

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ТЕХНОЛОГІЙ ЦИФРОВОГО РОЗВИТКУ**

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Інтерактивна 3D-головоломка на базі ігрового рушія
UnrealEngine 5»

на здобуття освітнього ступеня бакалавра
зі спеціальності 121 Інженерія програмного забезпечення
освітньо-професійної програми «Технології цифрового розвитку»

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

Софія ОГНЕВА

(підпис)

Виконала: здобувачка вищої освіти групи ТЦР-42

Софія ОГНЕВА

Керівник: Іван ОЛЕЙНІКОВ
Ст. викладач

Рецензент: _____

Київ 2026

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

Навчально-науковий інститут інформаційних технологій

Кафедра Технологій цифрового розвитку

Ступінь вищої освіти Бакалавр

Спеціальність 121 Інженерія програмного забезпечення

Освітньо-професійна програма «Технології цифрового розвитку»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Технологій цифрового розвитку

_____ Вікторія ЖЕБКА

« ____ » _____ 2026 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Огневій Софії Євгеніївні

1. Тема кваліфікаційної роботи: «Інтерактивна 3D-головоломка на базі ігрового рушія UnrealEngine 5»

керівник кваліфікаційної роботи старший викладач кафедри Іван ОЛЕЙНИКОВ,
затверджені наказом Державного університету інформаційно-комунікаційних
технологій від «20» лютого 2026 року № 51.

2. Строк подання кваліфікаційної роботи «19» травня 2026 року.

3. Вихідні дані до кваліфікаційної роботи: Теоретичні й практичні матеріали створення тривимірних інтерактивних застосунків, документація ігрового рушія UnrealEngine 5 та системи візуального редагування Blueprints, розбір сучасних 3D-головоломок та способів побудування просторових завдань, зразки фізичних взаємодій об'єктів у реальному часі.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Аналіз сучасного стану та тенденцій розвитку ігрових проектів у жанрі 3D-головоломок.
2. Порівняльний аналіз інструментарію ігрових рушіїв для розробки тривимірних ігор та обґрунтування вибору UnrealEngine 5.

3. Проектування архітектури та механік інтерактивної взаємодії гравця з ігровим середовищем.
4. Технічна реалізація ігрових систем: логіка персонажа від першої особи, система фізичних маніпуляцій об'єктами та побудова рівнів засобами Blueprints.
5. Тестування функціональності ігрових механік та оцінка продуктивності графічної підсистеми.

5. Перелік графічного матеріалу: *презентація*

1. Аналіз аналогів.
2. Вимоги до програмного забезпечення.
3. Програмні засоби реалізації.
4. Діаграма варіантів використання.
5. Алгоритм роботи застосунку.
6. Діаграма класів.
7. Екранні форми.
8. Апробація результатів дослідження

6. Дата видачі завдання «20» лютого 2026 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Підбір та аналіз науково-технічної літератури	27.02-07.04.2026	виконано
2	Аналіз та дослідження існуючих аналогів	27.02-10.04.2026	виконано
3	Огляд і порівняння ігрових рушіїв для 3D-розробки, обґрунтування вибору UnrealEngine 5	12.04-15.04.2026	виконано
4	Проектування архітектури	15.04-24.04.2026	виконано
5	Програмна реалізація застосунку	19.04-18.05.2026	виконана
6	Тестування застосунку	10.05.2026	виконано
7	Оформлення роботи: вступ, висновки, реферат	11.05-17.05.2026	виконано
8	Розробка демонстраційних матеріалів	23.04-17.05.2026	виконано
9	Попередній захист роботи	18.05.2026	виконано

Здобувач(ка) вищої освіти

(підпис)

Софія ОГНЕВА

Керівник

кваліфікаційної роботи

(підпис)

Іван ОЛЕЙНИКОВ

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 41стор., 1 табл., 2рис., 10 джерел.

Мета роботи-Підвищення якості ігрового враження та залучення гравців шляхом розробки, інтерактивної 3D головоломки з використанням сучасних технологій візуалізації та просторової взаємодії

Об'єкт дослідження -процес розробки 3D-головоломки та проектування інтерактивного ігрового оточення

Предмет дослідження – Методи,алгоритми та інструменталні засоби створення ігрових продуктів,а саме 3D головоломок із механіками просторової взаємодії на базі ігрового рушія UnrealEngine 5 та візуального пргграмуванняBlueprints

Короткий зміст роботи: У роботі обґрунтовано концепцію та створено функціональний прототип інтерактивної 3D-головоломки. Ігровий процес комбінує дослідження тривимірного простору з новими механіками впливу та оточення.

