Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерних наук
(повна назва)

Кафедра програмної інженерії
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

Ігровий програмний застосунок в жанрі Action RPG, піджанр Souls-like. Підсистема рівнів та завдань
(тема)

Виконав:
здобувач 4 року навчання,
групи ПЗПІ-21-6

Владислава ТИМОЩЕНКО
(Власне Ім'я, ПРІЗВИЩЕ)

Спеціальність 121 – Інженерія програмного забезпечення
(код і повна назва спеціальності)
Тип програми освітньо-професійна
Освітня програма Програмна інженерія
(повна назва освітньої програми)

Керівник доц. кафедри ПІ Віктор КАУК
(посада, Власне Ім'я, ПРІЗВИЩЕ)

Допускається до захисту
Зав. кафедри _____ Кирило СМЕЛЯКОВ
(підпис) (Власне Ім'я, ПРІЗВИЩЕ)

2025 р.

Харківський національний університет радіоелектроніки

Факультет комп’ютерних наук
Кафедра програмної інженерії
Рівень вищої освіти перший (бакалаврський)
Спеціальність 121 – Інженерія програмного забезпечення
Тип програми Освітньо-професійна
Освітня програма Програмна Інженерія
(шифр і назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«___» _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Тимощенко Владиславі Русланівні
(прізвище, ім’я, по батькові)

1. Тема роботи Ігровий програмний застосунок в жанрі Action RPG, піджанр Souls-like. Підсистема рівнів та завдань.
Затверджена наказом по університету від 19.05.2025 р № 397 Ст
2. Термін подання здобувачем роботи до екзаменаційної комісії 19.06.2025
3. Вихідні дані до роботи Розробити однокористувацький ігровий застосунок в жанрі Action RPG, піджанр Souls-like. Проєкт має включати підсистему рівнів та завдань, що реалізується в Unreal Engine 5. В роботі використовується Blueprint-система для створення взаємодій, діалогів, тригерів та логіки вибору. Ігровий світ складається з хаб-локації, ігрових рівнів та NPC, які впливають на сюжетні розгалуження.
4. Перелік питань, що потрібно опрацювати в роботі Вступ, аналіз предметної галузі, формування вимог до програмної системи, архітектура та проєктування програмного забезпечення, опис прийнятих програмних рішень тестування розробленого програмного забезпечення, висновки, додатки.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної галузі	21.05.2025	*виконано*
2	Створення специфікації ПЗ	23.05.2025	*виконано*
3	Проєктування ПЗ	24.05.2025	*виконано*
4	Розробка ПЗ	27.05.2025	*виконано*
5	Тестування ПЗ	29.05.2025	*виконано*
6	Оформлення пояснювальної записки	30.05.2025	*виконано*
7	Підготовка презентації та доповіді	10.06.2025	*виконано*
8	Попередній захист	13.06.2025	*виконано*
9	Нормоконтроль, рецензування	17.06.2025	*виконано*
10	Здача роботи у електронний архів	17.06.2025	*виконано*
11	Допуск до захисту у зав. кафедри	18.06.2025	*виконано*

Дата видачі завдання 18 квітня 2025р.

Здобувач (ка)_____ Владислава ТИМОЩЕНКО
(підпис)

Керівник роботи _____ доц. кафедри ПІ Віктор КАУК
(підпис) (Власне Ім’я, ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи бакалавра, 60 стор., 28 рис., 19 джерел.

ДИНАМІЧНА СКЛАДНІСТЬ, ВЗАЄМОДІЯ ГРАВЦЯ І ЗАВДАНЬ, ІГРОВА МЕХАНІКА, ІГРОВИЙ ІНТЕРФЕЙС, КВЕСТИ, ПІДСИСТЕМА ЗАВДАНЬ, ПІДСИСТЕМА РІВНІВ, ПРОГРЕСІЯ ГРАВЦЯ, ACTION RPG, GAME DESIGN, SOULS-LIKE.

Об’єкт розроблення — програмний ігровий застосунок у жанрі Action RPG з реалізацією підсистеми рівнів та завдань, орієнтований на ігрову модель Souls-like.

Мета роботи — розробити ефективну архітектуру та реалізацію підсистеми рівнів і завдань у межах гри жанру Action RPG, яка забезпечить глибокий геймплей, прогресію персонажа та нелінійність виконання завдань у відповідності до філософії Souls-like ігор.

Методи дослідження — аналіз та узагальнення підходів у сучасному ігровому дизайні, моделювання ігрової логіки, побудова UML-діаграм для архітектурних рішень, реалізація ігрових сцен з використанням рушія Unreal Engine 5 та візуальної системи Blueprint, створення ігрових рівнів, а також тестування готових рівнів у симульованих геймплей-сценаріях.

Сфера застосування — розробка комп’ютерних ігор, навчальні проєкти з геймдеву, дослідження в галузі інтерактивних систем та поведінкових моделей.

У результаті розроблено і протестовано підсистему рівнів, яка враховує складність, нелінійність, масштабність зон дослідження, а також підсистему завдань з підтримкою послідовних, паралельних і умовних квестів з інтеграцією в ігровий інтерфейс.

ABSTRACT

DYNAMIC COMPLEXITY, PLAYER-TASK INTERACTION, GAME MECHANICS, GAME INTERFACE, QUESTS, TASK SUBSYSTEM, LEVEL SUBSYSTEM, PLAYER PROGRESSION, ACTION RPG, GAME DESIGN, SOULS-LIKE.

Object of development — a software game application in the Action RPG genre with the implementation of level and task subsystems, oriented towards the Souls-like game model.

Purpose of the work — to develop an efficient architecture and implementation of level and task subsystems within an Action RPG game that ensures deep gameplay, character progression, and non-linear task execution in accordance with the philosophy of Souls-like games.

Research methods — analysis and generalization of approaches in modern game design, modeling of game logic, construction of UML diagrams for architectural solutions, implementation of game scenes using the Unreal Engine 5 and the Blueprint visual scripting system, creation of game levels, as well as testing finished levels in simulated gameplay scenarios.

Field of application — development of computer games, educational game development projects, research in the fields of interactive systems and behavioral models.

As a result, a level subsystem was developed and tested, which takes into account complexity, non-linearity, and scale of exploration zones, as well as a task subsystem supporting sequential, parallel, and conditional quests integrated into the game interface.

ЗМІСТ

ВСТУП

Ринок відеоігор сьогодні є однією з найдинамічніших і прибуткових сфер цифрових технологій. Значну популярність здобули ігри жанру Action RPG, зокрема Souls-like, що вирізняються складністю, глибокими бойовими механіками, атмосферністю та продуманим дизайном. Стандарти жанру сформувала студія FromSoftware (Dark Souls, Bloodborne, Elden Ring). Souls-like проєкти зазвичай поєднують складну структуру рівнів, нелінійний сюжет і прогресію персонажа.

Актуальність даної роботи полягає в необхідності створення функціональної та адаптивної підсистеми рівнів і завдань для гри в жанрі Action RPG, яка відповідатиме сучасним вимогам до ігор цього типу. Особливої уваги потребує реалізація туторіальних етапів, бойових рівнів з унікальними ворогами, включаючи босів, хаб-локацій для взаємодії з NPC та розвитку сюжету, а також елементів занурення (звуки, освітлення, відео тощо).

Ціль роботи — розробка та впровадження підсистеми рівнів і завдань як складової програмного ігрового застосунку у жанрі Souls-like. Дана підсистема має забезпечити послідовний ігровий прогрес, інтеракцію з елементами оточення, керування складністю та забезпечення наративного розвитку.

Можливі сфери застосування результатів — розробка інді-ігор, навчальні проєкти у сфері GameDev, дослідницькі проєкти, пов’язані з геймдизайном, або частина великого комерційного проєкту.

Розробка виконується у межах командного дипломного проєкту. Кожен з учасників реалізує окремі підсистеми гри. Представлена робота присвячена створенню саме підсистеми рівнів та завдань, що має критичне значення для загального функціонування гри та забезпечення гравця основною взаємодією з ігровим середовищем.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Аналіз предметної галузі

Жанр Action RPG займає одну з ключових позицій у сучасній індустрії відеоігор завдяки своїй здатності поєднувати динамічний бойовий процес із глибоким наративом, розгалуженою системою розвитку персонажа та нелінійною побудовою ігрового світу. Піджанр Souls-like, який отримав свою назву від серії Dark Souls, є своєрідним еталоном у створенні складних, глибоко механізованих ігор, що вимагають від гравця терпіння, стратегічного мислення та уважності.

Souls-like ігри вирізняються низкою характерних рис:

а) високий рівень складності, що створює відчуття виклику і досягнення;

б) нелінійна побудова рівнів, яка спонукає до дослідження та відкриття нових маршрутів;

в) відсутність традиційної системи збереження прогресу, що замінюється системою контрольних точок (бонфайрів, вівтарів тощо);

г) глибока бойова система, зосереджена на таймінгу, ухиленні, парируванні та витривалості;

д) атмосферність, яку підсилює музичне та звукове оформлення, освітлення та візуальні ефекти;

е) обмежена подача сюжету, що сприяє інтризі та бажанню досліджувати.

Такі проєкти, як Dark Souls, Bloodborne, Sekiro: Shadows Die Twice та Elden Ring, стали культовими не лише завдяки складності, але й завдяки ретельно побудованому ігровому світу, в якому кожна локація має власну ідентичність, складну структуру та логічно зв’язані елементи геймплею. (див. рис. 1.1)

Рисунок 1.1 – Структура ігрового світу Lordran у грі Dark Souls [1]

Одним із ключових компонентів таких ігор є система рівнів та завдань, що реалізується не через традиційні маркери квестів, а завдяки інтеракції з оточенням, предметами, монологами NPC, а також через наслідковість дій гравця. Це означає, що гравець може не лише втратити важливий елемент сюжету, не взаємодіючи з певним NPC, але й повністю змінити хід гри через виконання або ігнорування конкретного завдання.

Підсистема рівнів у таких проєктах має бути адаптивною, дозволяти побудову як лінійних, так і відкритих просторів, підтримувати можливість повернення до попередніх зон, створення умовних гейтів (наприклад, двері, які

відкриваються лише після певної події), а також дозволяти вбудовування ворожих юнітів і босів, з якими пов’язане просування гравця.

Підсистема завдань, у свою чергу, не повинна нав’язуватись, а органічно вплітатися в ігровий світ, надаючи гравцю право інтерпретації. Часто такі завдання створюються у вигляді низки умов: наприклад, гравець має знайти певний предмет, поговорити з конкретним NPC або перемогти ворога в іншій зоні.

При розробці таких підсистем в ігрових рушіях, зокрема Unreal Engine, важливу роль відіграє система Blueprint-логіки, яка дозволяє створювати ігрові механіки [2]. Навігація, колізії, тригери подій, реакції на події — усі ці елементи реалізуються та інтегруються безпосередньо у структуру рівнів. Основна увага приділяється створенню атмосфери за допомогою візуальних ефектів, освітлення, аудіосупроводу, а також розміщенню об’єктів і налаштуванню взаємодій, що забезпечують цілісний і логічний геймплей (див. рис. 1.2).

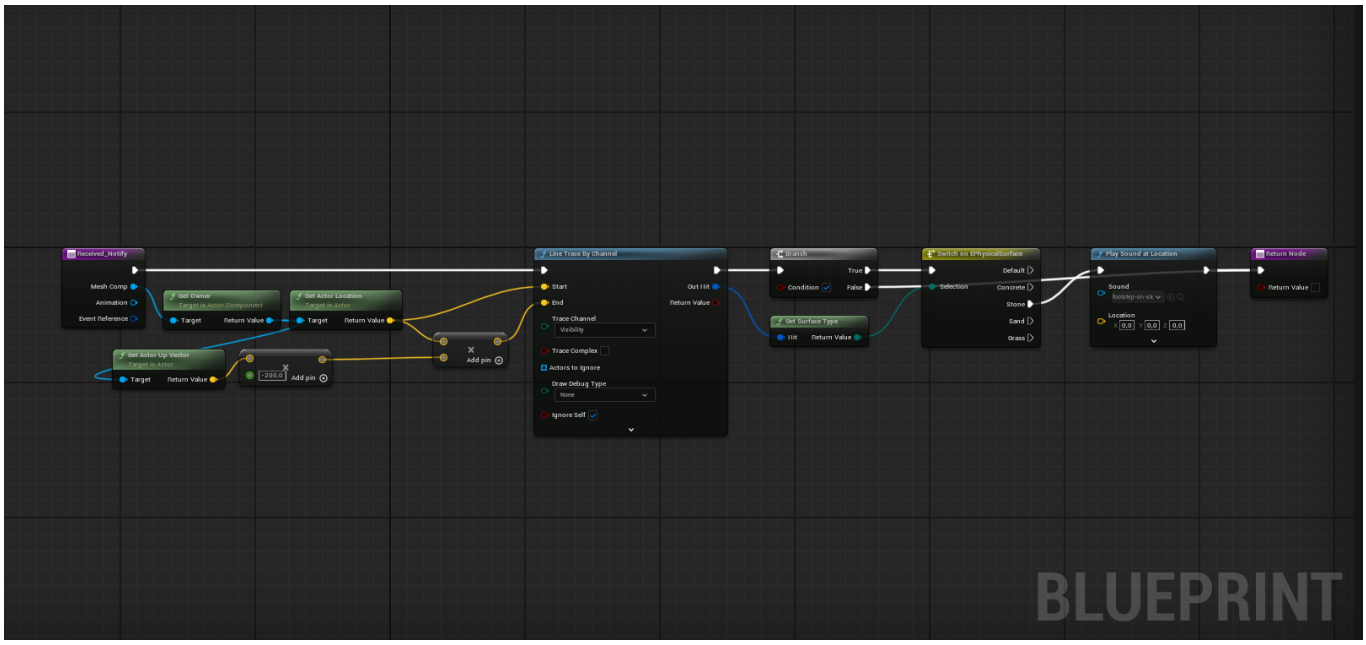


Рисунок 1.2 – Приклад логіки звуку кроків при ходьбі в Blueprint (виконано самостійно)

На сучасному етапі, інструменти Unreal Engine дозволяють створювати кінематографічні сцени, інтерактивне середовище, реалістичне освітлення та просторовий звук, що є необхідним для досягнення глибокого занурення — ключового чинника успішності Souls-like проєктів. У межах даної кваліфікаційної роботи реалізується саме така підсистема, що охоплює:

а) туторіальний рівень, де гравець навчається основам керування;

б) хаб-локацію, яка слугує безпечною зоною з NPC;

в) повноцінний бойовий рівень з босом;

г) завдання, які поступово вплітаються в сюжетну канву.

Це дозволяє не лише протестувати функціонал у різних сценаріях, але й показати практичну придатність підсистеми до розширення у майбутньому.

1.2 Виявлення та вирішення проблем

Жанр Souls-like, незважаючи на свою популярність, має ряд специфічних викликів, як для розробників, так і для гравців. Однією з основних проблем є високий поріг входу — новачки часто стикаються з надмірною складністю вже на початкових етапах гри. Це призводить до фрустрації, втрати мотивації та, як наслідок, до зниження утримання користувача. Багато гравців так і не проходять навіть першу локацію гри, що підтверджують статистичні дані Steam щодо досягнень у популярних Souls-like проєктах [3].

Ще однією проблемою є відсутність чітких інструкцій або туторіалів. У класичних Souls-іграх гравець повинен самостійно вивчати механіки бою, прокачування персонажа та навігації ігровим світом. Хоча такий підхід сприяє глибшому зануренню в гру, для нових користувачів це може стати серйозним бар’єром.

Для вирішення цих проблем у межах розроблюваного проєкту передбачено створення м'якого вступного рівня (туторіалу), який у зрозумілій та ненав’язливій формі навчить базовим механікам гри. Туторіал побудований так,

щоб користувач міг відразу взяти участь у простому бою, спробувати ухилення, атаку та взаємодію з об'єктами.

