

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

автоматизації і інформаційних технологій

(факультет)

інформаційних технологій

(кафедра)

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО АТЕСТАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ МАГІСТРА**

на тему: «Розробка 3д шутеру на двигуні UE5»

Гронський Олександр Андрійович
(прізвище, ім'я та по батькові магістра повністю)

Київ 2023 р.

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

автоматизації і інформаційних технологій

(факультет)

інформаційних технологій

(кафедра)

ЗАТВЕРДЖУЮ

Завідувач кафедри ІТ

Тетяна ГОНЧАРЕНКО

„__” _____ 2023 року

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО АТЕСТАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ МАГІСТРА**

на тему: " Розробка 3д шутеру на двигуні UE5"

Виконав: студент 2 – го курсу, групи КН – 1

Спеціальності: 122 «Комп’ютерні науки» .

(шифр і назва напрямку підготовки, спеціальності)

Магістрант Гронський Олександр Андрійович.

(прізвище та ініціали)

Керівник Саченко Ілля Анатолійович .

(прізвище та ініціали)

Рецензент _____ .

(прізвище та ініціали)

Київ, 2023 р.

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

Факультет: автоматизації і інформаційних технологій .

Кафедра: інформаційних технологій проектування та ПМ .

Освітній рівень: «магістр за ОПП» .

Спеціальність: 122 «Комп'ютерні науки» .

ЗАТВЕРДЖУЮ

Завідувач кафедри ІТ

Тетяна ГОНЧАРЕНКО

„__” _____ 2023 року

**З А В Д А Н Н Я
ДО ВИКОНАННЯ АТЕСТАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ МАГІСТРА**

1. Тема роботи: Розробка 3д шутеру на двигуні UE5 затверджена наказом ректора КНУБА № 1280/2 від «26» Червня 2023р.

2. Керівник роботи: Саченко Ілля Анатолійович.

3. Строк подання студентом роботи до захисту: 4 Грудня 2023р.

4. Зміст пояснювальної записки за розділами:

P.1. Дослідження жанру комп'ютерних ігор «шутер», огляд різноманітних технологій розробки комп'ютерних ігор та вибір конкретної технології для реалізації проекту.

P.2. Опис ігрового рушія Unreal Engine 5.

P.3. Розробка комп'ютерної гри у жанрі шутер.

P.4. Демонстрація відеогри.

5. Інформаційні слайди:

C.1. Огляд різноманітних технологій розробки комп'ютерних ігор

C.2. Інтерфейс

C.3. Об'єкти

С.4. Blueprints

6. Календарний план виконання атестаційної випускної роботи

Види робіт та їх зміст	Дата виконання
Р.1. <u>Дослідження жанру комп'ютерних ігор «шутер», огляд різноманітних технологій розробки комп'ютерних ігор та вибір конкретної технології для реалізації проекту.</u>	01.09.22
Р.2. <u>Опис ігрового рушія Unreal Engine 5.</u>	28.09.22
Р.3. <u>Розробка комп'ютерної гри у жанрі шутер.</u>	02.10.22
Р.4. <u>Демонстрація відеогри.</u>	14.10.22
Остаточне оформлення роботи	16.11.22
Направлення роботи на рецензування, перевірку на плагіат	24.11.22
Попередній захист роботи на кафедрі	30.11.22

7. Консультанти розділів атестаційної випускної роботи

Розділ	Прізвище, ініціали та посада консультанта	Перевірів	
		дата	підпис
Розділ 1.			
Розділ 2.			
Розділ 3.			
Розділ 4.			
Розділ 5.			

8. Дата видачі завдання: «22» Червня 2022р.

Керівник

(підпис)

Саченко І.А.

(прізвище та ініціали)

Магістрант

(підпис)

Гронський О. А.

(прізвище та ініціали)

РЕЗЮМЕ

Київський національний університет будівництва і архітектури

Гронський Олександр Андрійович

факультет автоматизації і інформаційних технологій,

група КН-1

Тема атестаційної випускної роботи: «Розробка 3д шутеру на двигуні
UE5»

освітній рівень: магістр,

спеціальність: 122 «Комп'ютерні науки»,

Науковий керівник: Саченко Ілля Анатолійович

Обсяг роботи. Атестаційна випускова робота магістра складається: розділів 4, стор. 75, таблиць 1, рис. 80, завдання, анотація, вступу, висновку та списку використаних джерел.

Актуальність теми. Розробка 3D шутера на Unreal Engine 5 (UE5) є захоплюючим викликом у галузі геймдеву. Завдяки передовим графічним технологіям, таким як Nanite Virtualized Geometry та система світла і тіней Lumen, UE5 відкриває широкі можливості для створення захоплюючих ігрових світів. Випущений у 2021 році, цей двигун став перспективним вибором для розробників, а його інструменти візуального програмування дозволяють креативно реалізувати ідеї та концепції в грі.

У вступі визначені основні напрямки дослідження, обґрунтовано актуальність теми, сформульовано мету та основні завдання системи, показано зв'язок із програмами, планами, темами.

У першому розділі «Дослідження жанру комп'ютерних ігор «шутер», огляд різноманітних технологій розробки комп'ютерних ігор та вибір конкретної технології для реалізації проекту» розглянуті різні шутери, особливості жанру, специфіки гри та обрана технологія розробки.

У другому розділі «Опис ігрового рушія Unreal Engine 5.» розглянуто рушій Unreal Engine 5 як потужний інструментарій для розробки ігор.

У третьому розділі «Розробка комп'ютерної гри у жанрі шутер» подано детальну інструкцію по користуванню рушієм Unreal Engine 5 та продемонстровані різні етапи розробки шутеру.

У четвертому розділі «Демонстрація відеогри» продемонстровано базовий геймплей в шуторі.

Ключові слова: шутер, Unreal Engine 5, Blueprints.

Якість оформлення проекту. Атестаційна випускна робота магістра оформлена у відповідності до діючих нормативних документів та методичних вказівок до виконання дипломних робіт для студентів спеціальності 122 «Комп'ютерні науки». Порушень та зауважень під час розробки та перевірки дипломної роботи не виявлено.

Загальний висновок стосовно роботи та присвоєння авторіві освітньо-кваліфікаційного рівня «магістр». Робота виконана якісно та на високому рівні, студент продемонстрував достатній рівень теоретичної підготовки та сформованих практичних навичок в області сучасних інформаційних технологій. Заслуговує оцінки «Відмінно».

Науковий керівник _____ / Саченко Ілля Анатолійович./
(підпис)

Посада, місце роботи: професор кафедри інформаційних технологій проектування та прикладної математики КНУБА

«23» Листопада 2022р.

АНОТАЦІЯ

Гронський О.А. «Розробка 3д шутеру на двигуні UE5».

Атестаційна випускна робота магістра за спеціальністю: 122 «Комп'ютерні науки». – Київський національний університет будівництва та архітектури. – Київ, 2022 р.

Атестаційна робота магістра присвячена створенню комп'ютерної гри в жанрі шутер на рушії Unreal Engine 5 із застосуванням інформаційних технологій. Результатом розробки є гра шутер в якій реалізовані базові вимоги до ігор цього жанру.

Ключові слова: шутер, Unreal Engine 5, Blueprints.

ANNOTATION

Gronsky O.A. «Experience of a 3D shooter on the UE5».

Attestation of the master's degree for the specialty: 122 “Computer Science”. – Kiev National University of Civil Engineering and Architecture. – Kiev, 2022 r.

The master's attestation work is devoted to the creation of a computer game in the shooter genre on the Unreal Engine 5 with the use of information technologies. The result of the development is a shooter game in which the basic requirements for games of this genre are implemented.

Keywords: shooter, Unreal Engine 5, Blueprints.

РЕЦЕНЗІЯ

на атестаційну випускную роботу

Студента Гронський Олександр Андрійович

Факультет автоматизації і інформаційних технологій

спеціальності 122 «Комп'ютерні науки»

Тема роботи: Розробка 3D шутеру на двигуні UE5.

Обсяг роботи: атестаційна випускова робота магістра складається: розділів 4, стор. 75, таблиць 1, рис. 80, завдання, анотація, вступу, висновку та списку використаних джерел.

Висновок про відповідність завданню: робота виконана у повній відповідності до завдання і у встановлений термін .

Актуальність обраної теми: Створення 3D шутера на Unreal Engine 5 (UE5) – це захоплюючий виклик у світі геймдеву, в якому передові графічні технології, включаючи Nanite Virtualized Geometry та систему світла і тіней Lumen, надають шалено широкі можливості для втілення захоплюючих ігрових світів. За своєю передовістю, випущений у 2021 році, UE5 стає важливим інструментом для розробників, а його інтуїтивно зрозумілі інструменти візуального програмування сприяють творчому втіленню ідей та концепцій в межах гри.

Практичне значення роботи: розробка 3D шутера на Unreal Engine 5 (UE5) має практичне значення в галузі геймдеву. Використання передових технологій, таких як Nanite Virtualized Geometry і Lumen, надає можливість створювати ігри з вражаючою графікою та реалістичністю. Розробка на UE5 дозволяє створювати інноваційні та конкурентоспроможні продукти, а візуальне програмування надає гнучкість для реалізації творчих ідей.

Якість оформлення роботи: випускна робота оформлений у відповідності до діючих нормативних документів та методичних вказівок для студентів спеціальності 122 «КН»

Зауваження та побажання: Зауважень не виявлено

Загальний висновок стосовно роботи та надання авторові освітнього ступеня “магістр”: робота виконана на високому рівні, студент продемонстрував високий рівень теоретичної підготовки та сформованих практичних навичок в області сучасних інформаційних технологій. Заслугує оцінки «відмінно».

Рецензент _____ / проф., к.т.н., Терентьев О.О. /
(підпис) (науковий ступінь, вчене звання, прізвище та ініціали)

Посада, місце роботи: професор кафедри інформаційних технологій проектування та прикладної математики КНУБА
«24» Листопада 2022р.

Зміст

<u>РОЗДІЛ 1. Дослідження жанру комп'ютерних ігор «шутер», огляд різноманітних технологій розробки комп'ютерних ігор та вибір конкретної технології для реалізації проекту</u>	11
1.1. Дослідження жанру комп'ютерних ігор «шутер»	11
1.2. Огляд різноманітних технологій розробки комп'ютерних ігор	17
1.3. Таблиця най популярніших ігор на рушіях Unity та Unreal Engine	20
1.4. Аналіз існуючих розробок	22
1.5. Вибір конкретної технології для реалізації проекту	27
<u>РОЗДІЛ 2. Опис ігрового рушія Unreal Engine 5</u>	29
2.1. Основні факти	30
2.2. Інтерфейс	30
2.3. Шаблони та демо-проекти	36
2.4. Редактор	37
2.5. Об'єкти	39
2.6. С++ чи Blueprints	39
2.7. Blueprints	40
2.8. Меши, матеріали та ефекти	42
2.9. Інтерфейс користувача (UI)	42
2.10. Звук	43
2.11. Збірка	43
<u>РОЗДІЛ 3. Розробка комп'ютерної гри у жанрі шутер</u>	44
3.1. Створення проекту	44
3.2. Налаштування камери	47
3.3. Шкала життя і обладунків гравця	51
3.5. Нанесення урону	53
3.6. Додавання повноцінного 3D персонажа з анімацією	54
3.7. Додавання зброї до гравця	58
3.8. Стрільба зі зброї	60
3.9. Додавання ворогів та знищення їх	65
3.10. Урон від ворогів	67
3.11. Штучний інтелект	68
<u>РОЗДІЛ 4. Демонстрація відеогри</u>	70

ВСТУП

Індустрія комп'ютерних ігор виникла в середині 1970-х років і вже досягла величезних розмірів, надаючи доходу в кілька мільярдів доларів щорічно. Це пояснюється поширенням комп'ютерних технологій та розвитком Інтернету, що зробило комп'ютерні ігри більш доступними для кінцевого користувача.

Для гравців, щоб просто насолоджуватися грою, потрібно мати лише комп'ютер або ігрову приставку та копію гри. Завдяки широкому поширенню мережі Інтернет, отримати копію гри можливо навіть не виходячи з дому. Більше того, споживачам не потрібно мати спеціальних знань для вибору гри, що відрізняє їх від інших видів розваг, де часто потрібна певна екіпіровка.

Важливо зауважити, що комп'ютерні ігри втратили своє позиціонування як просто засіб відпочинку. Сьогодні за допомогою ігрових технологій створюються спеціальні симуляційні комплекси для навчання та тренування фахівців у різних галузях, від лісорубів до пілотів реактивних літаків.

На жаль, в Україні ігрова індустрія розвинена менше порівняно з іншими країнами пострадянського простору через пізнє прийняття культури комп'ютерних розваг. Навіть при зростаючому попиті, кількість українських розробників, які можуть конкурувати на міжнародному ринку, залишається обмеженою.

Найбільш відомі українські компанії у цій сфері - 4A Games і GSC Game World. 4A Games випустили популярну серію ігор "Metro", яку грають люди з усього світу. GSC Game World славиться серією "Козаки" та "S.T.A.L.K.E.R.", особливо популярною в пострадянських країнах.

Є інші українські розробники, але їм важко привернути велику аудиторію. Україна також є домівкою для двох студій Ubisoft Entertainment, які активно працюють над великими ігровими проектами.

Розвиток технологій у цьому напрямку в Україні вважається перспективним. Випускна кваліфікаційна робота спрямована на розробку комп'ютерної гри на базі Unreal Engine 5.

Завдання включають вивчення особливостей шутерів, розробку сценарію та концепції, вибір засобів реалізації, підготовку анімації та реалізацію самої гри.

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ЖАНРУ КОМП'ЮТЕРНИХ ІГОР «ШУТЕР», ОГЛЯД РІЗНОМАНІТНИХ ТЕХНОЛОГІЙ РОЗРОБКИ КОМП'ЮТЕРНИХ ІГОР ТА ВИБІР КОНКРЕТНОЇ ТЕХНОЛОГІЇ ДЛЯ РЕАЛІЗАЦІЇ ПРОЕКТУ

1.1. Дослідження жанру комп'ютерних ігор «шутер»

Комп'ютерні ігри у жанрі "шутер" є ключовим елементом геймінгової індустрії. Їхня популярність почалася в середині 1970-х років і, з тих пір, стала значущою галуззю з мільярдами прибутку щорічно. Це дослідження присвячено аналізу історії, особливостей та впливу цього жанру.

1.1.1. Історія та Еволюція

Жанр "шутер" в історії відеоігор з'явився в середині 1970-х років і пройшов вражаючий шлях еволюції та розвитку. Початково це були прості аркади з абстрактними графічними елементами, де гравець мав стріляти по мішенях. З розвитком технологій та графічних можливостей жанр став більш реалістичним.

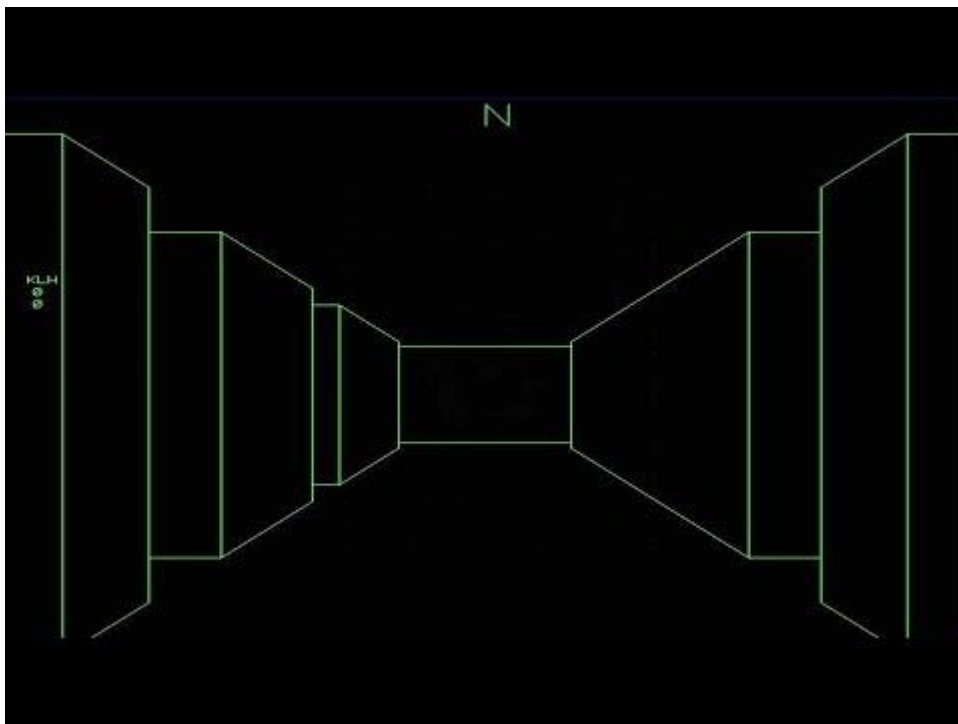


рис. 1.1. перший шутер у світі - Maze War

Перші "шутери" використовувалися для навчання військових та поліцейських, і лише потім вони перетворилися на розважальні ігри. В середині 1990-х років виникли перші 3D "шутери", які надали гравцеві можливість іммерсивного занурення в ігровий світ.

З появою інтернету у 2000-х роках мультиплеєрні режими стали стандартом для "шутерів". Це відкрило нові можливості для гравців, дозволяючи їм змагатися з іншими гравцями з усього світу.

Сучасні "шутери" використовують високотехнологічні графічні рішення, віртуальну та розширену реальність, а також високу ступінь реалізму. Жанр постійно вдосконалюється та адаптується до потреб гравців, що свідчить про його величезну популярність та важливість у світі відеоігор.

1.1.2. Характеристики "Шутерів"

Характеристики "шутерів" в комп'ютерних іграх визначають цей жанр як великою категорією ігор, що базується на вогневих сценах та активному залученні гравця в інтенсивні бойові події.

Однією з ключових характеристик є динаміка геймплею. "Шутери" відрізняються швидким темпом, де гравець повинен приймати рішення за долю секунди, реагуючи на ворожі атаки та змінюючи тактику у реальному часі. Ця динаміка створює напружені та захоплюючі моменти, де кожен вистріл може бути вирішальним.

Іншою важливою характеристикою є реалістичність вогневих сцен та зброї. Багато "шутерів" прагнуть максимально точно відобразити реальність, використовуючи різноманітні моделі вогнепальної зброї та її характеристики. Це включає в себе реалістичну модель пострілу, перезарядку та вплив різних типів зброї на геймплей.

Мультиплеєрні режими є ще однією визначальною характеристикою. Гравці можуть змагатися один проти одного або у складі команд в онлайн-баталіях, розвиваючи тактичні та стратегічні навички. Це додає соціальний аспект ігор, дозволяючи гравцям взаємодіяти та конкурувати між собою.

Характеристики графіки також важливі для "шутерів". Висока деталізація і реалістичність графіки допомагають створити іммерсивне враження від ігрового процесу, занурюючи гравця в фантастичний світ бойових операцій.

Загальна характеристика "шутерів" полягає в їхньому впливі на розвиток навичок гравців, таких як координація рухів, стратегічне мислення та реакційність. Цей жанр продовжує залишатися популярним завдяки поєднанню інтенсивності гри та соціальної взаємодії між гравцями.

1.1.3. Популярні Ігри та Серії

У світі комп'ютерних ігор існує багато популярних серій "не онлайн шутерів", які надають гравцям можливість насолоджуватися захоплюючим геймплеєм без прив'язки до мережевого з'єднання. Однією з таких серій є "BioShock".

"BioShock" відзначається унікальним поєднанням стрільби та рольового елемента. Гравці розгортають сюжет в атмосферному підводному світі, де

кожне рішення впливає на подальший хід подій. Гра отримала визнання за свою неперевершену атмосферу та креативний підхід до геймдизайну.

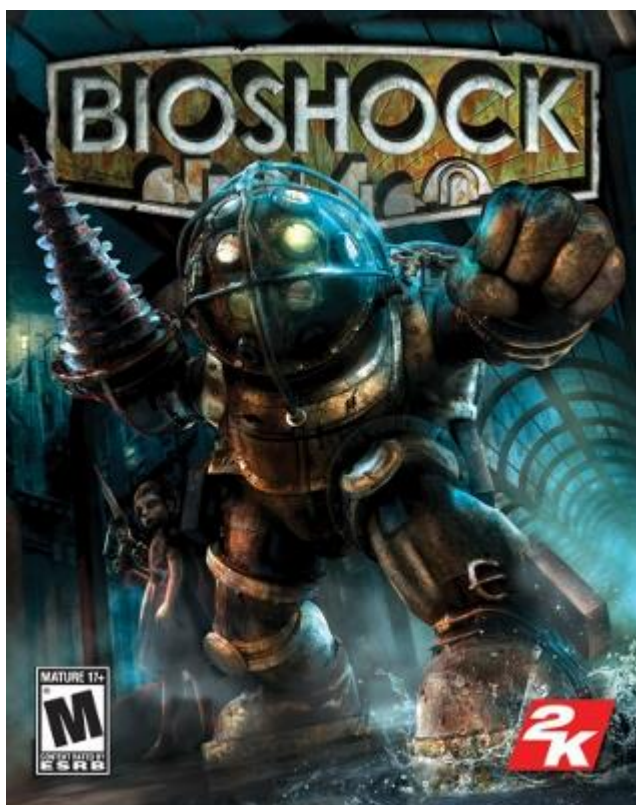


Рис. 1.2. BioShock

Іншою захоплюючою серією є "Far Cry". Гравці можуть насолоджуватися великими, відкритими світами та використовувати різні стратегії для подолання ворогів. Сюжети, що розгортаються в екзотичних місцях, і велика кількість доступної техніки додають грі захоплюючості та різноманітності.

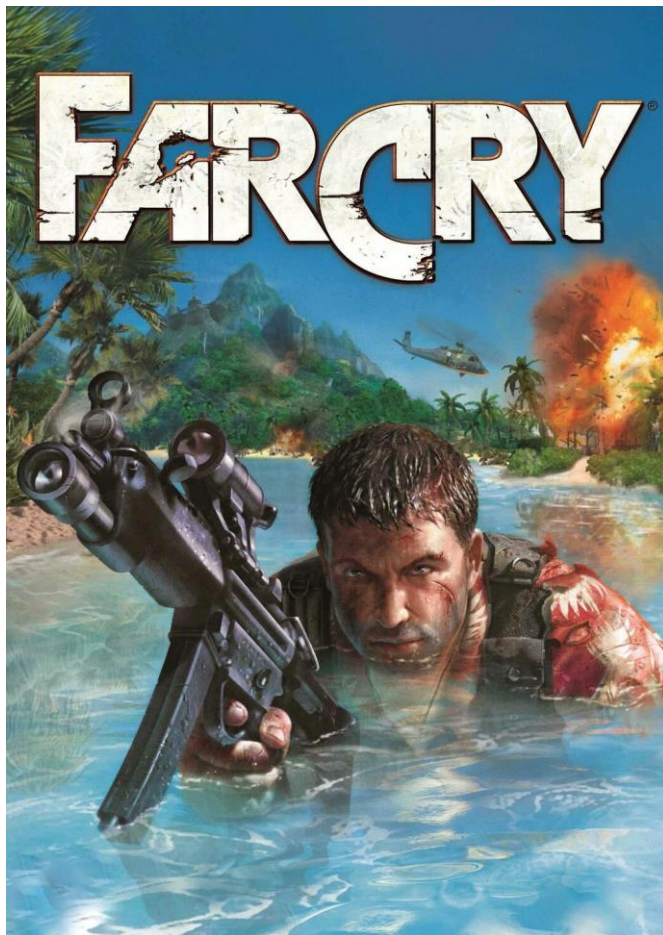


Рис. 1.3. Far Cry

"DOOM" - класична серія "не онлайн шутерів", яка визначається високою динамікою та екшеном. Гравці взяли на себе роль Doom Slayer, щоб боротися з безліччю демонів на Марсі та в інших адських локаціях. Гра славиться своєю кровопролитністю та адреналіновим геймплеєм.