Застосовано гнучку та модульну структуру ігрових систем із використанням патернів програмування. Здійснено огляд ринку сучасних 3D-головоломок, виконано порівняльний аналіз ігрових рушіїв,щообґрунтував вибір UnreaEngine 5 завдяки його системам нового поколінняLumen, Nanite та EnhancedInputSystem.

Сферою використання застосунку є індустрія створення незалежних ігор,а також освітній процес

КЛЮЧОВІ СЛОВА: 3D-ГОЛОВОЛОМКА, UNREALENGINE 5, BLUEPRINTS, ІНТЕРАКТИВНЕ СЕРЕДОВИЩЕ, ГЕЙМДИЗАЙН

ЗМІСТ

ВСТУП.....	8
1 АНАЛІЗ ІСНУЮЧИХ ЗАСОБІВ ТА ПОСТАНОВКА ЗАДАЧІ	11
1.1 Опис предметної області	11
1.2 Аналіз потреб користувача та постановка задачі.....	12
1.3 Формулювання цілей та функціональних вимог до системи	13
1.4 Нефункціональні вимоги	14
1.5 Дослідження існуючих аналогів	14
1.5.1 The Witness	14
1.5.2 The Talos Principle	14
1.5.4 Superliminal (Базова версія).....	15
1.5.5 Superliminal (Розширена версія).....	15
2 ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ	16
2.1 Архітектура програмної системи	16
2.2 Вибір технологій та інструментів реалізації.....	17
2.3 Розробка інтерфейсу користувача.....	18
2.4 Реалізація функціональних модулів.....	18
ЗАСОБИ РЕАЛІЗАЦІЇ	20
3.1 Огляд технологій.....	20
3.2 Порівняння аналогів технологій.....	23
4. АНАЛІЗ РЕЗУЛЬТАТІВ ТА ПЕРСПЕКТИВИ ПОДАЛЬШОГО РОЗВИТКУ ...	25
4.1 Аналіз результатів.....	25
4.2 Результати тестування системи	25
4.3 Перспективи подальшого розвитку системи	26
4.4 Практична та економічна цінність розробленої системи.....	26
ВИСНОВКИ	28
ПЕРЕЛІК ПОСИЛАНЬ	30
ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ.....	31
ДОДАТОК Б. ЛІСТИНГИ ПРОГРАМНИХ МОДУЛІВ	37

ВСТУП

Актуальність сфери інтерактивних розваг та геймдеву нині розвивається шаленими темпами. Особливу цікавість у гравців викликають тривимірні відеоігри, які не просто розважають, а й дозволяють повністю поринути в атмосферу та активно задіяти логічне мислення. Проте розробка таких проєктів це доволі складний процес, який вимагає точного проєктування механік взаємодії, динамічної зміни геометрії рівнів та правильного розрахунку фізичних зіткнень в реальному часі.

Сьогодні на ринку все ще багато рішень, які спираються на спрощені двовимірні алгоритми. Це суттєво обмежує ігровий досвід і заважає гравцеві відчувати повне занурення. Також, якщо створювати логіку рівнів та взаємодію акторів без гнучкої модульної архітектури, гра починає сильно навантажувати комп'ютер і стає практично непропорційною. Саме тому зараз так важливо створювати тривимірні ігри-головоломки, де просторова взаємодія об'єктів повністю автоматизована, а оригінальні механіки геймплею реалізовані на базі сучасного та потужного ігрового двигуна.

Об'єкт дослідження: процес програмної реалізації тривимірних інтерактивних застосунків у жанрі логічних ігор.

Предмет дослідження: програмне забезпечення та геймплейна логіка, що відповідають за автоматизовану взаємодію гравця з тривимірним середовищем рівня, з урахуванням фізичних зіткнень та тригерних станів об'єктів.

Мета роботи: розробити та зібрати інтерактивну 3D-головоломку в UnrealEngine 5, використовуючи оптимізовані механіки для просторових маніпуляцій.