Також проблемою є недостатня візуальна зрозумілість світу — відсутність підказок чи маркерів нерідко призводить до того, що гравець губиться або довго не може знайти шлях далі. У межах проєкту було вирішено реалізувати логічну структуру рівнів із яскравими візуальними орієнтирами та хабом, з якого гравець може переміщатися до різних зон (див. рис. 1.3).

Рисунок 1.3 – Рівень Hub, звідки гравець може переміщуватися в різні локації (виконано самостійно)

Таким чином, уже на етапі технічного планування були враховані основні недоліки жанру, та визначено шляхи їх подолання, що дозволить реалізувати гнучку, доступну та захопливу гру з духом Souls-like, але з меншою кількістю фрустрацій для новачків.

1.3 Постановка задачі

У сучасному ігровому просторі, насиченому великою кількістю однотипних проектів, все більше користувачів прагнуть до глибших вражень та

ігор, які не просто розважають, а й викликають емоційний резонанс, потребують стратегічного мислення та вміння адаптуватись до складних умов. Особливої популярності останніми роками набули так звані Souls-like ігри — складні, але захопливі за геймплеєм ігри, де кожна дія має значення, а гравець мусить постійно вчитись на своїх помилках, щоб досягти прогресу.

Однак незважаючи на популярність жанру, багато таких ігор мають ряд проблем, зокрема: надмірно високий поріг входу для новачків, відсутність адаптації для менш досвідчених гравців, а також одноманітність ігрових механік або занадто складна навігація в ігровому світі. Тому постає потреба створення нового програмного продукту, що поєднає глибину ігрового процесу з доступністю та інтуїтивно зрозумілим керуванням.

Метою проєкту є розробка набору ігрових рівнів у жанрі Souls-like, які:

а) зберігають ключові риси жанру: високу складність, непрямий наратив;

б) використовують адаптивні засоби дизайну, що допомагають новим гравцям поступово занурюватися у світ гри;

в) мають унікальний візуальний стиль, багаторівневу структуру локацій, атмосферне освітлення та саундтреки, що підсилюють загальне враження;

г) передбачають баланс між викликами та нагородами, забезпечуючи почуття задоволення після подолання складнощів;

д) допускають можливість доповнення і розширення у майбутньому.

Проєкт фокусується на створенні якісного ігрового середовища шляхом побудови архітектури рівнів [4], налаштування логіки взаємодії, інтеграції звукових та візуальних ефектів, а також використання Blueprint-системи для реалізації поведінки елементів оточення, подій та тригерів.

1.3.1 Цільова аудиторія

Основною цільовою аудиторією створюваного ігрового застосунку є люди, які:

а) вже знайомі або цікавляться жанром Souls-like і прагнуть отримати новий ігровий досвід;

б) віддають перевагу складним іграм із високим рівнем виклику, де результат залежить виключно від навичок гравця;

в) хочуть відчути атмосферу похмурого, містичного ігрового світу, в якому сюжет розкривається через середовище, підказки та спостереження;

г) мають можливість приділяти грі помірну кількість часу, але очікують від неї глибокого занурення;

д) не шукають швидких перемог, а натомість насолоджуються складним, але справедливим геймплеєм;

е) зацікавлені в інді-проєктах, де відчувається авторський підхід до дизайну, музики та побудови світу;

ж) перебувають у віковій категорії 16–40 років, як чоловіки, так і жінки;

и) готові долати труднощі, вчитись на помилках і поступово розвивати навички у грі.

Таким чином, застосунок спрямований не лише на вузьку нішу геймерів, а на широку аудиторію, яка прагне більш глибоких ігрових емоцій. Це дозволяє досягти високої залученості користувачів, сформувати активну спільноту навколо гри та забезпечити довготривалу актуальність проєкту.

1.3.2 Монетизація

Ефективна стратегія монетизації є ключовим аспектом успішного випуску будь-якого ігрового програмного застосунку. Враховуючи жанр гри та особливості цільової аудиторії, яка цінує якісний ігровий процес без нав’язливих обмежень, монетизація повинна бути ненав’язливою, прозорою та етичною. Метою є досягнення фінансової стабільності проєкту без погіршення користувацького досвіду.

Запропонована модель монетизації включає декілька напрямів:

а) пряма купівля гри (Premium Model) — користувачі одноразово сплачують фіксовану вартість за повний доступ до гри. Такий підхід відповідає очікуванням аудиторії Souls-like ігор, яка зазвичай надає перевагу повноцінному продукту без мікротранзакцій;

б) розширення контенту (DLC) — у майбутньому можуть бути додані платні доповнення у вигляді нових локацій, босів, зброї чи сюжетних розгалужень. Це дозволяє підтримувати інтерес гравців після завершення основної кампанії;

в) косметичні предмети — додаткові скін-паки для персонажа чи зброї, що не впливають на баланс гри, можуть бути запропоновані у вигляді опціональних покупок. Таким чином гравці отримують можливість персоналізувати свій досвід;

г) фізичні видання або мерч — у разі успішної популярності гри можлива реалізація брендованої продукції: футболок, постерів, артбуків, фігурок персонажів тощо;

д) платформи розповсюдження — гра буде поширюватись через Steam або аналогічні цифрові платформи, які забезпечують зручний спосіб оплати, систему відгуків, досягнень та інші функції для підтримки активної аудиторії.

Таким чином, монетизація базуватиметься не на штучних обмеженнях або pay-to-win механіках, а на створенні додаткової цінності для гравців. Це дозволить зберегти довіру спільноти, підтримувати якісний розвиток проєкту та забезпечити стійкий фінансовий прибуток.

1.4 Виявлення проблем

Після аналізу існуючих ігор у жанрі Souls-like можна виділити як переваги, так і недоліки кожного проєкту, що дозволяє визначити актуальні проблеми, які слід врахувати під час розробки нового ігрового застосунку.

а) плюси гри «Dark Souls III»:

1) висока складність та бойова система, яка мотивує гравця вивчати ворогів і вдосконалювати свої навички;

2) глибока міфологія та сюжет, що розкривається через навколишній світ і деталі;

3) атмосферний дизайн рівнів, де кожна локація візуально унікальна і логічно пов’язана з іншими.

б) мінуси гри «Dark Souls III»:

1) відсутність повноцінного навчального режиму або підказок для новачків, що створює поріг входу;

2) інтерфейс інвентаря перевантажений і незручний для швидкого керування під час бою;

3) недостатньо розвинена система локалізації для деяких мов, що ускладнює розуміння сюжету гравцями з інших регіонів.

в) плюси гри «Lies of P»:

1) візуально сучасна графіка з чіткою деталізацією оточення та персонажів;

2) гнучка система апгрейдів зброї та механік бою, яка дозволяє адаптувати стиль гри під себе;

3) більш дружній підхід до гравця через доступну навігацію по рівнях і зрозумілі місії.

г) мінуси гри «Lies of P»:

1) обмежена варіативність ворогів на початкових етапах гри;

2) деякі анімації бою виглядають повільними, що може призводити до дискомфорту в управлінні;

3) час завантаження між локаціями іноді надто довгий, що переривает ритм гри.

Крім порівняння існуючих ігор, було виявлено проблеми, які часто зустрічаються серед гравців жанру:

а) складність інтерфейсу для новачків, що знижує привабливість гри для ширшої аудиторії;

б) недостатня оптимізація продуктивності на середніх ПК, що робить гру недоступною для частини потенційних гравців;

в) відсутність внутрішньоігрових нагадувань або допомоги, яка могла б полегшити проходження новим користувачам;

г) перевантаженість HUD (інтерфейсу гравця) зайвою інформацією під час бою.

Виявлені недоліки стануть основою для формування вимог до функціональності нової гри. Метою є створення балансу між складністю, глибиною сюжету та зручністю використання, що дозволить залучити як досвідчених гравців, так і новачків.

2 ФОРМУВАННЯ ВИМОГ ДО ПРОГРАМНОЇ СИСТЕМИ

Підсистема рівнів у межах ігрового застосунку в жанрі Souls-like відповідає за побудову та структурування ігрових локацій [5], забезпечення атмосфери через звуковий супровід, освітлення, розміщення об'єктів та організацію геймплейної взаємодії в межах конкретного простору.

Функціональні вимоги:

а) підсистема має забезпечувати коректну роботу багаторівневої структури локацій у межах сценарію з лінійним просуванням гравця;

б) рівні мають містити контрольні точки (checkpoints) та логіку активації зон (trigger-зони);

в) підтримка інтеграції з аудіо- та візуальним оформленням через систему Blueprint;

г) звуковий супровід рівнів має змінюватися динамічно залежно від зони та подій (ambient, battle, special zones) [6];

д) всі елементи рівня повинні бути оптимізовані під різні графічні налаштування та зберігати візуальну цілісність на середніх і високих параметрах графіки;

е) реалізація освітлення повинна відповідати художньому стилю гри та підтримувати атмосферу відповідно до наративу.

Нефункціональні вимоги:

а) підсистема повинна бути оптимізована для стабільної роботи при роздільних здатностях від 1280×720 до 3840×2160;

б) час завантаження окремого рівня не повинен перевищувати 90 секунд при використанні SSD;

в) усі локації мають бути сумісні з системами Windows 10+ та Linux (ядро 5.0+), а також із рушієм Unreal Engine 5;

г) підтримка коректного відображення світла, тіней і післяобробки (post-processing) без зниження продуктивності на середніх конфігураціях ПК.

Технічні вимоги до апаратного забезпечення:

а) мінімальні:

1) Процесор: Intel Core i3-7100 або аналог;

2) ОЗП: 8 ГБ;

3) Відеокарта: NVIDIA GTX 750 Ti або аналог з підтримкою DirectX 11;

4) Диск: 20 ГБ (SSD бажано);

5) ОС: Windows 10 / Linux (KDE або GNOME).

Рекомендовані:

б) процесор: Intel Core i5-9600K або AMD Ryzen 5 3600;

1) ОЗП: 16 ГБ;

2) Відеокарта: NVIDIA GTX 1660 або AMD RX 590;

3) SSD-накопичувач обов'язковий;

4) ОС: Windows 11 або актуальна версія Linux.

3 АРХІТЕКТУРА ТА ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 UML проєктування ПЗ

Перед початком розробки набору рівнів у жанрі Souls-like було визначено основні функціональні сценарії взаємодії гравця з ігровим середовищем. Особливу увагу приділено тому, як гравець пересувається по рівнях, активує події, взаємодіє з об'єктами оточення, реагує на виклики та поступово розкриває атмосферу світу. На основі цього аналізу було побудовано діаграму прецедентів (Use-case діаграму), яка відображає ключові способи взаємодії гравця з ігровими елементами рівня (див. рис. 3.1).

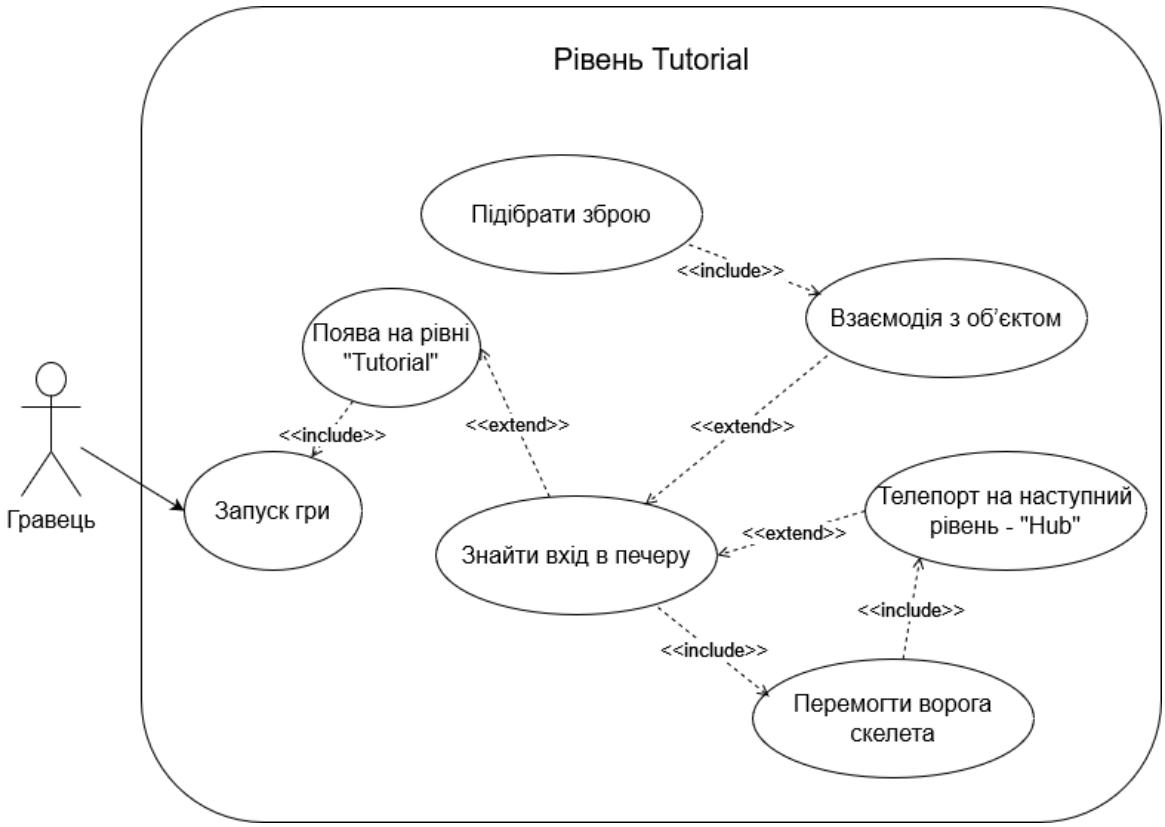


Рисунок 3.1 – Use-case діаграма взаємодії користувача з грою на рівні Tutorial (діаграма виконана самостійно)

У межах реалізованої підсистеми користувач (гравець) має змогу проходити туторіальний рівень, відкривати нові ігрові області, активувати

квести, перемагати босів, отримувати нагороди та взаємодіяти з NPC у хаб-локаціях. Кожна дія активує певні події у грі, що супроводжуються зміною стану гри, візуальними ефектами або звуковим супроводом.

Завдання у грі умовно поділені на основні та додаткові. Основні завдання відповідають за основну сюжетну лінію, тоді як додаткові — за розширення ігрового досвіду та опціональні винагороди.

Для забезпечення взаємодії між компонентами підсистеми та ігровим рушієм Unreal Engine використовуються візуальні скрипти Blueprint, які дозволяють реалізовувати логіку взаємодії, керування подіями, анімаціями, звуками та поведінкою об'єктів. Такий підхід дозволяє швидко створювати та тестувати механіки в межах конкретних рівнів, зосереджуючи увагу на геймдизайні, атмосфері та структурі оточення.

Одним із важливих елементів занурення в ігровий світ є звукове оформлення. Для досягнення атмосферності в грі реалізовано систему оточуючих звуків за допомогою Blueprint-логіки рушія Unreal Engine 5 [7].

Звуки, як-от тріск вогню, гул порталу або шелест вітру, активуються при вході гравця в спеціально створені зони з використанням Trigger Box. Для цього створено Blueprint-об’єкт, що реагує на подію входу гравця до зони. При вході запускається звук, який автоматично зупиняється при виході з тригера.

На рис. 3.2 показано фрагмент Blueprint-реалізації логіки відтворення звуку порталу.