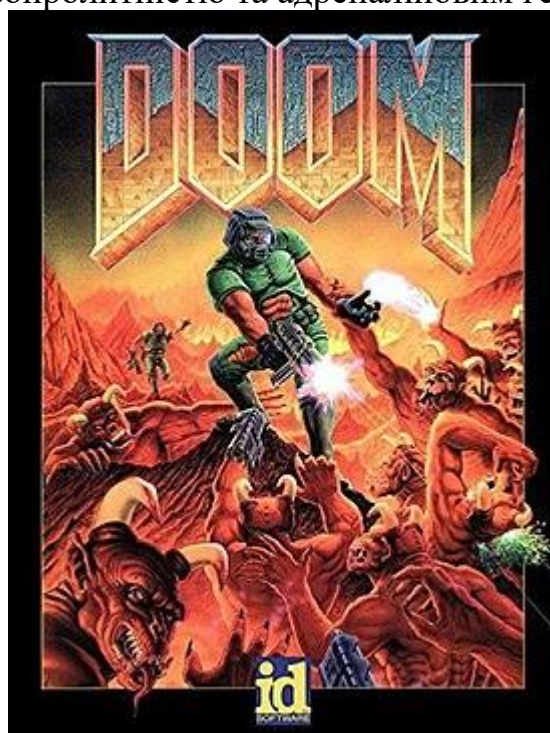


Рис. 1.4. DOOM

Ще однією надзвичайно популярною серією "шутерів" є "Resident Evil". Ця франшиза визначається якісним комбінуванням жаху та бойового геймплею. "Resident Evil" перетворилася зі своїх скромних початків у грандіозний проект, привертаючи увагу гравців своєю загадковою сюжетною лінією та страхітливими монстрами.



Рис.1.5. Resident Evil: Director`s Cut

Гравці відправляються в темні, занедбані локації, де їх чекають загадкові загрози та таємничі експерименти. Що робить "Resident Evil" унікальною серією - це поєднання стрільби та розгадування головоломок, що створює непередбачувану та напружену атмосферу для гравців.

Ці серії "шутерів" відмінно підходять для гравців, які шукають захоплюючий досвід від стрільби, не виходячи в онлайн-битви, і насолоджуються сюжетом та атмосферою гри. Вони підтверджують, що і без мережевого з'єднання геймерам доступно безліч захоплюючих пригод у світі "шутерів".

1.1.4. Вплив на Гравців та Геймінгову Спільноту

Вплив жанру "шутер" на гравців та геймінгову спільноту є важливою та значущою аспектом у світі комп'ютерних ігор. Цей жанр не лише розважає гравців, але також впливає на їхні навички, соціальні взаємодії та сприйняття геймінгу.

Однією з ключових особливостей "шутерів" є розвиток реакцій та стратегічного мислення гравців. Висока динаміка і швидкість геймплею змушують гравців швидко приймати рішення та взаємодіяти з великою кількістю інформації в реальному часі. Це сприяє покращенню рефлексів та здатності швидко адаптуватися до змін у грі.

З іншого боку, "шутери" часто включають онлайн-режим, де гравці взаємодіють один з одним. Це створює сприятливе середовище для соціальних взаємодій, спілкування та формування геймінгових спільнот. Гравці можуть об'єднуватися в команди, конкурувати або співпрацювати, що сприяє розвитку комунікативних навичок та сприяє формуванню онлайн-дружби.



Рис. 1.6. перший турнір по CS-GO у 2013(онлайн шутер)

Загальний вплив "шутерів" на геймінгову спільноту полягає в створенні захопливих та захоплюючих вражень для гравців, що додає динамізму та різноманіття у світ комп'ютерних ігор. Цей жанр впливає на культуру геймінгу, стимулюючи зміни та інновації, які підтримують інтерес до геймінгу серед різних поколінь гравців.

1.1.5. Тенденції та Майбутнє

Тенденції та майбутнє жанру "шутер" в світі комп'ютерних ігор обіцяють захопливі нововведення та технологічний розвиток. Останнім часом спостерігається ряд визначних тенденцій, які можуть визначити майбутнє цього жанру.

Однією з ключових тенденцій є зростання популярності віртуальної реальності (VR) у "шутерах". Гравці отримають можливість поглибленої іммерсії в гру, де їхні дії та реакції стають більш реалістичними та емоційно насиченими.

Ще однією важливою тенденцією є розвиток штучного інтелекту (ШІ) для створення більш складних та інтелектуальних опонентів у грі. Це дозволяє створювати більш реалістичні та викликові ситуації для гравців, підвищуючи рівень геймплею.

Модифікація та персоналізація героїв стають ще однією тенденцією, що набуває популярності. Гравці отримують можливість адаптувати свої персонажі до власних вподобань, що надає більше відчуття участі та індивідуальності у грі.

Щодо майбутнього "шутерів", очікується подальше вдосконалення графічних технологій, використання технологій штучного інтелекту та

продовження експериментів із сполученням різних ігрових жанрів. Такі тенденції обіцяють захопливий розвиток жанру та нові можливості для гравців у світі "шутерів".

1.2. Огляд різноманітних технологій розробки комп'ютерних ігор

Однією з ключових складових комп'ютерних ігор є їх рушій - це базове програмне забезпечення, що визначає їхній фундамент і визначає основні принципи розвитку та функціонування. Походження терміну "ігровий рушій" пов'язане із з'явленням ігор, подібних до першого шутера свого часу - Doom, у середині 90-х років. В цей час з'явилися ігрові рушії, які стали доступні для сторонніх розробників та звичайних користувачів для експериментів з власними ігровими концепціями.

Ігрові рушії стали технічно вдосконаленішими та складнішими з часом, а їхні програмні коди насичені різноманітними аспектами гри. Вони включають в себе жорстко фіксовані дані, такі як ігрова логіка, фізика об'єктів та правила відтворення об'єктів.

Рушій визначає велику кількість складових гри, і їх різноманіття визначає різноманітність віртуальних світів, навіть при використанні одного і того ж рушія. Ігровий рушій включає всі необхідні алгоритми для створення та правильного функціонування гри.

Сучасний ігровий ринок пропонує широкий вибір ігрових рушіїв, які відрізняються підтримуваними мовами програмування, функціональністю і вартістю ліцензій. Однією з популярних альтернатив є Unreal Engine, Unity і Game Maker: Studio, кожен з яких має свої переваги та особливості.

Game Maker: Studio є ігровим рушієм, розробленим та підтримуваним компанією YoYo Games. Початково написаний на Delphi та пізніше переписаний на C#, цей рушій став одним із найпопулярніших серед ігрових рушіїв для створення двовимірних ігор для різних платформ.

Основною мовою програмування для роботи з Game Maker: Studio є GML (Game Maker Language) - спеціально розроблена проста мова програмування, спрямована на полегшення процесу розробки. Це робить рушій доступним для широкого кола розробників, навіть для тих, хто не має великого досвіду в програмуванні.

Game Maker: Studio дозволяє створювати ігри для різних платформ, включаючи популярні операційні системи та ігрові консолі. Його популярність обумовлена не лише зручним інтерфейсом та простотою використання, але й ефективними можливостями для розробки якісних ігрових проектів.

Даний ігровий рушій має кілька версій поширення:

- Trial;
- Desktop;
- Web;
- UWP;
- Mobile.

Кожна з цих версій (крім Trial) відрізняється платформою, для якої буде створена гра. Версія Trial є безкоштовною і надається для тих, хто хоче спочатку випробувати ігровий рушій, має ряд обмежень на кількість використовуваних в грі об'єктів, а також дозволяє компілювати проект тільки для тесту на операційній системі Windows.

Unreal Engine - ігровий рушій, розроблений та підтримуваний компанією Epic Games, здобув свою популярність після випуску першої гри на цій платформі, "Unreal", у 1998 році. З того часу різні версії рушія застосовувалися при створенні більш як сотні ігор та проектів.

Написаний мовою програмування C++, Unreal Engine надає можливість розробки відеоігор для різних операційних систем і платформ, включаючи Microsoft Windows, Linux, Mac OS, Mac OS X, консолі Xbox, Xbox 360, Xbox One, PlayStation 2, PlayStation 3, PlayStation 4, PSP, PS Vita, Wii, Dreamcast, GameCube, а також для пристроїв, що працюють під управлінням iOS і Android.

Рушій підтримує модульну систему компонентів для полегшення портування та підтримує різні системи рендерингу (OpenGL, Direct3D, Pixomatic), аудіовідтворення (DirectSound3D, EAX, OpenAL), голосового відтворення тексту, розпізнавання мови, модулі для мережевої взаємодії та пристроїв введення. Unreal Engine доступний безкоштовно, але компанії-розробнику необхідно віддавати п'ять відсотків від прибутку з продажів гри, якщо їхній щоквартальний прибуток перевищує 3000 доларів.

Unity - це високоефективний інструмент для розробки ігор, що володіє широким функціоналом і користується великою популярністю серед розробників. Завдяки цьому рушію можна створювати як двовимірні, так і тривимірні ігри, а також додатки, що працюють на різних операційних системах, таких як Windows, OS X, Windows Phone, Android, Apple iOS, Linux.

Unity також підтримує ігрові приставки, такі як Wii, PlayStation 3, PlayStation 4, Xbox 360, Xbox One, а також MotionParallax3D дисплеї, призначені для відтворення віртуальних голограм. Можливість створення додатків для запуску в браузері здійснюється за допомогою спеціального модуля Unity Web Player, а також технології WebGL.

Останні версії Unity включають можливість розробки додатків для окулярів віртуальної реальності, що робить його сучасним інструментом для створення ігор на передовому рівні технологій. Редактор Unity приваблює своєю простотою завдяки Drag & Drop інтерфейсу, який дозволяє легко налаштовувати вікна для зручного налагодження гри безпосередньо в середовищі розробки. Рушій підтримує дві сценарні мови: C# та JavaScript (модифікація), що надає розробникам гнучкість у виборі підходящого інструменту для програмування. Обчислення фізики здійснюються за допомогою фізичного рушія PhysX від NVIDIA, що забезпечує високий рівень реалізму фізичної моделі в грі.

Проект в Unity організований у вигляді сцен (рівнів) - окремих файлів, які включають в себе ігрові світи з власним набором об'єктів, сценаріїв та налаштувань. Сцени можуть включати як самі об'єкти (моделі), так і порожні

ігрові об'єкти, які не мають своєї моделі, відомі як "пустушки". Кожен об'єкт містить набір компонентів, з якими взаємодіють скрипти, надаючи розробникам гнучкість у визначенні поведінки об'єктів у грі.

Unity пропонує безкоштовну версію, а також чотири різні збірки - стандартну Unity, Unity iOS Pro, Android Pro і командну ліцензію. Ці версії відрізняються як вартістю, так і функціональністю. Безкоштовна версія має обмеження, проте розробники можуть поширювати свої ігри, якщо річний дохід не перевищує 100 000 доларів. Unity 5, зокрема, видалила багато обмежень для безкоштовної версії, але умова обмеження річного доходу все ще діє.

Unity і Unreal Engine на сьогоднішній день є двома найпопулярнішими ігровими рушіями. Кожен із них має свої переваги, і вибір між ними залежить від конкретних потреб розробника та характеристик гри, яку він планує створити.

Якщо виникає потреба створити мобільну гру, Unity стане оптимальним вибором. Це підтверджується перевагами Unity серед розробників мобільних ігор та широким спектром плагінів, які дозволяють легко інтегрувати нативні можливості мобільних платформ: реклама, внутрішні покупки, аналітика, геймінгові центри тощо.

Якщо ваша мета - розробка двовимірної гри, Unity також виявиться відмінним вибором, оскільки цей рушій надає високий рівень підтримки для створення гравців у світі двовимірних ігор. Хоча Unreal Engine намагається привернути розробників мобільних додатків, обіцяючи ще більше можливостей для двовимірних ігор.

Якщо ваша мета - розробка красивої тривимірної гри, то Unreal Engine 5 буде кращим вибором, оскільки цей рушій виходить попереду Unity у розробці відеографіки на наступному рівні. Однак Unity також є потужним інструментом для створення тривимірних ігор, хоча його графічні можливості не досягають рівня Unreal Engine 5.

Unreal Engine 5 вражає своєю вбудованою системою постобробки, яка дозволяє вам не лише застосовувати bloom-ефект, тонування і згладжування до всієї сцени глобально, але й точно налаштувати ці ефекти для окремих її частин завдяки компоненту PostProcessVolume. У той час, як Unity пропонує стек постобробки, доступний для завантаження з магазину ресурсів рушія, важливо відзначити, що система в Unreal Engine 5 виходить за межі, надаючи більше гнучкості. Тут ефекти можна застосовувати як через стек, так і через скрипти, надаючи розробникам більше контролю над візуальним виглядом гри.

Щодо мов програмування, Unreal Engine 5 використовує потужну мову програмування C++, що відкриває безмежні можливості для розробників, особливо тих, хто звик до більш низькорівневого програмування. З іншого боку, Unity визначає свою основну мову як C#, що, хоча й надає зручний та ефективний інтерфейс для розробки, виявляється менш потужною у порівнянні з C++. Вибір між цими двома рушіями може залежати від

особистих вподобань та конкретних потреб розробника, оскільки кожна мова має свої унікальні переваги.

Також Unreal Engine 5 пропонує рішення для тих, хто уникає входження у високий поріг C++. Blueprint - це редактор візуального скриптування, який технічно дозволяє розробнику уникнути написання будь-якого коду. Це особливо зручно для швидкого створення прототипів та, при цьому, візуальне скриптування в Blueprint забезпечує достатньо потужні можливості для повноцінної розробки відеоігор. У порівнянні з Unity, якому відсутня власна система візуального скриптування, Unreal Engine 5 надає цю можливість, та навіть його відкритий код робить його привабливим для програмістів.

Крім того, для програмістів Unreal Engine 5 має ще один важливий плюс в порівнянні з Unity - це відкритий код.

Обрано Unreal Engine 5 для розробки проекту через його потужний функціонал у створенні тривимірних відеоігор. Планується розробка тривимірної комп'ютерної гри для платформи PC, і саме цей рушій дозволяє ефективно втілити різноманітні концепції у тривимірних іграх. Додатково, використання візуального середовища розробки скриптів (Blueprints) в Unreal Engine 5 обіцяє полегшити перші кроки у розробці комп'ютерних ігор.

1.3. Таблиця найпопулярніших ігор на рушіях Unity та Unreal Engine

Гра	Видавець	Дата випуску	Рушій
Hearthstone: Heroes of Warcraft	Blizzard Entertainment	Windows, macOS: 11 Березня 2014 iOS: 16 Квітня 2014 Android: 15 Грудня 2014	Unity
Cities Skylines	<u>Paradox Interactive</u>	10 Березня 2015	Unity
Cuphead	StudioMDHR Entertainment	29 Вересня 2017	Unity
Pillars of Eternity	<u>Paradox Interactive</u>	8 Серпня 2019	Unity
Pokemon Go	<u>Niantic</u>	6 Липня 2016	Unity

Subnautica	<u>Unknown Worlds</u> <u>Entertainment</u>	Windows, macOS: 14 Грудня 2014 (ранній доступ) 23 Січня 2018 (повна версія) PlayStation 4, Xbox One: 4 Грудня 2018 PS5, Series X/S, Switch: 14 Травня 2021	Unity
Kerbal Space Program	Squad Take-Two Interactive Private Division	27 Квітня 2015	Unity
S.T.A.L.K.E.R. 2: Heart of Chernobyl	GSC Game World	PC, Windows, Xbox Series X/S 2024, в розробці	Unreal Engine 5
Atomic Heart	Mundfish	PC, PlayStation 4, Xbox One 21 Лютий 2023	Unreal Engine 4
Five Nights at Freddy's: Security Breach	Steel Wool Studios	PC 17 Грудень 2021	Unreal Engine 4
ArcheAge II	XL Games	PC, Xbox One 2024, в розробці	Unreal Engine 5
Hogwarts Legacy	Avalanche Software	PC, Windows, PlayStation 4, PlayStation 5, Xbox One, Xbox Series X/S 10 Лютий 2023	Unreal Engine 4
Grand Theft Auto: The Trilogy — The Definitive Edition	Rockstar Games, Grove Street Games	PC, PlayStation 4, Xbox One, Nintendo Switch, Android 11 Листопад 2021	Unreal Engine 4

Little Nightmares II	Tarsier Studios	PC, PlayStation 4, Xbox One, Nintendo Switch, PlayStation 5, Xbox Series X/S 11 Лютий 2021	Unreal Engine 4
-----------------------------	-----------------	---	-----------------

1.4. Аналіз існуючих розробок

Unreal Engine, виходячи з свого багатовікового спадку, став не тільки культовим інструментом для розробників, але й справжнім символом геймдеву, перетворившись у важливу складову історії відеоігор. За понад чверть століття свого існування цей рушій визначав стандарти графіки та геймплею, створюючи кращі в світі гри.

Більше трьохсот ігор, зроблених на різних ітераціях Unreal Engine, свідчать про його неперевершену універсальність та потужність. Четверта версія рушія стала основою обраного проекту, надаючи йому той вражаючий потенціал, який вже здивував гравців та критиків у численних іграх.

Окрім технічних можливостей, Unreal Engine 5 також визначається своєю відкритістю, пропонуючи розробникам величезний простір для творчості та інновацій. У світі геймдеву Unreal Engine не просто інструмент — це визнана та невичерпна джерело ідей та можливостей для новаторських та захоплюючих відеоігор.

Кращими шутерами, які розроблені на базі Unreal Engine, можна вважати:

- Fortnite
- Gears of War 5
- Batman: Arkhem City
- Mass Effect 2

Fortnite - це відома і надзвичайно популярна онлайн-гра в жанрі "битва королівського стилю", розроблена і видана компанією Epic Games. Запущена в 2017 році, гра швидко стала явищем світової відеоігрової культури. Fortnite визначається унікальною комбінацією будівництва, виживання та стрілянини.



Рис. 1.7. Fortnite

У грі гравці висаджуються на острів разом із 99 іншими учасниками і змушені боротися за виживання, побудовувати структури, шукати ресурси та зброю, і знищувати супротивників. Одним із унікальних елементів гри є можливість будувати різні конструкції з матеріалів, зібраних на полі бою, що додає стратегічний елемент до геймплею.

Fortnite вражає вражаючим візуальним стилем, арт-дизайном та постійно оновлюваною геймплейною механікою. Гра також славиться тим, що постійно впроваджує нові сезони, змінює карту та додає унікальні події, що робить її непередбачуваною та захоплюючою для гравців. Fortnite став символом глобального ігрового співтовариства і продовжує збирати мільйони активних гравців у всьому світі.

Gears of War 5, розроблена студією The Coalition та випущена Microsoft, є винятковою грою в жанрі стрілянина з виділеною кампанією та мультиплеєром. Гра продовжує епічну сагу Gears of War, розповідаючи захоплюючу історію про війну людства проти жадливих загарбників, відомих як Locust.



Рис. 1.8. Gears of War 5

Gears of War 5 вражає гравців захоплюючим сюжетом, величезними бойовими механіками та вражаючою графікою. У грі гравці взянуть на себе роль героїні Кейт Діаз, яка розкриває секрети свого минулого, шукаючи відповіді на загадкові явища та загрози, пов'язані з розпадом світу. Мультиплеєрний режим гри пропонує різноманітність варіантів, включаючи сагу "Horde" та режим "Escape". Гравці можуть взаємодіяти в режимі кооперації, об'єднуючи зусилля в битві проти сильних противників або змагатися в інтенсивних онлайн-поєдинках.

Gears of War 5 – це не тільки стрілянина, але і вражаючий науково-фантастичний епос, що підкреслює якість та глибину ігрового досвіду для шанувальників серії та новачків у світі відеоігор.

Batman: Arkham City - захоплююча відеогра в жанрі екшн та пригод, розроблена студією Rocksteady Studios і видана Warner Bros. Interactive Entertainment. Вона є продовженням успішної Batman: Arkham Asylum і переносить гравців в відкритий світ Аркхем-Сіті, перетворений на величезний в'язницю-гетто.



Рис. 1.9. Batman: Arkhem City

У грі ви взямете на себе роль Лицаря Темряви - Бетмена, який має великий арсенал гаджетів та бойових прийомів. Arkham City пропонує захоплюючий сюжет, що розвивається в світі злочину та беззаконня, де Бетмен зіткнеться з численними ворогами, такими як Джокер, Дволикий та Харлі Квінн.

Гравці можуть вільно досліджувати місто, вирішувати головоломки, виконувати завдання та брати участь у захоплюючих бойових сценах. Величезний світ гри наповнений деталями, а реалістичний дизайн персонажів та динамічна система боротьби додають іммерсії та емоційність грі.

Batman: Arkham City - це не просто гра для фанатів коміксів, але й шедевр в ігровій індустрії, який вдало вдало поєднує елементи детективу, екшну та пригод у неперевершеному супергеройському досвіді.

Mass Effect 2 - епічний космічний рольовий екшн від BioWare, який вважається однією з найкращих гри в історії. Гра вийшла в 2010 році і стала другою частиною в захоплюючій трилогії Mass Effect.

