гри.

Отже мета, що полягала у покращенні ігрового процесу за рахунок вибору та застосування найбільш оптимального ігрового рушія шляхом реалізації гри з використанням системи Blueprint в Unreal Engine 4 була досягнута.