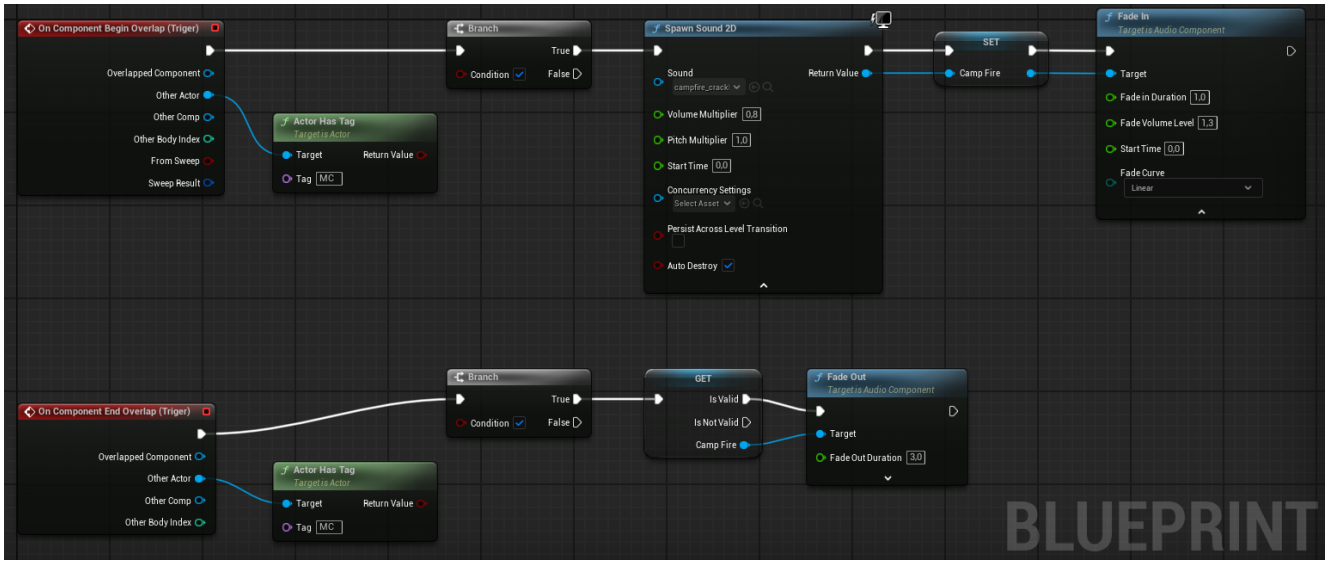


Рисунок 3.2 – Реалізація запуску звуку порталу при вході гравця у Trigger Box (виконано самостійно в Blueprint)

Аналогічна логіка використана для реалізації інших зон з унікальними оточуючими звуками, що дозволяє суттєво підвищити рівень занурення гравця в атмосферу гри. Такий підхід забезпечує гнучке налаштування і швидку інтеграцію аудіо-контенту безпосередньо з редактора.

3.2 Проєктування архітектури ПЗ

Архітектура програмного забезпечення гри є критично важливою для забезпечення стабільної роботи, масштабованості та логічної цілісності проєкту. У межах створення гри в жанрі Action RPG, піджанру Souls-like, архітектура ПЗ була спроєктована з урахуванням вимог до складного геймплею, нелінійного дизайну рівнів, системи завдань та адаптивної взаємодії з гравцем.

Базовою платформою для розробки виступає Unreal Engine 5, який надає потужні інструменти для створення інтерактивних середовищ та ігрових механік. Основна реалізація логіки здійснюється за допомогою системи візуального програмування Blueprints [8], що дозволяє гнучко проєктувати події, взаємодії та поведінку об’єктів. Такий підхід є особливо ефективним для

створення рівнів, де важлива швидка ітерація, художній контроль та точне налаштування атмосфери сцени.

Архітектура гри є модульною і включає такі основні компоненти [9]:

а) система рівнів (Level System) — відповідає за завантаження, перехід, ініціалізацію та налаштування середовищ гри. Кожен рівень представлений як окрема сцена з власними тригерами, параметрами освітлення та поведінкою об'єктів. Логіка переходу між рівнями реалізована на основі подій та перевірки виконаних завдань.

б) інтерфейс гравця (UI System) —включає спливаючі фрази при діалогах з NPC, а також підказки. UI оновлюється динамічно залежно від стану гравця і поточних подій.

в) NPC Controller — спеціальний клас, що керує поведінкою неігрових персонажів, у тому числі їхньою реакцією на події, діалогами, зміною поведінки після виконання певних завдань.

Архітектура побудована за принципами низького зв’язування та високої когезії, що дозволяє змінювати або вдосконалювати окремі компоненти гри без впливу на інші.

На рисунку 3.3 наведено загальну структурну схему архітектури підсистеми рівнів.

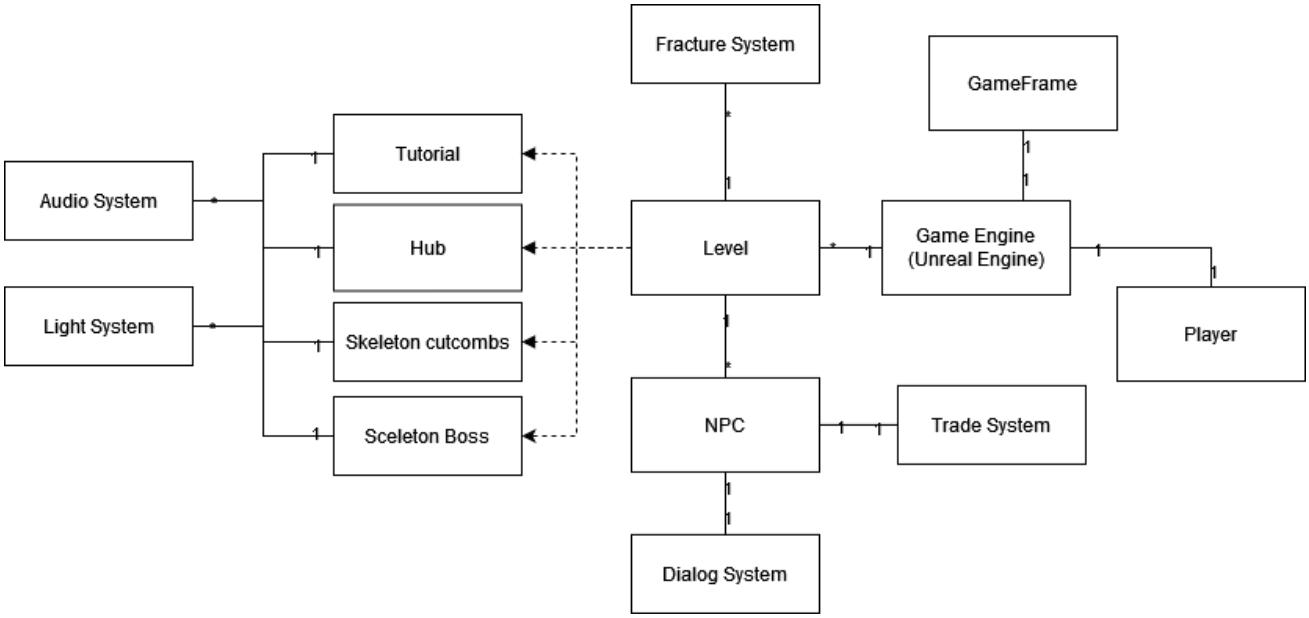


Рисунок 3.3 – Загальна архітектура підсистеми рівнів (виконано самостійно)

Завдяки обраному підходу архітектура ПЗ дозволяє забезпечити як стабільну роботу під час геймплею, так і зручність подальшого розширення функціональності. Це особливо важливо для реалізації рівнів і завдань, які відіграють ключову роль у побудові наративу гри та взаємодії з гравцем.

3.3 Реалізація ігрових подій

Однією з ключових подій у проєкті є перемога над босом, після чого активується портал, що дозволяє перейти до нового рівня. Для реалізації цієї логіки використовувалися Blueprint-и в Unreal Engine.

У Blueprint персонажа-боса було додано логіку, яка при досягненні нульового рівня здоров’я (0 HP), викликає подію активації порталу (див. рис. 3.4).

Активація порталу реалізована наступним чином:

а) портал за замовчуванням прихований та неактивний (Set Actor Hidden in Game, Disable Collision);

б) після події смерті боса виконується ланцюг нод:

1) Get Actor of Class → Портал;
2) Set Actor Hidden in Game → false;
3) Set Actor Enable Collision → true;
4) Play Sound at Location → запуск звуку активації порталу.

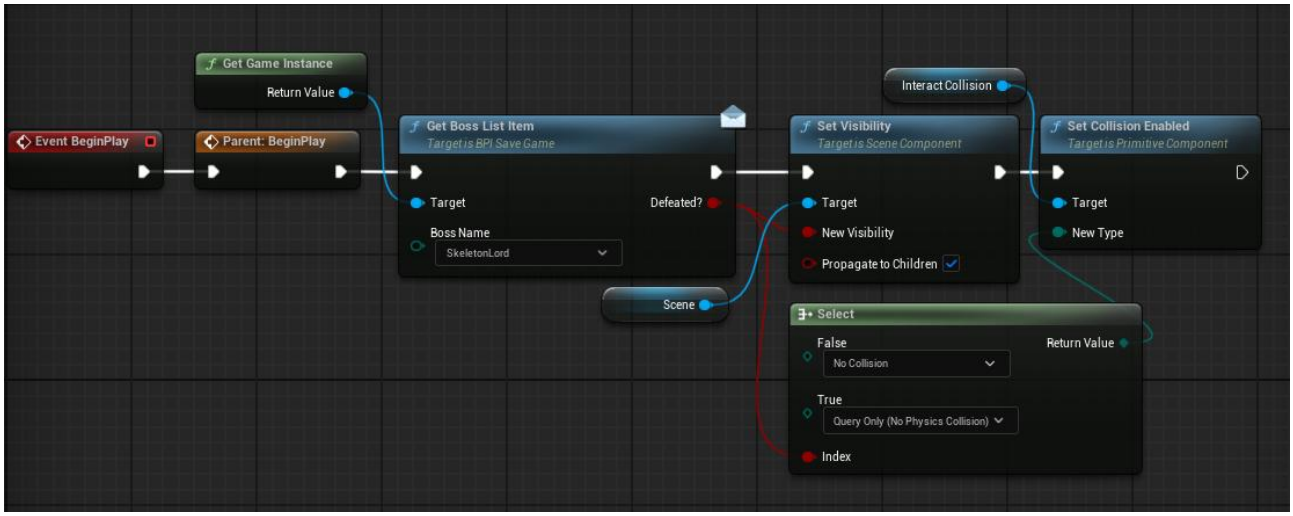


Рисунок 3.4 – Blueprint логіка активації порталу після смерті боса (виконано самостійно)

Цей підхід дозволяє реалізувати ігрову логіку виключно за допомогою візуального скриптингу в Blueprint-системі.

3.4 Приклади найцікавіших алгоритмів та методів

Під час розробки гри важливим етапом є створення та інтеграція підсистем, що забезпечують функціонування окремих рівнів. До таких підсистем належать: система звукового супроводу, механізми активації подій, логіка взаємодії з об’єктами середовища, перехід між рівнями тощо. Кожна з цих підсистем реалізується з урахуванням вимог до ігрового дизайну, технічної ефективності та зручності модифікації.

У цьому розділі розглянуто приклади реалізації окремих підсистем рівнів з використанням інструментів Unreal Engine, зокрема систем Blueprint.

3.4.1 Реалізація відтворення фонової музики в центральному хабі

Одним із важливих елементів реалізації підсистеми рівнів є система відтворення фонової музики в центральному хабі гри [10]. Її основна мета — забезпечення належного звукового супроводу, який сприяє формуванню відповідної атмосфери ігрового процесу. Для реалізації даного механізму були використані інструменти Sound Cue та Blueprint, що входять до складу рушія Unreal Engine.

У Blueprint-акторі, розміщеному в межах центрального хабу, було реалізовано логіку, яка реагує на появу гравця в певній зоні (див. рис. 3.5):

а) використано компонент Sphere Collision, який обробляє подію OnComponentBeginOverlap;

б) після виявлення гравця за допомогою вузла Cast To PersonCharacter відбувається запуск аудіокомпонента;

в) для уникнення монотонного звучання музика розпочинається з випадкового моменту за допомогою вузла Set Float Parameter → Random Float in Range.

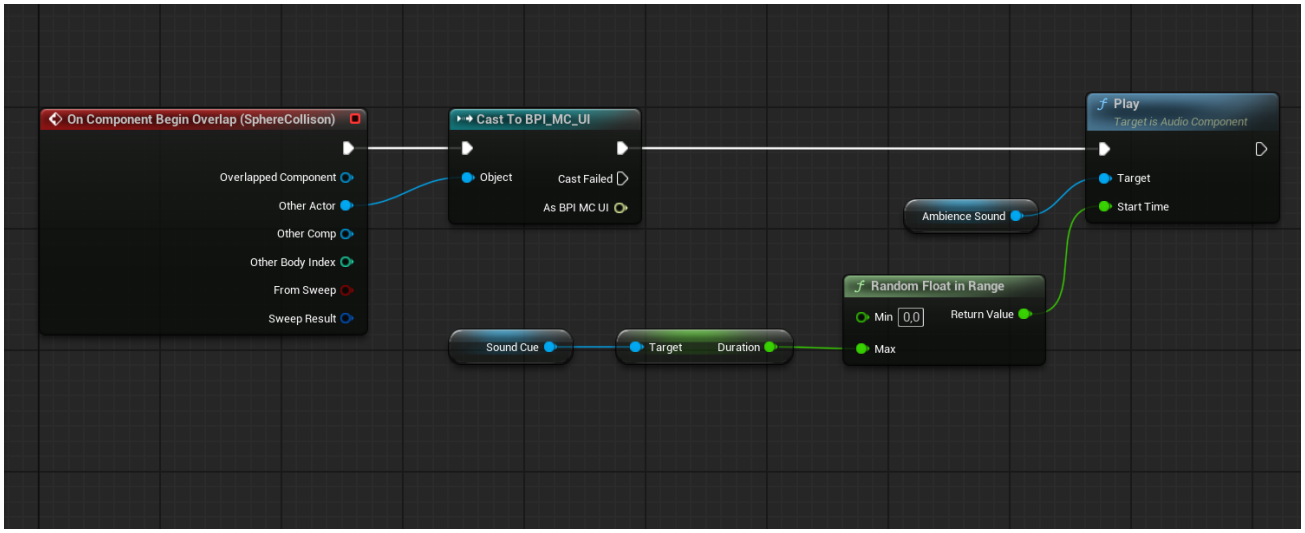


Рисунок 3.5 – Логіка в Blueprint для активації фонової музики під час входу гравця до зони (виконано самостійно)

Відтворення звуку здійснюється через систему Sound Cue, у якій реалізовано мікшування та випадковий вибір музичних фрагментів (див. рис. 3.6):

а) основні звукові доріжки об’єднуються за допомогою вузла Mixer;

б) інші треки вибираються випадковим чином за допомогою вузла Random;

в) результат передається через вихідний вузол Output до звукового компонента, який відповідає за відтворення.

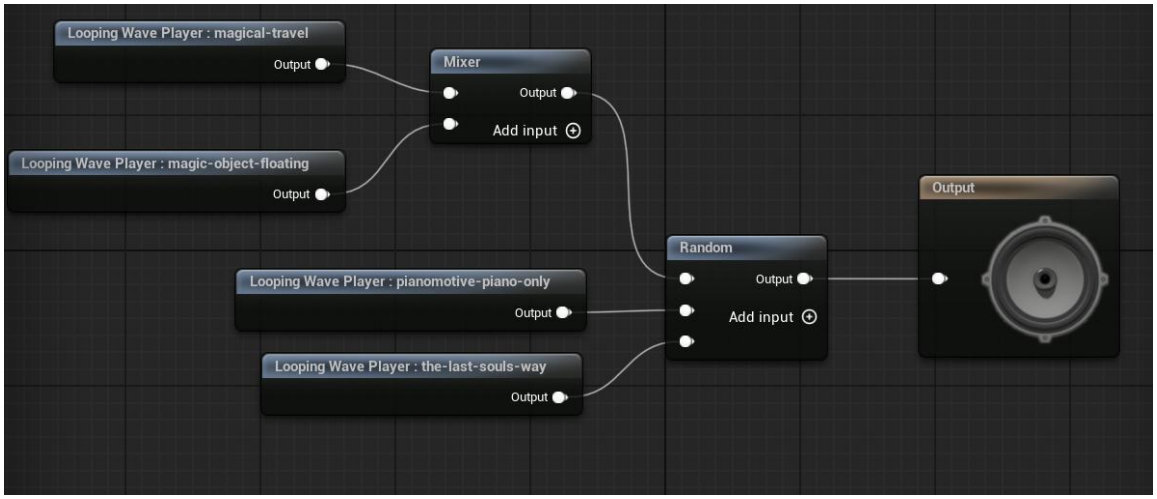


Рисунок 3.6 – Структура Sound Cue для реалізації фонової музики (виконано самостійно)

Завдяки такій реалізації забезпечується варіативність звучання та уникнення повторюваності, що позитивно впливає на загальне сприйняття ігрового середовища. Вся логіка реалізована засобами візуального скриптингу, що відповідає обраній архітектурі проєкту.