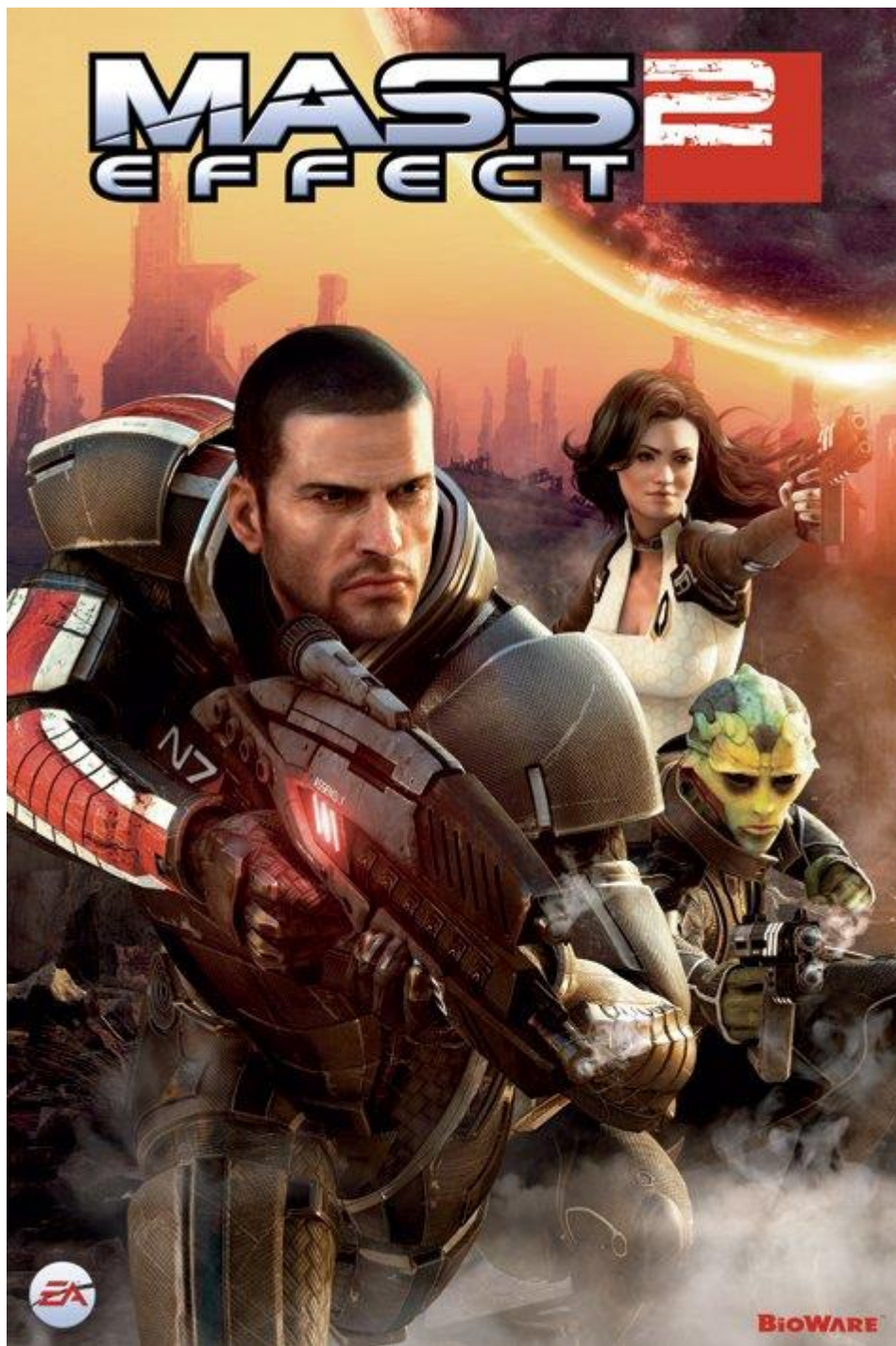


Рис. 1.10. Mass Effect 2

У Mass Effect 2 гравець взяме на себе роль Командора Шепарда, героя-командира Галактичного Флоту, який веде боротьбу проти загрози інопланетних рас, відомих як Реапери. Гра пропонує неперевершений космічний світ, де можна вільно досліджувати планети, збирати команду з унікальних персонажів різних рас і взаємодіяти з різними цивілізаціями.

Масштабна сюжетна лінія, різноманітні завдання та вибір, який впливає на події, роблять Mass Effect 2 надзвичайною. Гра має захоплюючу бойову систему, розвинуті характери та вражаючу атмосферу космічної опера. Важливі рішення гравця можуть впливати на завдання та навіть визначити, хто виживе у кінці гри.

Mass Effect 2 - це витончений та вражаючий шедевр, який вдало поєднує глибокий сюжет, науково-фантастичний світ та захоплюючі геймплейні елементи, забезпечуючи гравцям незабутню подорож у галактичних просторах.

1.5. Вибір конкретної технології для реалізації проекту

Розробка 3D шутерів завдяки Unreal Engine 5 стає винятковою та перевагою для талановитих геймдевелоперів. Перш за все, Unreal Engine 5 вражає своєю потужною графікою та передовими технологіями. Система Nanite Virtualized Geometry дозволяє створювати деталізовані об'єкти без втрати продуктивності, що робить оточення в грі дуже реалістичним та іммерсивним.

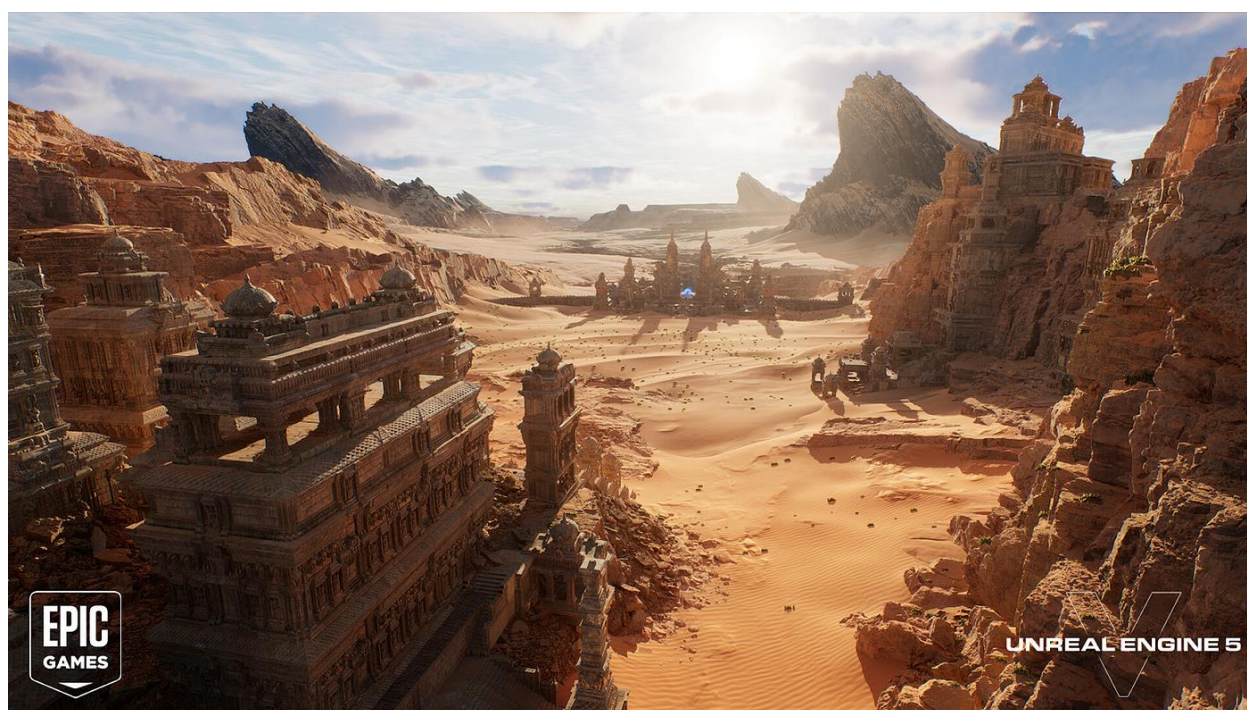


Рис. 1.11. Nanite Virtualized Geometry в Unreal Engine 5

Можливості системи Lumen Global Illumination дозволяють отримувати вражаюче реалістичне освітлення в реальному часі. Це важливо для шутерів, де гравець може перебувати в різних середовищах, від темних закутків до відкритих площадок.



Рис. 1.12. Lumen Global Illumination в Unreal Engine 5

Unreal Engine 5 пропонує зручний інтерфейс для розробки ігор та використовує мову програмування C++, що забезпечує потужність та гнучкість для розробників. Крім того, наявність візуального скриптування через Blueprint полегшує створення прототипів та реалізацію ідей без прив'язки до коду.

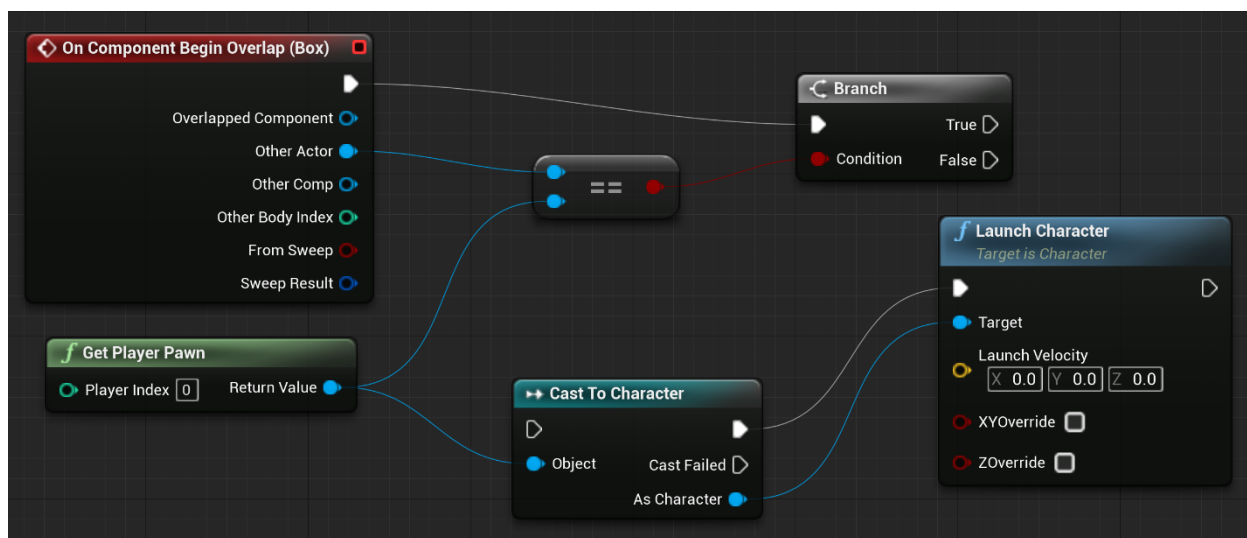


Рис. 1.13. Blueprint в Unreal Engine 5

Unreal Engine 5 також визначається великою спільнотою розробників та великим обсягом документації, що полегшує вивчення та розвиток навичок. Всі ці фактори роблять Unreal Engine 5 найкращим вибором для розробки 3D шутерів, дозволяючи творчо втілювати ігрові концепції у вражаючі та інноваційні відеоігри.

РОЗДІЛ 2. ОПИС ІГРОВОГО РУШІЯ UNREAL ENGINE 5

2.1. Основні факти

Розроблений у 1998 році компанією Epic Games, Unreal Engine є рушієм, спрямованим на першу чергу для жанру шутерів від першої особи, але при цьому не має обмежень і може застосовуватися для широкого спектру ігрових жанрів.

Перша версія рушія була представлена одночасно з грою Unreal, яка була розроблена на його основі. З того часу Unreal Engine пройшов численні зміни та вдосконалення, і завдяки йому було створено безліч ігор різних жанрів і для різних платформ. Остання на сьогоднішній день версія - Unreal Engine 5 - отримує активну підтримку від Epic Games.

Unreal Engine 5, розроблений компанією Epic Games, є передовим рушієм, спрямованим в першу чергу на жанр шутерів від першої особи. З необмеженою гнучкістю використання, рушії призначений для різноманітних типів ігор. Починаючи з першої версії, випущеної разом із грою Unreal, Unreal Engine пройшов через безліч змін і поліпшень. Версія Unreal Engine 5, активно підтримувана Epic Games, зазнала значних удосконалень.

Варто відзначити, що Unreal Engine 5 повністю безкоштовний для некомерційних проектів. У випадку комерційних відеоігор Epic Games збирає лише 5% прибутку в разі перевищення 3000 доларів в квартал. Рушії підтримує всі платформи та операційні системи, включаючи Windows, Linux, OS X, Android, Xbox One, Playstation 4 і Ouya, а також різні графічні API, такі як DirectX 11 і 12, OpenGL, Vulkan і JavaScript/WebGL.

Хоча Unreal Engine 5 може бути трошки складнішим у використанні порівняно з Unity, він надає вражаючі функціональні можливості та неперевершену графіку. Його універсальність робить його ідеальним інструментом для розробки великої різноманітності ігор.

У Unreal Engine 5 відсутні будь-які обмеження, дозволяючи вам втілити в життя будь-яку ідею, чи то складний 3D-пазл, чи великий проект з відкритим світом AAA-класу. Цей рушії користується популярністю серед студентів, інді-розробників та великих творчих груп, а його аудиторія стрімко зростає.

Unreal Engine 5 наділяє своїм користувачам повноцінним арсеналом інструментів для розробки ігор, який був створений самими розробниками ігор, спеціально для розробників ігор. Від невеликих двовимірних мобільних ігор до епічних консольних блокбастерів і імпресивних проектів віртуальної реальності, Unreal Engine 5 надає всі необхідні інструменти для запуску, розвитку та надання унікального характеру вашому проекту.

Однією з ключових переваг рушія є наявність безлічі вбудованих інструментів і функціональностей, не вимагаючи від вас написання або придбання плагінів. Достатньо завантажити редактор, і ви готові розпочати розробку, максимально використовуючи можливості Unreal Engine 5.

Unreal Engine 5 представляє собою повнокомплектну систему, що включає програми, редактори, обширний набір бібліотек, детальну

документацію, безліч уроків з розробки та використання рушія, а також різноманітний контент, який доступний як у платному, так і у безкоштовному варіантах для інтеграції у користувацькі проекти. Цей рушій надає широкі можливості для створення різноманітних ігор.

Аналогічно до інших рушіїв, Unreal Engine 5 включає в себе систему рендерингу, фізичний рушій, обробку звуку, роботу з моделями, текстурами, анімацією, управління ігровими ресурсами, управління пам'яттю, мережевий код та інші основні аспекти. Його особливості включають підтримку руйнівних об'єктів, ефективну обробку зіткнень між об'єктами складної форми, динамічне освітлення, розширений набір ефектів постобробки, можливість редагування гри під час її виконання, систему штучного інтелекту, підтримку складних ландшафтів, систему матеріалів, ієрархію об'єктів у грі та багато іншого.

Завдяки Unreal Engine 5 можна не лише легко та швидко створювати працюючі прототипи ігор, але й розробляти об'ємні відеоігри, які відповідають сучасним стандартам якості. Рушій поширюється у двох форматах: як готовий пакет програм і як вихідний код. У першому випадку користувачеві не потрібно власноруч пройти складний процес компіляції всього рушія, але при цьому обмежується можливість редагування його вихідного коду, якщо це необхідно. У другому випадку користувач отримує повний доступ до вихідного коду рушія на мові C++, створивши свою копію вихідного коду в системі GitHub. Якщо користувач вносить нові функції або удосконалює існуючі, він може пропонувати їх включення в наступну версію рушія.

2.2. Інтерфейс

Основний інтерфейс, відомий як редактор рівнів (level editor), зазвичай використовується для створення світів і рівнів, а також для розміщення активів (неподільних сутностей, що представляють частину ігрового контенту і мають конкретні властивості). Головний інтерфейс редактора включає сім ключових панелей: рядок меню (menu bar), панель modes, панель world outliner, панель details, панель content browser, панель інструментів редактора рівнів і панель viewport.

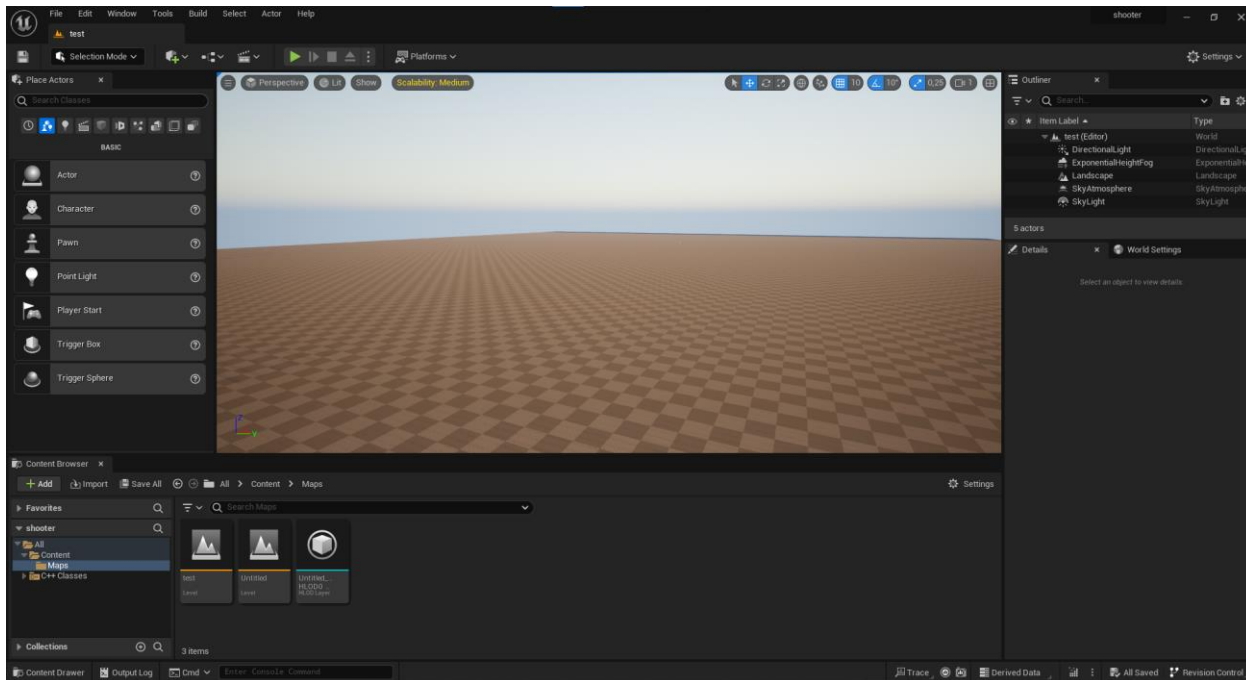


Рис. 2.1. Level Editor

Рядок меню(menu bar) містить основні команди та опції для управління проектом, створення об'єктів та редагування налаштувань.

- Панель modes(Знаходиться одразу над “Place Actors”) надає доступ до різноманітних режимів редактора, де можна вибирати та розміщувати різні об'єкти в грі.
- Панель outliner відображає ієрархію всіх об'єктів у світі, що значно полегшує їхнє вибір та управління.
- Панель details надає детальні властивості вибраного об'єкта чи рівня для редагування та налаштувань.
- Панель content browser дозволяє переглядати, імпортувати та організовувати різні активи та ресурси, які використовуються у грі.
- Панель інструментів редактора рівнів: містить інструменти для роботи з рівнями, їх обрання та редагування.
- Панель viewport: показує перегляд сцени чи рівня, де можна візуально редагувати та попередньо переглядати елементи гри.

Ці панелі разом утворюють потужний інструментарій для розробки ігор у середовищі Unreal Engine.

2.2.1. Рядок меню

Стрічка меню включає звичайні розділи File, Edit, Window і Help, як це зазвичай має місце у більшості сучасних програм. У File розташовані опції для завантаження та збереження проектів та рівнів. Edit включає основні операції копіювання та вставки, а також персоналізацію редактора та налаштування проекту. Window відкриває доступ до вікна перегляду проекту (вьюпорту) та інших панелей. У випадку, якщо ви закрили вікно чи панель, можна знову

відкрити їх, перейшовши до меню Window. Help містить посилання на зовнішні ресурси, такі як онлайн-документація та посібники.

2.2.2. Панель Modes

Панель Modes виводить на екран різноманітні режими редактора рівнів, що дає змогу спеціалізованим інтерфейсам редагування взаємодіяти з відповідними типами акторів і геометрії.

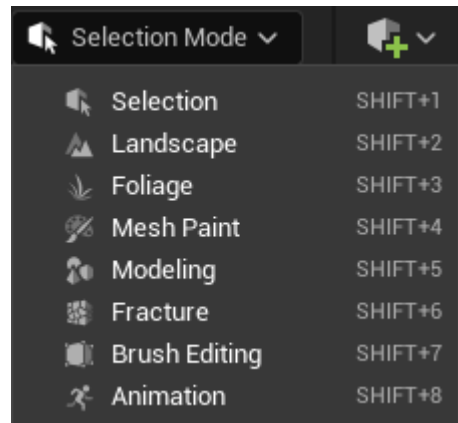


Рис. 2.2. Modes

Панель Modes включає різноманітні інструментальні режими редактора рівнів, які дозволяють модифікувати його функціональність. Тут ви можете обрати спеціалізовані завдання, такі як розташування нових ассетів у світі, формування ландшафтів, створення геометричних форм та об'ємів, генерація рослинності або фарбування мешів (моделей). Загалом редактор має п'ять режимів:

Selection - використовується для виділення та вибору об'єктів, акторів чи елементів у редакторі рівнів. Цей режим дозволяє розробникам швидко та ефективно вибрати елементи для подальших операцій, таких як переміщення, обертання чи зміна параметрів.

Landscape - призначений для редагування та формування ландшафтів у віртуальних світах. Цей режим дозволяє розробникам створювати рельєф, розташовувати текстури, та налаштовувати різноманітні аспекти поверхні землі для досягнення бажаного ландшафтного ефекту в ігровому середовищі.

Foliage - використовується для розміщення і редагування рослинності та інших об'єктів в ігровому світі. Цей режим надає розробникам зручний інтерфейс для додавання, налаштування та масового розміщення різноманітних рослин, що робить ландшафт в іграх більш реалістичним та насиченим.

Mesh Painting - використовується для фарбування текстур та матеріалів на поверхнях мешів (моделей). Це дозволяє розробникам створювати деталізовані та індивідуалізовані ефекти на поверхнях об'єктів, таких як

змінення кольору, насиченості та інших властивостей матеріалів, використовуючи інструменти фарбування.

Modeling - містить безліч інструментів для створення простих і складних 3D-ресурсів для всіх ваших проектів візуалізації. Ці інструменти призначені для оптимізації процесу створення ресурсів та сумісні з сітками, створеними з використанням традиційних робочих процесів, а також більш складними сітками, створеними за допомогою фотограмметрії.

Fracture - це режим редактора рівнів, який містить безліч інструментів, у тому числі ті, що використовуються Chaos Destruction System для створення, руйнування та управління Geometry Collections – типу ресурсу для моделювання руйнування в реальному часі у Unreal Engine.

Brush Editing - це найпростіший інструмент для побудови рівнів у Unreal. Концептуально найкраще розглядати геометричну кисть як заповнення та вирізування обсягів простору на вашому рівні. Раніше геометричні пензлі використовувалися як основний будівельний блок у дизайні рівнів. Однак тепер ця роль була передана статичним сіткам, які набагато ефективніші. Тим не менш, геометричні кисті, як і раніше, можуть бути корисними на ранніх стадіях розробки продукту для швидкого прототипування рівнів і об'єктів, а також для створення рівнів тими, хто не має доступу до інструментів 3D-моделювання. У цьому документі розповідається про використання геометричних пензлів та про те, як їх можна використовувати на ваших рівнях.

