

Міністерство освіти і науки України
Криворізький національний університет
Кафедра моделювання та програмного забезпечення

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття ступеня вищої освіти бакалавра
зі спеціальності 121 – Інженерія програмного забезпечення

На тему: Розробка комп'ютерної гри в жанрі метроїдванія за мотивами
української міфології

Засвідчую, що в цій
кваліфікаційній роботі немає
запозичень із праць інших
авторів без відповідних
посилань.
Студент гр. ПЗ-21-1
Р. Є. Ряпов

Керівник
кваліфікаційної роботи А. М. Стрюк

Завідувач кафедри А. М. Стрюк

Кривий Ріг
2025

Криворізький національний університет

Факультет: Інформаційних технологій

Кафедра: Моделювання та програмного забезпечення

Ступінь вищої освіти: бакалавр

Спеціальність: 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:

Зав. кафедри

А. М. Стрюк

«__» _____ 20__ р.

ЗАВДАННЯ

на кваліфікаційну роботу

студенту групи ІПЗ-21-1 Ряпов Роману Євгеновичу

1. Тема: Розробка комп'ютерної гри в жанрі метроїдванія за мотивами української міфології затверджено наказом по КНУ № 7 від «28» січня 2025 р.
2. Термін подання студентом закінченої роботи: «29» травня 2025р.
3. Вихідні дані по роботі: Проєкт передбачає розробку комп'ютерної гри в жанрі метроїдванія на базі Unreal Engine 5 з тематикою української міфології для платформи Windows.
4. Зміст пояснювальної записки: вступ, теоретичний розділ із аналізом української міфології та жанру метроїдванія, формалізацію вимог до гри, проєктування з використанням блок-схем, UML-діаграм і баз даних, розробку програмних модулів і інтерфейсу, а також висновки та перелік використаних джерел.
5. Перелік ілюстративного матеріалу: блок-схеми алгоритмів, UML-діаграми класів, фрагменти коду, таблиці бази даних

Календарний план:

№	Найменування етапів кваліфікаційної роботи	Термін виконання етапів роботи
1	Вивчення тематики та збір матеріалів з української міфології	27.03.2025 – 29.03.2025
2	Аналіз жанру метроїдванія та прикладів ігор	30.03.2025 – 09.04.2025
3	Постановка задачі, формування концепції гри	10.04.2025 – 18.04.2025
4	Розробка документації	19.04.2025 – 23.04.2025
5	Проектування архітектури	24.04.2025 – 30.04.2025
6	Розробка прототипу гри	01.04.2025 – 04.04.2025
7	Створення візуального стилю, графіки та UI	05.04.2025 – 13.05.2025
8	Реалізація логіки ігрового світу та ворогів	14.05.2025 – 20.05.2025
9	Інтеграція звукового супроводу	21.05.2025 – 23.05.2025
10	Оформлення пояснювальної записки	24.05.2025 – 29.05.2025

Дата видачі завдання: «28» Січня 2025 р.

Студент Р. Є. Ряпов/

Керівник роботи А. М. Стрюк

РЕФЕРАТ

УКРАЇНСЬКА МІФОЛОГІЯ, КОМП'ЮТЕРНА ГРА, ЖАНР МЕТРОЇДВАНІЯ, UNREAL ENGINE 5, ІГРОВИЙ ДИЗАЙН, ГЕЙМПЛЕЙ, ВІЗУАЛЬНЕ ОФОРМЛЕННЯ, ПРОЄКТУВАННЯ, ПРОГРАМНА РЕАЛІЗАЦІЯ, ТЕСТУВАННЯ.

Пояснювальна записка містить 32 сторінок, 7 рисунків, використані джерела.

Метою роботи є створення гри з освітньо-культурним наповненням, яка поєднує класичну ігрову механіку жанру метроїдванія та національні елементи фольклору України.

Для досягнення мети проведено аналіз міфологічної тематики, розроблено ігрову концепцію, спроектовано архітектуру гри з використанням UML-діаграм, створено графічні матеріали, реалізовано геймплей на рушії Unreal Engine 5 та проведено тестування.

До основних характеристик розробленого програмного продукту належать: двовимірна структура світу з нелінійною побудовою рівнів, підтримка системи збережень, система бойової механіки, а також адаптивне управління персонажем.

Гра призначена для персональних комп'ютерів (Windows) та орієнтована на поціновувачів інді-ігор, студентів, викладачів та всіх, хто цікавиться українською культурою й ігровим мистецтвом.

У висновках підтверджено доцільність поєднання культурної тематики з ігровою індустрією, продемонстровано функціональність реалізованих рішень та запропоновано напрями подальшого розвитку проєкту.

ABSTRACT

UKRAINIAN MYTHOLOGY, VIDEO GAME, METROIDVANIA GENRE, UNREAL ENGINE 5, GAME DESIGN, GAMEPLAY, VISUAL DESIGN, MODELING, SOFTWARE IMPLEMENTATION, TESTING.

The explanatory note contains 32 pages, 7 figures, sources.

Purpose of the work to create an educational and culturally inspired game combining classical metroidvania mechanics with national folklore elements of Ukraine.

To achieve the goal, a mythology analysis was conducted, the game concept was developed, architecture was designed using UML diagrams, graphic assets were created, gameplay was implemented in Unreal Engine 5, and final testing was carried out.

Key features of the developed product include: a two-dimensional non-linear world structure, save system, combat mechanics, and responsive character controls.

The game is intended for Windows-based personal computers and targets indie game enthusiasts, students, educators, and anyone interested in Ukrainian culture and digital art.

The conclusions confirm the relevance of combining cultural themes with the game industry, demonstrate the effectiveness of the implemented solutions, and outline possible directions for future development.

Зміст

ВСТУП	7
1 ТЕОРЕТИЧНИЙ РОЗДІЛ.....	8
1.1 Українська міфологія як об'єкт вивчення та дослідження.....	8
1.2 Актуальність розробки комп'ютерної гри за мотивами української міфології.....	10
1.3 Особливості жанру метроїдванія.....	11
1.3.1 Основні риси жанру метроїданія.....	12
1.3.2 Метроїдванія у сучасному ігровому контексті	13
1.3.3 Переваги жанру адаптації української міфології	13
1.3.4 Додаткові напрямки розвитку метроїдвань.....	14
1.4 Формалізація вимог до комп'ютерної гри в жанрі метроїдванія за мотивами української міфології	15
1.4.1 Концепція гри та геймплей	15
1.4.2 Візуальне й звукове оформлення	16
1.4.3 Освітньо-культурна складова	16
2 РОЗДІЛ ПРОЕКТУВАННЯ	18
2.1 Розробка блок-схем алгоритмів програмного комплексу.....	18
2.2 Розробка UML-діаграм класів програмного комплексу	21
2.2.1 Класи ядра ігрового процесу	21
2.2.2 Класи взаємодії зі світом.....	23
2.2.3 Класи користувацького інтерфейсу	24
2.3 Розробка моделі візуального інтерфейсу програмного комплексу	26
2.4 Проектування таблиць бази даних і вибір типів даних полів	28
3. РОЗРОБКА ПРОГРАМНИХ МОДУЛІВ ПРОЄКТУ	30
3.1 Розробка основних класів програмних модулів проекту:.....	30
3.2 Розробка візуального інтерфейсу програмного комплексу	34
ВИСНОВОК.....	39
ПЕРЕЛІК ПОСИЛАНЬ	40

ВСТУП

Сучасна індустрія комп'ютерних ігор є однією з найдинамічніших сфер цифрової культури, яка поєднує інженерні, художні, наративні та психологічні аспекти. Окрім розважальної функції, ігри дедалі частіше використовуються як засіб передачі знань, збереження культурної спадщини та формування національної ідентичності. В умовах зростання суспільного інтересу до української міфології актуальним стає створення тематичних ігрових продуктів, що відображають національні фольклорні мотиви у доступній, інтерактивній формі.

Українська міфологія, незважаючи на свою глибину та унікальність, залишається недостатньо представленою у масовій культурі, особливо в сегменті цифрових ігор. Використання образів, персонажів та сюжетних архетипів українського фольклору у відеогрі дозволяє не лише створити захопливий ігровий досвід, а й привернути увагу молоді до культурних коренів свого народу.

Метою цієї роботи є розробка комп'ютерної гри в жанрі метроїдванія, сюжет і візуальний стиль якої базуються на українській міфології. Гра має поєднувати класичну ігрову механіку з елементами дослідження, боїв та розвитку персонажа у нелінійному світі.

Галузь застосування результатів роботи охоплює індустрію комп'ютерних ігор, освітні та культурно-просвітницькі проєкти, інді-розробки, а також потенційно може слугувати платформою для інтеграції елементів культурної спадщини у сучасне медіасередовище.

1 ТЕОРЕТИЧНИЙ РОЗДІЛ

1.1 Українська міфологія як об'єкт вивчення та дослідження

Українська міфологія як об'єкт вивчення та дослідження Українська міфологія – це найбагатший і менш досліджений пласт культурної спадщини, який досі зберігає багато білих плям. Вона формувалася протягом тисячоліть, починаючи з індоєвропейських вірувань та завершуючи християнізацією Київської Русі. Основними джерелами її дослідження є усна народна творчість – пісні, казки, думи, легенди, а також етнографічні дослідження XIX–XX століть. Зокрема, значний внесок у збирання та аналіз українського фольклору зробили такі вчені, як Володимир Гнатюк, Михайло Драгоманов, Павло Чубинський.

Однією з ключових особливостей української міфології є її тісний зв'язок із природним середовищем. У традиційних віруваннях селян панували образи парфумів лісу, води, полів, будинки, які втілювали природні сили. Найвідоміші персонажі – Мавки, Русалки, Льоші, Будинки – були не просто фантастичними істотами, а важливою частиною побутової уяви, що регулює відносини між людиною та довкіллям.

До християнізації Русі основну роль міфології грали язичницькі божества, серед яких Перун – бог грому та війни, Велес – покровитель тварин і багатства, Дажбог – сонячне божество, і Мокош – богиня родючості та жіночої долі. Прийняття християнства 988 року не знищило повністю давні вірування, а скоріше інтегрувало їх: багато дохристиянських образів трансформувалися і збереглися у нових формах. Наприклад, обрядові пісні на Івана Купала, святкування Колодія, вірування в духів-покровителів будинку (будинкових) – це залишки язичницької системи у християнському обрамленні.