Для досягнення цієї мети в бакалаврській роботі вирішено такі завдання:

- Розібрати наявні аналоги серед тривимірних головоломок, звернувши особливу увагу на те, як там влаштована взаємодія гравця з геометрією кімнат та логічними завданнями;

- Визначити ключові технічні параметри для проектування 3D-простору та з'ясувати, як фізичний двигун впливає на стабільність обробки зіткнень інтерактивних об'єктів;
- Зібрати ігровий застосунок на UnrealEngine 5, налаштувати керування персонажем від першої особи та прописати логіку просторових маніпуляцій;
- Провести комплексне тестування готової гри: перевірити точність спрацьовування тригерів, стабільність фізики, частоту кадрів та загальну ефективність роботи системи.

Методи дослідження. Для виконання поставлених завдань на різних етапах розробки було застосовано комплекс методів. Метод системного аналізу використовувався для вивчення предметної області та аналізу існуючих ігор-аналогів. Метод проектування сприяв побудові модульної архітектури майбутнього ігрового застосунку. Метод програмного експерименту дозволив виконати налаштування фізичних колізій та оптимізацію тривимірних об'єктів. Методи тестування системи були задіяні для перевірки стабільності гри та вимірювання параметру частоти кадрів.

Наукова новизна розробки полягає у створенні тривимірного ігрового застосунку, що поєднує гнучку логіку взаємодії з об'єктами від першої особи з сучасними інструментами автоматизації процесу розробки. Запропоновано підхід до швидкого наповнення 3D-простору унікальним контентом із застосуванням генеративних нейромереж. Окрім цього, розроблено оптимізовану систему відстеження станів рівня, яка використовує тригери та трасування променів, що дозволяє реалізовувати складні просторові головоломки без надмірного навантаження на апаратні ресурси системи.

Теоретична цінність роботи полягає в узагальненні сучасних підходів до проектування 3D-ігор у жанрі головоломок та систематизації методів інтеграції штучного інтелекту в класичний процес геймдеву (на етапах препродакшну та продакшну). Матеріали дослідження розширюють розуміння взаємодії фізичних

рушіїв та візуальних скриптів при обробці просторових маніпуляцій у реальному часі.

Практична цінність полягає в тому, що розроблена гра є готовим функціонуючим прототипом з відкритим каркасом. Створена база, яка включає налаштоване керування, фізику кубів та логіку роботи дверей, може слугувати навчальним посібником для студентів або основою для розробки комерційних ігор під Windows. Система передбачає можливості масштабування: додавання складних оптичних ілюзій, зміну габаритів об'єктів через перспективу, впровадження підтримки VR-шоломів або підключення розширеної системи збереження прогресу.

Апробація результатів була проведена шляхом комплексного тестування ігрової збірки в локальному середовищі на базі операційної системи Windows. Під час перевірки були протестовані всі ключові механіки: плавне переміщення персонажа, коректне захоплення та переміщення інтерактивних об'єктів, реакція підлогових кнопок на вагу предметів, функціонування головного меню та швидкість завантаження ігрових локацій. Всі виявлені дефекти у фізиці колізій та взаємодії елементів були своєчасно усунені, що забезпечило стабільну частоту кадрів та високий рівень надійності гри.

Таким чином, розроблена тривимірна гра-головоломка повною мірою відповідає поставленим технічним завданням, функціонує стабільно, має інтуїтивно зрозуміле керування та слугує чудовою основою для подальшого розвитку та впровадження нових ігрових механік.

1 АНАЛІЗ ІСНУЮЧИХ ЗАСОБІВ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Опис предметної області

Ігри-головоломки - один із найстаріших жанрів, які люди придумали для комп'ютерів. Якщо дивитися на історію, за сорок років тут змінилося майже все [1]. Почалося все з Тетрісу в 1984 році, який написав Олексій Пажитнов [1]. Ця гра стала хітом змусивши гравців не просто думати, а реагувати на швидкість, заклавши основу для екшн-головоломок [1]. Потім розробники почали ділити жанр на шматки: з'явилися ігри з плитками (Tile-Matching) на зразок SameGame (1985 рік), Bejeweled (2001 рік) чи Threes (2014 рік) [1]. Окремо виділився пошук предметів - від Alice: AnInteractiveMuseum (1991 рік) до HiddenFolks (2017 рік) з її чорно-білою графікою, намальованою від руки [1]. Були й фізичні експерименти, наприклад, TheIncredibleMachine (1993 рік), де будували складні ланцюгові реакції, WorldofGoo (2008 рік) з липкими краплями та BridgeConstructor: Portal (2017 рік) [1]. Раніше все це працювало у 2D, на плоскому екрані. Зараз комп'ютери стали потужнішими, і загадки перейшли у 3D [2, 3]. Для розробників це означає, що проектувати рівні та підхід має бути інакшим. Рішення тепер може ховатися десь за шафою, залежати від висоти або кута огляду. У Steam зараз доволі багато таких прикладів. Наприклад, TheWitness дає гравцеві цілий безлюдний острів, де лінії на терміналах треба малювати, дивлячись на форму дерев чи тіні від каміння [5]. У TheTalosPrinciple ми граємо за робота серед руїн, копіюємо свої дії у часі та керуємо лазерами [6]. А гра Superliminal взагалі ламає звичну логіку ігрового звичок, використовуючи оптичні ілюзії: там предмет стає більшим, якщо підняти його і подивитися на далеку стіну [4, 7]. Гра зроблена на Unity як симулятор сну, де немає смерті, але є чітка зміна настрою - від звичайної лікарні до повного сюрреалізму [4, 7]. Даний дипломний проект спрямований на створення схожого тривимірного простору, але вже на базі UnrealEngine 5.