Animation - це режим у Unreal Engine, який надає нові інструменти, панелі та можливості редактора, які допомагають у робочих процесах анімації. Використання цього режиму при анімації за допомогою Control Rig пропонує більш орієнтований на анімацію редактор із вкладками, які допомагають вибирати елементи керування, перетворювати дисплеї та запускати інструменти.

2.2.3. Панель World Outliner

Панель Outliner відображає всіх акторів на поточному рівні у вигляді ієрархічного дерева. Ви можете обрати актора, натискавши на його назву в панелі Outliner, після чого його властивості з'являться на панелі Details. Якщо подвійно натиснути на назву, панель Viewport автоматично фокусується на вибраному ассеті.

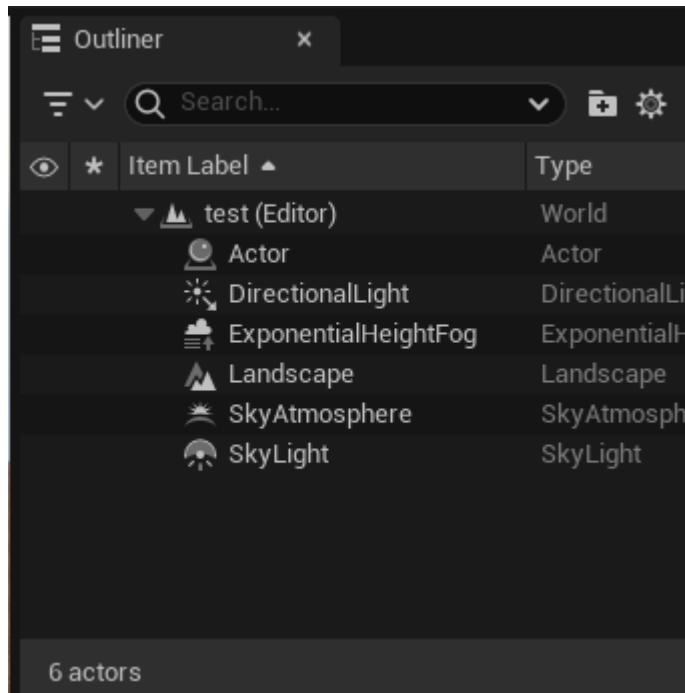


Рис. 2.3. Outliner

2.2.4. Панель Details

Панель Details є однією з найбільш часто використовуваних у Unreal Engine 5. Ця панель присутня практично в кожному підредакторі і відображає всі редаговані властивості обраних акторів на панелі Viewport. Хоча ці властивості залежать від типу обраного актора, існують загальні характеристики, що притаманні більшості акторів. Серед типових властивостей можна виділити назву актора, блоки трансформації для переміщення, повороту і масштабування акторів, а також властивості відображення.

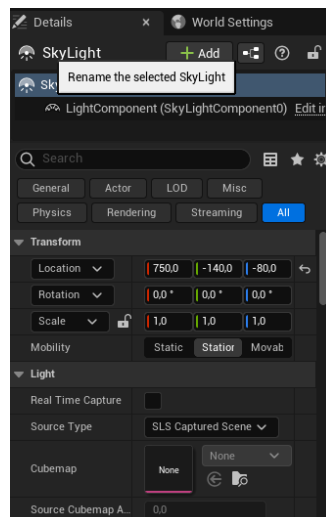


Рис. 2.4. Details

2.2.5. Панель Content Browser

Панель Content Browser представляє собою основну область управління активами в проекті. Цей браузер використовується для вирішення завдань, пов'язаних із вмістом, таких як створення, перегляд, зміна, імпорт і організація. Він також дозволяє керувати папками та виконувати основні операції з активами, такі як перегляд посилань, переміщення, копіювання і перейменування. У Content Browser є рядок пошуку і фільтраційні прапорці для швидкого пошуку активів.

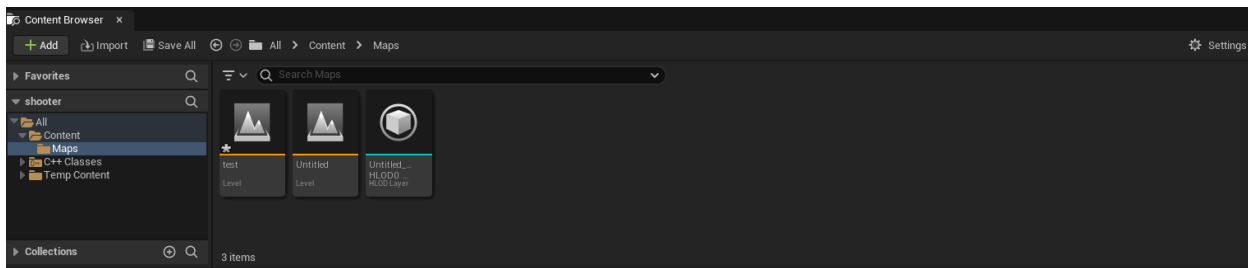


Рис. 2.5. Content Browser

Представте собі панель Content Browser як іграшкову коробку, що містить нескінченні активи. При необхідності ви можете взяти з неї екземпляр (копію) актива і помістити його на рівень. Після розміщення екземпляра на рівень, він стає актором. Початковий екземпляр розміщеного актора є точною копією оригінального актива, що міститься на панелі Content Browser. У лівій частині панелі Content Browser знаходиться панель Source, яка відображає ієрархію папок контенту. Її можна розгорнути або згорнути, натиснувши по іконці в лівому верхньому кутку під зеленою кнопкою Add New. Права сторона панелі Content Browser називається Asset Management і відображає активи в обраній папці на панелі Source.

2.2.6. Панель Viewport

Віюпорти (Viewports) - це панелі, призначені для конструювання та перегляду створених світів. Панель Viewport використовується для переміщення по поточному рівню. Ця панель має безліч різних режимів, розташувань і налаштувань, які допомагають створювати і редагувати рівні, а також керувати ними.

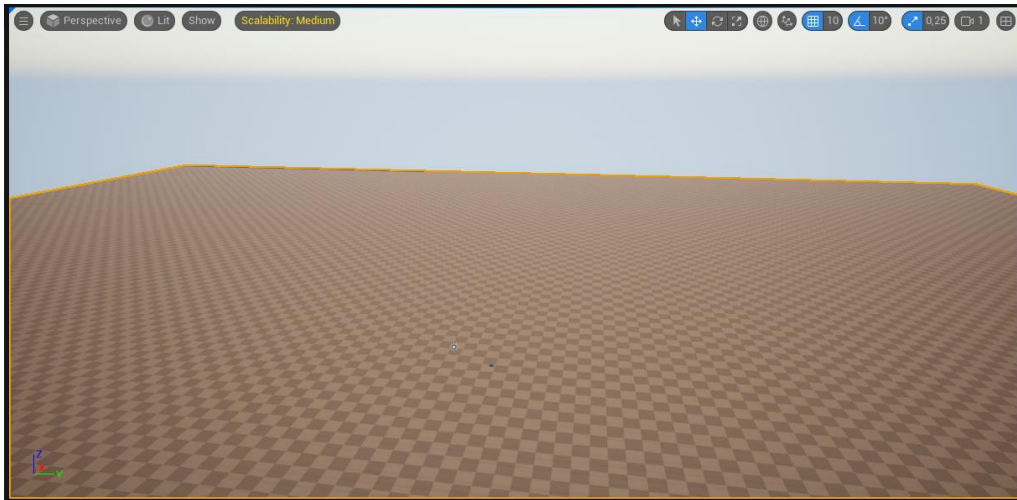


Рис. 2.6. Viewports

2.3. Шаблони та демо-проекти

Відразу після запуску рушія вас вітає вибір шаблону. Хоча ви завжди можете обрати порожній шаблон, часто зручніше вибрати готовий, який вже відповідає обраному жанру комп'ютерної гри. Усього доступні дев'ять шаблонів:

- FirstPerson - шаблон під шутери від першої особи;
- Flying - шаблон під прості симулятори літаків;
- Puzzle - шаблон для логічної гри;
- Side Scroller - шаблон для платформера;
- 2D Side Scroller - шаблон для двовимірного платформера;
- Third Person - шаблон з видом від третьої особи;
- Top Down – шаблон з видом зверху і управлінням мишею;
- Twin Stick - шаблон з аркадою видом зверху;
- Vehicle - шаблон для створення авто-симуляторів;
- Vehicle Advanced - шаблон для створення авто-симуляторів з поліпшеною системою підвіски.

Необхідно відзначити також наявність демонстраційних проектів у лаунчері Epic Games (в розділі Learn / Вивчити). Тут представлено значний обсяг проектів на різноманітні теми, які включають повноцінні прототипи ігор з власною логікою та графікою.

Ці демо-проекти ідеально підходять для початківців, оскільки їх можна вивчати або використовувати як основу для власних проектів. Важливо

відзначити, що всі готові шаблони та демо-проекти можна використовувати в комерційних цілях.

2.4. Редактор

Вперше користуючись редактором Unreal Engine 5, можна відчути легку плутанину – основний редактор насичений різноманітними налаштуваннями, і кожен ассет має власний внутрішній редактор для редагування контенту. Наприклад, окремий редактор статичних мешів дозволяє генерувати низькополігональні моделі, а редактор анімацій - створювати пози або змінювати анімації.

Спочатку може здатися, що редактор має занадто багато функцій, але це скоріше перевага: всі необхідні інструменти знаходяться в одному місці. Хоча для створення дійсно якісного контенту (моделі, звуки, текстурі і т.д.) часто доводиться використовувати зовнішні програми, в Unreal Engine 5 є інструменти для прототипування, що дозволяють зібрати гру без використання інших програм. Наприклад, за допомогою Brush'ей можна створити базовий світ та об'єкти, а потім легко замінити їх на більш якісні або залишити відповідно до стилістики проекту.

2.5. Об'єкти

Під час створення рівнів або сцен у Unreal Engine 5, об'єкти розташовуються на карті, переміщаються для створення оточуючого середовища, і їх параметри та властивості змінюються для досягнення необхідного вигляду та поведінки. Нижче перераховано типи об'єктів, які найчастіше використовуються та зустрічаються під час розробки рівнів і сцен у Unreal Engine 5.

- **StaticMesh Actor (СтатикМеш)** - це простий тип об'єкта, який відображає тривимірну модель на сцені. Назва зовсім не означає, що об'єкт не може переміщатися - під "статичним" розуміється статична геометрія об'єкта. Даний об'єкт може переміщатися або змінюватися будь-якими способами в процесі. Ці об'єкти в основному використовуються як об'єкти навколишнього світу або декорації при створенні сцени.
- **Brush** – це об'єкт (також відомий як BSP-браш), який відображає тривимірний і статичний примітив на сцені. Ці об'єкти можуть бути змінені в режимі редагування геометрії в редакторі рівнів. Браші використовуються головним чином для швидкого макетування середовища і створення рівнів для перевірки ігрового процесу.
- **SkeletalMesh Actor** – це тип об'єктів, який відображає анімований об'єкт із скелетом, геометрія якого може бути деформована, головним чином через використання анімаційних послідовностей, створених і імпортованих з інших програм для створення 3D-анімації. Ці об'єкти часто використовуються для створення персонажів або інших рухомих

істот, а також для реалізації складної механіки та інших аспектів, які вимагають зміни форми або відображення різноманітного руху. Також можуть використовуватися в поєднанні з Matinee для створення анімацій.

- **Player Start** - об'єкт, який визначає, в якому місці буде починати гравець і де спочатку буде з'являтися головний персонаж.
- **Trigger** – це об'єкти, які використовуються для виклику подій при взаємодії інших об'єктів в рівні. Іншими словами, вони викликають події, що відповідають за певні дії в рівні. Всі тригери за замовчуванням подібні і відрізняються лише формою зони впливу тригера на об'єкти, які його активують - коробка, капсула, сфера.
- **Matinee** – це об'єкти, які дозволяють створювати анімацію властивостей об'єктів в часі за допомогою анімаційного редактора Matinee. Вони використовуються для створення як динамічних моментів в ігровому процесі, так і анімованих сцен, що вбудовуються в рівень. Система ґрунтується на спеціалізованих анімаційних доріжках, де можна розміщати ключові кадри для задання значень певних властивостей об'єктів рівня. Редактор Matinee схожий на нелінійні редактори відео, що робить його зрозумілим для професіоналів в області відео.
- **Point Light** – працює дуже подібно звичайної електричної лампочки, поширюючи світло у всіх напрямках. Однак, для економії ресурсів, процес спрощений до освітлення у всіх напрямках тільки з однієї точки простору.
- **Spot Light** – це джерело світла, яке поширює світло з однієї точки у формі конуса. Користувач може вибрати один з двох конусів для освітлення: конус з внутрішнім кутом і конус з зовнішнім кутом. У межах конуса внутрішнього кута світло досягає повної яскравості. Коли ви виходите за межі внутрішнього радіуса, потрапляєте в конус зовнішнього кута, де освітлення пом'якшується, створюючи півтінь або ефект пом'якшення світла навколо світлової плями. Радіус освітлення розділяє ці конуси, створюючи схожість з роботою ліхтаря.
- **Directional Light** – це джерело світла, яке моделює світло, що розсіюється від джерела, розташованого на дуже великій відстані. Такий тип світла генерує паралельні тіні, що ідеально підходять для симуляції сонячного світла. При розміщенні вибирається один з трьох видів рухливості.
- **Particle Emitter** – це об'єкт, який використовується для створення різноманітних ефектів, таких як дим, вогонь, іскри тощо, шляхом генерації частинок у формі спрайтів (обличчям до камери) або об'єктів. Поведінка цих частинок визначається та контролюється за допомогою спеціальної системи, відомої як Particle System, яку можна створювати в Content Browser та налаштовувати в редакторі Cascade.
- **Ambient Sound** – використовується для відтворення звуків в просторі. Ці звуки можуть бути циклічними чи одноразовими, мати

просторову акустичну модель, а також ефекти посилення та затухання. Усе це налаштовується в звуковому сигналі і не вбудовано в сам об'єкт звуку.

- Decal - це спеціальний об'єкт, який накладає матеріал на іншу поверхню без необхідності використання додаткової геометрії. Це може використовуватися, наприклад, для додавання різних слоїв, тріщин, вм'ятин або бруду на геометрію, зробивши поверхню більш реалістичною і менше повторюваною.
- Camera Actor - це об'єкт, який використовується для відображення сцени на екрані. Він може існувати самостійно і використовуватися разом з інструментом Matinee, а також може бути включений як компонент всередині інших об'єктів, наприклад, всередині персонажа.
- Volume - це спеціальна область, в межах якої застосовуються певні правила. Ці правила можуть впливати як на геймплей (наприклад, завдання шкоди персонажу або зміна його типу руху), так і на редагування (визначення областей для обчислення світла або навігації для штучного інтелекту).
- Target Point - це об'єкт, який використовується для маркування конкретної точки в просторі для подальшого використання в коді або в блупринтах. Наприклад, дизайнер рівня може розмістити таку точку, яка визначає місце, куди персонаж повертається після смерті, і програміст може використовувати цей об'єкт для отримання координат відновлення персонажа.

2.6. C++ чи Blueprints

При розробці комп'ютерних ігор важливою частиною є написання логіки гри. У Unreal Engine 5 для цього можна використовувати мову програмування C++, а також інструмент візуального програмування, відомий як Blueprints. Blueprints дозволяють створювати логіку гри за допомогою візуальних блок-схем із вузлів, що спрощує розробку і дозволяє реалізовувати функціональність без прямого програмування на мові C++.

Використання Blueprints може бути великим плюсом для розробників, особливо для тих, хто не є програмістами за освітою. Blueprints надають візуальний інтерфейс для створення логіки гри, що робить процес доступнішим для широкого кола творчих фахівців. Вони дозволяють уникнути деяких типових помилок, що можуть виникнути при програмуванні на мові C++, і забезпечують розробників більшою безпекою від "вильотів" гри, оскільки помилки в логах дають можливість вчасно виявляти і виправляти проблеми.

Але незважаючи на низький поріг входу і наочність, блупринти - повноцінний інструмент об'єктно-орієнтованого програмування (ООП).

Blueprints є потужним інструментом для візуального програмування та створення логіки гри, але вони можуть мати свої обмеження, особливо в області взаємодії з зовнішніми ресурсами та сторонніми бібліотеками.

Програмування на C++ надає більше гнучкості і можливостей, особливо коли йдеться про більш складні аспекти гри, такі як робота з мережею, робота з файловою системою або використання сторонніх бібліотек. Водночас, це вимагає глибшого розуміння програмування та архітектури гри.

Кожен підхід має свої сильні сторони, і в реальних проектах розробники часто комбінують використання Blueprints та C++ для оптимального використання переваг кожного з них.

Ваш вибір використовувати Blueprints для розробки логіки гри виправданий, особливо якщо ви новачок у розробці ігор або не маєте значного досвіду у програмуванні на C++. Blueprints дозволяють легко визначати логіку гри за допомогою візуального інтерфейсу, що полегшує вивчення та розвиток вашого проекту.

Щодо документації, важливою перевагою використання Unreal Engine є активна спільнота розробників. Багато питань та проблем можуть бути вирішені шляхом пошуку в інтернеті або звертання до форумів та спільноти. Однак важливо мати на увазі, що розвиток компетентностей в розробці гри - це поетапний процес, і навіть початковий досвід з Blueprints може вам добре послужити у майбутньому.

Якщо ви вирішили розвивати свій проект та групу великою мірою на Blueprints, це не тільки прискорить розробку, але й забезпечить легкість відлагодження та розширення проекту. Удачі вам у розробці вашої гри!

2.7. Blueprints

Blueprints у Unreal Engine є важливими елементами для розробки гри. Це візуальний інтерфейс, що дозволяє створювати різні типи об'єктів та розробляти ігрову логіку, не використовуючи коду. Основна перевага полягає в тому, що вони дозволяють розробникам, які не мають глибоких знань програмування, ефективно та швидко реалізовувати ідеї та функціональність гри.

Кожен Blueprint може включати в себе не тільки графічні аспекти, але й логічні складові. Це робить їх потужним інструментом для розробки різноманітних ігрових систем, включаючи рух персонажів, системи штучного інтелекту, обробку подій, анімацію, ефекти та багато іншого.

Blueprint дозволяють розробникам створювати та експериментувати з ідеями швидше і ефективніше, що робить їх важливим інструментом для тих, хто тільки починає свій шлях у галузі розробки ігор або хоче прискорити процес розробки.

Blueprints надають вбудований в редактор інтерфейс для візуального програмування, що дозволяє розробникам будувати логіку гри за допомогою

блок-схем та графічних елементів. Це зроблено шляхом з'єднання різних блоків, що представляють події, функції, змінні та інші логічні елементи.

Цей візуальний підхід до програмування полегшує розуміння та модифікацію логіки гри, особливо для тих, хто не має глибоких знань у програмуванні. Розробники можуть працювати з blueprints, не пишучи жодного рядка коду. Вони можуть визначати події, реагування на них, і створювати складні логічні структури, просто перетягуючи та з'єднуючи блоки.

Це дозволяє прискорити розробку, полегшує відладку та робить процес створення гри більш доступним для широкого кола розробників.

Два найбільш використовуваних типи блупринтів:

- Level Blueprint;
- Class Blueprint.

Кожен рівень включає в себе свій власний Level Blueprint. Level Blueprint використовується для маніпулювання об'єктами на сцені під час гри, а також для управління Matinee, стрімінгом рівнів, чекпоінтами та іншими системами, що відносяться до даного рівня. Level Blueprint може взаємодіяти із Class Blueprint, що також присутні на сцені.

Class Blueprint дозволяють створювати складні об'єкти для подальшого розміщення на сцені, такі як, наприклад, двері, що відкриваються, ящики з предметами, кнопки і інше. Наприклад, кнопка, розташована на підлозі, і двері – це різні блупринти, кожен з яких включає в себе конкретний скрипт для взаємодії з гравцем, відтворення анімації та звуків, відкриття дверей і інші дії.

В даному випадку, натискання кнопки спричиняє активацію події всередині блупринта дверей, що призводить до їх відкриття. Таким самим чином можна взаємодіяти з будь-якими іншими блупринтами, а також викликати події не лише для гравця, але й для інших об'єктів або Level Blueprint. Class Blueprint можуть бути абсолютно автономними, і для їхньої роботи не потрібно зовнішнього впливу; вони можуть виконувати різноманітні дії самостійно.

Ігрові персонажі також представлені у вигляді Класових Блупринтів, які використовуються для створення всіх необхідних елементів та логіки для майбутнього персонажа. Це дозволяє налаштовувати параметри камери, встановлювати управління персонажем, включаючи мишу чи навіть сенсорні екрани, а також розробляти різноманітні можливості, які доступні персонажу.

При створенні Блупринта Персонажа (Character Blueprint) вам вже буде доступний набір властивостей, які налаштовують пересування, стрибок, плавання і падіння. Вам лише потрібно буде додати управління та визначити, як саме ваш персонаж буде взаємодіяти з оточенням.

Створення інтерфейсу виконується за допомогою блупринта, схожого на класовий, але він прив'язується до геймплею, не вимагаючи розміщення на рівні. За допомогою блупринта інтерфейсу можна налаштувати так, щоб він зчитував змінні з інших блупринтів і відображав різні елементи, такі як бали,

показник здоров'я персонажа, кількість наявних патронів і т.д. Через блупринт HUD (інтерфейсу) також можна створювати кнопки для різноманітних меню.

Під час створення будь-якої логіки з використанням блупринтів ви взаємодієте з Редактором Блупринтів. Існують різні типи редакторів, які використовуються для редагування конкретних блупринтів. Основний функціонал редактора залишається схожим (графіки, змінні і т.д.), але деякі блупринти, зокрема Level Blueprint, можуть відрізнятися у відсутності власних властивостей або компонентів.

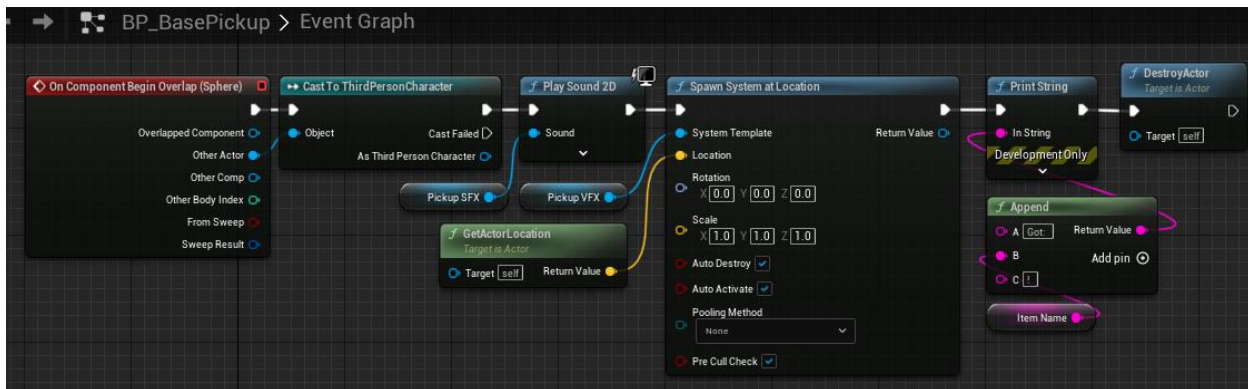


Рис. 2.7. Blueprint

2.8. Меши, матеріали та ефекти

Припустимо, що у вас вже є прототип ігрового світу і геймплей. Тепер вам потрібно імпортувати моделі та налаштувати матеріали для них, щоб досягти відповідного вигляду. Один із способів — скористатися готовими ассетами.