Українську міфологію також було ізолювано – на неї активно впливали культури сусідніх народів: слов'янських (польських, білоруських), балтійських, тюркських. У цих контактах відбувався взаємний обмін

образами, сюжетами та символами. Наприклад, образ русалки має паралелі у білоруському, польському та навіть угорському фольклорі.

Однак за всієї цієї глибини та різноманітності українська міфологія досі залишається маловідомою. У шкільній програмі їй приділяється мала увага, а масової культури вона виникає лише фрагментарно. Культуролог Ігор Козловський зазначає, що "українці досі не до кінця усвідомлюють багатство власної міфологічної спадщини, яка може бути основою національного відродження".

Сьогодні особливу роль у популяризації української міфології можуть відігравати засоби масової культури – кіно, література, музика та особливо комп'ютерні ігри. Вже є приклади вдалого використання міфологічних мотивів у творах сучасних авторів, зокрема Лада Лузіна у своєму циклі «Київські відьми» поєднує магію та історію столиці; групи DakhaBrakha та Dakh Daughters використовують елементи обрядової музики у поєднанні з авангардом; а мультфільм "Мавка. Лісова пісня" (2023) представивши адаптацію фольклору на міжнародному рівні.

Однак саме комп'ютерні ігри мають унікальний потенціал для глибокого занурення користувача в міфологічний світ. Ігрові наративи дозволяють не лише спостерігати за подіями, а й безпосередньо взаємодіяти з міфологічними істотами, проходити випробування, що ґрунтуються на народних віруваннях, та досліджувати символіку українських легенд. Прикладом такого підходу є гра «Black Book» (2021), створена студією «Morteshka». Хоча вона базується на північноросійському фольклорі, в ній простежуються численні спільні риси зі слов'янською, у тому числі українською, міфологією: є образи домових, лісовиків, чаклунства, ритуалів та глибокого шанування предків та природи. Успіх гри серед гравців та критиків доводить, що інтерес до слов'янських міфів існує і може бути успішно реалізований через інтерактивний наратив.

Активне використання української міфології у сучасних комп'ютерних іграх не лише сприяє збереженню нематеріальної культурної спадщини, а й дозволяє новітнім поколінням дізнатися про свою історію у захоплюючій та

доступній формі. Це поєднання глибокого змісту та сучасних технологій може стати важливим інструментом національної ідентичності у світі.

1.2 Актуальність розробки комп'ютерної гри за мотивами української міфології

У сучасному світі комп'ютерні ігри стали не просто розвагою, а найважливішим елементом культурної комунікації та самовираження. Вони здатні формувати уявлення про країну, її народ, історію та цінності. Через інтерактивні світи, гравці з усього світу набувають унікальної можливості ознайомитися з культурним нащадком інших народів. Тому розробка комп'ютерної гри, натхненної українською міфологією, є не лише творчим проектом, а й важливим культурним кроком.

Ігри мають сильний вплив на формування позитивного іміджу України у світі. Наприклад, серія ігор «S.T.A.L.K.E.R.», розроблена українською студією GSC Game World, прославила Чорнобильську зону відчуження як унікальне та загадкове місце, перетворивши її на відомий символ у геймерській культурі. Незважаючи на пост апокаліптичний антураж, гра викликає інтерес до української природи, архітектури, навіть мови — в останніх версіях доступні україномовні локалізації. Інший яскравий приклад — стратегія «Козаки», що знайомить гравців із військовими кампаніями XVII–XVIII століття, роллю козацтва та геополітичною ситуацією у Східній Європі. Ці ігри не лише розважають, а й сприяють формуванню поваги до української історії та культури.

Слід зазначити, що комп'ютерні ігри відіграють значну роль у формуванні національної самосвідомості українців, особливо молоді. В умовах війни та культурної експансії з боку агресора культурні продукти, створені українцями про Україну, стають засобом збереження ідентичності, джерелом натхнення та гордості. Коли гравець бачить у грі знайомі ландшафти, чує українську мову, стикається зі своїми міфами — це створює ефект приналежності, сприяє зміцненню національного духу.

Світовий досвід показує, що відеоігри здатні ефективно популяризувати міфологію та фольклор. Яскравим прикладом є серія «The Witcher» (CD Projekt Red), заснована на романах Анджея Сапковського, що спираються на польський та загальнослов'янський фольклор. Гравці знайомитимуться з образами віл, лісовиків, упирів, відьом тощо. Гра стала світовим хітом, а персонажі слов'янської міфології – відомими у всьому світі. Ще один приклад - серія «Assassin's Creed» (Ubisoft), що дозволяє гравцям досліджувати різні історичні епохи, культури та релігії, наприклад, Стародавній Єгипет, Грецію або Скандинавію. В іграх не тільки реконструюються архітектура та побут, а й інтегруються елементи вірувань та міфології, які мають великий пізнавальний потенціал.

Незважаючи на такі приклади успішного використання міфології у світовому ігровому просторі, українська міфологія, як і раніше, залишається на периферії уваги розробників. Немає жодної великої гри, яка б системно й повно відображала багатство українського фольклору, пантеону божеств чи демонології. І хоча з'являються інді-проекти, такі як «Serpent Rogue», вони переважно тільки торкаються тематики, не розкриваючи її глибоко і масштабно.

Тому розробка комп'ютерної гри у жанрі метроїдванія за мотивами української міфології не лише актуальна, а й необхідна. Вона має потенціал стати інструментом культурної дипломатії, джерелом пізнання іноземної аудиторії та водночас способом переосмислення власної ідентичності для самих українців. У поєднанні з елементами сучасного геймдизайну та художнього підходу такий проект здатний вивести український фольклор на міжнародну арену та подарувати йому нове життя в цифровому вимірі.

1.3 Особливості жанру метроїдванія

Жанр метроїдванія (англ. «Metroidvania») — це один із найвпливовіших напрямків у світі інді-ігор та авторських проектів, що виник на перетині класичних платформерів, пригодницьких ігор та рольових елементів,

поєднуючи їхні найхарактерніші особливості. Назва жанру походить від двох революційних ігор — *Metroid* (Nintendo, 1986) і *Castlevania: Symphony of the Night* (Konami, 1997), які задали канонічні риси цієї гібридної форми, включаючи дослідження великих, взаємопов'язаних світів з нелінійною структурою, поступовим відкриттям нових областей і поверненням до вже відомих локацій. Саме ці ігри започаткували ключовий принцип відкритого дослідження світу з обмеженням доступу, яке знімається в міру розвитку персонажа, отримання нових здібностей або інструментів, що відкривають нові шляхи та можливості для взаємодії з оточенням.

1.3.1 Основні риси жанру метроїданія

Жанр метроїдання має ряд характерних рис, що сформувалися протягом десятиліть еволюції ігрового дизайну:

а) Нелінійний світ, що поступово розкривається. Світ гри зазвичай великий і зв'язний, часто поділений на тематичні зони (ліс, печера, храм, замок тощо), доступ до яких відкривається поступово. Це приваблює гравця до повторного відвідування локацій та дозволяє геймдизайнерам вибудовувати глибокий інтерактивний наратив.

б) Розвиток персонажа через механіку. Персонаж у метроїдванії не тільки накопичує досвід або покращує характеристики, як у звичних RPG, але й набуває нових навичок, що прямо змінюють спосіб дослідження світу. Наприклад, після отримання подвійного стрибка гравець може дістатися раніше недоступних платформ.

в) Симбіоз наратива та ігрового простору. Історія в метроїдванії рідко подається через довгі діалоги або катсцени. А сюжет інтегрований у світ: через архітектуру, візуальні підказки, позиції ворогів, секрети, об'єкти. Гравець стає дослідником як простору, а й сенсу.

г) Висока вага до атмосфери. Звукове оформлення, стиль графіки та музика в метроїдваніях надзвичайно важливі для створення емоційного зв'язку

з гравцем. Вдалий приклад — Hollow Knight, який через похмуру візуальну мову і звуковий фон створює відчуття глибокої самотності в загиблому світі.

д) Метод "гейтинга" (gating) - обмеження прогресу через відсутність певного інструменту, здібності чи знання. Цей метод дозволяє гравцеві не просто рухатися за сценарієм, а набувати досвіду через відкриття та винахідливість. Часто гравець бачить область, в яку не може дістатися, і пам'ятає її на майбутнє – це один із найефективніших способів стимулювати дослідження.

1.3.2 Метроїдванія у сучасному ігровому контексті

Останніми роками жанр метроїдванія отримав нове дихання, особливо у сфері незалежної розробки. Ігри типу Axiom Verge (Thomas Happ, 2015), Ender Lilies (Binary Haze Interactive, 2021), Salt and Sanctuary (Ska Studios, 2016) та Blasphemous (The Game Kitchen, 2019) демонструють величезну жанрову. Деякі з них інтегрують souls-like елементи (високий рівень складності, респаун ворогів після смерті, відновлення ресурсів на контрольних точках), інші – сильний емоційний наратив чи інноваційні бойові системи.

Blasphemous, зокрема, є прикладом того, як локальна культура (у цьому випадку іспанський католицизм, містицизм і бароко) може лягти в основу естетики, лор-будівлі і навіть бойових механік. Її успіх демонструє, що поєднання геймплейної класики з глибокою фольклорною кулею здатне не лише зацікавити гравця, а й сприятиме популяризації національної культури за межами країни походження.

1.3.3 Переваги жанру адаптації української міфології

Українська міфологія, з її багатим пантеоном істот, складною структурою світобудови (навіть у триєдиному форматі — Нав, Яв, Прав), циклічністю, уявленням про ініціацію, смерть та відродження чудово підходить для механік метроїдванії. Шкіра зони гри може бути уособленням

різних пластів світогляду: ліс – як територія Лісовик та Мавок, вода – володіння Русалок та Водяників, гори – місце сили богів та духів. Наприклад:

- а) Нав - темні, підземні області, пов'язані зі смертю, привидами, відьмами, випробуваннями;
- б) Яв – світ живих, людські поселення, місця звичних конфліктів;
- в) Прав - сакральні вершини, дерева життя, небесні сфери, доступні тільки після духовного піднесення героя.