Створення ігор зараз - це довгий процес із трьох етапів: препродакшн, продакшн і постпродакшн[3]. Спочатку команда пише дизайн-документ (GDD), який служить інструкцією для всіх [3]. На етапі виробництва програмісти пишуть код, художники роблять моделі, а музиканти додають звук [3] об'єднуючи це в одну картину та створюючи атмосферу. Щоб робити це швидше, існують нові інструменти штучного інтелекту, наприклад платформа Meshy, яка допомагає з текстурами та за лічені хвилини генерує 3D-моделі речей чи героїв за простим описом, після чого їх завантажують у движок [3]. Програм для цього є безліч, а поочатківці часто беруть прості конструктори, які працюють без коду, де логіку збирають як конструктор: Construct 2, ClickteamFusion, Stencyl чи GameEditor [8]. Microsoft колись зробила KoduGameLab, щоб вчити дітей візуальному програмуванню, а для простого 3D з готовою фізикою та мережею є 3D Rad [8]. Для серйозних комерційних ігор беруть Unity, на якому написана та ж Superliminal, або складніші UnrealEngine та CryEngine, де можна зробити реалістичну графіку, круте освітлення і налаштувати фізику матеріалів [3, 7, 8]. З огляду на це, створення інтерактивної головоломки вимагає правильного вибору рушія та розуміння того, як поєднати логіку загадок із тривимірним оточенням, щоб сама гра не виглядала нудною і однотипною [9].

1.2 Аналіз потреб користувача та постановка задачі

Сучасні гравці звикли до якісної картинки та хочуть, щоб ігровий світ реагував на кожну дію. Просто клацати мишкою по статичному екрану вже нікому не цікаво. Людям подобається досліджувати простір, знаходити потаємні предмети, рухати речі, бачити, як змінюється світло, і відчувати фізику предметів від першої особи [4]. Головне для користувача - щоб керування було зручним, персонаж не застрягав у стінах, не провалювався крізь текстуру, а правила гри зчитувалися інтуїтивно без довгих текстових інструкцій [4, 8]. Також гра має нормально працювати на звичайних комп'ютерах без надто розвинутих наповнювачей, не гальмувати і не вилітати під час завантаження нових кімнат, або самої гри [9]. Тому

задача цього диплому - спроектувати та зібрати працездатну тривимірну головоломку в UnrealEngine 5 [3, 8].

Потрібно створити серію кімнат, де гравець буде вирішувати логічні завдання: переносити об'єкти, тиснути на кнопки, відкривати проходи та взаємодіяти з геометрією рівнів [4]. Також важливо налаштувати стабільну фізику і колізії, щоб предмети поводитися природно, і продумати логіку збереження прогресу та предметів, щоб гравцеві не доводилося починати все спочатку після перезапуску програми [3, 9].

1.3 Формулювання цілей та функціональних вимог до системи

Мета роботи - отримати готову комп'ютерну гру-головоломку, де логічні задачі інтегровані в тривимірні локації [3, 4]. Щоб гра працювала як треба, вона повинна виконувати конкретний набір функцій. Перш за все, персонаж має вміти ходити та взаємодіяти з предметами, крутити головою на 360 градусів за допомогою миші та впиралися в стіни, а не проходити крізь них [8].