Рушій підтримує формати 3D-моделей .fbx і .obj, а також всі популярні формати текстур. Процес імпорту є достатньо простим — ви можете просто перемістити файли у папку з проектом, і імпорт буде виконаний автоматично.

Також необхідно правильно налаштувати матеріали, що означає роботу з шейдерами. Зазвичай шейдери розробляються за допомогою коду в інших програмах, але в Unreal Engine 5 є візуальний інструментарій, який дозволяє вам визначати інструкції для шейдерів за допомогою вузлів.

Не забуваймо і про візуальні ефекти. В даний момент в Unreal Engine 4 є два редактори для їх створення: модульний і нодовий. У першому редакторі система частинок і ефектів збирається з модулів і налаштувань. У другому можна створювати ефекти, використовуючи ноди редактора, який дуже схожий на редактор матеріалів. Там можна гнучко налаштовувати ефекти для досягнення потрібного результату.

2.9. Інтерфейс користувача (UI)

У Unreal Engine 5 доступний спеціальний редактор віджетів - UMG. Як і весь рішення, він є інтуїтивно зрозумілим та потужним. Можна розміщувати елементи з фіксованими положеннями, створювати блочну верстку,

вбудовувати віджети всередині інших, і так далі. Надається широкий спектр налаштувань, що дозволяє переробляти або анімувати практично будь-що.

UUMG надає можливість створювати меню, HUD та інші елементи інтерфейсу для користувача, і всі ці компоненти можна налаштовувати через блупринти.

2.10. Звук

Робота зі звуком в грі також не є складною. Достатньо імпортувати його у форматі Wav 16 та відтворювати в потрібному місці за допомогою блупринтів (або коду на C++). Альтернативно, можна використовувати компоненти звуку і вмикати/вимикати їх у потрібні моменти. Також у Unreal Engine 5 можна міксувати та створювати складні звукові ефекти, використовуючи ноди редактора.

2.11. Збірка

Unreal Engine 5 дозволяє легко компілювати проект для різних платформ, включаючи ПК, консолі, мобільні пристрої та навіть HTML5. Для компіляції для ПК достатньо всього лише одного кліка, але для інших платформ може бути необхідно встановлення додаткового SDK, інструкції щодо якого можна знайти у документації.

РОЗДІЛ 3. РОЗРОБКА КОМП'ЮТЕРНОЇ ГРИ У ЖАНРІ ШУТЕР

Unreal Engine 5 встановлюється та запускається за допомогою Epic Games Launcher. Перед тим, як розпочати розробку шутера, ми завантажуємо Animation Starter Pack з магазину в лаунчері. Цей спеціальний пакет додає до гри різноманітну анімацію, таку як ходьба, стрибки, стрільба і багато іншого. Він є абсолютно безкоштовним і розроблений самою компанією, яка створила рушій Unreal Engine.

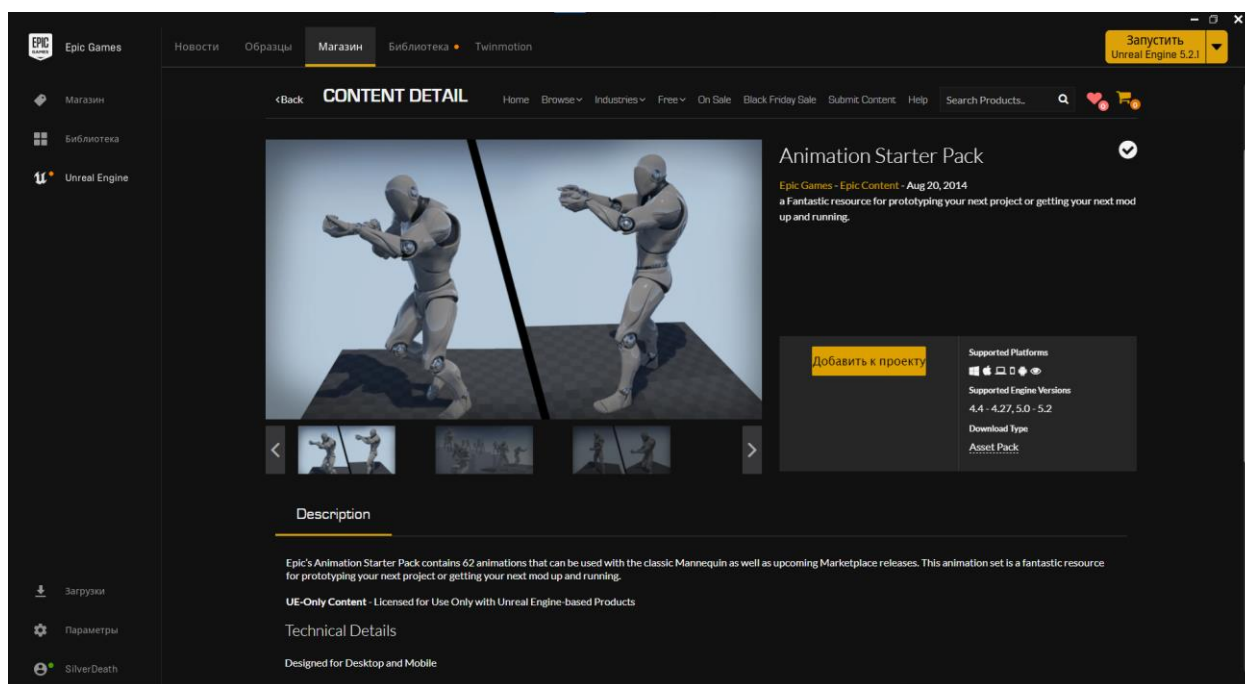


Рис. 3.1. Animation Starter Pack

3.1. Створення проекту

Спочатку потрібно обрати тип проекту. Оскільки ми будемо розробляти комп'ютерну гру, обираємо тип Games.

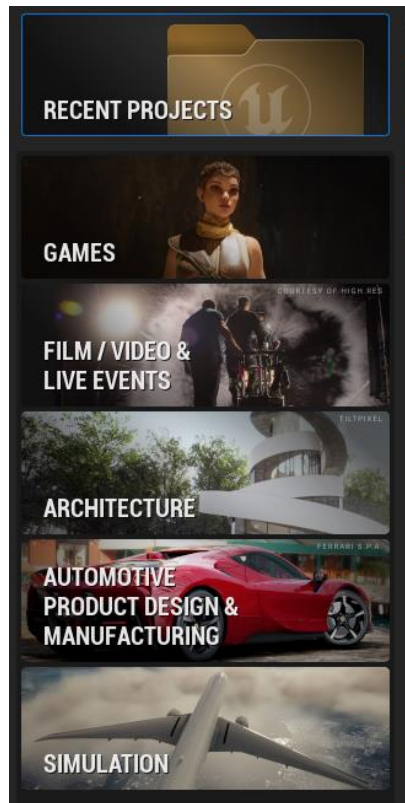


Рис. 3.2. Категорії проекту

У наступному вікні відразу ж пропонується на вибір кілька заготовлених шаблонів. Вибираємо шаблон Third Person. В такому випадку ми відразу отримаємо готового ігрового персонажа з видом від третьої особи.

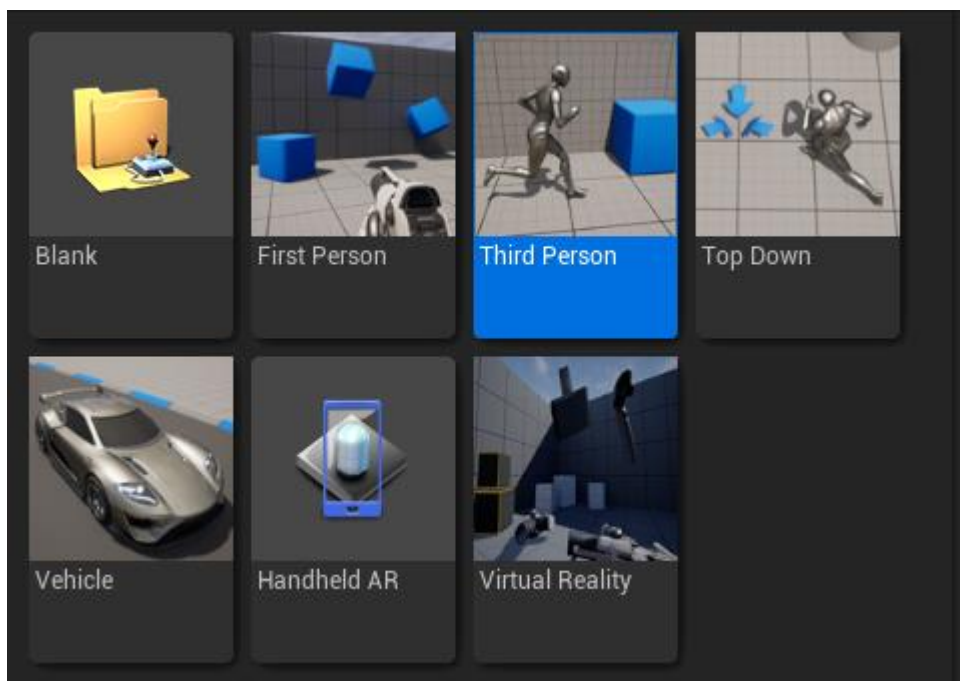


Рис. 3.3. Вибір шаблону

У наступному вікні слід визначити основні параметри проекту. Ми вказуємо, що для розробки будуть використовуватися Блупринти, а проект розроблятиметься спрямовано на персональні комп'ютери. Встановлюємо

максимальну якість графіки, проте вимикаємо трасування променів. Також вказуємо, що нам необхідний Starter Content – стандартний пакет, який містить прості розміщені меші з основними матеріалами та текстурами. Після цього визначаємо розташування та назву проекту.

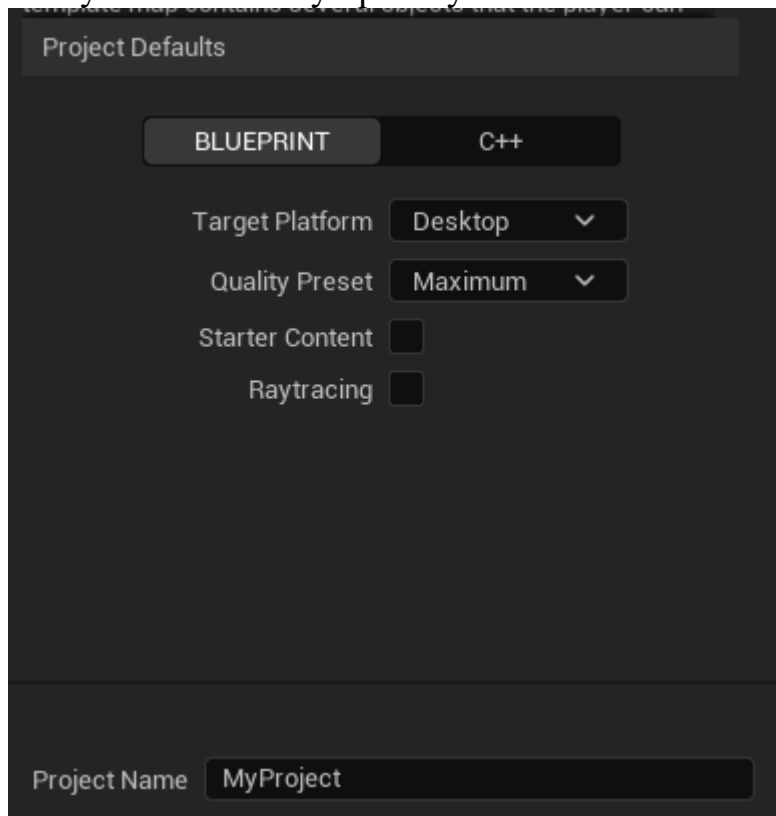


Рис 3.4. Налаштування проекту

Після створення проекту автоматично відкривається редактор створеного рівня, де ви можете помітити деякі вже наявні елементи (актори):

- Directional Light - це штучне джерело світла, яке симулює ефект світла, що йде з великої відстані;
- SphereReflectionCapture - реєструє вигляд оточення з конкретної позиції. Після цього це захоплення використовується для відображення на поверхнях, які перебувають у зоні впливу цього об'єкта;
- AtmosphericFog - забезпечує ефект розсіювання світла через атмосферу планети. Це може призвести до більш реалістичного вигляду зовнішніх рівнів;
- SkyLight - захоплює віддалені області рівня та використовує цю інформацію для освітлення сцени. Це означає, що зовнішній вигляд неба та його освітлення або відображення будуть відповідати, навіть якщо небо виходить за межі атмосфери, або якщо хмари шаркують поверхню небесного склепіння або вдалеку високих гір;
- ThirdPersonCharacter - гравець у грі з анімаціями, які вже налаштовані та готові до управління;
- FollowCamera - відображає перспективу гравця, тобто те, як гравець сприймає оточуючий світ;

- **StaticMeshComponent** - використовується для створення інстансів **UStaticMesh**, де **StaticMesh** є частиною геометрії, складеної з статичного набору багатокутників. Цей компонент є основним елементом для формування геометрії світу рівнів в **Unreal Engine 5**;
- **LightmassImportanceVolume** - контролює зону, в яку **Lightmass** (створює світлові карти із складними світловими взаємодіями, такими як затінення областей і дифузна інтеррефлексія) випромінює фотони. Це дозволяє концентрувати його у визначеній області, що потребує деталізованого непрямого освітлення;
- **PostProcessVolume** - це особливий вид обсягу в **Unreal Engine 5**. Кожен обсяг постобробки представляє собою один із типів шарів, які можуть бути створені через ігровий код (наприклад, ефект попадання), застосовані до інтерфейсу користувача (наприклад, меню паузи), пов'язані з камерою (наприклад, віньєтка) або використовувані в **Matinee** (**matinee**; ефект старовини). Кожен окремий шар має свою вагу, що дозволяє ефективно керувати ними та змішувати для досягнення бажаних ефектів.

На поточному етапі на рівні розташовано простір із підлогою, оточений стінами, і ігровий персонаж. Крім того, є декілька об'єктів, на які гравець може стрибати, переміщатися по них та зістрибувати з них.

Останнім етапом перед початком розробки є включення анімаційного пакету **Animation Starter Pack** у проект.

3.2. Налаштування камери

Оскільки було вирішено розробляти гру з видом від першої особи, але наразі у нашій грі використовується вид від третьої особи, необхідно внести корективи. Це можна виправити, перемістивши камеру, прикріплену до персонажа, так, щоб гравець сприймав світ крізь очі персонажа, а не ззаду, як це реалізовано в шаблоні. Для цього переходимо на панель **Content Browser** і відкриваємо папку "**Blueprints**" (рис. 3.5), де знаходяться всі об'єкти з блупринтами у нашій грі (рис. 3.6). Потім відкриваємо **ThirdPersonCharacter** і переходимо до вкладки **Viewport** у новому вікні.

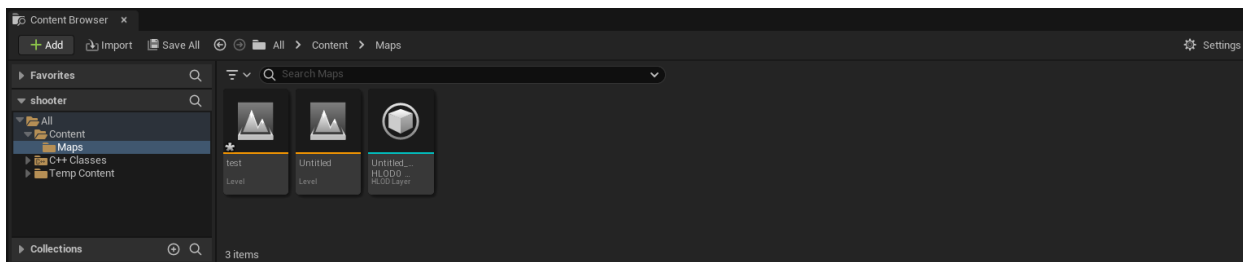


Рис 3.5. Content Browser

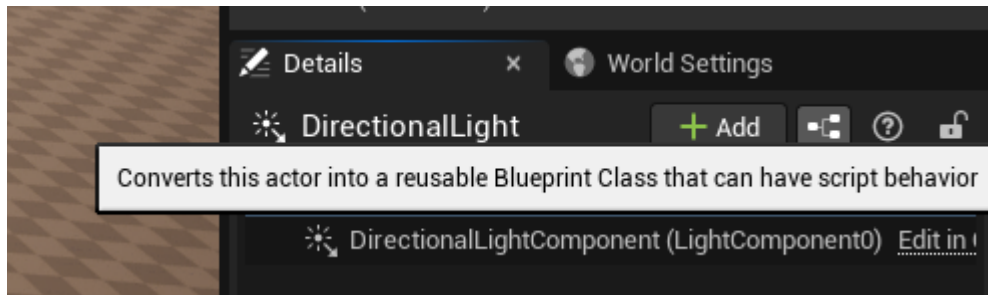


Рис 3.6. Папка Blueprints

Спочатку, переміщаємо камеру у відповідне положення, а потім призначаємо її дочірнім об'єктом персонажа (рис. 3.7). Оскільки камера є статичною, а персонаж має анімацію, необхідно забезпечити, щоб камера рухалася разом з рухами голови персонажа. Для цього в параметрі Parents Socket камери обираємо "head", щоб камера відстежувала анімаційні рухи голови персонажа та повторювала їх. Після цього камера автоматично обертається та зміщується (рис. 3.8), тому потрібно повернути її на необхідне місце.



Рис. 3.7. Переміщення камери



Рис. 3.8. Автоматичне зміщення камери

За замовчуванням, компоненти камери не реагують на обертання контролера. Щоб виправити це, необхідно вибрати камеру у редакторі, а потім на панелі параметрів активувати опцію "Use Pawn Control Rotation" (рис. 3.9). Далі, у редакторі ThirdPersonCharacter, також на панелі параметрів, потрібно включити опцію "Use Rotation Controller Yaw", щоб при обертанні камери тіло персонажа також повороталося (рис. 3.10).

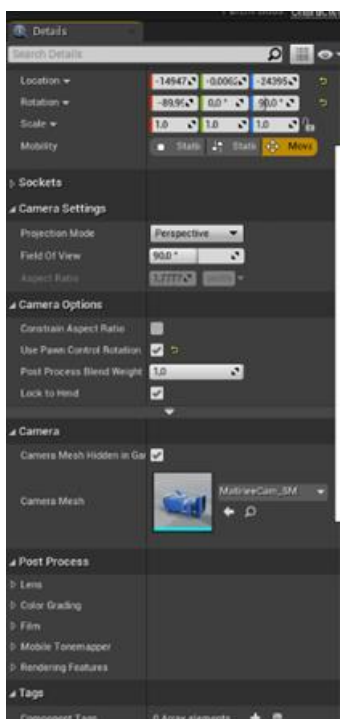


Рис. 3.9. Use Pawn Control Rotation



Рис. 3.10. Use Rotation Controller Yaw

Тестуємо, натиснувши кнопку Play на панелі інструментів. Можна побачити, що тепер в грі вид від першої особи.



Рис. 3.11. Вид від першої особи

3.3. Шкала життя і обладунків гравця

У грі буде дві шкали: шкала життя і шкала обладунків. Обидві шкали будуть зменшуватися при отриманні урону гравцем. Спочатку внаслідок урону буде зменшуватися шкала обладунків, яку можна часом відновлювати. Однак якщо вона повністю вичерпається, то почне зменшуватися шкала життя гравця, яка не має властивості відновлення. У випадку, якщо шкала життя знизиться до нуля відсотків, гравець втратить життя.

Відкриваємо блупринт ThirdPersonCharacter (гравця) і в новому вікні переходимо до вкладки Event Graph. Спочатку треба додати дві змінні типу Float: Armor (броня) та Health (життя) (рис. 3.12).

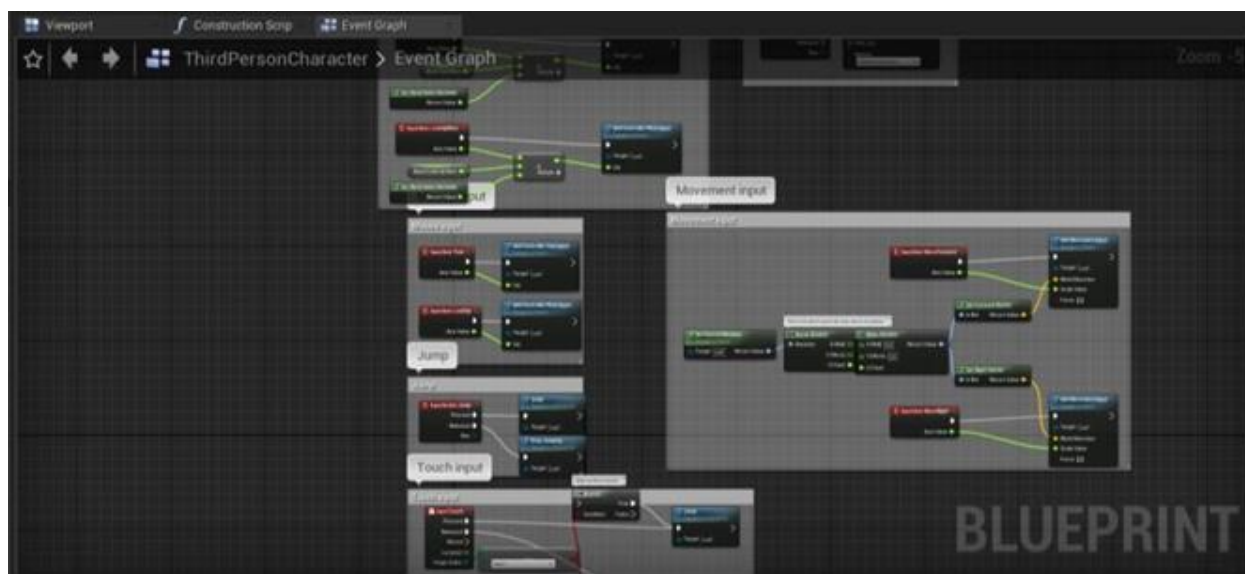


Рис. 3.12. Редактор ThirdPersonCharacter

Далі можна провести компіляцію блупринта, натиснувши кнопку "Compile" на панелі інструментів, а також встановити початкові значення для змінних. Давайте припустимо, що для обох змінних початковим значенням буде 1.0. Після цього виконуємо знову компіляцію та зберігаємо цей блупринт.