Міфологічні образи можна інтерпретувати як босів, NPC, союзників, і навіть метафоричні етапи становлення головного героя. Німфа - не просто ворог, а дух, що охороняє кордон між світами. Відьма — носій стародавнього знання, що ставить перед героєм загадки. Цей підхід дозволяє не лише представляти фольклор як естетику, а й вбудовувати їх у геймплейні рішення.

1.3.4 Додаткові напрямки розвитку метроїдвань

Жанр постійно розвивається. Нові ігри експериментують із процедурною генерацією (як у *Dead Cells*, 2018), інтерактивною музикою, мультиплеєром або елементами survival. Такі гібридні рішення справляють унікальні ігрові враження, зберігаючи при цьому дух відкритого дослідження. Це відкриває можливості для розширення концепту української метроїдванії, де, наприклад, можна адаптувати народні пісні в ігровий саундтрек або створити системи магії на базі традиційних замовлянь та обрядів.

Метроїдванія — це не просто жанр, це платформа для інтерактивного міфотворення. Його гнучкість, відкритість до наративних експериментів, здатність вибудовувати довгу й глибоку емоційну арку робить його ідеальним вибором для адаптації української міфології у вигляді комп'ютерної гри. Через відповідний ігровий досвід можна не лише розважати гравця, а й відкрити йому цілий пласт культури, що досі залишається маловідомим навіть в Україні.

1.4 Формалізація вимог до комп'ютерної гри в жанрі метроїдванія за мотивами української міфології

Формалізація вимог до комп'ютерної гри є необхідним етапом, який визначає її концептуальні й технічні рамки. У випадку створення гри в жанрі метроїдванія за мотивами української міфології, особливої уваги потребує гармонійне поєднання геймплейних рішень із культурним контекстом. Цей розділ окреслює головні засади майбутнього проєкту, починаючи з базових механік і завершуючи наративною структурою.

1.4.1 Концепція гри та геймплей

Запропонована гра належатиме до жанру метроїдванія, що означає нелінійну побудову світу з численними локаціями, які поступово відкриваються гравцю. Прогресія в грі відбуватиметься шляхом отримання нових умінь і здібностей — наприклад, можливості здійснювати подвійний стрибок, проходити крізь духовні перешкоди або використовувати артефакти для подолання перешкод. Кожне нове вміння відкриватиме доступ до нових ділянок карти, що формуватиме постійне відчуття відкриття і занурення в ігровий світ.

Головним героєм гри є двоєдушник — персонаж, що водночас належить до світу живих і світу мертвих. У традиційних українських віруваннях двоєдушники були наділені здатністю бачити духів, переходити між вимірами та спілкуватися з потойбіччям. Така подвійна природа створює внутрішній конфлікт, який гравець переживає разом із героєм: людське начало змагається з демонічною сутністю, що поступово пробуджується. Саме ця особливість формує унікальну геймплейну динаміку — одні здібності герой може застосовувати лише у світі живих, інші — лише у світі духів. Крім того, перехід між цими світами стає не лише механікою, а й важливою складовою сюжету.

Наратив гри базується на концепції української трисвітності — Яви, Нави та Прави. Герой проходить ініціаційний шлях, зустрічаючи на своєму

шляху істот української міфології: Мавку, Вія, Чугайстра, Домовика, Песиголовця, Жар-птицю та багатьох інших. Кожна зустріч з такими персонажами розкриває не лише сюжет, а й певні аспекти духовного шляху героя. Босами виступають потужні сутності, що охороняють переходи між світами або блокують спогади самого героя. Кожен з них має символічне значення та відображає страхи або випробування, пов'язані з подвійною природою двоєдушника.

1.4.2 Візуальне й звукове оформлення

Візуально гра стилізована під українське народне мистецтво: графіка черпає натхнення з орнаментів, дереворитів, вишивки та писанкарства. Кольорова палітра обмежена, з акцентом на світло-тіньовий контраст, що створює атмосферу казковості та містики. Інтерфейс витриманий у стилі стародавніх рукописів — мінімалістичний, але глибоко символічний.

Музичне оформлення ґрунтується на автентичному фольклорі: у саундтреку використовуються сопілка, трембіта, дримба, вокалізація з елементами поліфонії. Звукові ефекти створюють ілюзію перебування в архаїчному лісі — чути шепіт вітру, тріск гілок, далекі голоси, що нагадують обрядові пісні.

Гра розробляється під платформу ПК із перспективою портування на консолі. Мовна підтримка включає українську як основну мову, а також англійську для міжнародної аудиторії. Збереження прогресу реалізовано через контрольні точки — ліжка, які розташовані в безпечних зонах. Саме у ліжках гравець не лише відпочиває та відновлює сили, а й бачить символічні сни, у яких розкриваються фрагменти давніх легенд і підказки для подальшого проходження.

1.4.3 Освітньо-культурна складова

Крім розважальної функції, проєкт виконує важливу культурну та освітню місію. Він створений не лише для захоплюючого проходження, а й як інструмент популяризації української міфології серед широкого кола

гравців — як українських, так і міжнародних. Гравець не просто взаємодіє з фантастичними істотами — він поступово занурюється в багатий світ давніх вірувань, обрядів і уявлень, що формували духовний світ українців упродовж століть.

У грі реалізовано спеціальний глосарій, який містить пояснення ключових міфологічних понять: від назв істот і духів до традиційних символів, магічних обрядів і концептів потойбічного світу. Гравець зможе дізнатись, хто такий Чугайстер, чому Песиголовець був втіленням страху перед чужинцем, що символізують тризуб, обереги, мак чи калина, а також як саме функціонували народні уявлення про Яву, Наву та Праву. Глосарій органічно вбудований у геймплей: кожна нова зустріч або знахідка відкриває новий термін, що робить процес пізнання природним і мотиваційним.

У такий спосіб запропонована гра поєднує динамічний геймплей жанру метроїдванія з глибоким культурним зануренням. Вона не лише розважає, а й формує повагу до національної спадщини, а образ двоєдушника виступає символом тонкого зв'язку між матеріальним і духовним, між реальним і уявним. Цей герой дозволяє розкрити теми потойбіччя, духовності, внутрішньої боротьби та ініціації, які є наскрізними у традиційній українській картині світу.

2 РОЗДІЛ ПРОЕКТУВАННЯ

2.1 Розробка блок-схем алгоритмів програмного комплексу

У рамках проєкту було створено кілька ключових блок-схем, які відображають основні аспекти роботи гри:

Алгоритм збереження та завантаження гри — відображає логіку запису поточного стану гри (локація, характеристики героя, прогрес, відомі легенди тощо) у момент відпочинку на ліжку, а також процедуру завантаження цих даних при новому запуску.



Рисунок 2.1 — Алгоритм збереження та завантаження гри

Блок-схема ігрового циклу є графічним зображенням, яке деталізує основні етапи, що відбуваються під час виконання циклу гри. Вона включає обробку введення з клавіатури або геймпада, що дозволяє гравцеві керувати персонажами чи взаємодіяти з ігровим світом. Після цього відбувається оновлення стану всіх ігрових об'єктів відповідно до введення користувача, внутрішньої логіки гри та зовнішніх чинників.

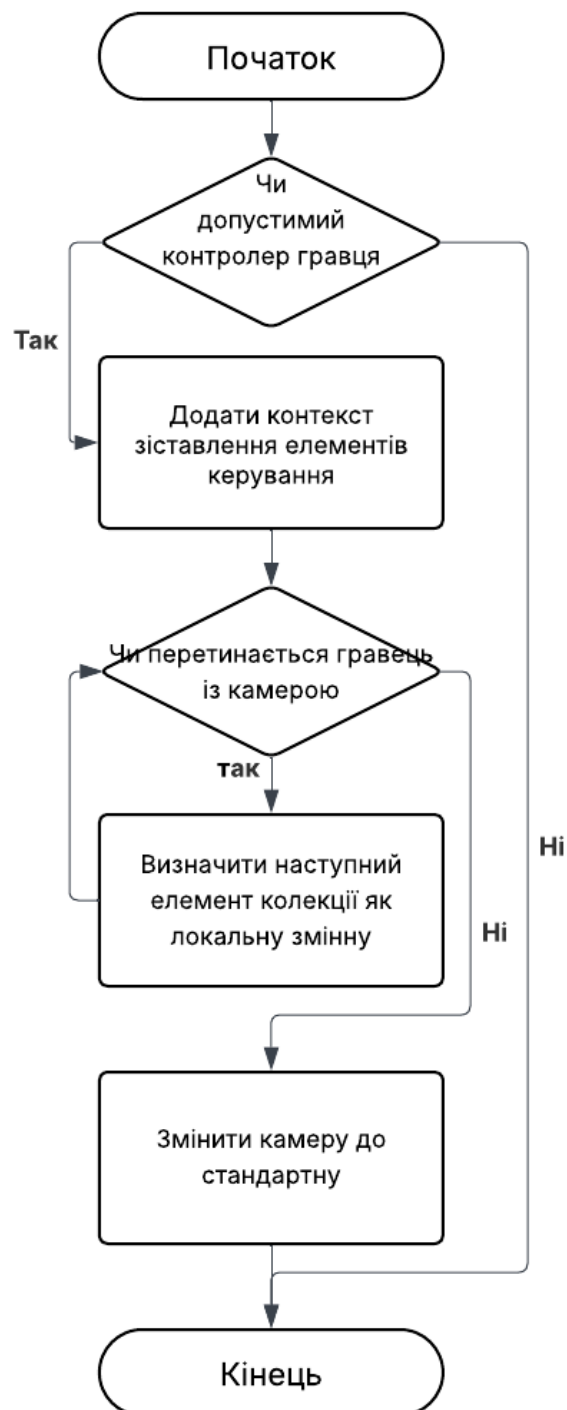


Рисунок 2.2 — Алгоритм ігрового циклу

Алгоритм бою з ворогами описує послідовність дій і логіку, яка реалізується під час бойових сутичок у грі. Він включає механізми виконання атак гравцем і противниками, використання спеціальних здібностей або навичок, які можуть впливати на перебіг бою. Окрім цього, алгоритм враховує зміну рівня здоров'я як у персонажа гравця, так і у ворогів чи босів, відображаючи втрату здоров'я при отриманні ушкоджень.