3.4.2 Реалізація системи діалогу з NPC

У межах розробки ігрової логіки було реалізовано систему озвученого діалогу з NPC, що активується при натисканні гравцем клавіші E у зоні взаємодії. Такий підхід дозволяє створити більш глибоке занурення в сюжетну складову

гри, зокрема, в даному випадку — мотиваційне звернення NPC до гравця із закликом спуститися в катакомби.

Логіка взаємодії зі сторони гравця реалізована у Blueprint-і персонажа наступним чином (див. рис. 3.7):

а) після натискання клавіші E здійснюється пошук усіх об’єктів, які перебувають у зоні перекриття (Get Overlapping Actors), із фільтрацією за класом SevorogNPC;

б) далі запускається цикл For Each Loop, у якому для кожного об’єкта перевіряється, чи реалізує він інтерфейс BPI Dialogue;

в) якщо інтерфейс реалізовано, викликається функція Talk, що ініціює діалог.

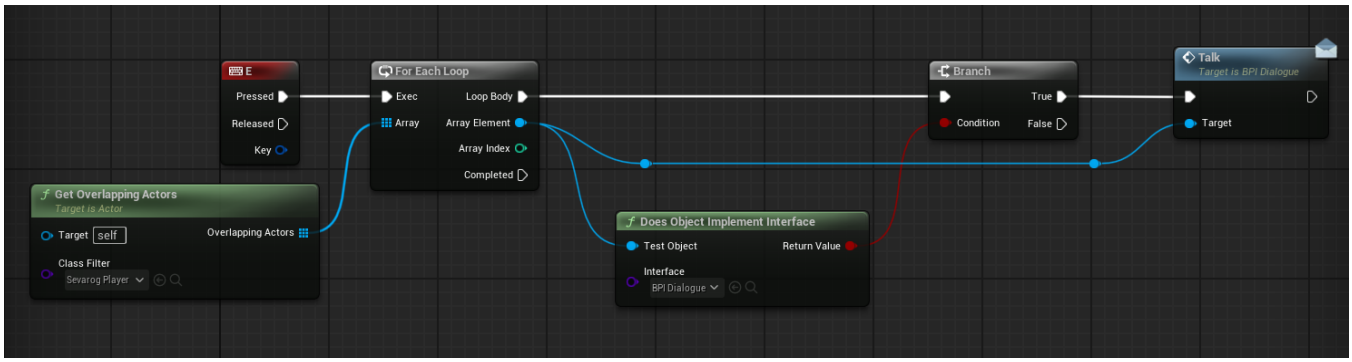


Рисунок 3.7 – Blueprint логіка запуску монологу NPC при натисканні клавіші E гравцем (виконано самостійно)

Логіка зі сторони NPC реалізована у вигляді функції Talk, що є частиною інтерфейсу BPI Dialogue. У Blueprint-і NPC передбачено такі змінні:

а) Dialogue – масив текстових фраз;

б) DialogueSounds – масив відповідних аудіофайлів для кожної фрази;

в) UI – посилання на елемент інтерфейсу, через який виводиться текст;

г) TalkIndex – числова змінна, що визначає поточну фразу для відтворення.

При виклику функції Talk виконується така послідовність дій (див. рис. 3.8 – 3.10):

а) перевірка наявності наступної фрази у масиві;

б) вивід відповідного тексту на екран через елемент UI;

в) програвання відповідного аудіофайлу з масиву DialogueSounds;

г) збільшення значення TalkIndex для переходу до наступної репліки під час наступної взаємодії.

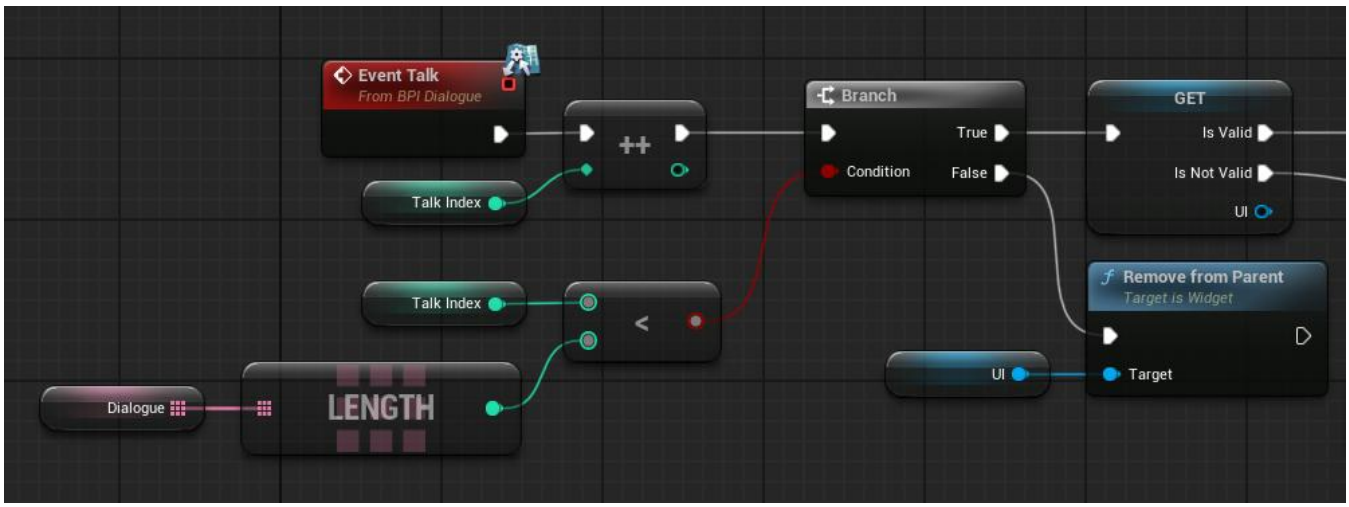


Рисунок 3.8 – Blueprint логіка NPC, що відповідає за програвання реплік та аудіо. Частина 1 (виконано самостійно)

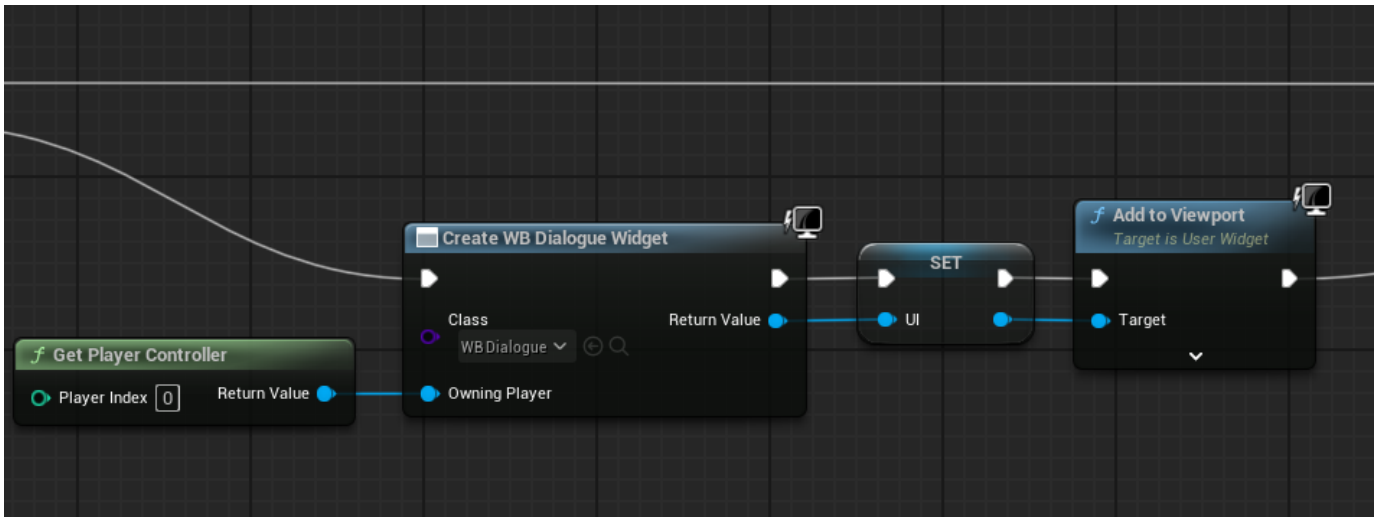


Рисунок 3.9 – Blueprint логіка NPC, що відповідає за програвання реплік та аудіо. Частина 2 (виконано самостійно)

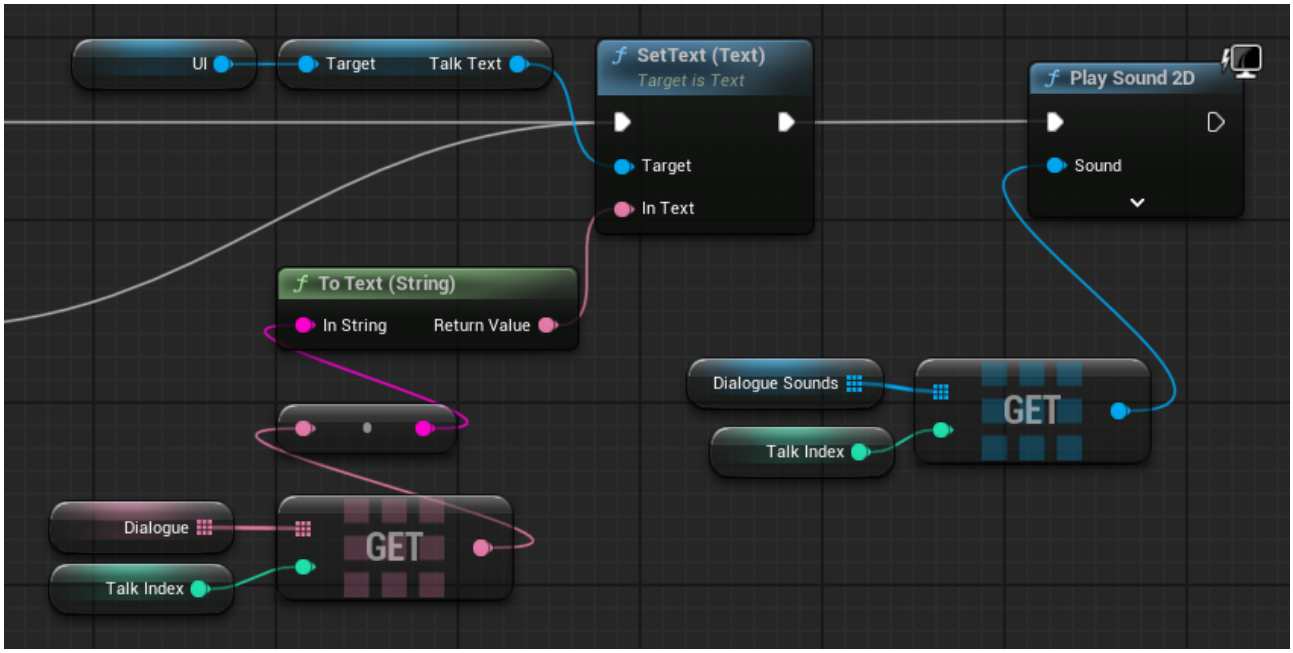


Рисунок 3.10 – Blueprint логіка NPC, що відповідає за програвання реплік та аудіо. Частина 3 (виконано самостійно)

Таким чином, реалізовано просту, але ефективну систему монологу, яка поєднує візуальний інтерфейс, аудіосупровід та взаємодію гравця з NPC. Це сприяє більш глибокому зануренню у світ гри та формує сюжетний зв’язок між гравцем та оточенням.

Кінцевий результат взаємодії можна побачити у візуальному інтерфейсі гри (див. рис. 3.11) : після натискання клавіші E на екрані з’являється відповідна текстова репліка NPC у вигляді діалогового вікна, супроводжена озвученням. Це дозволяє гравцеві зосередитись на змісті, не відволікаючись на читання тексту без голосового супроводу.

Рисунок 3.11 – Відображення монологу NPC під час гри (виконано самостійно)

Реалізована система діалогу є важливим елементом наративної складової гри, оскільки забезпечує структуровану та інтуїтивно зрозумілу взаємодію гравця з персонажами. Такий підхід дозволяє ефективно передавати сюжетну інформацію, підсилює емоційне сприйняття ігрових подій та забезпечує узгодженість між геймплеєм і візуально-звуковим оформленням.

3.5 Створення UI/UX

У межах проєкту було реалізовано базовий інтерфейс користувача, зосереджений на зручності взаємодії та мінімальному відволіканні гравця від геймплею. Елементом UI є система текстових підказок, яка активується натисканням клавіші H [11].

Принцип роботи:

а) при першому натисканні на клавішу H виводиться на екран блок текстової інформації з корисними підказками щодо керування персонажем або подальших дій.

б) повторне натискання H приховує підказки, щоб не перевантажувати інтерфейс зайвою інформацією.

в) підказки реалізовані за допомогою Blueprint Widget у Unreal Engine (див. рис. 3.12) [12];

г) логіка реалізована за допомогою умовного перемикача (FlipFlop), що дозволяє контролювати видимість віджета (див. рис. 3.13).

Рисунок 3.12 – Елемент UI текстові підказки (виконано самостійно)

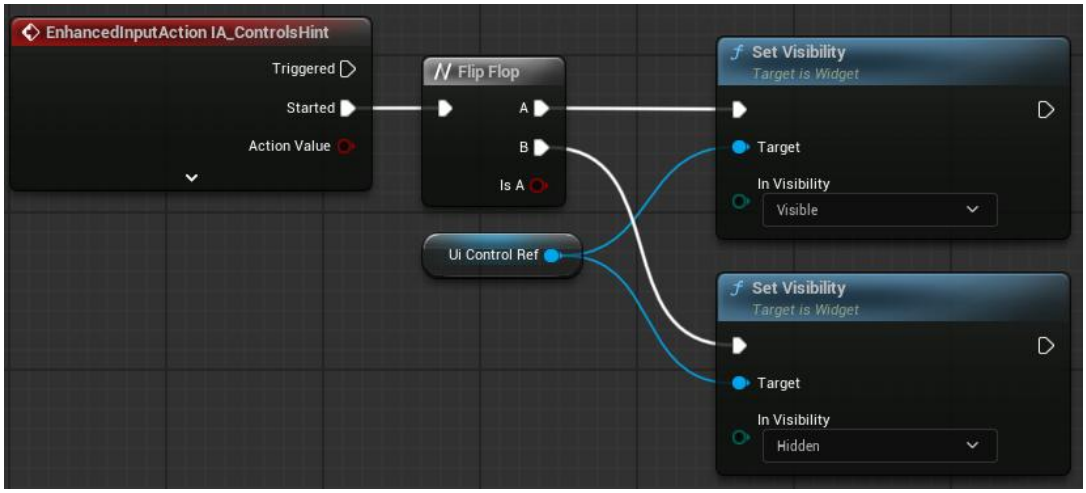


Рисунок 3.13 – Реалізація логіки підказок в Blueprint (виконано самостійно)

Такий підхід сприяє інтуїтивному UI, дозволяючи гравцеві самостійно вирішувати, коли отримувати додаткову інформацію, не порушуючи загальної динаміки гри.