Для створення інтерфейсу гравця зі шкалами життя створюємо клас типу Widget Blueprint, який ми назвемо HUD. У редакторі інтерфейсу додаємо два елементи Progress Bar та налаштовуємо їх розмір, положення та колір. Перший Progress Bar буде відображати броню і матиме жовтий колір, а другий буде представляти шкалу життя з червоним кольором (рис. 3.13).

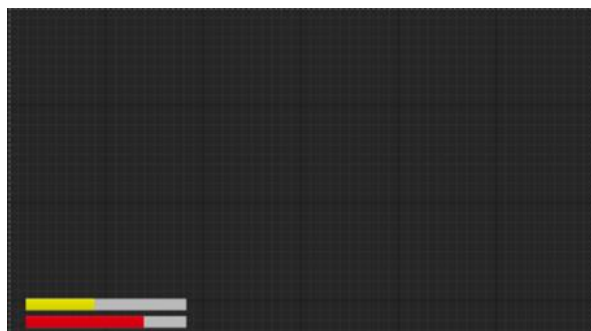


Рис. 3.13. HUD

Тепер треба прив'язати інтерфейс до змінних. Для цього використовується Blueprints – візуальне програмування. Було створено по одній спеціальній функції для кожної шкали, які звертаються до гравця, як до об'єкту, дізнаються значення потрібної змінної та повертають його (рис. 3.14, 3.15).



Рис. 3.14. Функція для шкали броні

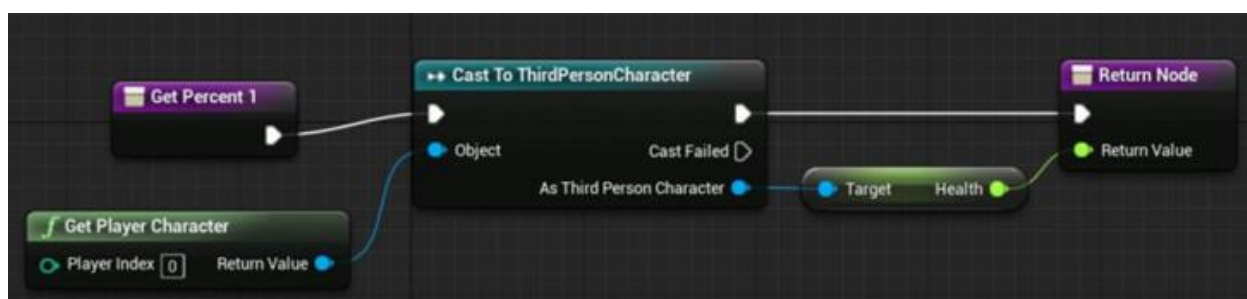


Рис. 3.15. Функція для шкали життя

Тепер необхідно налаштувати відображення створених шкал на екрані під час гри. Для цього розширюємо блупринт персонажа (рис. 3.16).



Рис. 3.16. Скрипт для відображення шкал на екрані

Далі перевіряємо, що все працює (рис. 3.17).

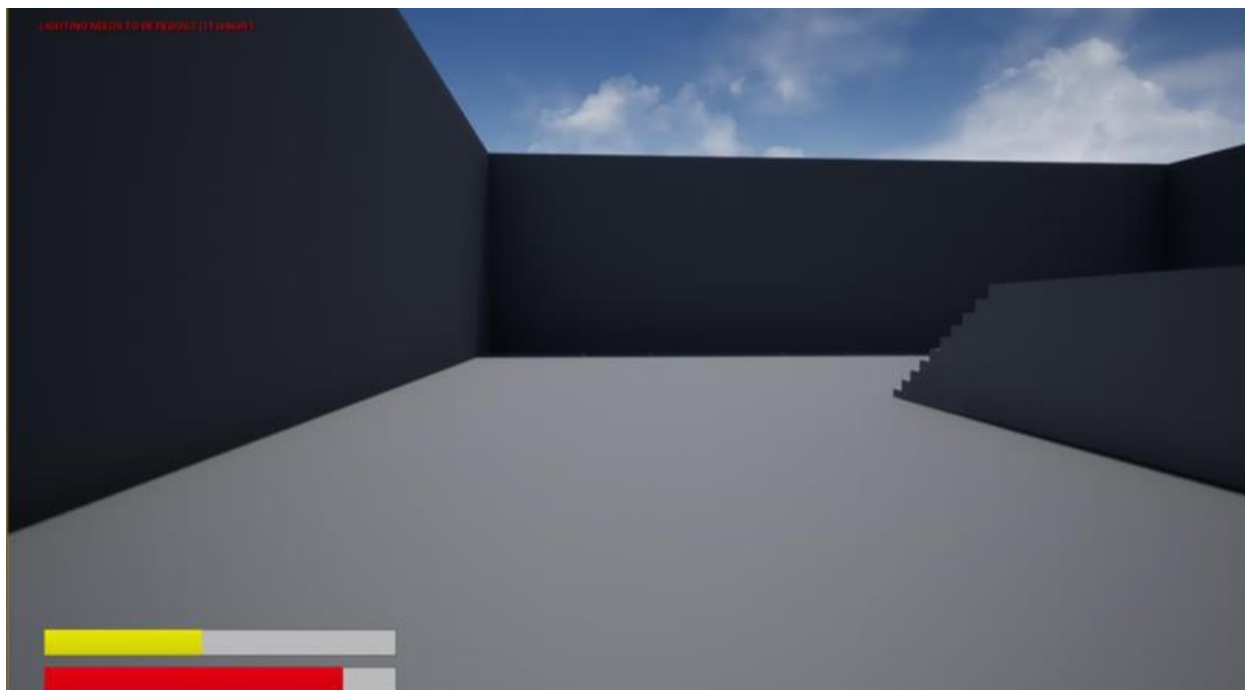


Рис. 3.17. Відображення шкал на екрані

3.4. Відновлення обладунків

Відновлення обладунків реалізується шляхом розширення блупринта персонажа, а саме створення графіків (скриптів). Було вирішено здійснити так, щоб якщо рівень обладунків менше 100%, то кожен секунду вони будуть відновлюватись на 1% (див. рис. 3.18).

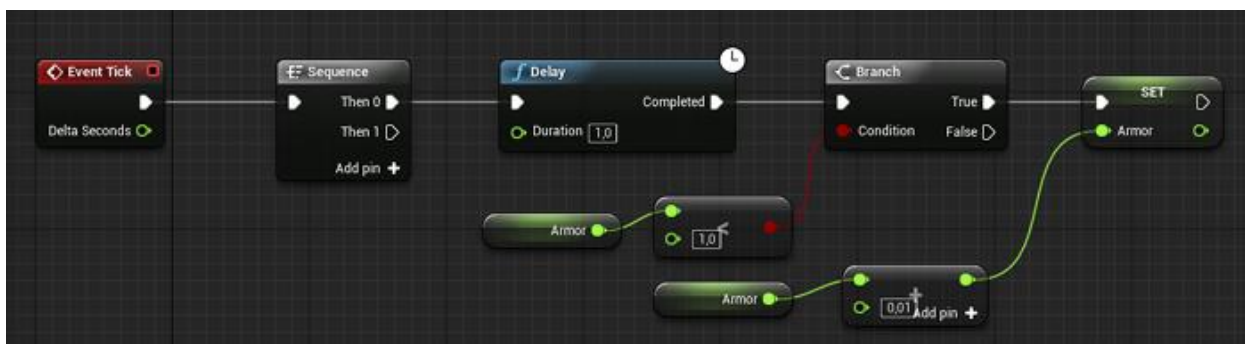


Рис. 3.18. Скрипт відновлення броні

3.5. Нанесення урону

Тепер для блупринта персонажа гравця треба створити функцію, яку можна буде визивати. Було вирішено розробити функцію, яка віднімає 5% від значення обладунків, а якщо обладунки мають нульове значення, то відніматися буде 5% від рівня життя персонажа (див. рис. 3.19).

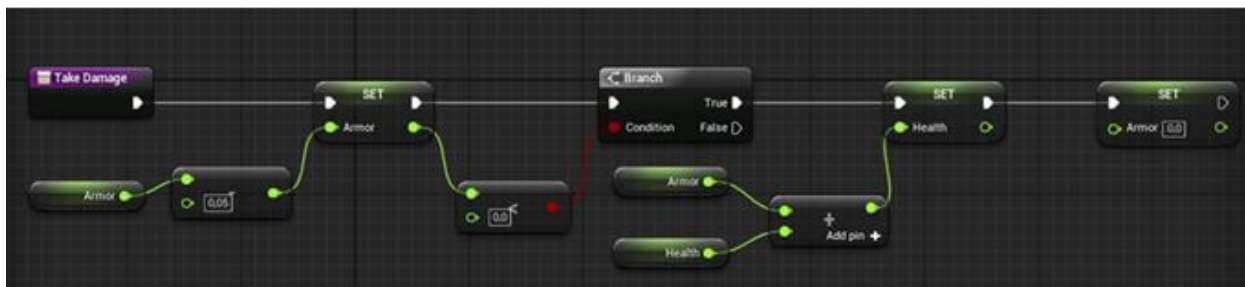


Рис. 3.19. Функція нанесення урону

3.6. Додавання повноцінного 3D персонажа з анімацією

В даний момент наш персонаж обладнаний анімаціями для бігу, стрибків та поворотів. Для простого шутера цього може виявитися достатнім, але перед тим, як продовжити, потрібно адаптувати анімації так, щоб вони виглядали реалістично, коли персонаж тримає в руках автомат. Це завдання досить трудомістке і вимагає високого рівня професіоналізму. Тому виникла необхідність завантажити готового персонажа із відповідними анімаціями для шутера, який доступний відкритою ліцензією в Інтернеті. Проте це також потребує налагодження.

Спочатку формуємо асет Animation Blueprint. Після відкриття цього блупринта ми маємо доступ до редактора (рис. 3.20), де можна налаштовувати, які анімації будуть активними та в який момент. На один час може бути активною лише одна анімація, проте можна створити State Machine (скінчений автомат), щоб налаштувати переходи від однієї анімації до іншої в залежності від ігрових ситуацій.

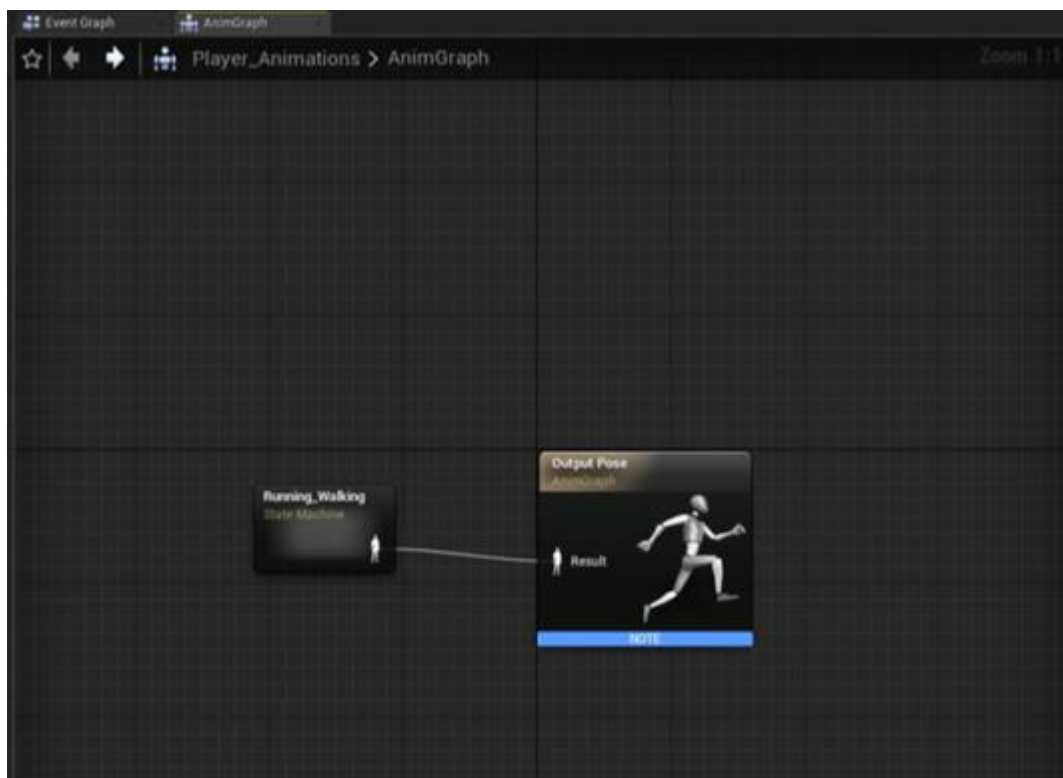


Рис. 3.20. Редактор Animation Blueprint

Маємо два стани: Idle та Run_Walk (рис. 3.21).

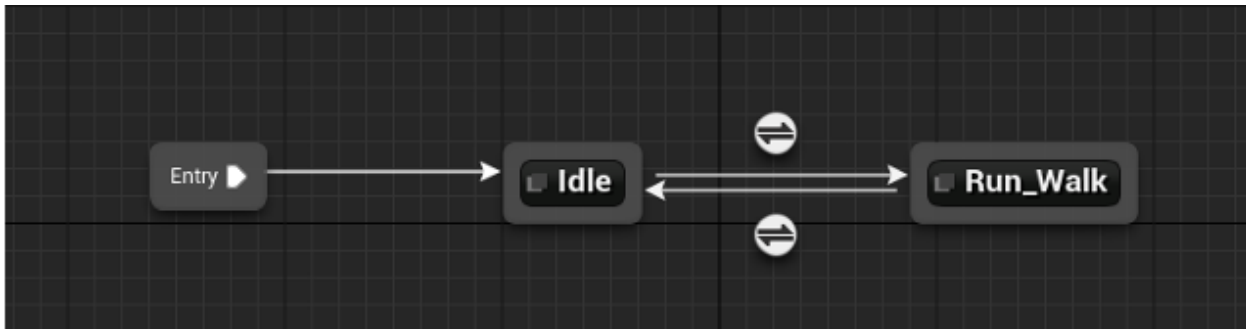


Рис. 3.21. State Machine

Під час стану "Idle" буде відтворюватися анімація людини, що стоїть на місці (рис. 3.22).

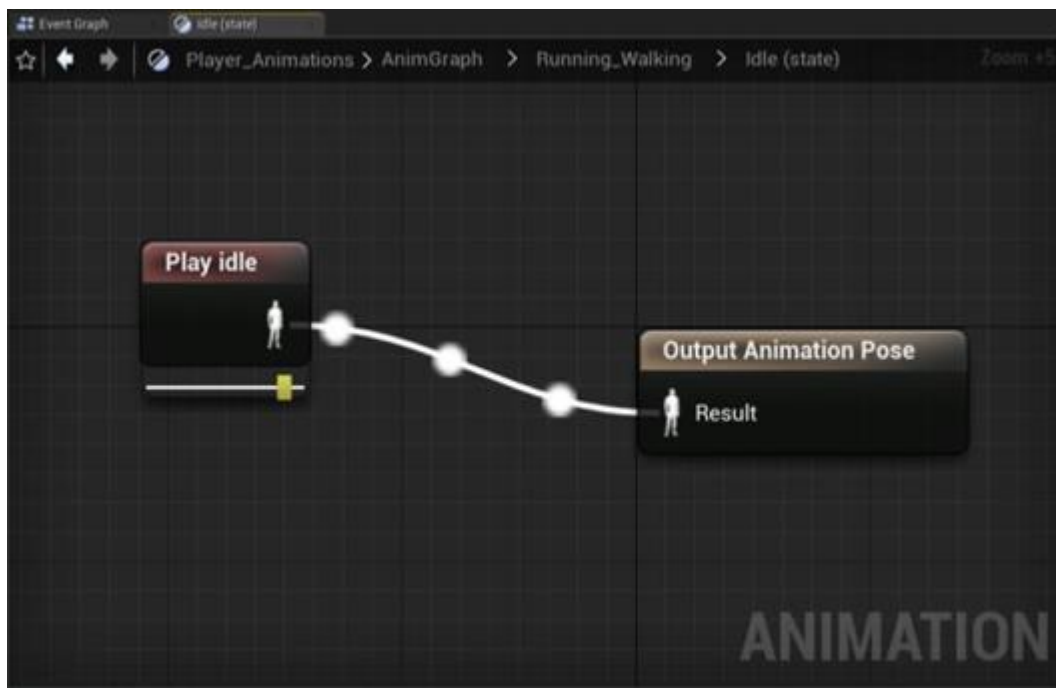


Рис. 3.22. Стан Idle

Перед тим як додати анімацію ходьби, важливо розуміти, як і куди наш персонаж може рухатися, а також з якою швидкістю. Ми створюємо асет Blend Space, і в його редакторі налаштовуємо анімації так, щоб вони змінювалися в залежності від швидкості та напрямку руху гравця (рис. 3.23).



Рис. 3.23. Редактор Blend Space

Повертаємося до редактора створеного раніше кінцевого автомату. У стані Run_Walk ми додаємо раніше створений Blend Space, де вказуємо напрям та швидкість руху персонажа. Ці дані буде розраховувати, тому створюємо дві змінні, в які будемо записувати всі необхідні параметри (рис. 3.24).

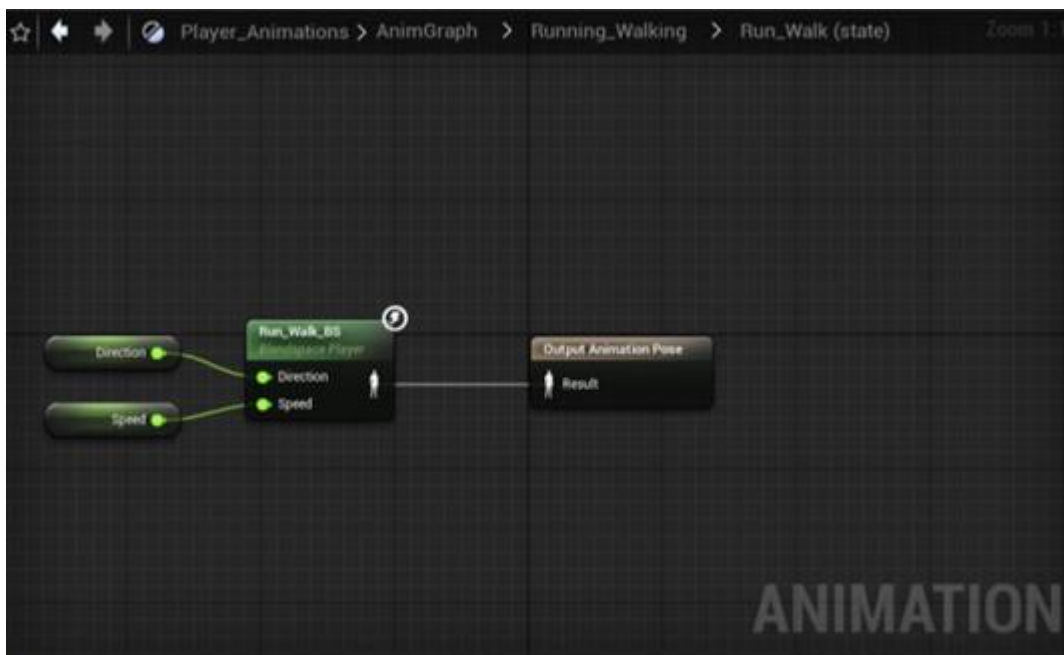


Рис. 3.24. Стан Run_Walk

Переходимо до вікна Event Graph та реалізуємо скрипт, за допомогою якого буде встановлюватися і напрям, і швидкість персонажа (рис. 3.25).

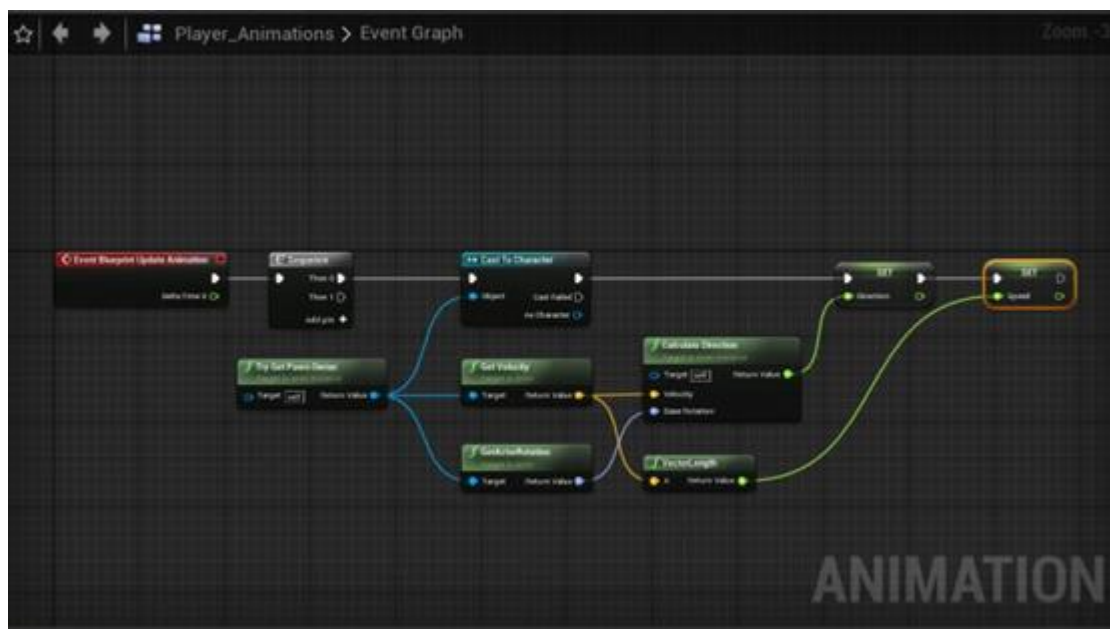


Рис. 3.25. Скрипт встановлення напрямку і швидкості персонажа

Тепер необхідно вказати умови переходу між станами стояння на місці та бігу. Ми налаштуємо усе так, що якщо швидкість гравця перевищує

десять, то відтворюється анімація бігу (рис. 3.26), і якщо вона стає менше десяти, то запускається анімація стояння на місці (рис. 3.27).

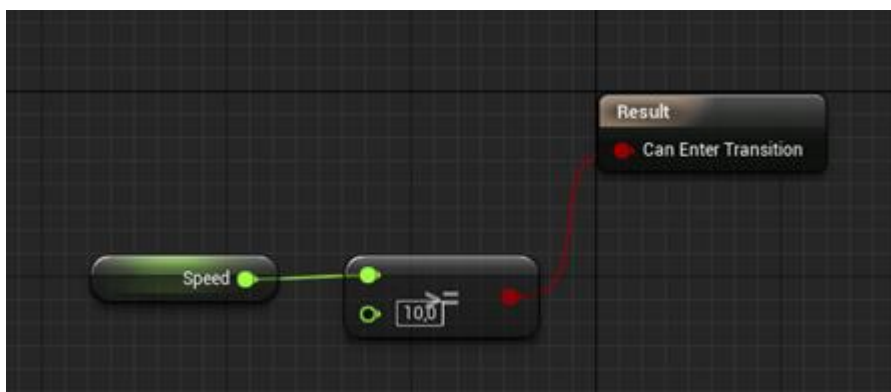


Рис. 3.26. Правило, при якому запускається анімація бігу



Рис. 3.27. Правило, при якому запускається анімація стояння на місці.