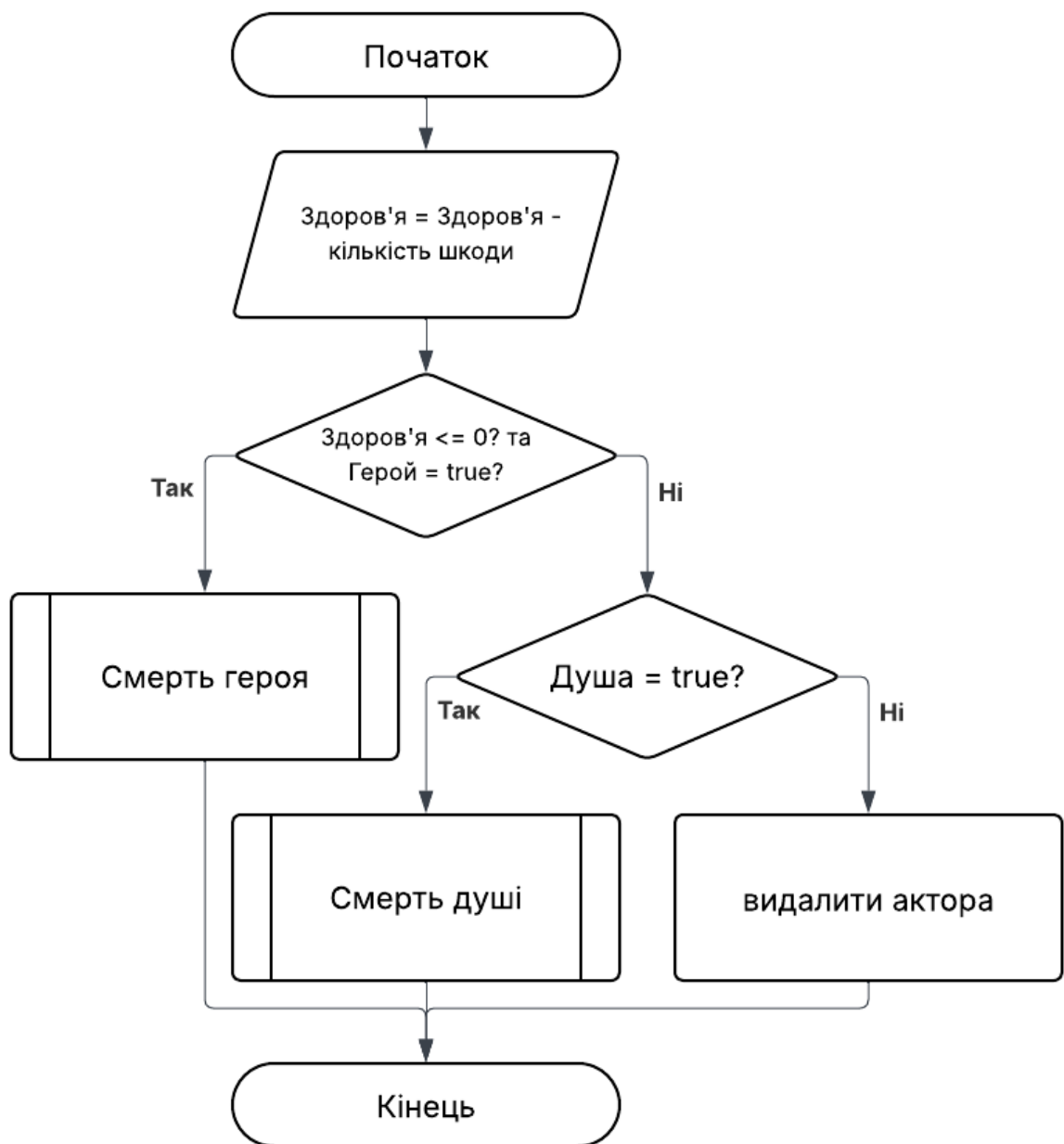


Рисунок 2.3 — Алгоритм бою з ворогами

2.2 Розробка UML-діаграм класів програмного комплексу

На етапі проєктування архітектури програмного забезпечення гри важливим інструментом моделювання стала UML-діаграма класів (Unified Modeling Language), яка дозволяє описати основні компоненти системи, їхні атрибути, методи та взаємозв'язки між ними.

2.2.1 Класи ядра ігрового процесу

Представлена UML-діаграма описує спадковість, атрибути та методи основних ігрових сутностей, таких як персонажі, вороги, та їхні форми. У центрі цієї архітектури знаходиться базовий клас CharParent, який служить спільним предком для всіх персонажів у грі. Він має лише одну властивість — рівень здоров'я — і метод ReciveDamage(DamageAmount), який дозволяє обробляти отримані ушкодження. Це забезпечує єдиний механізм для зменшення здоров'я як у головного героя, так і в його супротивників.

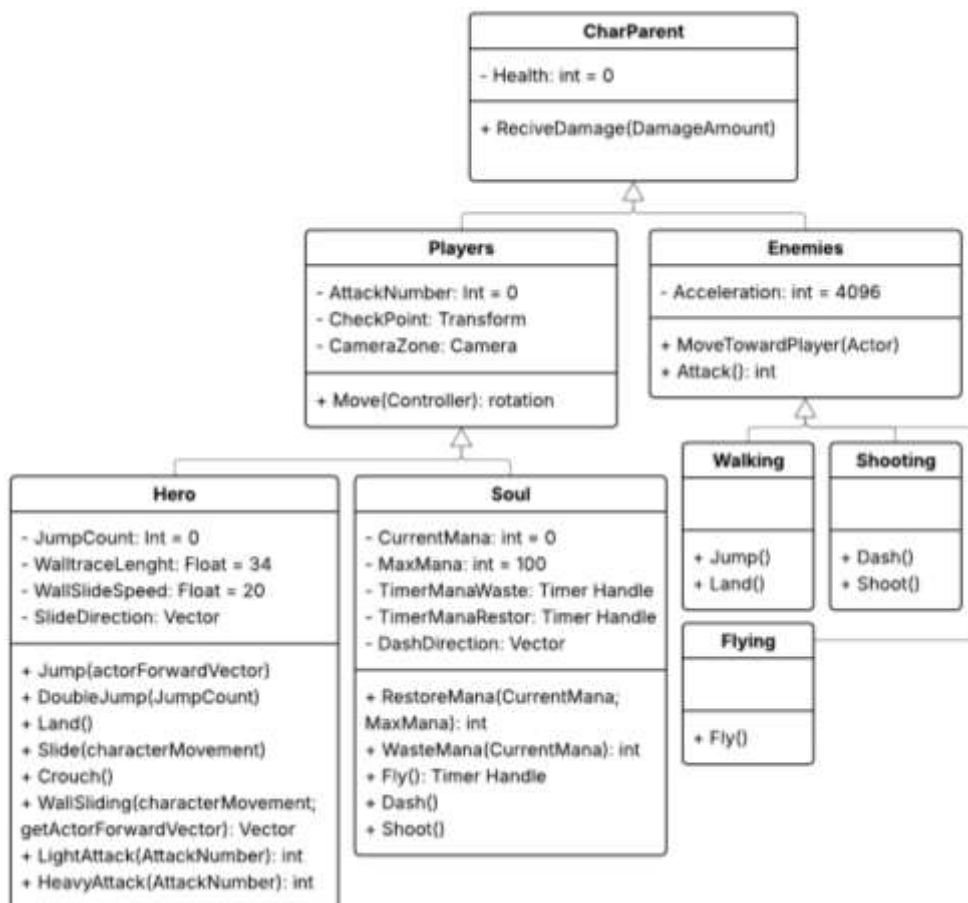


Рисунок 2.4 — Діаграма класів ігрового процесу

Від CharParent наслідуються два основні напрями: гравець і вороги. Клас Players — це узагальнення всіх станів і функцій керованого гравцем персонажа. До його складу входять атрибути, пов'язані з атаками, системою збереження прогресу та камерою спостереження за героєм. Метод Move(Controller) дозволяє здійснювати переміщення залежно від вхідних даних контролера, змінюючи при цьому орієнтацію персонажа.

Від класу Players походять два спеціалізовані класи — Hero та Soul. Клас Hero репрезентує фізичне втілення головного героя — двоєдушника у світі Яви. У ньому реалізовані складні механіки переміщення та бою: подвійний стрибок, ковзання по стінах, присідання, легка та важка атака тощо. Усі ці дії залежать від конкретних атрибутів, таких як JumpCount, WalltraceLength, WallSlideSpeed і SlideDirection. Завдяки цьому гравець може активно взаємодіяти з платформеними викликами гри.

Клас Soul, у свою чергу, уособлює духовну форму головного героя в потойбічному світі — Наві. Його відмінною рисою є використання мани як ресурсу для здібностей: польоту, ривка та стрільби. Для цього клас має атрибути CurrentMana, MaxMana, а також таймери для витрат і відновлення мани. Методи Fly(), Dash() і Shoot() дозволяють створити унікальний стиль геймплею з упором на вертикальність і дистанційні атаки, на противагу фізичному бою в Яві.

Клас Enemies описує базову логіку супротивників. Основними атрибутами є прискорення руху та методи MoveTowardPlayer(Actor) і Attack(), які дозволяють ворогам переслідувати гравця та завдавати шкоди. Цей клас є фундаментом для побудови більш конкретних типів ворогів. У діаграмі представлено три окремі типи: Walking, Flying та Shooting. Вони не мають власних атрибутів, але розширюють функціональність Enemies відповідно до свого стилю поведінки. Наприклад, Walking має методи Jump() і Land() для ворогів, які переміщуються пішки; Flying — метод Fly() для повітряних істот; Shooting — методи Dash() і Shoot() для дальніх атак. Такий підхід забезпечує гнучкість і розширюваність ігрового світу.

2.2.2 Класи взаємодії зі світом

Представлена UML-діаграма зображує структуру основних класів і зв'язків у програмному комплексі гри. У центрі архітектури знаходиться клас **WorldZone**, який представляє окрему зону світу гри — наприклад, ліс, болото чи село. Кожна така зона має свою назву, набір ворогів, неігрових персонажів, фрагменти легенд і прапорець, що вказує, чи була ця зона вже відкрита. Клас реалізує методи **LoadZone()** і **UnloadZone()** для керування ресурсами під час входу та виходу з цієї зони.