До списку основних функцій входять:

- Можливість підійти до інтерактивного предмета (наприклад, куба), схопити його та перенести в інше місце [4];
- Перевірка логічних умов: гра повинна розуміти, коли куб стоїть на потрібній платформі-тригері [6];
- Активація ігрових елементів, наприклад, відкриття дверей після вирішення загадки [5];
- Робота головного меню, де можна почати гру, змінити налаштування або вийти [3];
- Скидання рівня, якщо гравець зайшов у глухий кут і хоче спробувати знову [7];
- Запис та завантаження поточного стану гри, щоб прогрес не зникав [9].

1.4 Нефункціональні вимоги

Ці вимоги відповідають за якість роботи програми, її швидкість та стабільність. Гра повинна видавати стабільну частоту кадрів (FPS) на комп'ютерах середньої потужності, щоб картинка непадвисала [3, 8]. Самі рівні мають завантажуватися швидко, а переходи між ними не повинні змушувати гравця довго дивитися на екран загрузки [9]. Продукт орієнтований на роботу під операційною системою Windows [8]. Код гри та структуру рівнів треба зробити гнучкими, для того, щоб пізніше можна було легко додати нові кімнати, механіки чи предмети, не ламаючи те, що вже створено [3]. Інтерфейс меню та підказок має бути простим, читабельним, без зайвих елементів, які заважають грати чи псують атмосфесру [8]. Програма має працювати надійно, без раптових вильотів та збоїв у файлах збереження чи злітання текстур[9].

1.5 Дослідження існуючих аналогів

Для розробки власного проєкту було переглянуто та зіграноу популярні тривимірні головоломки, які вже продаються у Steam [5, 6, 7], наприклад:

1.5.1 TheWitness

Ця гра пропонує великий відкритий світ без чітких підказок, куди йти [5]. Гравець просто ходить островом і розгадує лабіринти на екранахвзаємодіючи з об'єктами [5]. Головний плюс - те, як навколишній світ підказує рішення (форми хмар, лінії на стінах) [1]. Але для невеликого проєкту така структура не підходить: відкритий світ важко оптимізувати, він вимагає занадто багато ресурсів комп'ютера і може заплутати гравця [9].

1.5.2 TheTalosPrinciple

Тут рівні побудовані послідовно, а для вирішення завдань використовуються технічні пристрої - відбивання лазерів, деактиватори кулеметів та запис власних дій

у часі [6]. Гра виділяється чіткою логікою і цікавим сюжетом про роботів [1]. Головний мінус цього аналога - технічна складність, так як реалізувати систему, яка записує рухи гравця, а потім відтворює його "привида", в рамках дипломної роботи за короткий час дуже важко [3, 9].

1.5.4 Superliminal (Базова версія)

Гра заснована на зміні розміру предметів через зміну ракурса [4, 7]. Гравець бере річ, підносить до очей, і вона змінює свої габарити у просторі [4]. Плюс гри в тому, що механіка дуже оригінальна, а атмосфера спокійна, без ворогів [4]. Мінус - математика прорахунку розміру об'єктів залежно від кута огляду камери вимагає складних специфічних скриптів, які важко налаштувати без багів фізики [7, 8].

1.5.5 Superliminal (Розширена версія)

Це оновлена версія гри в Steam, де автори покращили графіку, додали нові ефекти та виправили помилки сумісності з новими відеокартами [7]. Цей приклад показує, що ігрові рушії треба постійно оновлювати [3].

Це підтверджує мій вибір UnrealEngine 5, оскільки його архітектура дозволяє спочатку зробити прості механіки, а потім покращувати візуальну частину без переписування всієї логіки з нуля [9].

2 ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ

2.1 Архітектура програмної системи

Архітектурасформовуваної тривимірної інтерактивної головоломки базується на модульному принципі, що є стандартом для сучасних ігрових систем [3, 9]. Головним ядром системи є ігровий рушій, який керує всіма підсистемами та забезпечує їхній взаємозв'язок [8]. Архітектура поділяється на кілька ключових рівнів [4, 8]:

- Рівень введення даних;
- Рівень ігрової логіки;
- Рівень фізичної симуляції;
- Рівень відображення.

Така структура дозволяє ізолювати окремі блоки коду, що спрощує процес тестування та виправлення помилок під час створення [3]. Основою взаємодії всередині архітектури є об'єктно-орієнтований підхід, де кожен елемент гри (гравець, книга, двері, ключ) є окремим програмним об'єктом із власними властивостями та функціями [9]. Об'єкти «спілкуються» між собою за допомогою системи подій та тригерів [6]. Наприклад, при вході об'єкта "шафа" у зону дії об'єкта "книга", генерується подія, яку зчитує модуль керування дверима[4]. Це забезпечує гнучкість системи та дозволяє створювати різноманітні комбінації логічних рішень та завдань на різних рівнях гри [1].