4 ОПИС ПРИЙНЯТИХ ПРОГРАМНИХ РІШЕНЬ

4.1 Рівнева ізоляція через розділення на окремі мапи

У процесі розробки гри було прийнято рішення розміщувати кожен ігровий рівень на окремій карті (рівні) в Unreal Engine. Такий підхід забезпечує логічне та технічне розділення контенту, що спрощує структуру проєкту, полегшує налагодження та оптимізує процес редагування.

Кожна мапа відповідає окремому рівню гри і має власну сцену, освітлення, розміщення об’єктів та логіку взаємодії. Розділення рівнів на окремі карти має наступні переваги:

а) простота редагування — під час розробки можна працювати з одним рівнем, не завантажуючи зайвий контент, що пришвидшує відкриття редактора;

б) краща організація — кожен рівень зберігається в окремому файлі, що полегшує навігацію у проєкті та дозволяє краще структурувати роботу;

в) оптимізація продуктивності — завдяки завантаженню лише однієї мапи зменшується споживання пам’яті під час редагування чи тестування;

г) підготовка до системи переходу між рівнями — такий підхід дозволяє легко реалізувати механізм переходів між сценами, наприклад, через Blueprint-функцію Open Level.

Рівні зберігаються у відповідній структурі проєкту, а саме /All/Game/LevelPrototyping/Levels (див. рис. 4.1 та 4.2) та мають унікальні назви, які відображають зміст або послідовність проходження (LV_Tutorial, LV_Hub, LV_SkeletonCatacombs, LV_SkeletonBoss). Перехід між рівнями у грі реалізовано через портали, які викликають завантаження наступної карти і переносять гравця на неї.

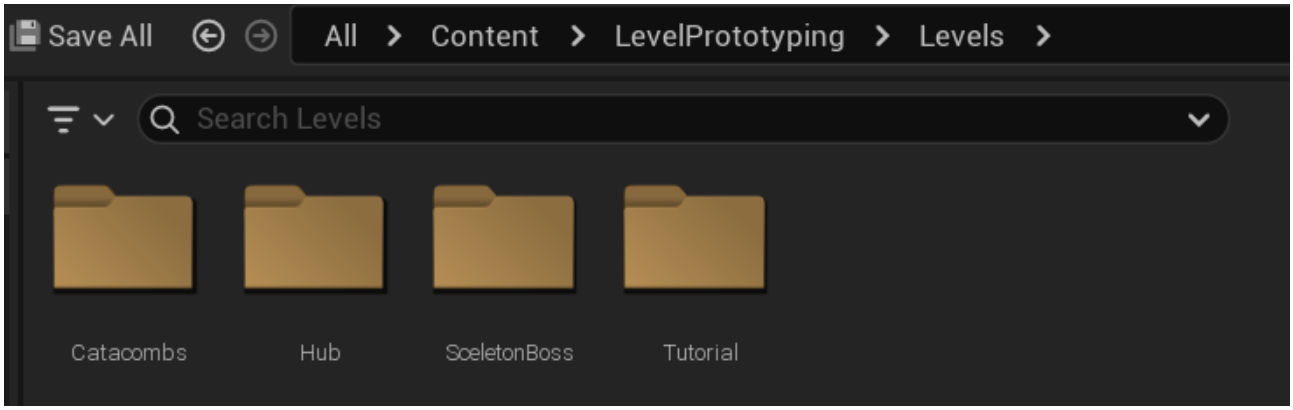


Рисунок 4.1 – Структура каталогів з рівнями в Content Browser Unreal Engine (виконано самостійно)

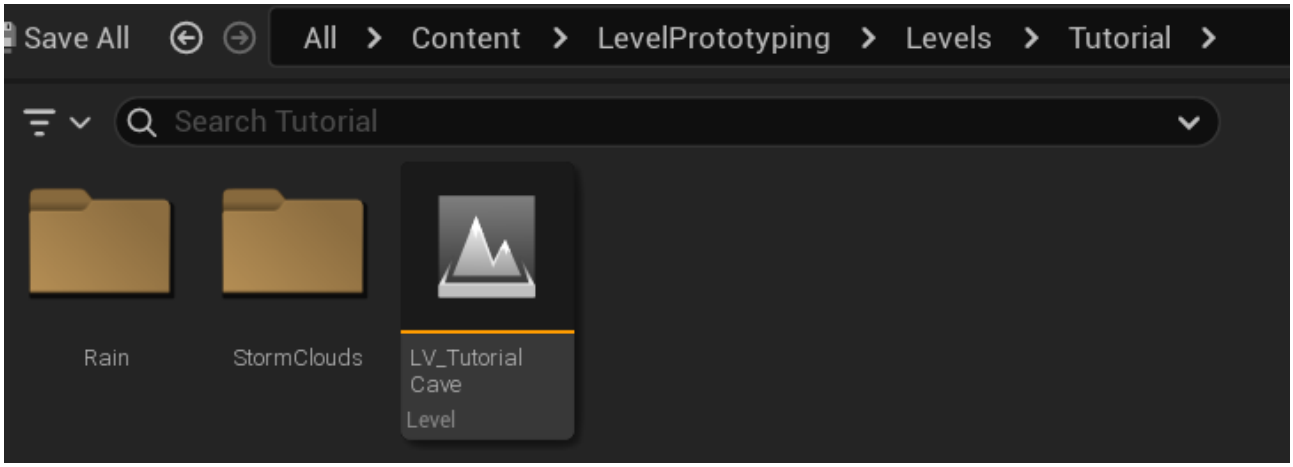


Рисунок 4.2 – Приклад змісту папки з певним рівнем(виконано самостійно)

Такий підхід до організації проєктної структури сприяє зручному управлінню контентом, спрощує навігацію під час розробки та забезпечує модульність у створенні й редагуванні рівнів. Розділення кожної локації на окрему мапу також дозволяє оптимізувати процеси завантаження, тестування та подальшого масштабування гри.

4.2 Реалізація візуального оформлення та освітлення

З огляду на обраний жанр та атмосферу гри, одним із ключових рішень при розробці стало створення гнітючої, депресивної обстановки з використанням

відповідного освітлення, кольорової гами та погодних умов. Гра орієнтована на ПК-платформу, тому реалізація візуальних ефектів виконувалася з акцентом на якість графіки без обмежень, притаманних мобільним платформам.

Хаб-рівень (див.рис. 4.3) - це умовно безпечна зона під великим куполом. Основним елементом інтерфейсу навігації між рівнями є портали, кожен з яких веде на окрему мапу. Освітлення хабу – приглушене, з акцентом на портали, які мають легке світіння, що візуально підкреслює їхню важливість.

Рисунок 4.3 – Атмосфера в рівні Hub (виконано самостійно)

Перший рівень гри – катакомби (див.рис. 4.4). Він побудований як заплутаний лабіринт із системою вузьких проходів, які легко змушують гравця втратити орієнтацію.

Рисунок 4.4 - Візуалізація першого рівня-катакомб із туманом та факелами (виконано самостійно)

Загальна атмосфера підтримується такими засобами:

а) освітленням: використано точкові джерела світла з анімацією вогню для імітації тьмяного освітлення від факелів;

б) кольорами: у різних частинах катакомб встановлено джерела синього, зеленого та червоного світла, що викликають почуття тривоги та напруженості (див. рис. 4.5);

Рисунок 4.5 - Атмосферне кольорове освітлення (виконано самостійно)

в) туманом: використано Exponential Height Fog з затемненням у нижніх шарах, що обмежує видимість та посилює страх невідомості;

г) погодними ефектами: у вступній сцені та на рівні туторіала реалізовано дощ, грім та майже повну темряву, що задає тон усьому проходженню (див. рис. 4.6 та 4.7)

Рисунок 4.6 – Рівень-туторіал з дощем, блискавкою та затемненням неба (виконано самостійно)

Рисунок 4.7 – Вступна сцена [13]

Усі ці елементи були реалізовані за допомогою стандартних інструментів Unreal Engine: Post Process Volume, Point Light, Fog, Niagara для ефекту дощу, а також Sound Cue, що додають фонові звуки і музику на фон задля глибини та атмосфери.

4.3 Рівень з фінальним босом

Один із ключових моментів геймплею – це фінальна битва з босом. Для цього було створено окремий рівень з особливою атмосферою та геометрією [14]. Арена розміщена в просторій печері з круглим вівтарем посередині, що виконує роль місця битви. Вода у центрі створює ефект дзеркального відображення зоряного неба, а бруківка навколо неї формує зони для маневрування гравця під час бою (див. рис. 4.8).

Рисунок 4.8 – Арена для бою з босом у фінальному рівні (виконано самостійно)

Для підсилення атмосфери було застосовано спеціальне освітлення: злегка синє та помаранчеве світло факелів, а також частинки світла над водою, які створюють містичне враження. У цьому рівні не лише зростає складність, але й

наративна напруга, оскільки битва з босом слугує фінальним випробуванням для гравця.

Важливу роль у створенні емоційної напруги відіграє музичний супровід. Під час бою з босом звучить напружена, динамічна композиція, що створює відчуття небезпеки. Після перемоги над босом музика змінюється на спокійну, що символізує завершення конфлікту та емоційне полегшення [15].

5 ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Тестування програмного забезпечення здійснювалося на всіх етапах розробки з метою виявлення та усунення дефектів, що могли вплинути на стабільність роботи гри, ігровий досвід користувача та коректність взаємодії між елементами сцени. Основну увагу було зосереджено на тестуванні функціональних компонентів, зокрема системи колізій, освітлення, звукових тригерів, а також взаємодії ігрових об’єктів після проведення геометричних операцій типу Boolean.

Тестування проводилось вручну без застосування автоматизованих інструментів, безпосередньо у середовищі Unreal Engine 5 у режимі попереднього перегляду (Play-in-Editor).

5.1 Перевірка системи колізій

Після об'єднання або обрізання об'єктів за допомогою Boolean-операцій [16] деякі з них втрачали коректні фізичні межі. Було здійснено перевірку та коригування:

а) фізичних колізій статичних об'єктів (стіни, колони, підлога, меблі);

б) усунення невидимих бар’єрів або дефектів зіткнення, що перешкоджали руху гравця (див. рис. 5.1).

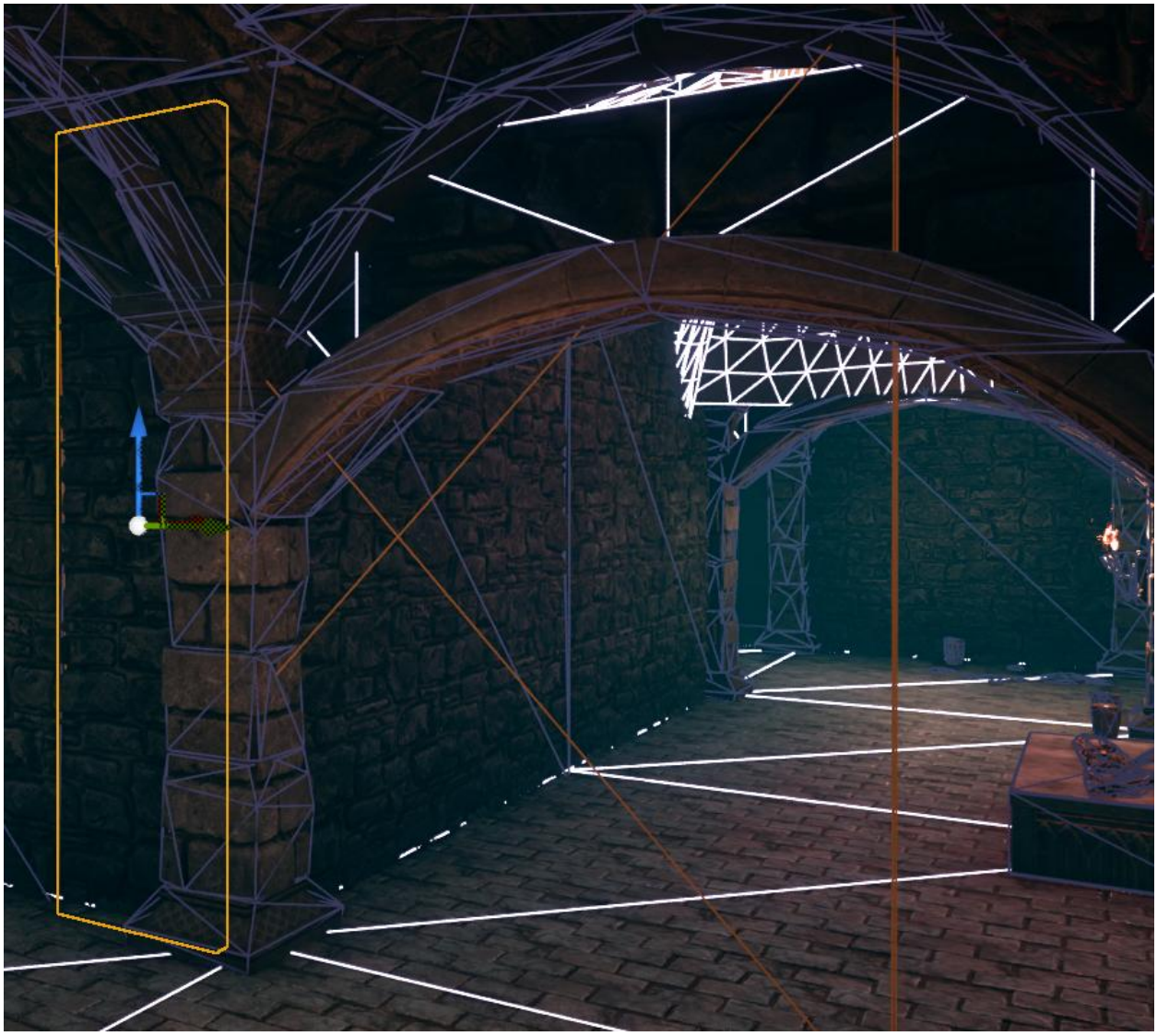


Рисунок 5.1 – Приклад некоректної колізії стіни: жовтим позначено фактичні межі, помаранчевим – колізію, яка блокує прохід гравця (виконано самостійно)

У проблемних випадках було застосовано ручне налаштування Collision Complexity [17], де обиралось Use Complex Collision as Simple. Приклад зображено на рисунку 5.2.