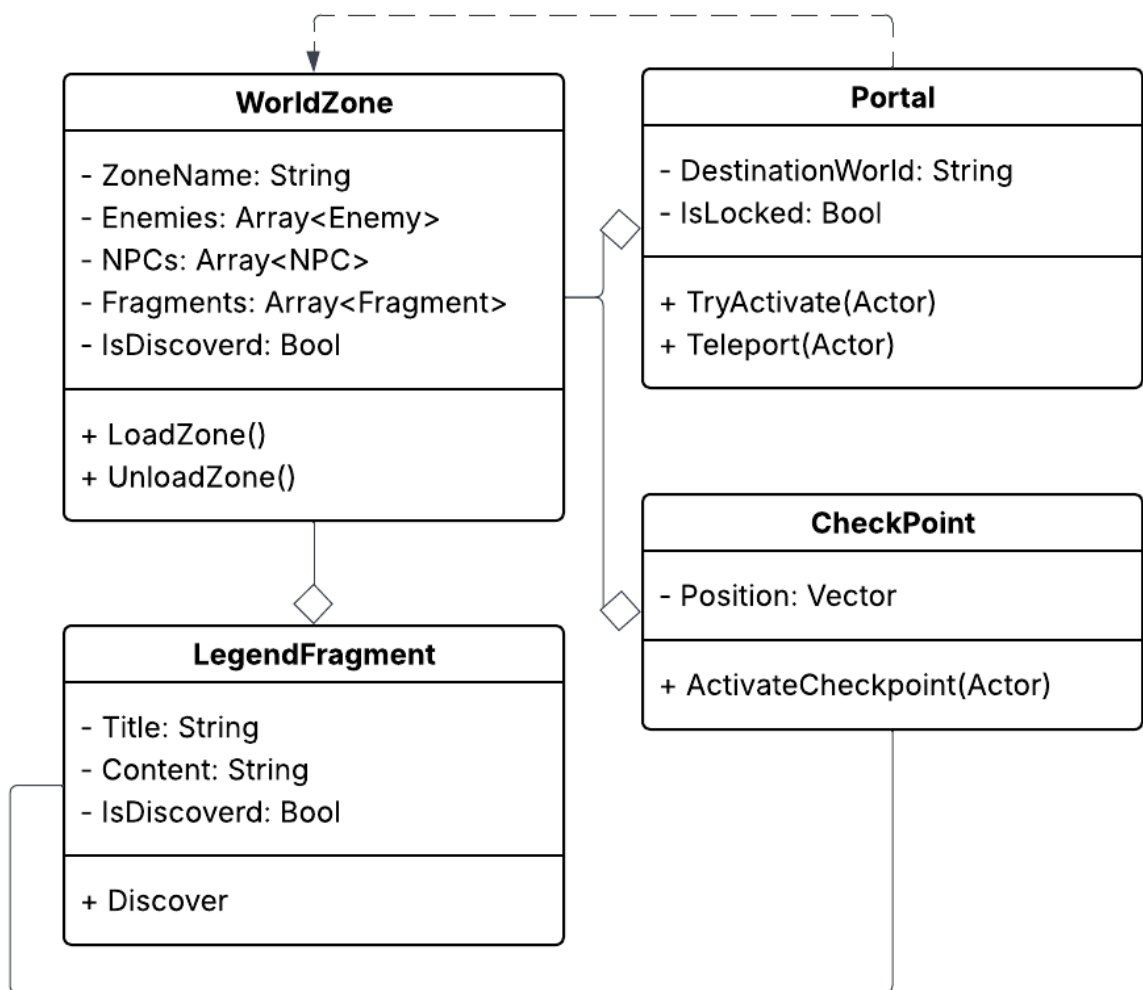


Рисунок 2.5 — Діаграма класів взаємодії зі світом

WorldZone пов'язаний із трьома іншими класами через агрегативні зв'язки, що означає, що він містить у собі об'єкти цих класів, але не володіє ними повністю — вони можуть існувати незалежно.

Першим з них є Portal, який слугує для переміщення між різними світами, зокрема між Явою та Навою. Він має поля DestinationWorld для збереження назви цільового світу та IsLocked, що вказує, чи активний портал. Основні методи — TryActivate(Actor) для перевірки доступності portalу і Teleport(Actor) для безпосереднього переміщення гравця.

Другим пов'язаним класом є CheckPoint. Це об'єкти збереження, представлені в грі як ліжка. Вони мають координати (Position) та метод ActivateCheckpoint(Actor), який дозволяє зберегти гру й отримати додаткову міфологічну інформацію у вигляді сну або фрагмента легенди.

Третім класом є LegendFragment. Цей клас зберігає окремі фрагменти знань про міфологію: назву, зміст і статус відкриття. Метод Discover() активує фрагмент і, можливо, передає його до UI для відображення гравцю.

Важливо, що LegendFragment асоційований не лише з WorldZone, а й з CheckPoint, що свідчить про можливість відкриття легенди при взаємодії з точкою збереження.

2.2.3 Класи користувацького інтерфейсу

Представлена UML-діаграма зображує класи інтерфейсу користувача які мають три ключові компоненти, кожен із яких виконує специфічну функцію в межах взаємодії гравця з грою. Ці компоненти — HUDController, DialogueManager та GlossaryUI — створені з урахуванням принципів модульності, гнучкості та простоти розширення.

HUDController (Head-Up Display Controller) відповідає за відображення базової інформації про стан гравця під час ігрового процесу. Це зокрема рівень здоров'я, запас мана, активні здібності або інші ресурси. Клас оперує елементами інтерфейсу, такими як повзунки (sliders) та текстові поля, й постійно оновлює їх відповідно до актуального стану персонажа. Для отримання цих даних HUDController звертається до об'єкта гравця, зазвичай через клас Actor або подібну логіку. Цей елемент інтерфейсу є постійно

активним, але може тимчасово приховуватись, коли запускаються діалоги або кат-сцени, що потребують більшої уваги користувача.

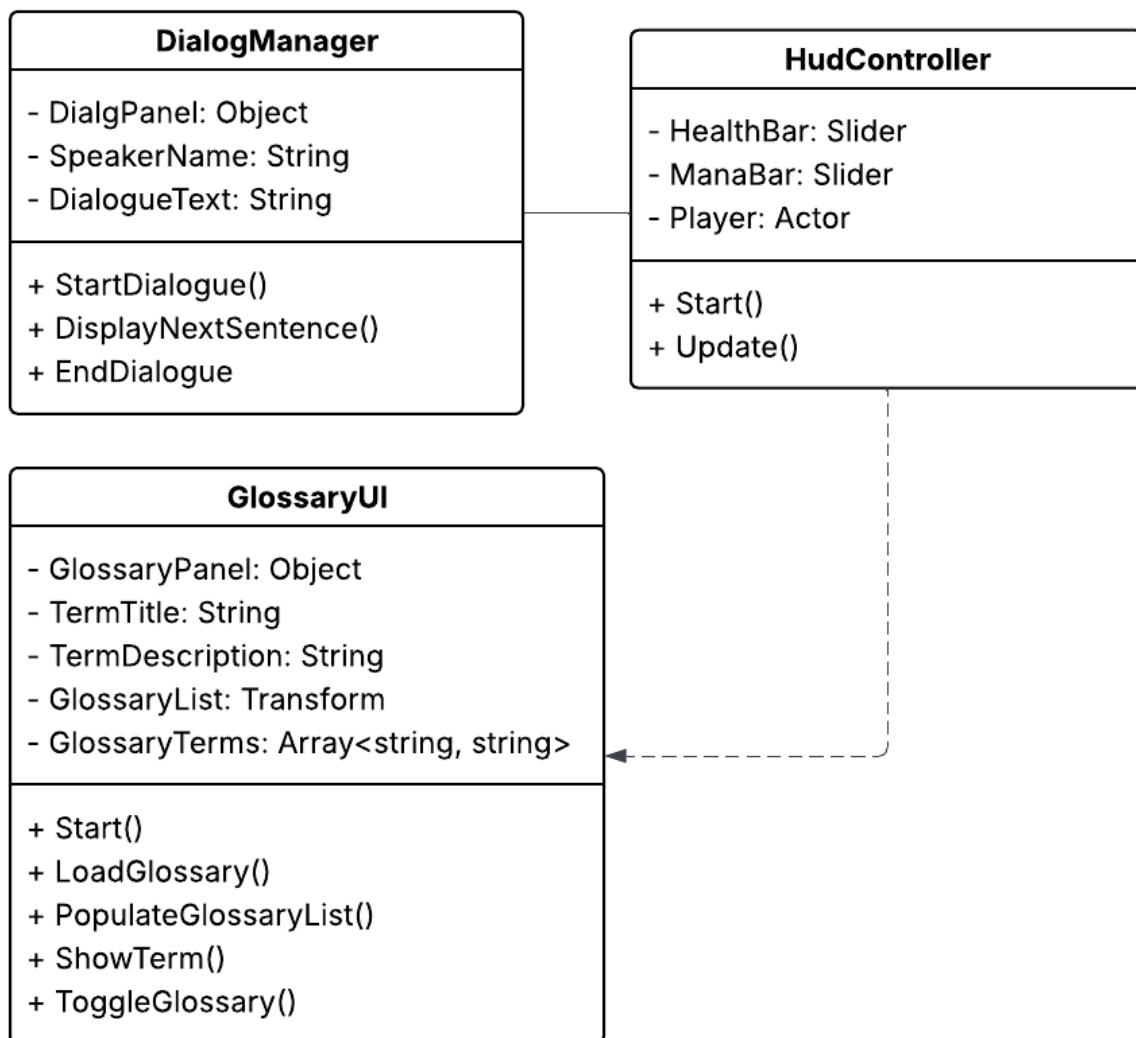


Рисунок 2.6 — Діаграма класів користувацького інтерфейсу

DialogueManager — це менеджер діалогів, який забезпечує відображення реплік персонажів у вигляді текстових блоків із підписаним іменем мовця. Клас керує чергою фраз, дозволяє послідовно перегортати їх за допомогою натискань кнопки або клавіш, і завершує діалог після вичерпання всіх рядків. Основною задачею є надання гравцю можливості брати участь у сюжетних розмовах та дізнаватися більше про світ гри. У певних випадках, коли під час діалогу згадується важливе поняття чи термін, **DialogueManager** може звертатися до глосарію (**GlossaryUI**), щоб автоматично показати пояснення або запропонувати гравцеві ознайомитися з ним пізніше.