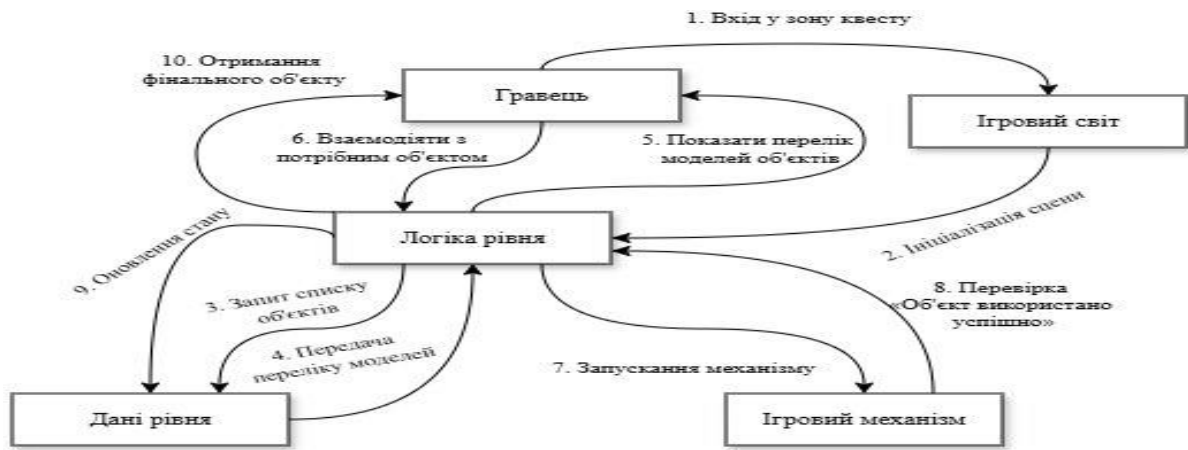


Рис. 2.1. Архітектура логіки гри

2.2 Вибір технологій та інструментів реалізації

Вибір технологій для цього проєкту обрано не випадково, а за необхідністю створення якісного тривимірного простору з надійною фізикою та сучасними візуальними ефектами [8]. Для реалізації було обрано ігровий рушій UnrealEngine 5 [3]. На відміну від легких конструкторів типу Construct 2 чи мобільно-орієнтованого рушія Unity, UnrealEngine 5 надає вбудовані інструменти для роботи з тривимірним світлом, великими наборами моделей та складною фізикою безпосередньо з коробки [7, 8]. Рушій має потужну систему візуального програмування Blueprints, що дозволяє швидко та надійно налаштовувати логіку взаємодії об'єктів без необхідності написання важкого текстового коду [8]. Для створення тривимірного контенту та наповнення рівнів декораціями використовуються можливості сучасної платформи з різними 3D моделями Fab [3]. Це дозволяє значно прискорити етап продакшину, оптимізуючи їх для імпорту в UnrealEngine 5 [3]. Програмний комплекс доповнюється стандартними інструментами операційної системи Windows для збірки кінцевого виконуваного файлу гри [8]. Такий технологічний момент є оптимальним для успішної реалізації проєкту в обмежені терміни [9].

2.3 Розробка інтерфейсу користувача

Інтерфейс користувача для інтерактивної 3D-головоломки проектується за принципом мінімалізму, щоб не відволікати гравця від дослідження тривимірного простору та вирішення логічних завдань, тим самим не відволікаючи його від занурення у атмосфери рівнів [4, 8]. Інтерфейс поділяється на два основні блоки: початкова сцена гри та внутрішньоігровий інтерфейс [3]. Початкова сцена гри містить легкі налаштування для запуску нової гри, та виходу з програми [9]. Воно розробляється з використанням простих і чітких шрифтів та контрастних елементів керування [8]. Оскільки гра не містить ворогів та механік смерті, на екрані відсутні складні смужки здоров'я чи енергії [4]. Головним елементом HUD є центральний приціл, який змінює свій зовнішній вигляд, коли гравець наводиться на об'єкт, з яким можна взаємодіяти [4, 7]. Також у системі реалізуються спливаючі текстові підказки, які з'являються на початкових етапах гри для пояснення базових правил керування щоб гравець не заплутався, але управління і без того залишається інтерактивним [8].