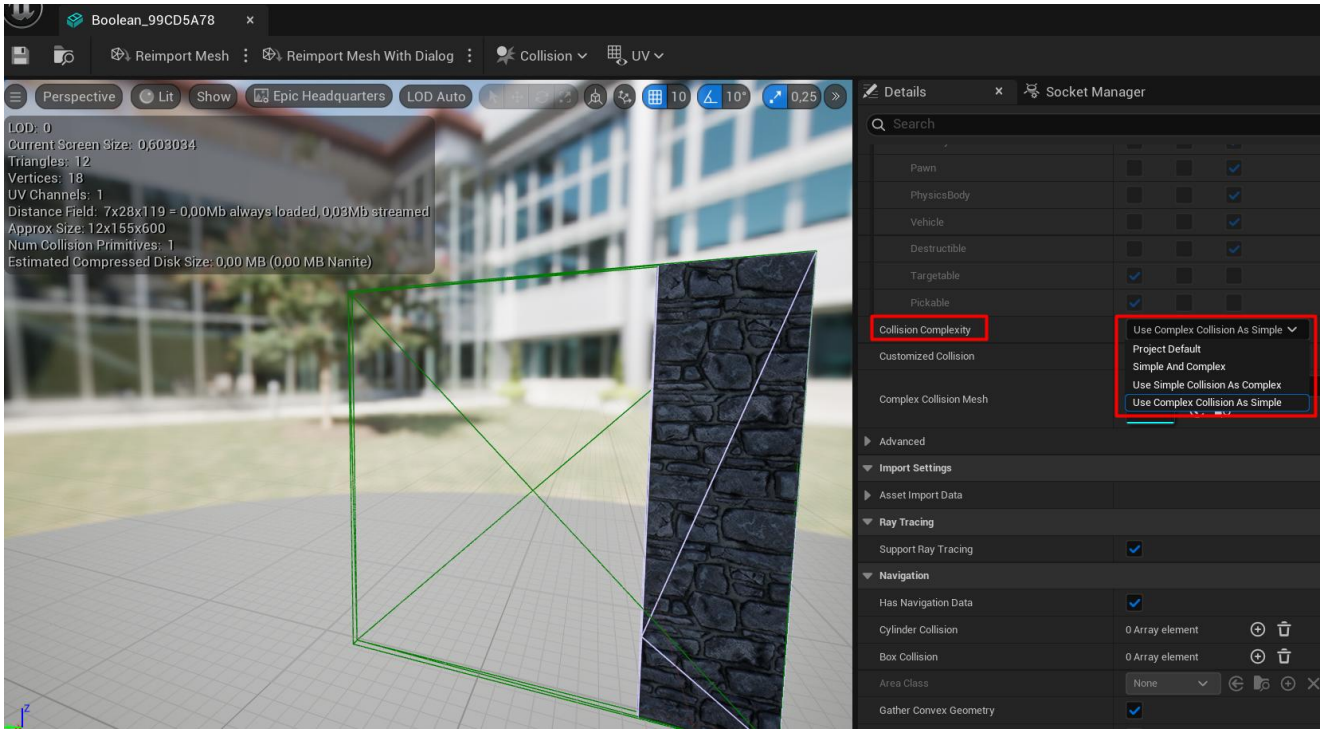


Рисунок 5.2 – Візуалізація оновленої колізії: стара колізія позначена зеленим кольором (виконано самостійно)

Застосування точного налаштування колізій дозволило усунути помилки зіткнення, що впливали на ігровий процес, та забезпечити коректну взаємодію гравця з оточенням. Це є критично важливим для уникнення непередбачуваної поведінки персонажа під час пересування рівнями й підвищення загальної стабільності гри.

5.2 Тестування освітлення

Світлотехнічне тестування охоплювало детальну перевірку якості та коректності роботи системи освітлення у різних умовах і сценах гри, з особливим акцентом на створення емоційного візуального настрою та відповідності технічним вимогам продуктивності. Тестування включало:

а) перевірку розташування та типів джерел світла: ретельно протестовано основні типи освітлення — Point Light, Spot Light (див. рис. 5.3), та Directional Light. Особливу увагу приділено коректності позиціонування джерел світла у тривимірному просторі, їх впливу на

навколишнє середовище, а також зменшенню кількості непотрібних перетинів або засвічення.

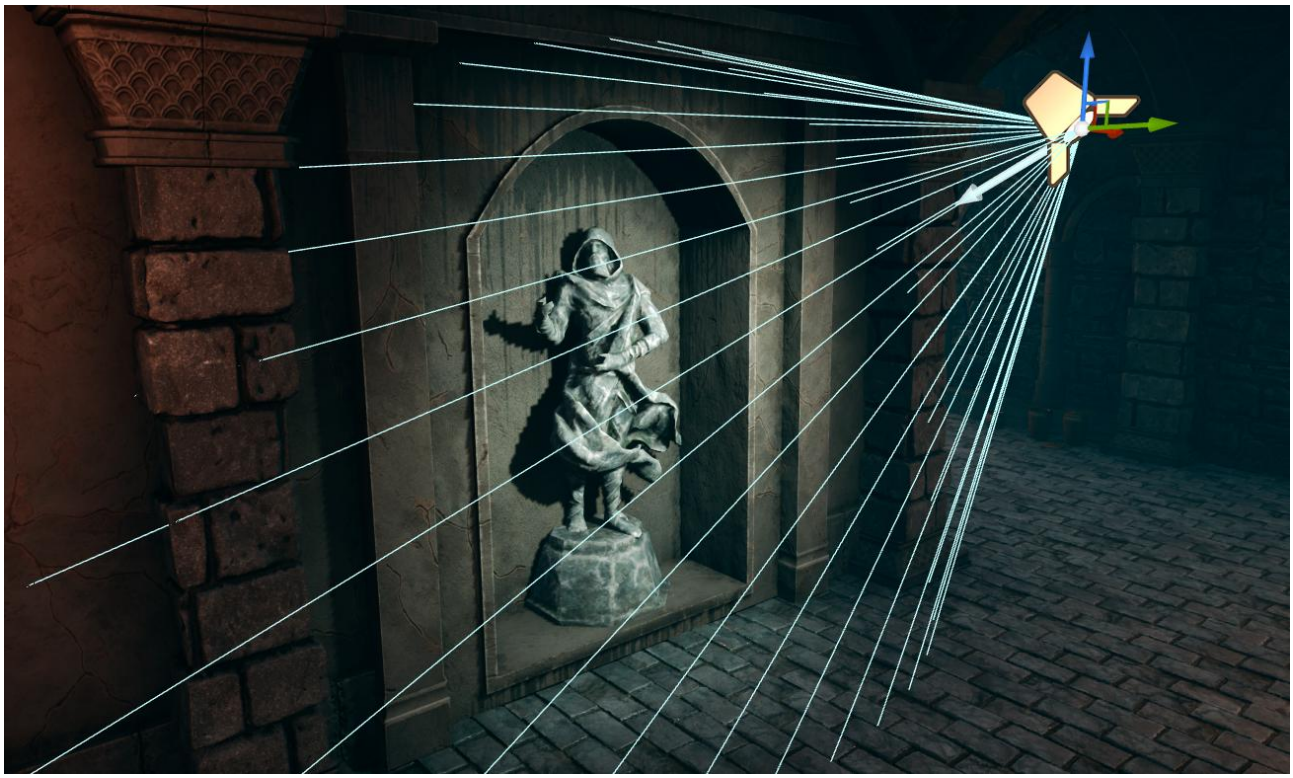


Рисунок 5.3 – Приклад використання Spot Light (виконано самостійно)

б) налаштування колірних температур: перевірено відповідність освітлення атмосфері сцени за допомогою різних температур світла (від теплого жовтуватого до холодного блакитного). Наприклад, тепле освітлення використовувалось у хаб-зонах для створення відчуття безпеки, тоді як холодне світло застосовувалось у катакомбах, підземеллях чи лабораторіях для формування напруженої або тривожної атмосфери.

Поєднання освітлення з візуальними ефектами: протестовано інтеграцію світлових джерел з FX-елементами, зокрема: м’яке світло від факелів і свічок, об’ємне світло в тумані, підсвітка магічних частинок біля порталів (див. рис. 5.4) тощо. Це дозволило підсилити враження присутності в ігровому просторі та глибше передати емоційний стан локацій.

Рисунок 5.4 – Підсвітдка магічних частинок біля порталу (виконано самостійно)

Освітлення у проєкті налаштовувалось вручну для кожної сцени відповідно до її художнього задуму, із використанням постобробки (Post Process Volume) для додаткової корекції кольору, контрасту та ефектів. У результаті вдалося досягти візуального стилю, що поєднує гнітючу атмосферу з естетично привабливим виглядом, не перевантажуючи систему.

5.3 Тестування звукового супроводу

Особливу увагу під час тестування було приділено звуковим тригерам та динамічній зміні музичного супроводу відповідно до стану гри, з метою забезпечення повного занурення гравця в атмосферу ігрового світу. Проведено наступні види тестування:

а) коректне спрацювання звукових ефектів: протестовано різноманітні звуки навколишнього середовища, зокрема звук дощу, вітру, горіння

факелів, скрипу дверей тощо. Перевірено їх активацію у відповідні моменти та відповідність візуальному контексту сцени.

б) 3D-позиціювання звуків у просторі: перевірено коректність локалізації джерел звуку відносно позиції гравця. Звуки змінювали гучність та панораму відповідно до переміщення гравця по рівню, що створює ефект просторової присутності. Особливо тестувались звуки, пов'язані з ворогами, NPC, водоспадами, вогнищами тощо.

в) плавність зміни музичного супроводу залежно від стану гри: реалізовано та протестовано сценарії, у яких змінюється фоновий аудіотрек на основі ігрових подій (наприклад, перехід від спокійної мелодії до напруженої в момент бою з босом, і навпаки — після перемоги). Перевірялась відсутність різких переходів між композиціями, застосовувались кросфейди та згасання (fade in / fade out) для досягнення плавного аудіо-досвіду.

Звукові ефекти були інтегровані за допомогою Audio Components, прив’язаних до об’єктів сцени. Активація звуків здійснювалась через Trigger Box [18] у поєднанні з Blueprint Logic, яка відповідала за визначення контексту та стану гри. У складних випадках використовувались Sound Cue для створення складних аудіо-сценаріїв із випадковими або накладеними звуками.

Результатом тестування стало виявлення та усунення незначних проблем із несвоєчасною активацією окремих ефектів, а також вдосконалення переходів між музичними темами, що значно покращило загальне аудіо-сприйняття гри.

ВИСНОВКИ

У межах виконання кваліфікаційної роботи бакалавра було виконано важливу частину роботи над ігровим застосунком, зосереджену переважно на розробці рівнів, реалізації подій, створенні UI-підказок та організації звукового супроводу.

На основі поставленого технічного завдання було реалізовано ряд ігрових сцен з унікальними механіками та логікою. Застосовуючи можливості візуального програмування через Blueprint-систему Unreal Engine, вдалося налаштувати взаємодію гравця з ігровим середовищем, зокрема — події при вході в зону тригера, активацію звуків та відкриття нових порталів після проходження босів.

Окрему увагу приділено створенню елементів користувацького інтерфейсу. Було реалізовано систему підказок, які з’являються при натисканні клавіші H та можуть бути приховані повторним натисканням. Це покращує взаємодію користувача з ігровим середовищем та підвищує зручність проходження рівнів.

Таким чином, внесок у розробку проєкту полягав у створенні логіки ігрових рівнів, інтеграції UI-елементів, додаванні звукових ефектів та проєктуванні подій, які впливають на атмосферу та динаміку гри. Отримані знання та практичний досвід стали важливим етапом професійного розвитку у сфері геймдизайну та технічної реалізації ігрових подій.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Steam Community. Посібник DARK SOULS™: maps [Електронний ресурс] – URL: https://steamcommunity.com/sharedfiles/filedetails/?l=ukrainian&id=3094082097 (дата звернення: 09.05.2025)

2. Unreal Engine. Blueprint Tutorials in Unreal Engine [Електронний ресурс] – URL: https://dev.epicgames.com/documentation/en-us/unreal-engine/blueprint-tutorials-in-unreal-engine (дата звернення: 18.05.2025)

3. Guzsvinecz T. The Soulsification of Video Games [Електронний ресурс] // Multimedia Tools and Applications. – 2024. – URL: https://link.springer.com/article/10.1007/s11042-024-19628-4 (дата звернення: 06.06.2025)

4. Unreal Engine. Community Tutorial: Level Design Fundamentals [Електронний ресурс] – URL: https://dev.epicgames.com/community/learning/tutorials/3VKJ/unreal-engine-fortnite-level-design-fundamentals (дата звернення: 02.06.2025)

5. Unreal Engine. Level Designer Quick Start in Unreal Engine [Електронний ресурс] – URL: https://dev.epicgames.com/documentation/en-us/unreal-engine/level-designer-quick-start-in-unreal-engine (дата звернення: 21.05.2025)

6. Unreal Engine. Sound Design: What do you want to know? [Електронний ресурс] – URL: https://forums.unrealengine.com/t/sound-design-what-do-you-want-to-know/131707 (дата звернення: 01.06.2025)

7. Unreal Engine. Working with Audio [Електронний ресурс] – URL: https://dev.epicgames.com/documentation/en-us/unreal-engine/working-with-audio-in-unreal-engine (дата звернення: 22.05.2025)

8. Unreal Engine. Quick Start Guide for Blueprints Visual Scripting in Unreal Engine [Електронний ресурс] – URL:

https://dev.epicgames.com/documentation/en-us/unreal-engine/quick-start-guide-for-blueprints-visual-scripting-in-unreal-engine (дата звернення: 18.05.2025)

9. Unreal Engine. Community Tutorial: How to create Modular and Scalable UI Systems? [Електронний ресурс] – URL: https://dev.epicgames.com/community/learning/tutorials/rmv5/unreal-engine-how-to-create-modular-and-scalable-ui-systems (дата звернення: 18.05.2025)

10. Unreal Engine. Tutorial: Begin Play | Audio [Електронний ресурс] – URL: https://dev.epicgames.com/community/learning/tutorials/0ODw/unreal-engine-begin-play-audio (дата звернення: 01.06.2025)

11. Unreal Engine. Building Your UI in Unreal Engine [Електронний ресурс] – URL: https://dev.epicgames.com/documentation/en-us/unreal-engine/building-your-ui-in-unreal-engine (дата звернення: 21.05.2025)

12. Unreal Engine. Community Tutorial: Card widget UI [Електронний ресурс] – URL: https://dev.epicgames.com/community/learning/tutorials/r207/unreal-engine-card-widget-ui (дата звернення: 22.05.2025)

13. YouTube. Stormy Ocean - Blender 3.1 [Електронний ресурс] – URL: https://www.youtube.com/watch?v=WGBxC7khWLM (дата звернення: 08.06.2025)

14. Unreal Engine. Introduction to Level Design [Електронний ресурс] – URL: https://www.unrealengine.com/en-US/blog/introduction-to-level-design (дата звернення: 29.05.2025)

15. Unreal Engine. Ambient and Procedural Sound Design [Електронний ресурс] – URL: https://dev.epicgames.com/community/learning/courses/qR/unreal-engine-ambient-and-procedural-sound-design (дата звернення: 24.05.2025)

16. Unreal Engine. Boolean Tool In Unreal Engine [Електронний ресурс] – URL: https://dev.epicgames.com/documentation/en-us/unreal-engine/boolean-tool-in-unreal-engine (дата звернення: 8.06.2025)

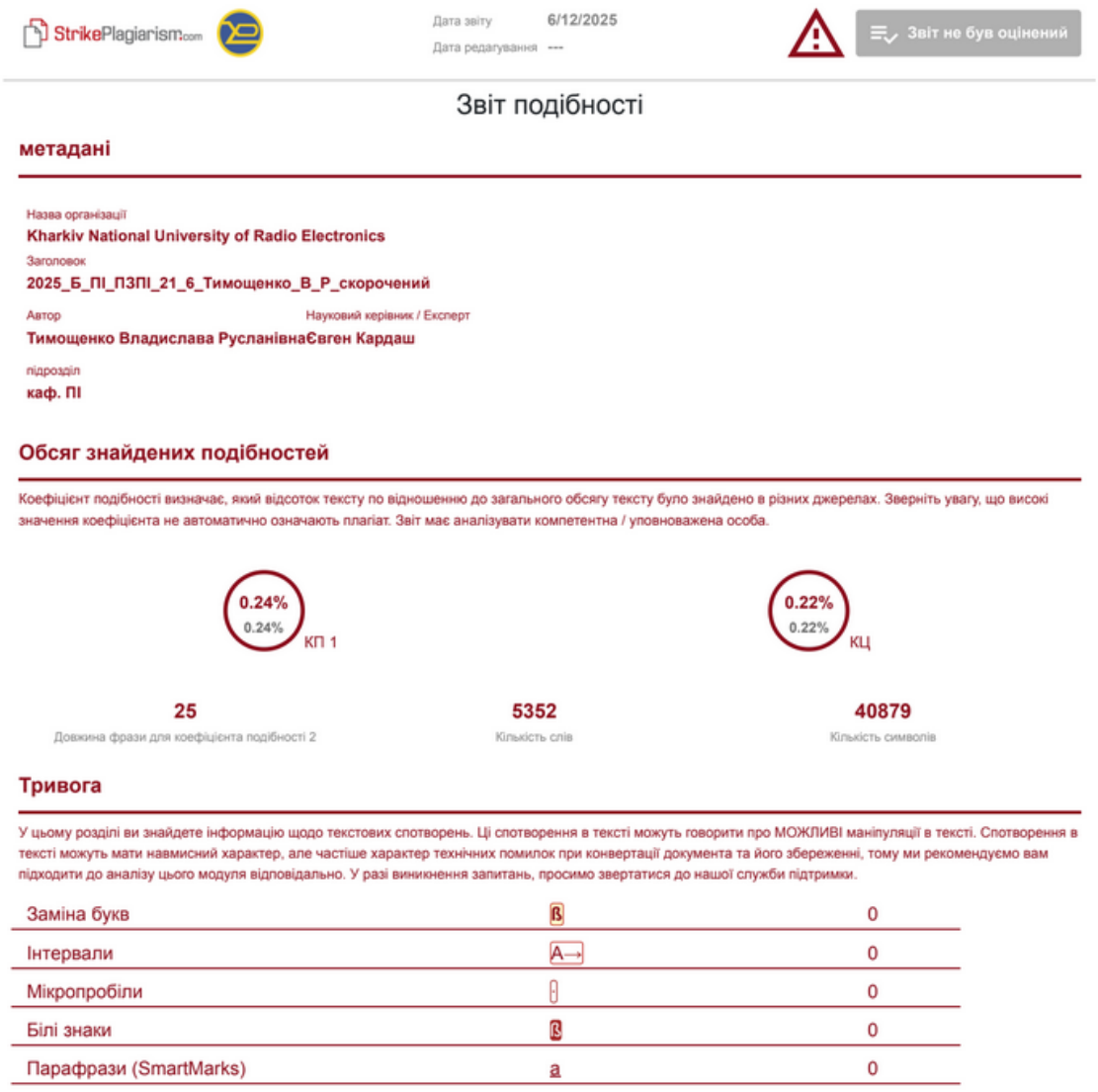
17. Unreal Engine. Simple versus Complex Collision [Електронний ресурс] – URL: https://dev.epicgames.com/documentation/en-us/unreal-engine/simple-versus-complex-collision-in-unreal-engine (дата звернення: 24.05.2025)