GlossaryUI — це візуальний інтерфейс глосарію, який містить перелік ключових термінів, пов'язаних із внутрішньоігровим світом. Користувач має можливість переглядати список слів або назв, натискати на них та отримувати детальні описи. Такий функціонал є особливо корисним для ігор з глибокою міфологічною або сюжетною складовою. Глосарій може відкриватися безпосередньо гравцем, наприклад, через меню паузи, або автоматично — у відповідь на репліки в діалозі, що містять важливі поняття. Клас працює з колекцією термінів, яку зазвичай завантажує під час старту сцени, й генерує інтерфейсні кнопки для кожного з них.

2.3 Розробка моделі візуального інтерфейсу програмного комплексу

Візуальний інтерфейс є невіддільною частиною користувацького досвіду, і в контексті розробки гри за мотивами української міфології він відіграє не лише функціональну, а й естетичну та символічну роль. Завдяки візуальному оформленню інтерфейсу гравець не просто взаємодіє з механіками гри, а занурюється у глибокий, автентичний світ народних образів, орнаментів і вірувань, що створюють особливу атмосферу містики й давнини. Тому під час створення моделі інтерфейсу було вирішено поєднати сучасну зручність, знайому гравцям жанру метроїдванія, з унікальною стилістикою, що черпає натхнення зі слов'янської естетики, зокрема української народної культури.

Інтерфейс оформлений у стилістиці витинанок, гравюр і традиційних символів, візуально споріднених з язичницькими мотивами — дерева життя, солярних знаків, вишитих елементів, що символізують захист, життєвий шлях і зв'язок із предками. Панель здоров'я головного героя-двоєдушника стилізована під оберіг або намисто з ладанкою, в якому кожен сегмент уособлює стан тіла і духу. Духовна енергія або “душевна сила”, яку гравець витрачає на спеціальні здібності, позначається мерехтливим світлом чи серпанком, що змінюється залежно від активного стану — чи перебуває герой у світі живих (Ява) чи мертвих (Нава).

Інтерфейс мінімалістичний, щоб не перевантажувати екран і дозволити гравцеві повністю зосередитися на дослідженні навколишнього світу. Всі елементи HUD делікатно інтегруються в загальний візуальний стиль — вони не відволікають, але залишаються інтуїтивно зрозумілими. Мапа гри відкривається не з самого початку, а лише після здобуття відповідного предмета — артефакту або амулета, що надає змогу “бачити шлях”. Карта оформлена у вигляді зістареного пергаменту з міфологічними знаками, що відображають розташування важливих об’єктів — вівтарів, порталних переходів, місць боїв з босами, капищ та інших сакральних точок.

У грі реалізовано повноцінне меню інвентаря, оформлене у вигляді полотняної торбини або скрині, прикрашеної різьбленням і традиційними узорами. Тут гравець може керувати предметами, переглядати зібрані фрагменти легенд, налаштовувати спорядження та читати опис кожного об’єкта. Додатково реалізований глосарій — інтерактивна “книга знань”, що автоматично поповнюється при знаходженні нових термінів, символів або персонажів. Кожен термін супроводжується коротким поясненням і художньою ілюстрацією, створеною в етнічній манері. Це дозволяє гравцеві не лише краще розуміти контекст гри, а й поступово занурюватися в українську міфологічну традицію.

Система діалогів у грі створена з акцентом на глибоку атмосферність. Репліки подаються на тлі текстурованих дощечок або сувоїв, з ефектом ручного письма. Усі ключові слова виділяються, а після першої згадки зберігаються в глосарії. Діалоги супроводжуються звуковими ефектами, які імітують стародавній шепіт, дзвін або дерев’яний скрип, підсилюючи відчуття, ніби гравець спілкується з істотою, що існує поза межами часу.

Навігаційне меню гри, включаючи стартове, паузу та налаштування, оформлено як стилізоване капище або духовне місце з кам’яними або дерев’яними об’єктами, навколо яких в’ється дим, і чути слабе тремтіння голосів. Ефекти анімацій повільні та плавні, що відповідає тональності міфологічного світу — час у ньому тече інакше, розмірено та глибоко.

Окремої уваги заслуговує інтерфейс переходу між світами Яви і Нави. У момент перемикання змінюється загальна кольорова палітра інтерфейсу: теплі жовто-червоні відтінки живого світу змінюються на холодні синьо-сірі тіні Нави. Елементи інтерфейсу ніби розчиняються в тумані або трансформуються, відображаючи зміну стану героя. Це створює ефект справжньої мандрівки між реальностями і підсилює відчуття трансформації, що тісно пов'язане з образом головного героя-двоєдушника.

2.4 Проектування таблиць бази даних і вибір типів даних полів

У рамках розробки гри в Unreal Engine 5 використовується вбудована система збереження даних, яка базується на класі USaveGame. Це дозволяє реалізувати ефективне збереження стану гри без необхідності створення зовнішньої реляційної бази даних. Проте навіть у такій системі важливим є етап проектування структури збережених даних: які саме змінні зберігаються, як вони організовані, які типи даних використовуються, і як забезпечується розширюваність структури в майбутньому.

Для реалізації системи збереження було створено власний клас, що наслідує USaveGame, наприклад, UMySaveGame. У цьому класі описані всі поля, які мають бути збережені при переході гравця до контрольної точки, завершенні сесії або зміні рівня. До таких полів належать: координати гравця (FVector), його рівень здоров'я (float), поточні здібності та перетворення (TArray<FName>), зібрані предмети (TArray<FItemData>), прогрес сюжетної лінії (TMap<FName, bool>), а також інформація про переміщення між світами Яви і Нави (наприклад, логічне поле bIsInNava).

Вибір типів даних здійснюється відповідно до внутрішньої логіки Unreal Engine та потреб проєкту. Для координатних і просторових даних доцільно використовувати FVector, для об'єктів ротації — FRotator, для флагів — bool, для лічильників — int32, а для складніших об'єктів — користувацькі структури (USTRUCT) або масиви (TArray). Наприклад, щоб зберегти інвентар гравця, створено структуру FItemData, яка містить інформацію про

ідентифікатор предмета, його кількість, стан, іконку та опис. Це дозволяє гнучко обробляти зміну кількості або властивостей предметів.

Збереження відбувається через функцію `UGameplayStatics::SaveGameToSlot`, а завантаження — через `LoadGameFromSlot`. Назви слотів ідентифікуються за допомогою унікального імені або індексу, що дозволяє створювати кілька збережень або автоматичне збереження (autosave).

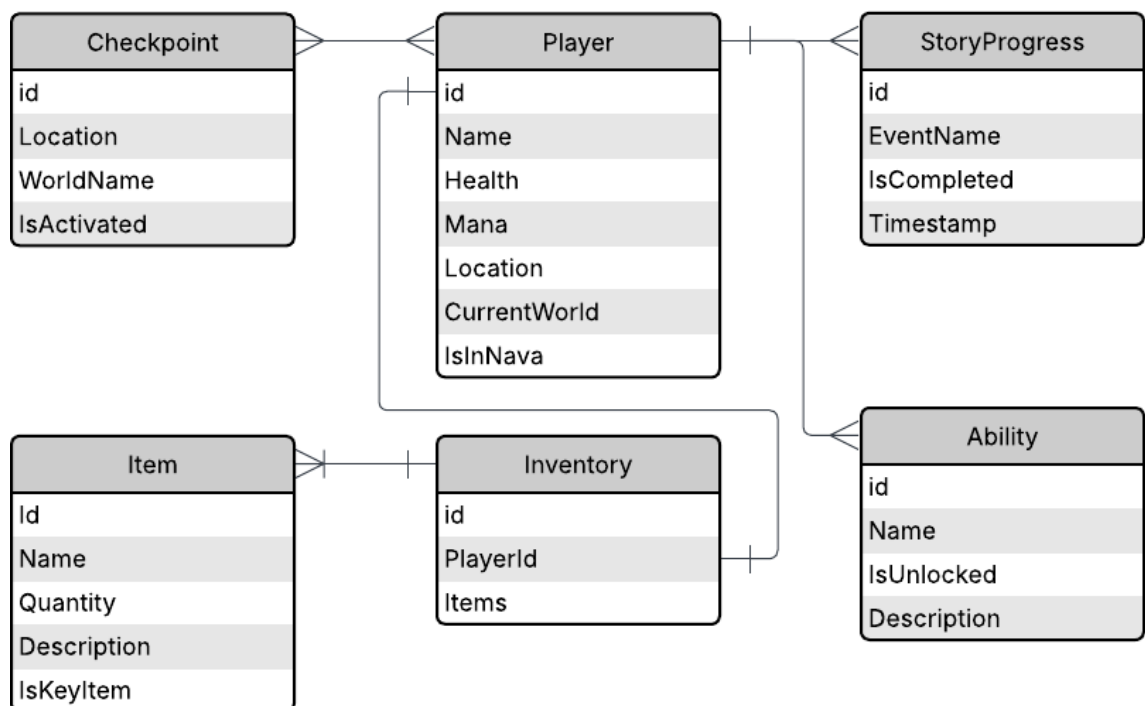


Рисунок 2.7 — ER-Діаграма таблиць бази даних

Цікавою особливістю реалізації системи збереження є використання контрольних точок, які в грі реалізовані як ліжка — символи переходу між світами або відпочинку духу. Підходячи до ліжка, гравець автоматично активує функцію збереження, що викликає `SaveGameToSlot` з оновленими даними про його стан, положення, активні квести й доступні області мапи. Це створює ефект органічного вбудування технічної функції в художній ігровий світ.

3. РОЗРОБКА ПРОГРАМНИХ МОДУЛІВ ПРОЄКТУ

3.1 Розробка основних класів програмних модулів проєкту:

У рамках розробки основних класів програмних модулів проєкту було реалізовано надзвичайно детальну, багатоетапну логіку поведінки персонажів у грі на рушії Unreal Engine 5 із використанням Blueprints. Центральними вузлами цієї комплексної архітектури виступають три класи — BP_CharParent, BP_Hero та BP_Soul, кожен із яких виконує свою чітко окреслену роль у загальній ігровій структурі та взаємодіє з іншими компонентами через події, сигнали, змінні та таймери, реалізовані засобами движка.