За останнім етапом ми замінюємо стандартного персонажа на того, що має необхідну для нас анімацію (рис. 3.28). Після початку гри можна помітити, що руки персонажа розташовані відповідним чином, готові утримувати зброю (рис. 3.29).



Рис. 3.28. Заміна стандартного персонажа на тимчасову модель



Рис. 3.29. Вид від першої особи

3.7. Додавання зброї до гравця

У шутері обов'язково повинна бути представлена зброя для гравця, і на вибір існує багато різних видів. Під час розробки вирішили тимчасово скористатися автоматом, зупинившись на моделі автомата АК-47, яку вдалося знайти серед готових об'єктів у відкритому доступі. Після завантаження моделі було виконано її додавання до проекту. Тепер залишалось лише прив'язати автомат до ігрового персонажа.



Рис. 3.30. Клас Gun

Далі треба за допомогою редактора скелету гравця прив'язати автомат до правої руки ігрового персонажу (рис. 3.31).

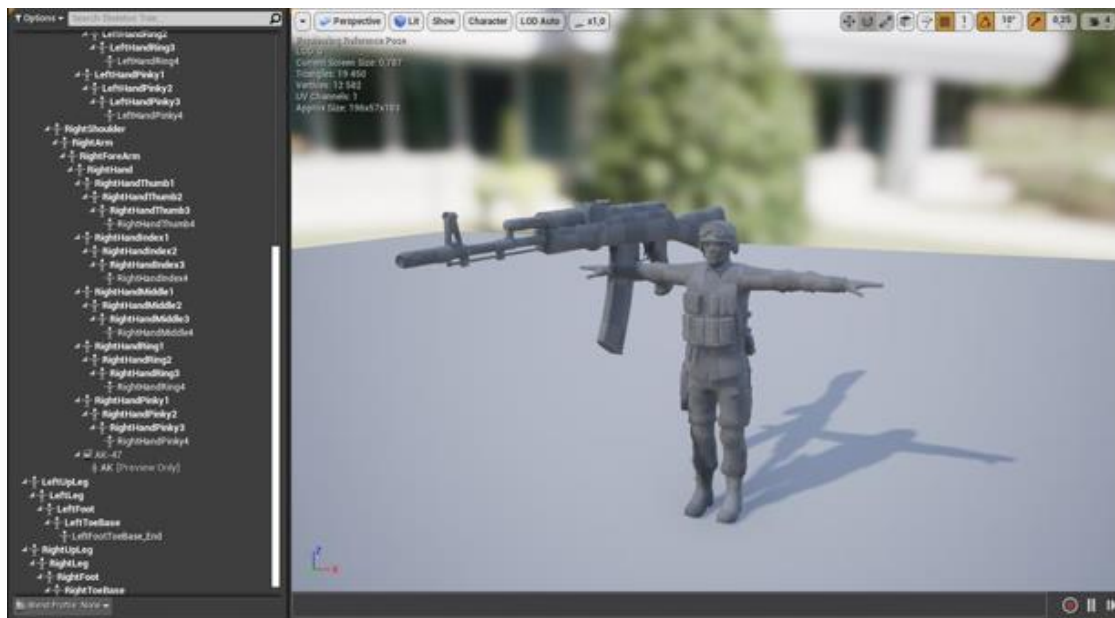


Рис. 3.31. Прив'язка автомату до руки персонажа

Як видно, необхідно налаштувати розташування автомата. По-перше, слід активувати анімацію "Idle_aiming" у гравця, а потім, використовуючи переміщення, повороти та зменшення розмірів, розмістити автомат у руках персонажа (див. рис. 3.31).



Рис. 3.32. Налаштування положення автомату

Для того щоб гравець отримав автомат після початку гри, необхідно розширити блупринт ThirdPersonCharacter за допомогою кількох скриптів (рис. 3.33). Після цього проводимо перевірку, щоб забезпечити коректну роботу (рис. 3.34).

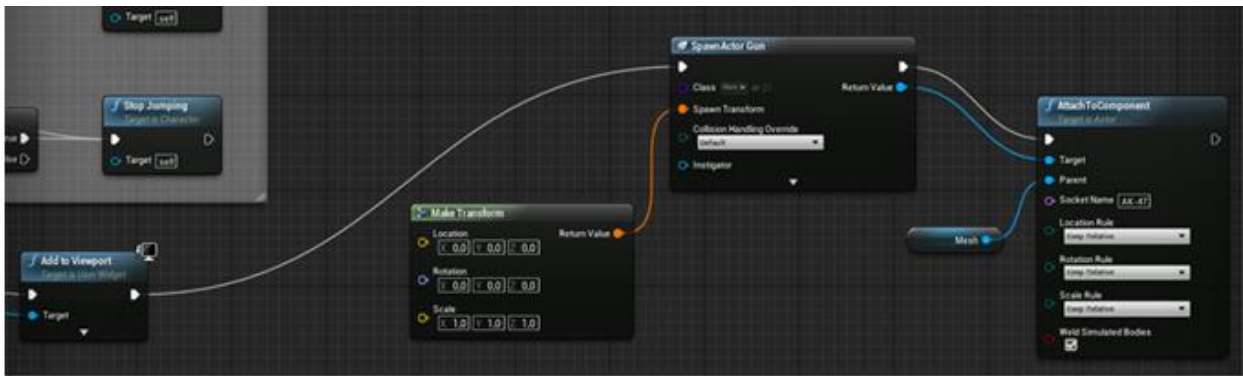


Рис. 3.33. Додання скриптів



Рис. 3.34 Перевірка роботи

3.8. Стрільба зі зброї

На початковому етапі необхідно створити новий клас, який буде названий `Bullet`. Основним компонентом цього класу буде меш `Shape_NarrowCapsule`, який служитиме обгорткою для кулі, випущеної гравцем через автомат. Після цього додаємо компонент `ProjectileMovement` та конфігуруємо початкову та максимальну швидкість кулі, а також визначаємо вплив гравітації на неї (рис. 3.34).

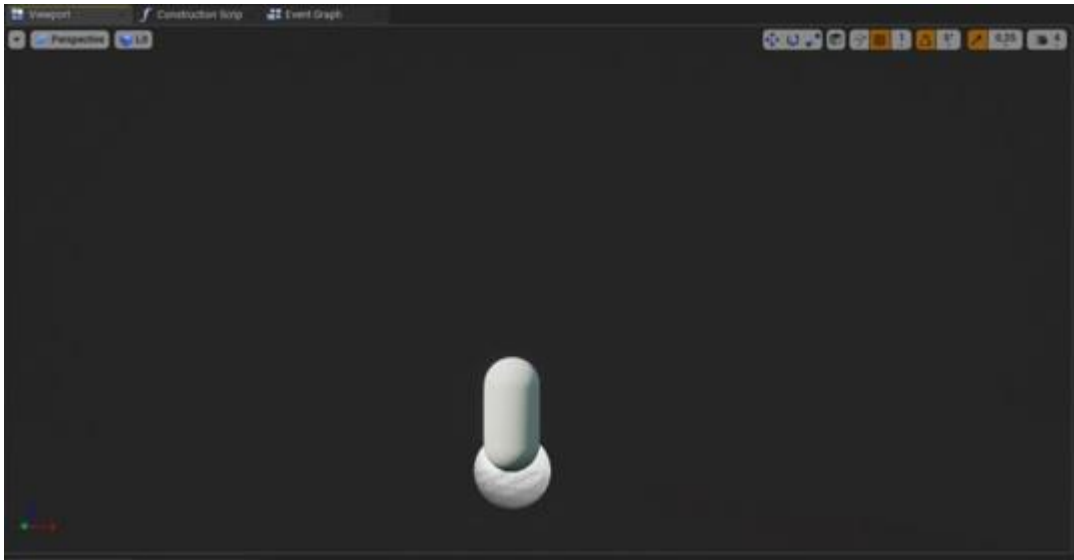


Рис. 3.35. Клас Bullet

Під час стрільби кулі повинні виходити з дула автомата. Для досягнення цього ефекту спочатку необхідно чітко визначити точку, в якій кулі будуть з'являтися. Для цього ми відкриваємо редактор скелету нашої зброї, додаємо так званий сокет та переміщаємо його в кінець дула (рис. 3.36).

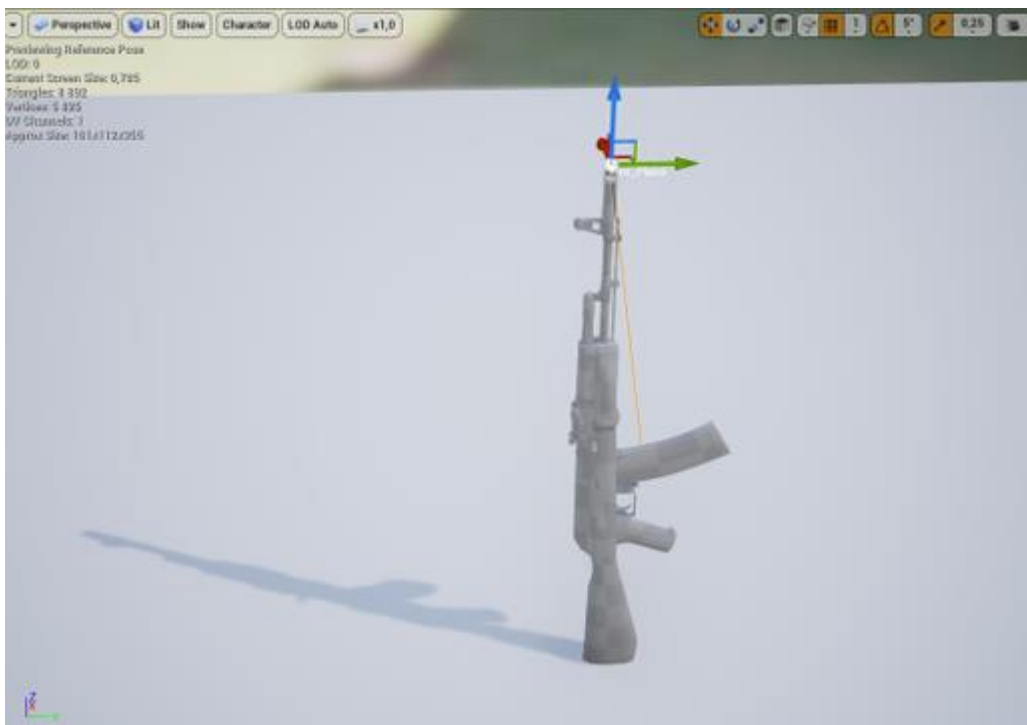


Рис. 3.36. Задання точки вильоту куль

Тепер відкриваємо блупринт клас Gun та створюємо нову подію, під час якої кулі будуть вилітати з дула автомата (рис. 3.37).

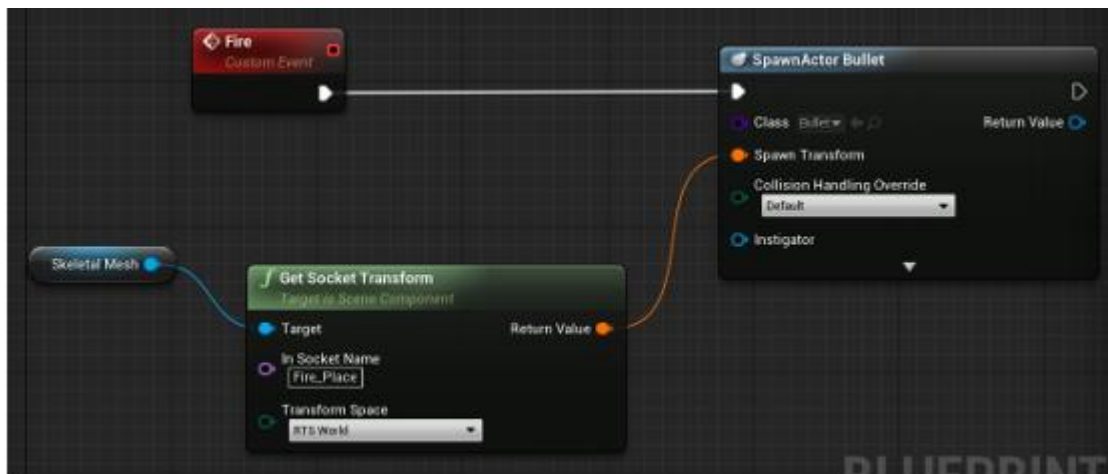


Рис. 3.37. Подія вильоту куль з автомату

Після створення події важливо вказати, які кнопки мають відслідковуватися, щоб активувати цю подію. Це робиться через налаштування введення (Input) в параметрах проекту. Ми вказуємо, що подія "Fire" повинна відбуватися при натисканні лівої кнопки миші (рис. 3.38).

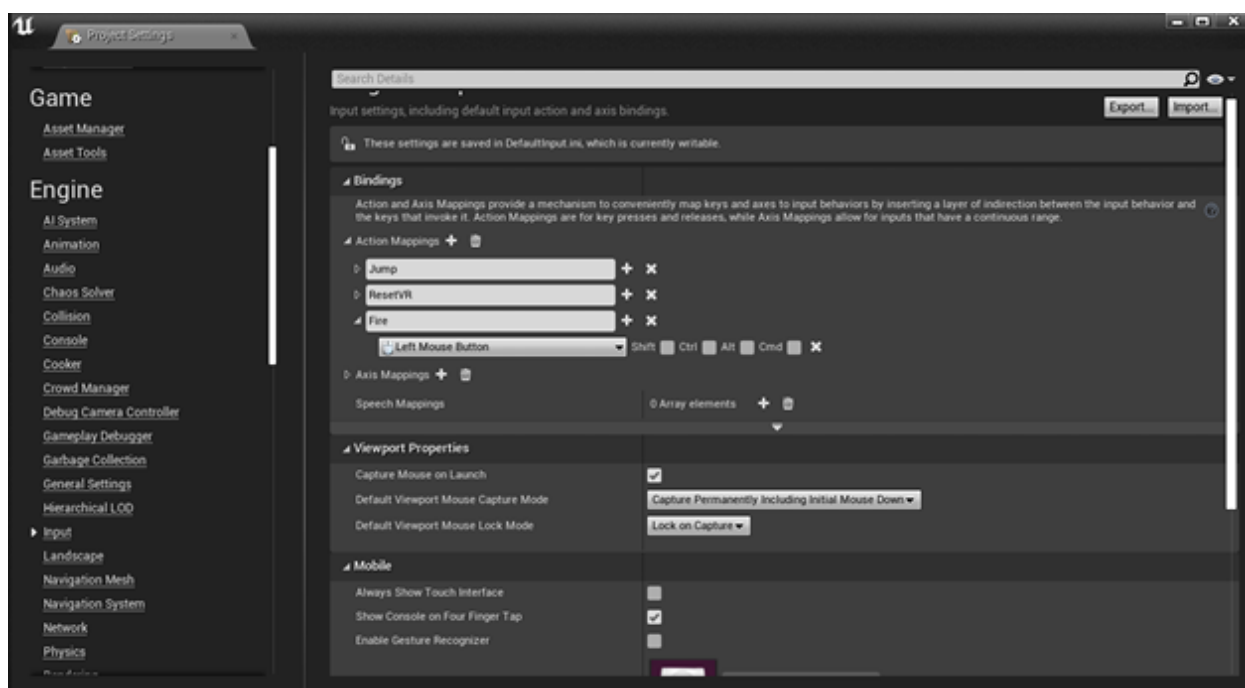


Рис. 3.38. Відстеження кнопок

Тепер у блупринті гравця необхідно налаштувати виклик події, яка призведе до вистрілів з автомату. Спочатку створюємо посилання на автомат, щоб мати можливість взаємодіяти з подією в зброї (рис. 3.39).

Після цього додаємо подію, яка при натисканні лівої кнопки миші ініціює подію "Fire" у зброї гравця (рис. 3.40).

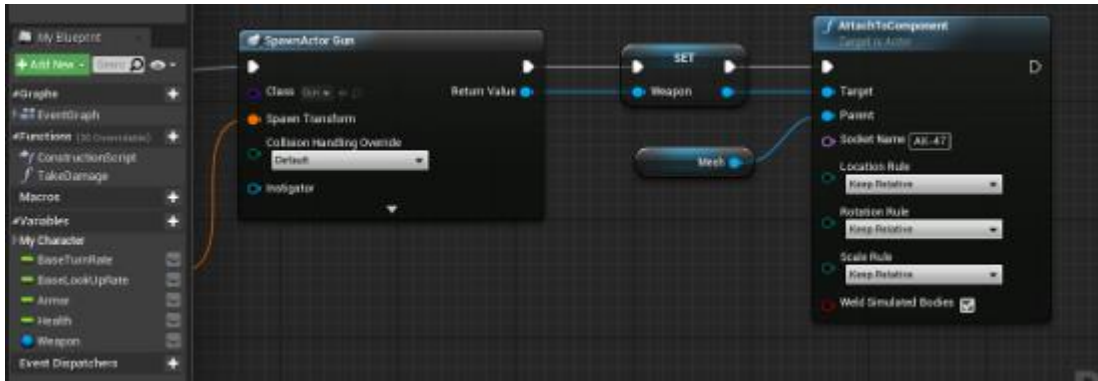


Рис. 3.39. Посилання на автомат

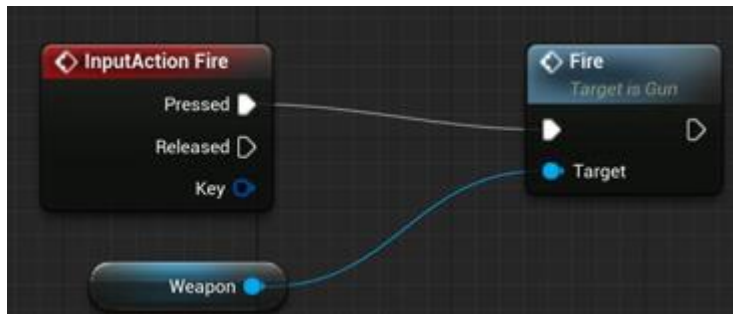


Рис. 3.40. Подія, яка запускає подію вильоту куль з автомату

Перед перевіркою правильності стрільби встановимо невелике значення (200) швидкості руху куль, щоб легше слідкувати за їхньою траєкторією. Під час перевірки виявлено, що кулі рухаються не у напрямку вперед, як очікувалося, а вліво (рис. 3.41).

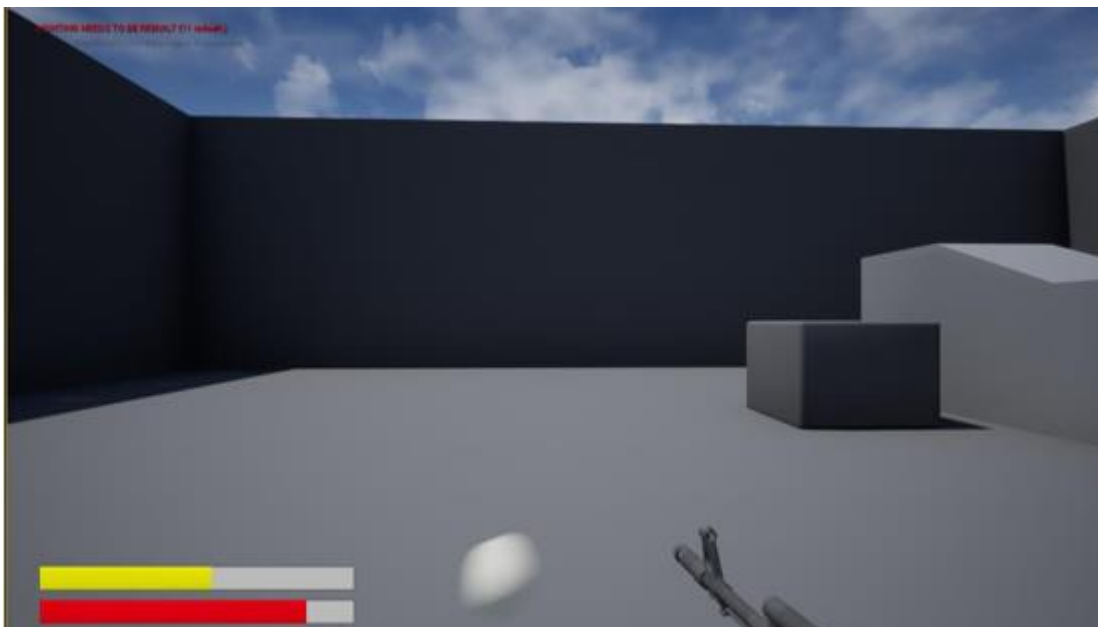


Рис. 3.41. Кулі летять вліво

Цю проблему вдалося вирішити, виконавши поворот сокета "Fire" в редакторі скелету автомата. Тепер кулі рухаються вперед, але їх передня частина вилітає боком, а не спрямована вперед (рис. 3.42).



Рис. 3.42. Кулі летять вперед, але боком

Цю проблему вдалося вирішити, змінивши орієнтацію кулі в редакторі блупринта Bullet (рис. 3.43).



Рис. 3.43. Кулі летять як треба

Для більшого реалізму залишилося зменшити розміри куль та збільшити швидкість їх польоту (рис. 3.44).

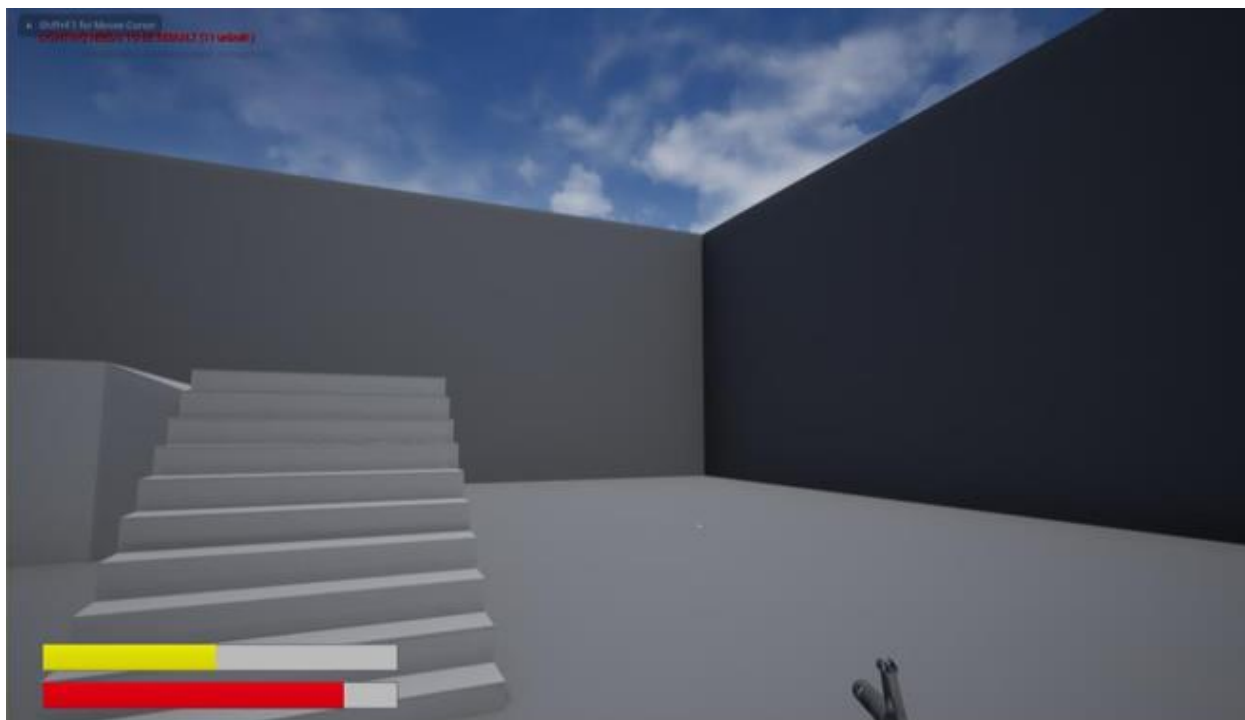


Рис. 3.44. Кінцевий варіант вильоту куль

3.9. Додавання ворогів та знищення їх

Головний ігровий персонаж повністю готовий. Тепер потрібно додати йому ворогів. Створюємо блупринт класу типу `Character` під назвою `Enemy_AI` зі стандартним видом персонажа, зі стандартними анімаціями та змінною `Health` (рис. 3.45).