18. Unreal Engine. Course: Implementing UI for Level Design [Електронний ресурс] – URL: https://dev.epicgames.com/community/learning/tutorials/KKdD/how-to-use-trigger-box-unreal-engine-5-tutorial? (дата звернення: 9.06.2025)

19. GitHub Репозиторій з роботою. GitHub. URL: https://github.com/NureTymoshchenkoVladyslava/2025_B_PI_PZPI-21-6_Tymoshchenko_V_R (дата звернення: 17.06.2025)

ДОДАТОК А

Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ



StrikePlagiarism.com

Дата звіту 6/12/2025
Дата редагування ---

Звіт не був оцінений

Звіт подібності

метадані

Назва організації
Kharkiv National University of Radio Electronics
Заголовок
2025_Б_ПІ_ПЗПІ_21_6_Тимощенко_В_Р_скорочений
Автор
Тимощенко Владислава Русланівна
Науковий керівник / Експерт
Євген Кардаш
підрозділ
каф. ПІ

Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.

0.24% 0.24% КП 1

0.22% 0.22% КЦ

25	5352	40879
Довжина фрази для коефіцієнта подібності 2	Кількість слів	Кількість символів

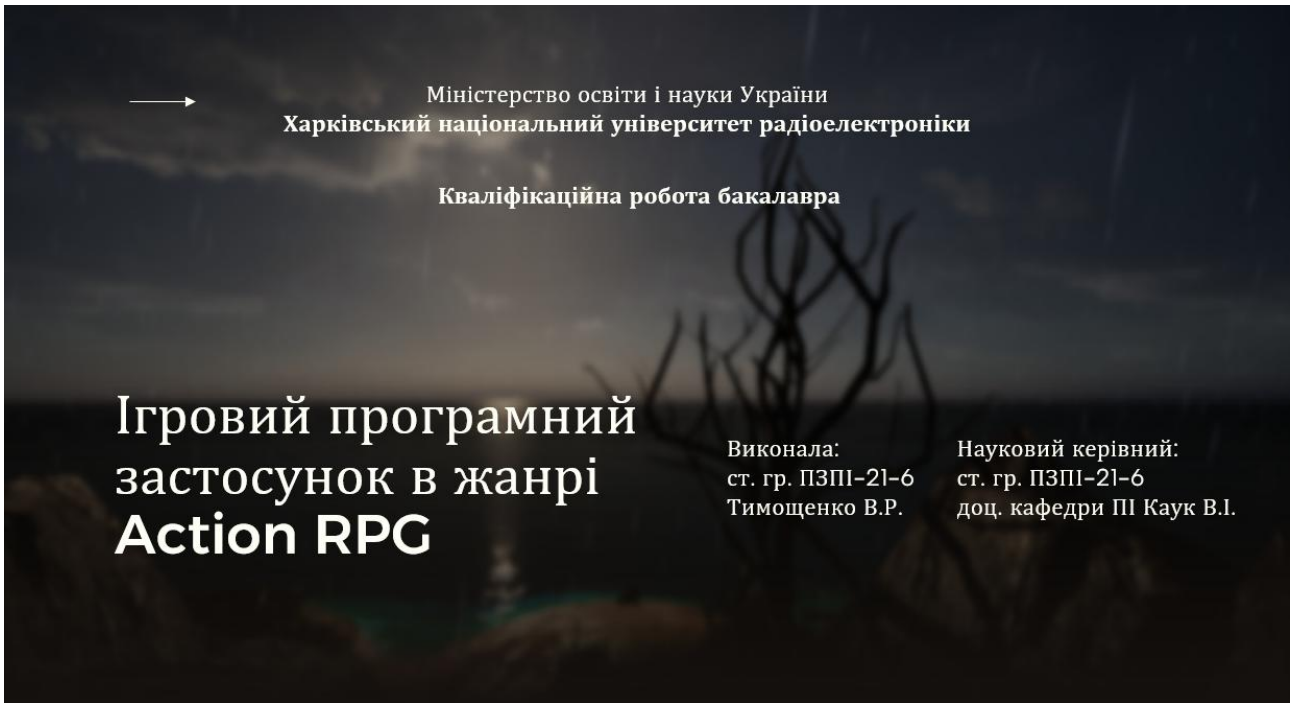
Тривога

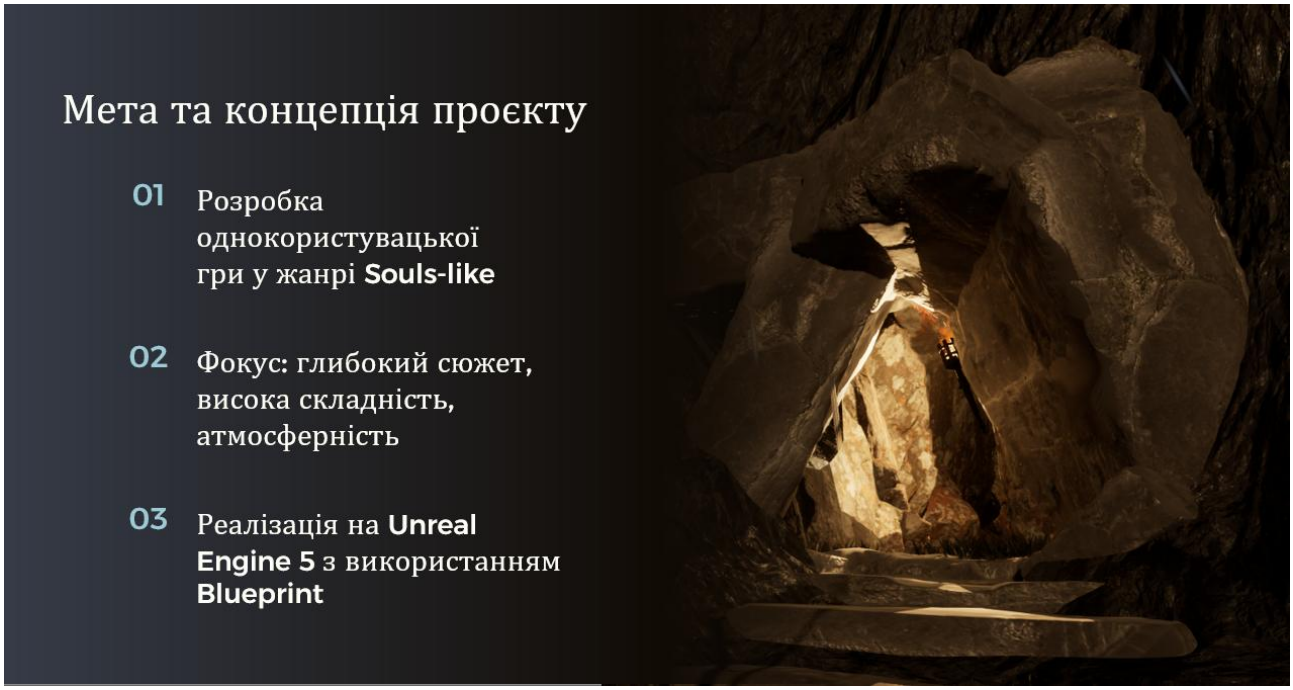
У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про МОЖЛИВІ маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

Заміна букв	ß	0
Інтервали	A⟶	0
Мікропробіли		0
Білі знаки	ß	0
Парафрази (SmartMarks)	a	0

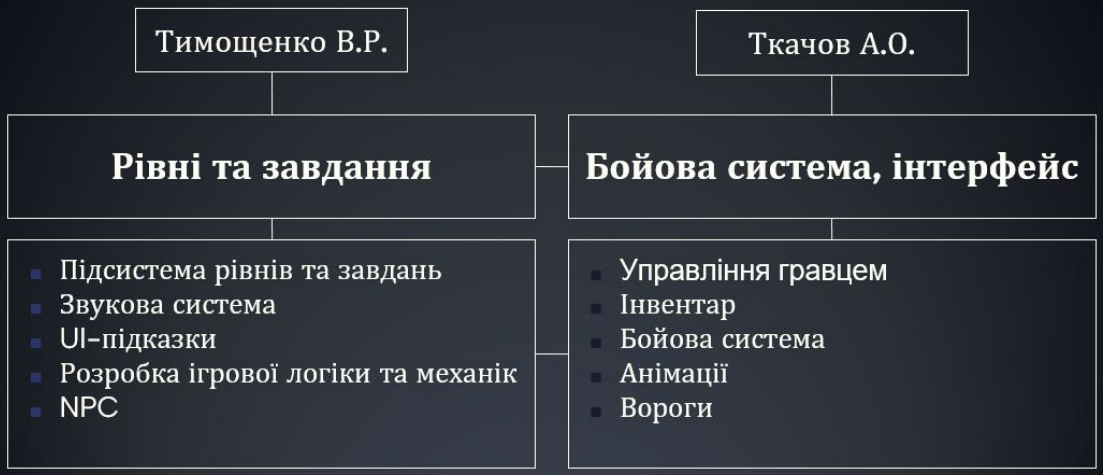
ДОДАТОК Б

Слайди презентації





Командна розробка – розподіл підсистем



Актуальність проєкту

Популярність жанру Souls-like

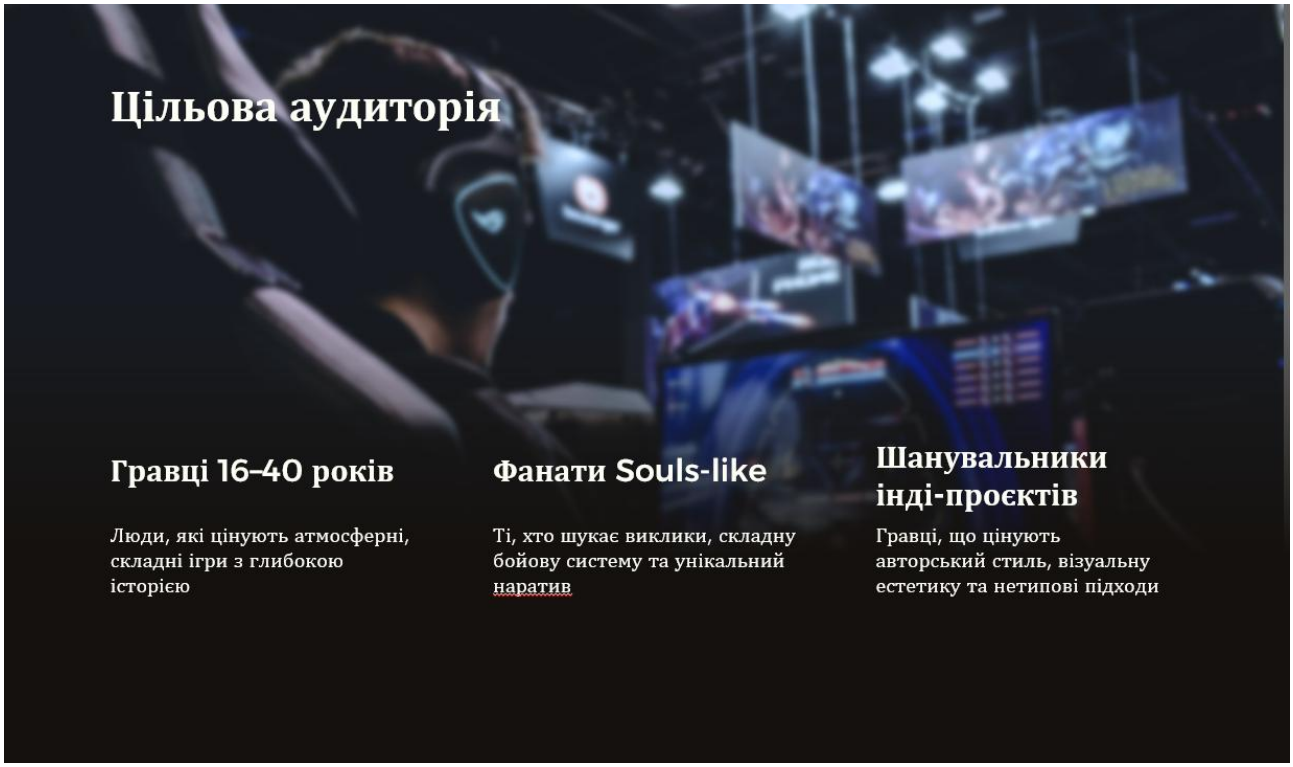
Ігри цього типу мають стабільний попит серед фанатів складного, вдумливого геймплею.

Високий поріг входу

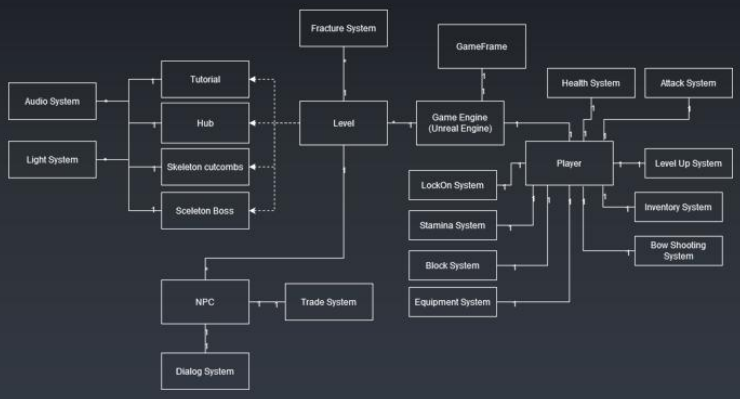
Новачки часто стикаються з труднощами через відсутність підказок та пояснень.