Почнемо з першого, базового класу BP_CharParent, який слугує фундаментом для всіх похідних персонажів у грі. Саме тут закладається центральна механіка отримання шкоди, що відбувається через функцію ReceiveDamage. Коли відбувається контакт із ворожим об'єктом чи активація колайдери супротивника, до BP_CharParent надходить сигнал із інформацією про величину ушкодження. Цей сигнал зчитує внутрішню змінну Health, після чого відбувається її зменшення на відповідне значення. Важливо, що процес зменшення здоров'я супроводжується одночасним запуском візуальних та аудіоефектів — наприклад, коротких спалахи світла на краях екрану чи сплеску часток крові, а також відтворенням характерного звукового ефекту шкоди.

Після того, як значення Health змінено, система не миттєво перевіряє стан персонажа на смерть, а чекає визначену кількість часу, реалізовану через Delay. Це дає змогу відтворити плавні переходи між станами: по-перше, надати шанс анімації «відскоку» або «падаючого удару» завершитися коректно, а по-друге, додати динамічного затримання, коли екран може тремтіти або забарвитися в холодні відтінки, перш ніж остаточно оголосити про смерть. Після завершення цієї затримки відбувається перевірка булевої змінної IsDeath: якщо Health впав нижче нульового порога, запускається механізм обробки смерті. Але оскільки BP_CharParent – це абстрактний батьківський клас, він не приймає остаточного рішення самостійно: залежно

від того, який саме клас наслідує BP_CharParent (BP_Hero чи BP_Soul), викликається або функція HeroDeath, або SoulDeath, кожна з яких містить власний сценарій подальших дій.

Сам механізм переходу між героями та «душами» реалізований теж на рівні BP_CharParent. Коли герой гине, в BP_CharParent міститься логіка, яка автоматично викликає функцію створення нового актора типу BP_Soul. Після того як новий об'єкт — «душа» героя — з'являється у світі, рушник керування передається саме йому через виклик Possess у PlayerController. Це відбувається надзвичайно акуратно: старий актор BP_Hero не просто безпосередньо знищується, а спершу йому дається кілька фреймів на програвання анімації «мертвого тіла». Завдяки такій послідовності дій гравець бачить, як його герой впав, але в ту ж мить з'являється ефірна форма душі й починає свій рух по рівню.

Наступним у ланцюгу йде клас BP_Hero, що реалізує весь спектр управління живим персонажем до його можливої загибелі. Тут було проведено велику роботу над механікою переміщення: окрім звичайного бігу та ходьби з різною швидкістю, закладена можливість виконати стрибок із фіксованою силою. Ще більш глибокою є реалізація подвійного стрибка. Змінна Jump Max Count контролює максимальну кількість стрибків у повітрі: якщо герой зачепив поверхню, лічильник стрибків обнуляється, але до досягнення ліміту в повітрі можна сягати додаткових стрибків, що надає грі відчуття більшої свободи. Коли ж мова заходить про ковзання, у BP_Hero присутні два окремі режими: WallSliding та Slide. При першому герой може торкатися стіни під певним кутом і плавно сповзати вниз, контролюючи швидкість падіння; при другому — нахиляючись у присіданому положенні, швидко ковзати по землі, уникаючи перешкод із меншим зіткненням корпусу.

Не менш ґрунтовно пропрацьовані бойові дії героя: у BP_Hero передбачено як «легку» атаку, так і «важку», кожна з яких викликає окремі анімації та обробляє свої власні колайдери для реєстрації пошкоджень ворогам. Перемикачі станів (State Machine) у Blueprints слідкують за тим, чи

герой зараз атакує, чи може розпочати комбо, чи перебуває в стані відскоку від блоку, що дозволяє органічно поєднувати різні рухи в одну анімаційну послідовність. Важливо, що кожна атака має свою затримку «відкриття вікна» (hit window) та фази, що створює багат шарову динаміку протистояння.

Ще одним ключовим елементом BP_Hero є система збереження гри. На події Event BeginPlay перевіряється наявність раніше збереженого слоту через виклик LoadGameFromSlot із використанням класу BP_SaveGame. Якщо дані знайдені, то позиція героя, його поточний рівень здоров'я, ману, досвід та всі інші параметри миттєво підвантажуються в відповідні змінні. На екрані при цьому може з'явитися ефект завантаження, а сам герой автоматично телепортується в ту точку рівня, де його залишили в попередньому сеансі. Якщо ж дані не знайдені, запускається стандартна процедура початку нової гри з дефолтними параметрами.

Однією з найбільш оригінальних механік, реалізованих у BP_Hero, є можливість «відродження» за допомогою душі. Якщо після смерті героя гравець керує BP_Soul і активне збереження виявлено, відбувається зворотний процес переселення: створюється новий актор BP_Hero у тій же точці, де перебуває душа, гравцю передається контроль над цим актором, а душа одразу знищується. Ця механіка потребує дуже точної взаємодії з класом збереження, адже потрібно не лише відновити всі параметри героя, але й очистити всі тимчасові стани душі, її анімації, ефекти й активні таймери.

І, нарешті, клас BP_Soul, що демонструє особливий тимчасовий режим, у який потрапляє гравець після смерті героя. Досить докладно пропрацьовані дві основні здібності: політ і ривок. Подія IA_Fly є одночасно і простою за своєю суттю, і надзвичайно гнучкою в реалізації. При початковому натисненні перевіряється булева змінна Is Flying. Якщо вона дорівнює false, через функцію Set Movement Mode персонаж переходить у режим Flying, а через Create Event і Set Timer by Event ініціюється виклик функції WasteMana() із циклічністю в 0.1 секунди. У середині WasteMana() відбувається віднімання певної кількості одиниць мани (змінна Current Mana зменшується), одночасно

може запускатися візуальний індикатор виснаження ресурсу на HUD чи ефект розсіювання часток. Якщо ж під час польоту натиснути кнопку знову, відбувається переведення в режим Falling і через Clear and Invalidate Timer by Handle відбувається скасування циклічного витрачання мани, що дає гравцю свободу самостійно вирішувати, коли зупинитися.

У діаграмі EventGraph BP_Soul також міститься блок Event BeginPlay, де відбувається встановлення початкових значень змінних Health, Max Mana та Current Mana. Одразу після цього запускається подія Create Event / Set Timer by Event для RestoreMana() з інтервалом 0.1 секунди. Ця функція перевіряє, чи не досягнуто вже максимальної мани; якщо ні, додає до Current Mana невелику кількість пунктів (наприклад, 1 одиницю або певний відсоток). Таким чином, у механіці душі створено збалансований цикл, в якому гравець змушений рахуватися з власними ресурсами: летіти можна довго, але не безмежно, оскільки ману потрібно витратити і поповнювати.

Ривок у BP_Soul — подія IA_Dash — реалізовано через ще одну пару булевих змінних: Can Dash (дозвіл на виконання ривка) та Is Dash (активний стан ривка). Після натискання кнопки перевіряється Can Dash; якщо true, то встановлюється Is Dash = true, зберігається напрямок останнього руху гравця, отриманий через Get Last Movement Input Vector, і передається в змінну Dash Direction. Далі створюється подія ResetDash() із затримкою в 0.3 секунди: саме по його завершенні Can Dash знову стає true, що означає відновлення здатності ривка. Відтворення самого руху відбувається у Event Tick: щок cadрово перевіряється, чи активний Is Dash, і якщо так — викликається Add Movement Input із Dash Direction та масштабом 1.0. Такий підхід гарантує миттєве, вибухове переміщення душі в обраному напрямку, яке триває лише одну мить, але може бути вирішальним у ситуації, коли потрібно уникнути небезпеки або дістатися потрібного об'єкта.

Нарешті, стрільба в BP_Soul прив'язана до події IA_Shoot: перевірка змінної Can Shoot, установка Is Attacking = true, виконання Delay на 0.7 секунди і повернення змінних у початковий стан. Ця проста, але водночас

ефективна механіка обмежує частоту пострілів, позбавляючи гравця можливості безупинно «спамити» атаку, і разом із тим дозволяє точно розрахувати траєкторію вистрілу чи кількість одночасно випущених снарядів, залежно від реалізації функції Shoot() в самому Blueprint.

Узагальнюючи, BP_Soul демонструє високий рівень деталізації та багатоступеневу обробку кожної дії: від ініціації через Enhanced Input System до складних взаємодій із внутрішніми таймерами, булевими прапорцями та візуальними ефектами. Цей мультишарний підхід дозволяє створити ігрову механіку, що одночасно є гнучкою для розробників і інтуїтивною для гравців, а також органічно вписується в загальну архітектуру BP_CharParent і BP_Hero, утворюючи єдиний злагоджений механізм, здатний надати користувачу незабутній досвід у віртуальному світі.

3.2 Розробка візуального інтерфейсу програмного комплексу

У процесі створення візуального інтерфейсу користувача для гри в Unreal Engine 5 я зосередився на тому, аби кожен елемент інтерфейсу не просто виконував свою базову функцію, а дійсно відображав атмосферу української міфології та підсилював ігровий досвід. Створення всіх графічних компонентів розпочалося із побудови загальної ієрархії виджетів у редакторі UMG (Unreal Motion Graphics). Перший і найважливіший виджет, який був реалізований, — це WBP_HUD (Widget Blueprint для Heads-Up Display), у якому розташовано кілька ключових графічних елементів: індикатор здоров'я, індикатор душевної енергії, спливаючі підказки та часові таймери для активації тимчасових ефектів. Для індикатора здоров'я було зроблено наступне: у вікні виджета додано компонент ProgressBar і задано його стиль, скопійований із старовинного деревориту, — з коричневим фоном, обрамленим стилізованим орнаментом, та зеленуватим заповненням, що нагадує колір молодого листя. Далі у Blueprint'і самого героя (клас BP_Hero) було створено змінну HealthPercent, яка приймає значення між 0.0 і 1.0 і в залежності від поточного здоров'я персонажа динамічно змінюється. У Event

Graph героя під час кожного отримання або відновлення здоров'я спрацьовувала функція UpdateHealthHUD, яка за допомогою референсу на віджет викликала метод SetPercent(HealthPercent) для оновлення внутрішнього стану ProgressBar. Щоб забезпечити зв'язок між класом героя і віджетом, в BP_Hero наприкінці функції BeginPlay було додано ноди Create Widget (клас — WBP_HUD) та Add to Viewport, а результат (посилання на створений віджет) зберігався у змінній HUDReference. Таким чином, вже на старті рівня інтерфейс з'являвся на екрані, і герой міг динамічно відправляти до нього оновлені дані зі своєї логіки.