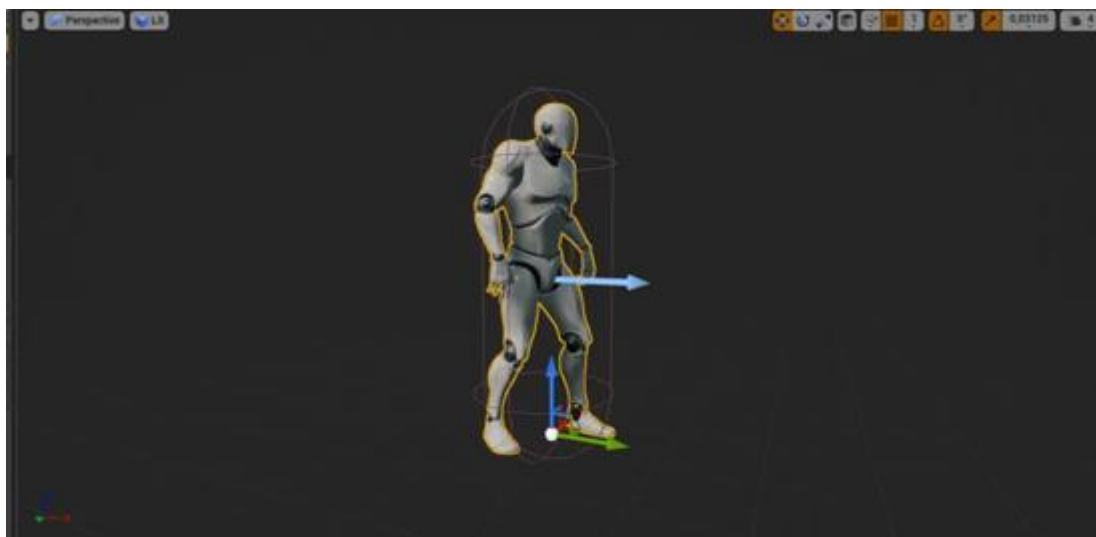


Рис. 3.45. `Enemy_AI`

Ми використовуємо візуальне програмування через `Blueprints` для налаштування персонажа так, щоб при кожному попаданні кулі в нього віднімалося життя (рис. 3.46). Якщо рівень життя зменшиться до нуля, то ворог зникне (помре).

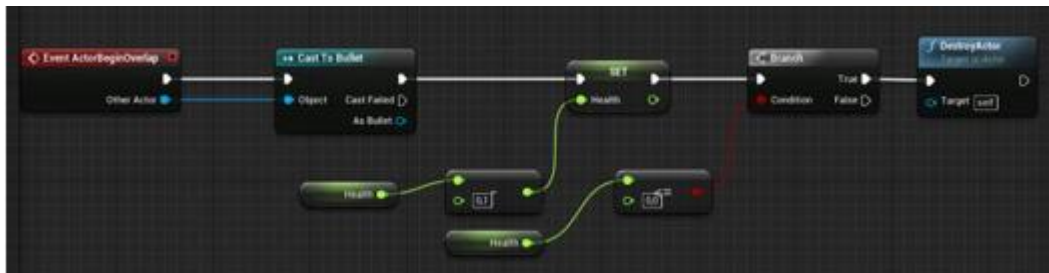


Рис. 3.46. Скрипт віднімання життя у ворогів

Перевірка роботи скрипта пройдена успішно (рис. 3.47, 3.48).
 Виявилося, що для вбивства ворога достатньо 10 куль.



Рис. 3.47. Перед гравцем три вороги



Рис. 3.48. Гравець вбив одного ворога, випустив в нього 10 куль

3.10. Урон від ворогів

Уявімо, що наші вороги – цілі ближнього бою і можуть тільки атакувати руками. Отже, урон по гравцю повинен наноситися тоді, коли він стоїть поруч з ворогом. Щоб це реалізувати, створюємо блупринт з назвою Rain та додаємо до нього компонент Capsule Collision.

Цей компонент – область у формі капсули. Потрібно його налаштувати так, щоб гравець, увійшовши в капсулу, отримував урон. У нашому випадку, коли гравець увійде в капсулу, то запуститься функція TakeDamage, яка була написана раніше. Реалізуємо все це за допомогою візуального програмування у редакторі Net Graph (рис. 3.49).

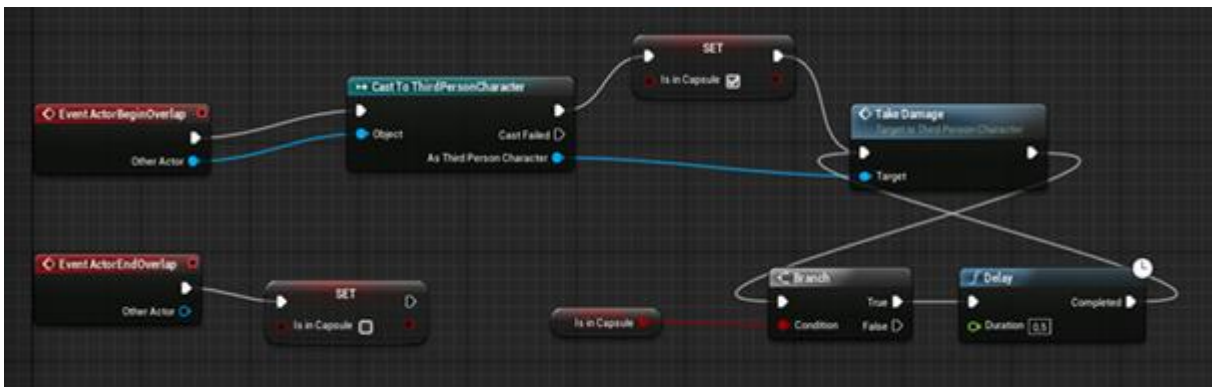


Рис. 3.49. Скрипт для капсули, що наносить урон гравцю

Наступним кроком є пов'язання класу Rain з класом Enemy_AI. Тепер кожен ворог матиме область навколо себе, яка буде завдавати урон гравцеві, емулюючи ближній бій (рис. 3.50).

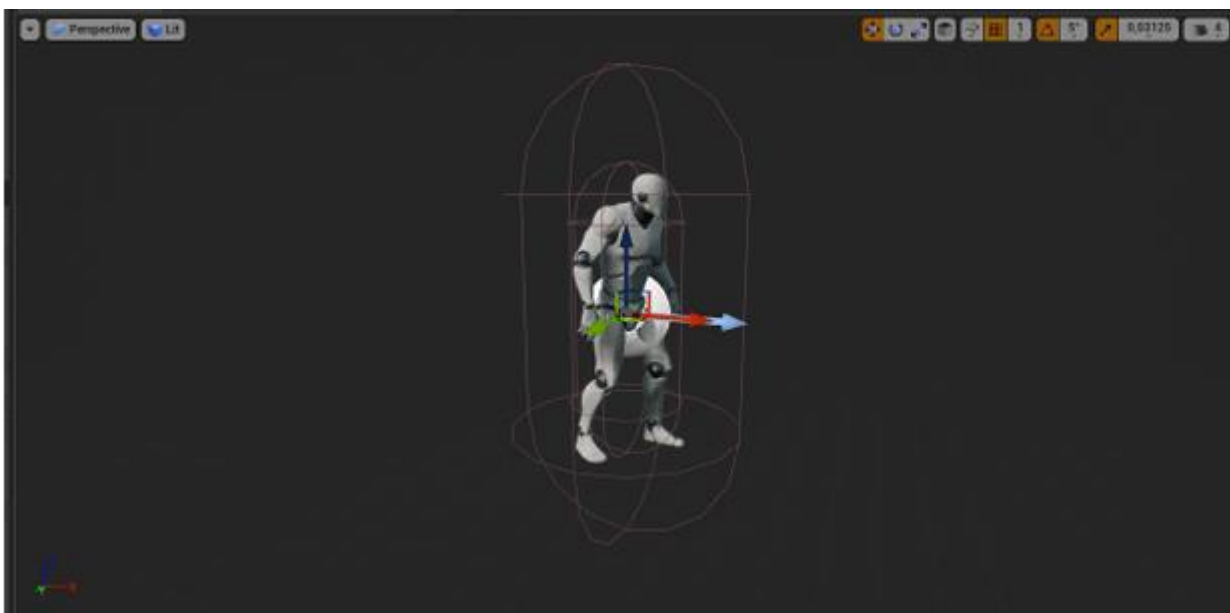


Рис. 3.50. Ворог з зоною навколо себе, яка нанесе урон гравцю

3.11. Штучний інтелект

Для того щоб вороги могли відстежувати гравця, їм потрібно вказати, де можна рухатися, а де заборонено. Додамо на рівень Nav Mesh Bounds Volume з панелі Modes. Розтягнемо його на весь рівень, забезпечивши персонажам із штучним інтелектом можливість рухатися по всьому рівню. Зелений колір на ілюстрації вказує на область, доступну для руху (рис. 3.51).

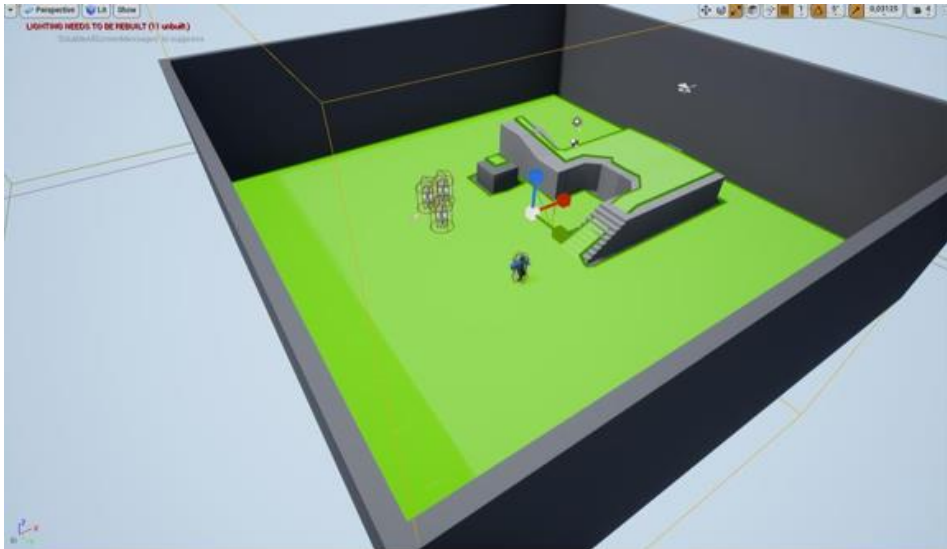


Рис. 3.51. Nav Mesh Bounds Volume

Додаємо компонент Pawn Sensing до блупринта Enemy_AI. Цей компонент надає можливість персонажу із штучним інтелектом виявляти гравця, і, як тільки він його помічає, ми можемо використовувати цю інформацію, наприклад, для вказівки штучному інтелекту пристосуватися та визначити напрямок руху. У налаштуваннях Pawn Sensing встановлюємо кут огляду на 180 градусів (рис. 3.52). Це забезпечить постійне відстеження гравця персонажами із штучним інтелектом.

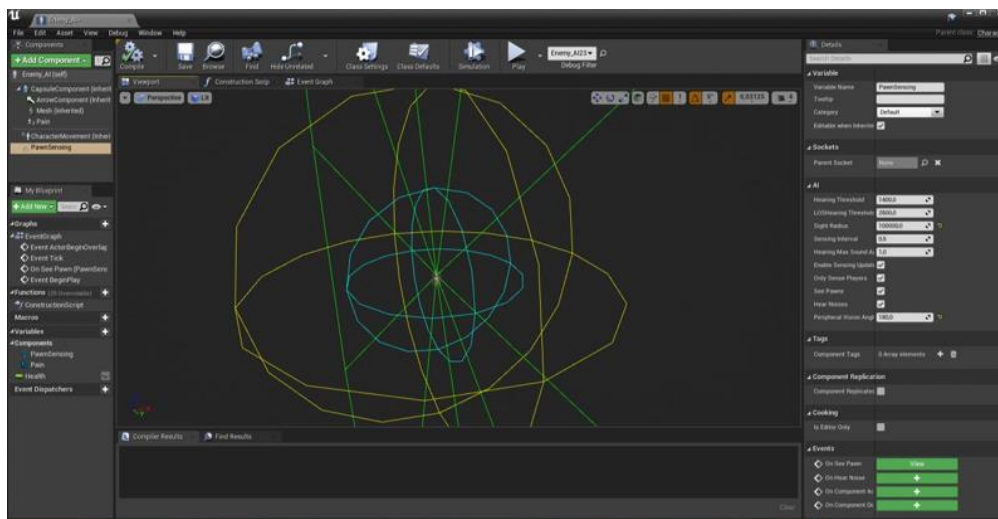


Рис. 3.52. Pawn Sensing

Тепер налаштуємо реакцію на подію, коли гравець потрапляє в поле зору штучного інтелекту. В такому випадку нехай персонаж із штучним інтелектом розпочне переслідування гравця, намагаючись максимально наблизитися до нього. Також важливо підібрати швидкість для ворогів так, щоб вона була більшою за максимальну можливу швидкість гравця (рис. 3.53).

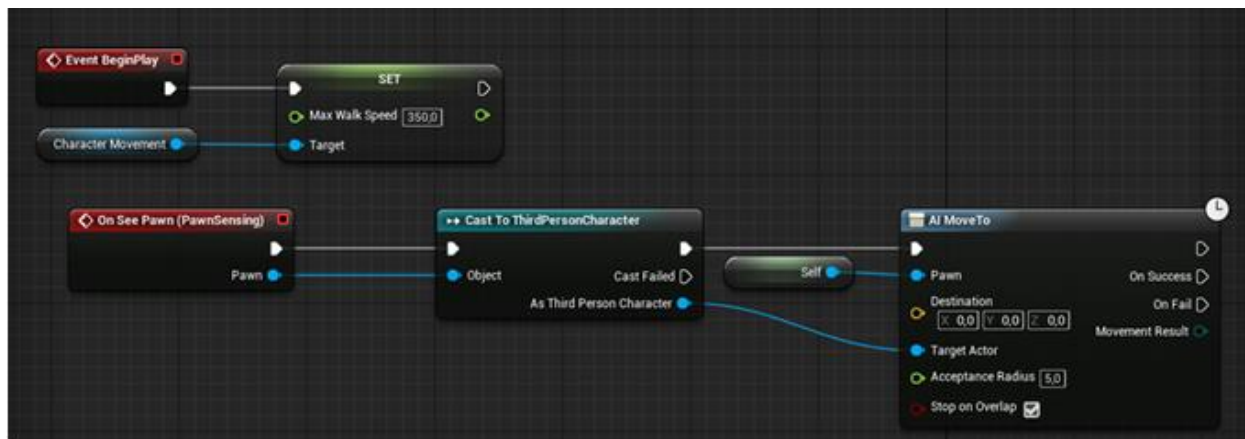


Рис. 3.53. Скрипт, який змушує ворогів постійно бігати за гравцем

Компілюємо та запускаємо гру. Все працює так, як і потрібно, вороги біжать за гравцем і якщо наздоганяють його, то наносять урон.

РОЗДІЛ 4. ДЕМОНСТРАЦІЯ ВІДЕОГРИ

При початку гри гравець з'являється у просторі великої кімнати, оточений численними живучими ворогами, які миттєво виявляють інтерес до його усунення (див. рис. 4.1). Вороги обмежені можливістю лише бігати за гравцем (див. рис. 4.2) і завдавати йому урон, якщо вони йому дістають (див. рис. 4.3, 4.4). Гравець, зі свого боку, може переміщатися і вбивати ворогів, використовуючи автомат (див. рис. 4.6). Коли гравець уникає ворогів і вони не можуть його догнати, його обладунки повільно відновлюються (див. рис. 4.5). Основна мета гравця - вижити, вбивши усіх ворогів і не витрачаючи усю свою шкалу здоров'я (див. рис. 4.7).

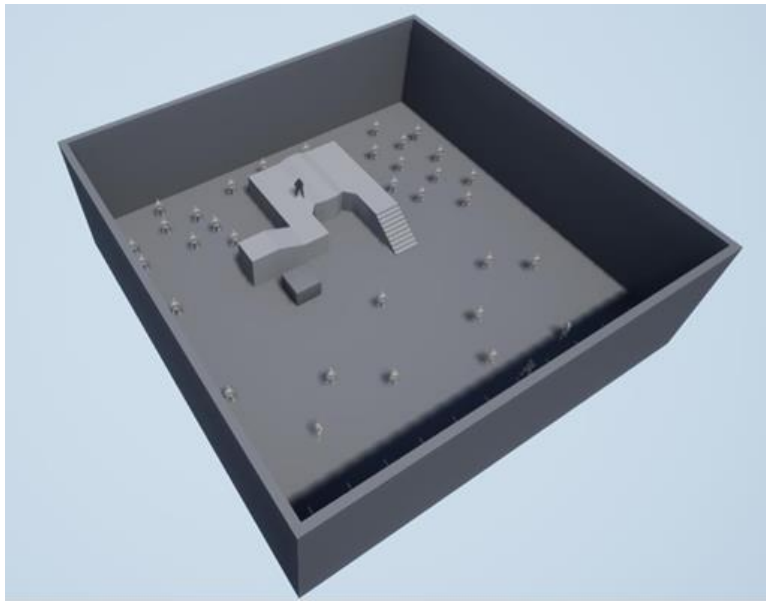


Рис. 4.1. Ігровий рівень спочатку гри, вид згору



Рис. 4.2. Початок гри, вороги почали переслідувати гравця



Рис. 4.3. Вороги наздогнали гравця та почали наносити урон



Рис. 4.4. У гравця закінчилася шкала броні та почало закінчуватися життя



Рис. 4.5. Гравцю вдалося відбігти на безпечну дистанцію від ворогів, шкала броні почала відновлюватися



Рис. 4.6. Гравець ліквідував більшу кількість ворогів



Рис. 4.7. Гравець вбив всіх ворогів та вижив

ВИСНОВКИ

Під час експлорації різних джерел проводилося дослідження концепції комп'ютерних ігор, яке включало класифікацію ігор за чотирма критеріями. У додаток до цього був розроблений алгоритм для процесу розробки відеоігор.

Також було вивчено популярні інструменти для розробки, їх порівняно, і визначено найбільш актуальні для новачків. Вибір пріоритетних інструментів базувався на доступності та функціональності.

Під час аналізу наявних розробок було проведено порівняння, в результаті чого були визначені переваги та недоліки кожної з них. В процесі цього аналізу стало очевидним, що при створенні комп'ютерної гри з простою ігровою механікою слід приділяти увагу додатковим аспектам, таким як сюжет та графічне оформлення. Це важливо для того, щоб зберегти інтерес потенційних гравців і подовжити життєвий цикл гри.

Після вибору інструментів для розробки розпочалось освоєння Unreal Engine 4 та реалізація самого проекту. Під час розробки глибше вивчено можливості гейм-руші Unreal Engine 4 та використання візуального програмування за допомогою Blueprints. Набуто необхідних знань та навичок, таких як:

- створення рівнів;
- створення і написання скриптів;
- налаштування об'єктів;
- створення штучного інтелекту.

Вивчення робочого середовища Unreal Engine 4 має важливе значення, оскільки індустрія розробки ігор набуває все більшого значення у нашому суспільстві. Гри вже давно перестали бути лише засобом розваг, вони активно використовуються у різних сферах, таких як наука та освіта. Таким чином, розвиток в цьому напрямку можна вважати одним із ключових в сучасному суспільстві.

ПЕРЕЛІК ПОСИЛАНЬ

1. Carnall B. Unreal Engine 4.X By Example / Benjamin Carnall., 2016. – 506 с.
2. Lee J. Learning Unreal Engine Game Development / Joanna Lee., 2016. – 274 с.
3. L. Newton P. Unreal Engine 4 AI Programming Essentials / Peter L. Newton., 2016. – 188 с.
4. Sewell B. Blueprints Visual Scripting for Unreal Engine / Brenden Sewell., 2015. – 188 с.
5. Valcasara N. Unreal Engine Game Development Blueprints / Nicola Valcasara., 2015. – 352 с.
6. Unreal Engine Game Development Blueprints, 2015. – 352 с.
7. A.Moniem M. Mastering Unreal Engine 4.X / Muhammad A.Moniem., 2016. – 384 с.
8. Tavakkoli A. Game Development and Simulation with Unreal Technology / Alireza Tavakkoli., 2015. – 742 с.
9. Edmonds M. Mastering Game Development with Unreal Engine 4 / Matt Edmonds., 2018. – 356 с.
10. Куксон А. Разработка игр на Unreal Engine 4 за 24 часа / А. Куксон, Р. Даулінгсока, К. Крамплер., 2019. – 528 с.
11. Lee J. Unreal Engine: Game Development from A to Z / J. Lee, J. P. Doran, N. Misra., 2016. – 837 с.
12. Unreal Engine 4 Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.unrealengine.com/en-US/index.html>.
13. Тьюторіал по Unreal Engine [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: <https://habr.com/ru/post/344394/>.
14. Навігація Вьюпорта в Unreal Engine 4 [Електронний ресурс] – Режим доступу до ресурсу: <https://www.native-game.com/ue4-docs/ue4manual/navigacija-v-juporta-v-unreal-engine-4/>.
15. Как Импортировать Анимацию в Unreal Engine 4 [Електронний ресурс] – Режим доступу до ресурсу: <https://www.native-game.com/ue4-docs/ue4manual/kak-importirovat-animaciju-v-unreal-engine-4/>.