Мета проєкту – баланс

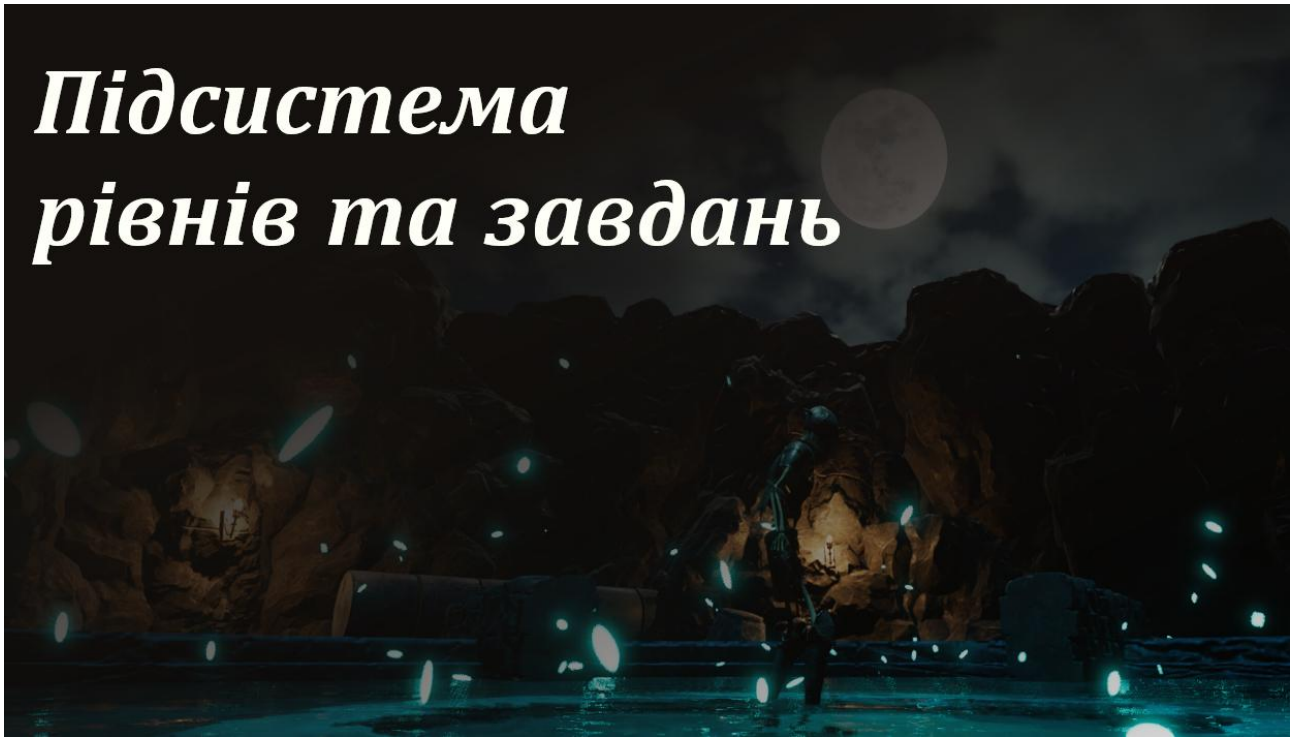
Поєднуємо глибину жанру з доступністю через туторіал, логіку рівнів і інтерактивність.

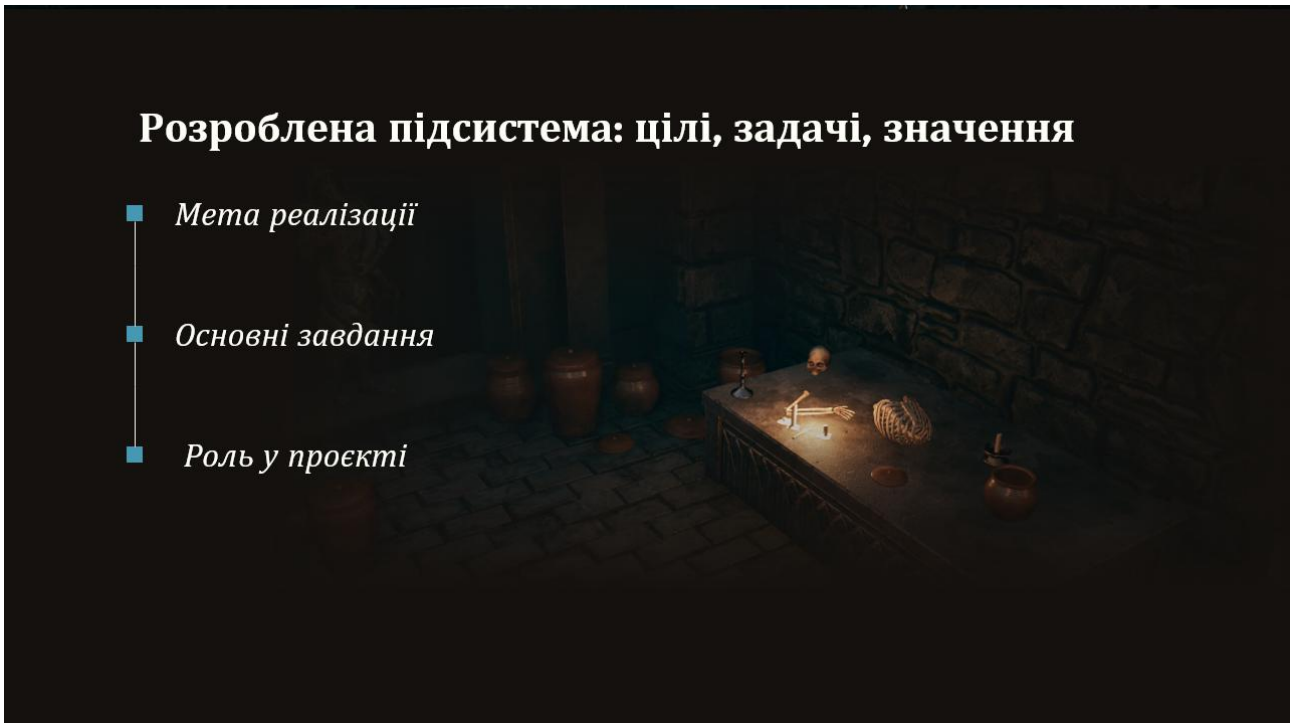


Користь та застосування



- ***Навчальний приклад***
- ***База для подальшого розвитку***
- ***Практичне застосування в геймдеві***


Підсистема
рівнів та завдань


Розроблена підсистема: цілі, задачі, значення
Мета реалізації
Основні завдання
Роль у проєкті

Структура рівнів та їх особливості

Хаб-локація

Безпечна зона з NPC. Містить портали до бойових рівнів, виконує роль навігаційного центру.

Рівень-туторіал

Початковий навчальний рівень. Гравець знайомиться з керуванням, боєм та взаємодією з об'єктами.

Катакомби

Бойовий рівень у вигляді лабіринту. Наповнений ворогами, атмосферою напруженості та загадковості.

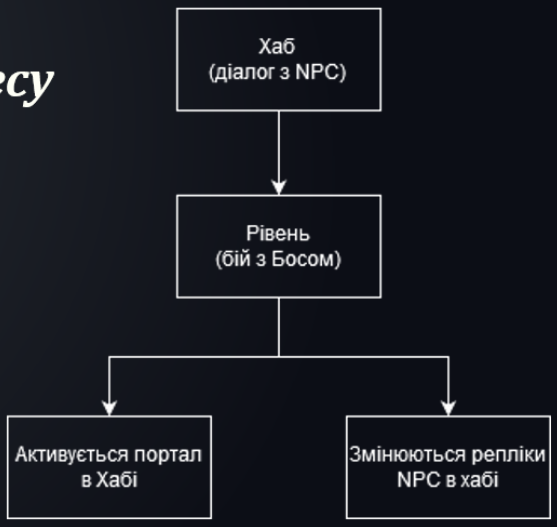
Фінальний рівень з босом

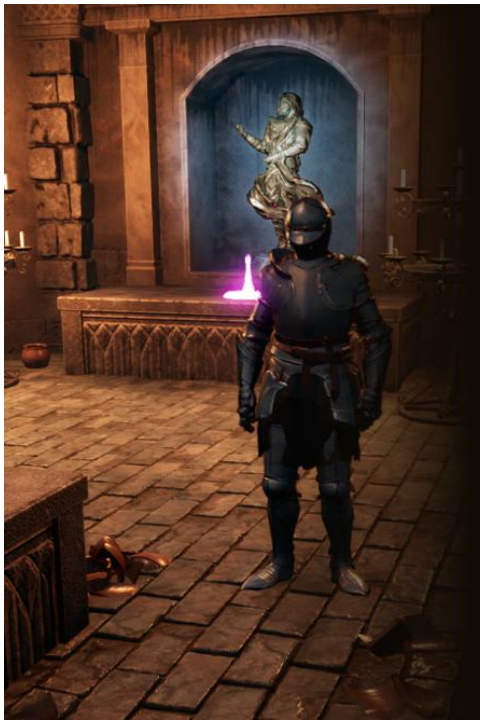
Простора арена для ключового бою. Має унікальне оформлення, динамічне освітлення і логіку відкриття порталу після перемоги.

Сюжетні завдання як частина ігрового прогресу

Сюжет подається через події, діалоги та взаємодії з оточенням, що відповідає стилю Souls-like:

- Діалоги з NPC
- Бій з Босом
- Розблокування нового порталу





Механіка ламання глечиків

Однією з цілей було реалізувати інтерактивне середовище, що підсилюватиме живість світу й створюватиме ефект занурення

01 *Fracture*

02 *Активація*

03 *Damage-куб*

Взаємодія з NPC — історії, вибір, наслідки

NPC слугують джерелом сюжетних натяків, підтримують атмосферу гри та орієнтують гравця на подальші дії.

Нелінійна поведінка

Взаємодія з NPC може призводити до різних сценаріїв:

- Доступ до додаткових функцій (торгівля)
- Зміни у розташуванні персонажів

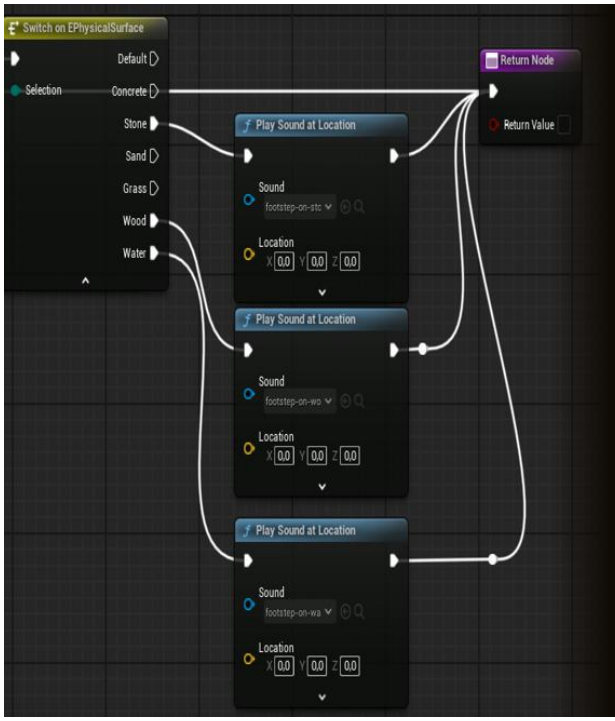
Інтеграція в ігровий процес

- Діалоги побудовані відповідно до подій у грі
- NPC реагують на дії гравця, підсилюючи ілюзію живого світу
- Реалізація відповідає жанровим особливостям Souls-like (непряме керування сюжетом)

Атмосфера — освітлення, звукове оформлення, ефекти

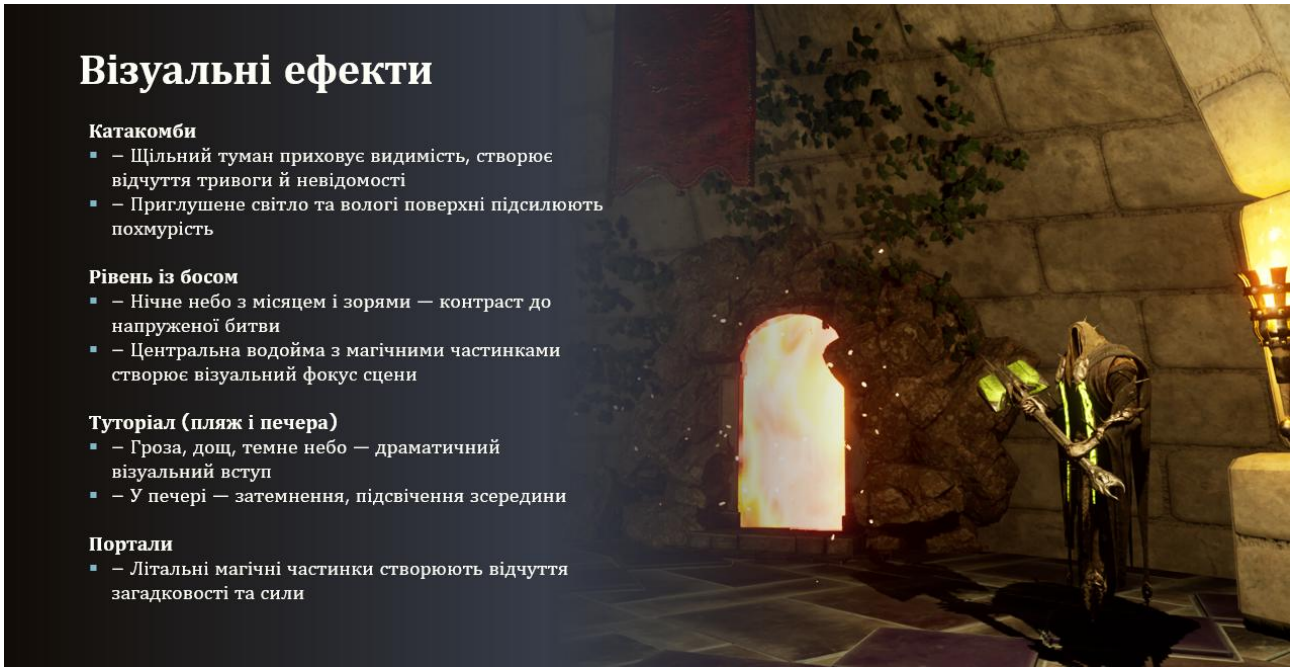
Освітлення

- Використано локальні джерела світла (факели, світлячки, м'яке підсвічування)
- Контраст між хабом (теплі тони) і катакомбами (холодні, приглушені)



Звукове оформлення

- Озвучено анімації переміщення гравця: ходьба, біг, рух зі зброєю / без зброї
- Фонове музичне оформлення:
  - Хаб-локація — спокійна, медитативна мелодія
  - Катакомби — тиск, фонова напруга
  - Арена з босом — драматична музика, яка змінюється на спокійну після перемоги
  - Навчальний рівень (туторіал):
    - Пляжна частина — звуки грози та хвиль
    - Печера — тривожний звуковий фон
- Окремо створені звукові ефекти:
  - Факели, полум'я, портали


Візуальні ефекти
Катакомби
– Щільний туман приховує видимість, створює відчуття тривоги й невідомості
– Приглушене світло та вологі поверхні підсилюють похмурість
Рівень із босом
– Нічне небо з місяцем і зорями — контраст до напруженої битви
– Центральна водойма з магічними частинками створює візуальний фокус сцени
Туторіал (пляж і печера)
– Гроза, дощ, темне небо — драматичний візуальний вступ
– У печері — затемнення, підсвічення зсередини
Портали
– Літальні магічні частинки створюють відчуття загадковості та сили


Реалізовані інтерфейси Система торгівлі
Реалізовано магазин у NPC: натискання Q відкриває вікно з товарами.
NPC озвучує одну з випадкових фраз при відкритті магазину.
Показується назва, опис, ціна та зображення предмета.
Повторне Q — купівля за наявності валюти.
Gideon Trader
Heal Potion
Potion Affect
Potion to restore some health (55%).
Potion Price
150 souls
Press [Q] to buy

Інтерфейс підказок

Натискання H відкриває/закриває короткі підказки з управління.
Допомагає гравцеві швидко орієнтуватися без виходу з гри.

ВИСНОВКИ по реалізованій частині

- Реалізовано ключові елементи геймплею
- Інтегровано інтерфейси та механіки
- Досягнуто цілісності підсистеми