Одночасно з індикатором здоров'я у тому самому WBP_HUD було додано індикатор душевної енергії — компонент ProgressBar, який спочатку мав прозорий фон і текстуру, схожу на візерунок стародавньої вишивки. Цей прогрес-бар прив'язаний до змінної SoulEnergyPercent у герої, яка відображає поточний запас енергії для використання спеціальних здібностей у стані духа. Як тільки гравець переходить у режим душі (здійснивши перехід із класу BP_Hero до BP_Soul), у BP_Soul також викликається функція UpdateSoulHUD, що змінює цей бар. Якщо ж герой повертається до живого стану, бар поступово відновлюється залежно від відновлюваних ресурсів.

Після того, як базовий HUD було налагоджено, була почата робота над екраном поразки (Game Over). Для цього створено окремий User Widget під назвою WBP_GameOver. У ньому графічно розміщено кілька елементів: фон, який симулює затемнення простору навколо персонажа з розмиттям країв; текст «Game Over» з використанням старослов'янських шрифтів, стилізованих під різьблену дерев'яну табличку; а також дві кнопки — «Restart Level» та «Exit to Main Menu». Логіка відображення екрану Game Over реалізована у Blueprint'і BP_CharParent під час спрацьовування функції HeroDeath. Коли змінна IsDeath стає істинною, виконується послідовність: спочатку через ноду Set Game Paused гра переходить у паузу, потім з контексту геймплею викликається Create Widget (клас — WBP_GameOver), а отриманий віджет додається до в'юпорту через Add to Viewport. Паралельно, щоб уникнути

зайвих маніпуляцій у майбутньому, у змінній `GameOverWidgetRef` зберігається посилання на створений віджет, що дозволяє в перспективі підписатися на події натискання користувача, наприклад, коли гравець вирішить перезапустити рівень. Для кнопки «Restart Level» налаштовано обробник подій `OnClicked`, який викликає функцію `Open Level` із вказаною назвою поточного рівня. Для кнопки «Exit to Main Menu» — відповідний виклик на відкриття головного меню. Щоб виглядало атмосферніше, перед викликом `Open Level` було додано невеликий таймер (нібито персонаж провалюється в темряву) через `Delay`, після чого плавно перестрибували на екран головного меню також із використанням віджета `WBP_FadeOut`.

Меню паузи (`WBP_PauseMenu`) розроблялося паралельно з `Game Over`. Головною особливістю цього меню стало те, що його виклик не просто призупиняє гру, а й тимчасово змінює інтерфейс героя: аналогічно до затемнення під час `Game Over`, екран підсвічується холодними відтінками сірого та блакитного, що повторюють кольори північних українських зим. Меню складається з п'яти кнопок — «Continue», «Restart Level», «Settings», «Main Menu» і «Quit Game». Перші дві функціонують за стандартною схемою: «Continue» через ноду `Set Game Paused(false)` знімає паузу й видаляє віджет з в'юпорту (`Remove from Parent`), а «Restart Level» викликає `Open Level` безпосередньо. Кнопка «Settings» підключена до власного підменю налаштувань звуку та графіки — також було реалізовано `Slider` для гучності музики, `Slider` для гучності звукових ефектів та випадаючий список `ComboBox` для вибору якості тіней або роздільної здатності текстур. Важливо, що всі ці налаштування миттєво зберігаються у `SaveGame` через клас `BP_SettingsSave`, тож гравець бачить результат налаштування в реальному часі. Кнопка «Main Menu» — це виклик `Open Level` для сценки головного меню, і тут також було додано плавну анімацію зникнення кнопок через `Widget Animation`, де кожна кнопка зсувається за межі екрана під кутом і зменшується в розмірі, створюючи враження старовинної книги, що закривається.

Коли гравець перебуває у процесі переходу між рівнями або при запуску гри, використовуються віджети WBP_FadeIn та WBP_FadeOut, які реалізують плавний перехід із затемненням чи появою простору. Технічно це виглядає так: у батьківському GameMode у функції BeginPlay створюється WBP_FadeIn за допомогою Create Widget, додається віджет до в'юпорту, запускається відповідна анімація (Animation Timeline у самому віджеті, де керується властивістю Opacity чорного фону), а коли анімація завершена, через ноду OnAnimationFinished автоматично викликається Remove from Parent, щоб інтерфейс зник. Аналогічно, перед викликом функції Open Level запускається WBP_FadeOut, чекає на її завершення (Delay із тривалістю анімації) і вже потім змінюється рівень. У підсумку ці плавні переходи створюють відчуття «повільного дихання» світу, коли персонаж із глибоким подихом перетинає кордони локації.

Ще одним важливим елементом UI, пов'язаним із міфологічною складовою, стала реалізація виджета WBP_LorePopup — невеликого спливаючого вікна, яке відображається кожного разу, коли гравець взаємодіє з артефактом чи встеляє спеціальну зону, де з'являються фрагменти легенд. Коли герой знаходиться неподалік певного об'єкта (ALoreObject), у його Blueprint є область OnComponentBeginOverlap, яка перевіряє, чи це активний об'єкт міфології. Якщо так, виконується нода Create Widget (клас — WBP_LorePopup), і в цей новостворений віджет передаються дані через змінну LoreText (тип Text), після чого додаються до його в'юпорту. Через таймер (Delay) цей віджет поступово зникає, демонструючи текст із ефектом «плавного розгортання» — використовується RichTextBlock з персоналізованими стилями, щоб ключові слова, наприклад «Мавка», «Вій» чи «Коцїй», були підкреслені чи виділені певним кольором. Після закінчення таймера допоміжно викликається Remove from Parent, знищуючи віджет із екрану.

Усі ці взаємодії між Blueprints забезпечують синхронність між логікою гри та графічним інтерфейсом. Зокрема, код із Event Tick у героя спостерігає

за станом змінної `bIsInSpiritWorld` і, якщо вона змінюється, викликає функцію `UpdateSpiritOverlay` — цей метод додає або видаляє над головою персонажа напівпрозорий фільтр (`WBP_SpiritOverlay`), що створює ефект «примарного бачення» Нави. У самому `WBP_SpiritOverlay` реалізовано невеликий анімаційний цикл, де текстури із затумененими контурами предметів прослизують у кадрі, створюючи ілюзію, що герой бачить приховані знаки.

Особливу увагу було приділено не лише створенню самих віджетів, а й їх відлагодженню на різних роздільних здатностях екрана. Для цього кожен UI-елемент був обгорнутий у `Size Box` або `Canvas Panel` із прив'язаними величинами в відсотках (`Percentage`), щоб інтерфейс “тягнувся” і не розтягувався непропорційно. Також на початку кожного рівня в `BP_PlayerController` є виклик `Set Input Mode Game and UI`, що забезпечує одночасну роботу геймплейного керування та реагування на натискання миші по елементах інтерфейсу.

ВИСНОВОК

На основі аналізу української міфології, особливостей жанру метроїдванія та сучасних підходів до розробки комп'ютерних ігор було спроектовано і реалізовано прототип гри, що поєднує класичну геймплейну структуру з елементами національного фольклору.

У результаті виконання роботи розроблено комп'ютерну гру в жанрі метроїдванія з використанням рушія Unreal Engine 5, створено базову архітектуру проєкту, реалізовано основні модулі ігрової механіки, візуальне оформлення, користувацький інтерфейс і базові алгоритми взаємодії з ігровим світом.

Розробка дозволяє продемонструвати ефективність поєднання технічних засобів сучасної ігрової індустрії з культурно-освітнім контентом, сприяє популяризації української міфології, а також може слугувати основою для подальшого розширення проєкту, його комерціалізації або використання в навчальних цілях.

ПЕРЕЛІК ПОСИЛАНЬ

Малоруські народні перекази й оповідання / М. Драгоманов; Південно-Зап. Відд. Імперат. Рус. Геогр. О-ва. - Київ: Тип. М. П. Фріца, 1876. - XXV, 434, [2] с.

Праці етнографічно-статистичної експедиції в Західно-Руський край. Південно-Західний відділ. Матеріали і дослідження : [у 7 т.] / зібрав П. П. Чубинський. - Санкт-Петербург : [б.и.], 1872 - 1878.

Етнографічний збірник / вид. Етногр. коміс. Наук. т-ва ім. Шевченка. - Львів : 3 друк. НТШ, 1895 - 1929.

Українці: народні вірування, повір'я, демонологія / [упоряд., прим., біогр. нариси: А. П. Пономарьов, Т. В. Косміна, О. О. Боряк ; вступ. ст. А. П. Пономарьов]. - 2-ге вид. - Київ: Либідь, 1992. - 637, [1] с. : іл. - (Пам'ятки історичної думки України).

Мавка. Лісова пісня [Відео] / sweet.tv. - 2023. - 1 відеофайл. - Режим доступу: <https://sweet.tv/cartoon/25931-mavka-lisova-pisnya>.

Black Book [Електронний ресурс] / Red Cerberus. - 2021. - Відеогра. - Режим доступу: https://store.steampowered.com/app/1138660/Black_Book/?l=ukrainian.

GSC Game World. [S.T.A.L.K.E.R. серія ігор] [Електронний ресурс] / GSC Game World. - 2007–2025. - Відеогра. - Режим доступу: <https://www.gsc-game.com/index.php?lang=ua>.

Cossacks: European Wars [Електронний ресурс] / GSC Game World. - 2001. - Відеогра. - Режим доступу: <https://www.gog.com/game/cossacks>.

The Witcher 3: Wild Hunt [Електронний ресурс] / CD Projekt Red. - 2015. - Відеогра. - Режим доступу: <https://thewitcher.com/witcher3>.

Assassin's Creed Origins [Електронний ресурс] / Ubisoft. - 2017. - Відеогра. - Режим доступу: <https://www.ubisoft.com>.

Українська міфологія [Електронний ресурс] // Bestiary.us. - Режим доступу: <https://www.bestiary.us/cult/slavjanskaja-mifologija/ukrainskaja-mifologija>