2.4 Реалізація функціональних модулів

Реалізація функціональних модулів у середовищі UnrealEngine 5 виконується шляхом створення окремих класів та логічних блоків [8].

Першим реалізується модуль керування персонажем від першої особи [4]. Він відповідає за обробку сигналів від клавіш пересування та миші, налаштування висоти погляду та швидкість руху [8]. Другим важливим кроком є модуль взаємодії з фізичними об'єктами [3], цей модуль у реальному часі прораховує промінь із центру екрана гравця, якщо промінь перетинає інтерактивний предмет на території взаємодії об'єкта, активується функція взаємодії, або захоплення, що дозволяє переносити річ [4, 7]. Третім модулем є логічна система тригерів та активаторів [6], він реалізує поведінку підлогових кнопок та лазерних приймачів [6]. При потраплянні потрібного об'єкта в зону тригера, модуль змінює стан пов'язаних

елементів [5]. Четвертим є модуль менеджменту рівнів, який відповідає за плавний перехід між кімнатами, повне очищення пам'яті від старих об'єктів та завантаження нової тривимірної сцени без критичних падінь продуктивності комп'ютера та пропагуванні[3, 9].

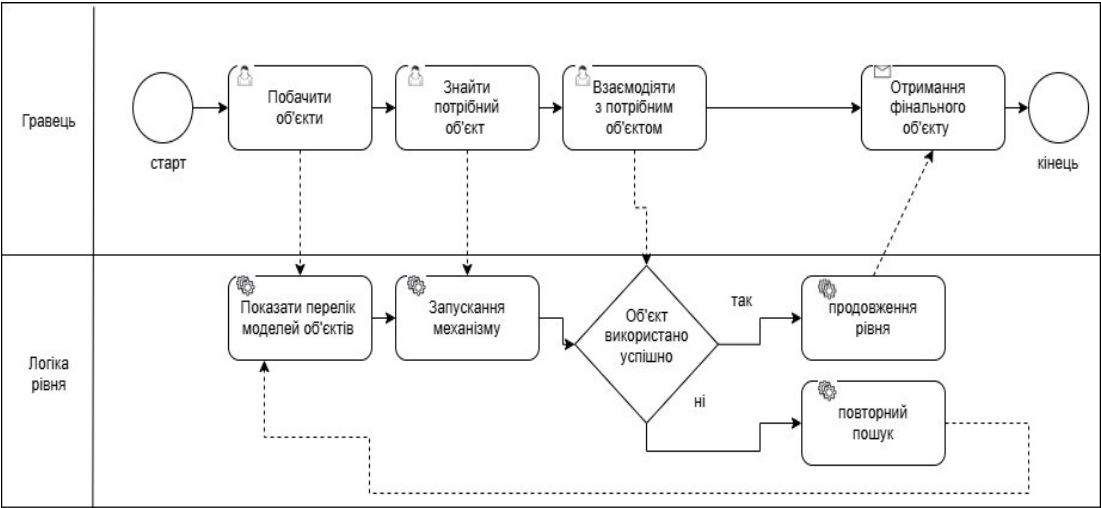


Рис.2.2.Діаграма процесів взаємодії у грі

3 ЗАСОБИ РЕАЛІЗАЦІЇ

3.1 Огляд технологій

У сучасному високотехнологічному світі процеси проектування, розробки та подальшого супроводження складних інтерактивних тривимірних застосунків потребують залучення комплексного, всебічного та глибоко інтегрованого підходу до вибору відповідного інструментарію [8]. На сьогоднішній день глобальний ринок комп'ютерних технологій та програмного забезпечення для індустрії геймдеву і віртуальної реальності характеризується наявністю величезної кількості різноманітних засобів розробки [3]. Кожна окрема програма, середовище або утиліта має свої специфічні внутрішні архітектурні особливості, унікальні обчислювальні алгоритми, вимоги до апаратної конфігурації персональних комп'ютерів, а також суттєві відмінності у порозі входження для кінцевого розробника [8, 9].

Саме тому на початкових етапах життєвого циклу створення будь-якого програмного продукту виникає критична необхідність проведення детального теоретичного аналізу наявних технологій [9]. Неправильний або необґрунтований вибір базового інструменту на перших кроках може призвести до виникнення системних помилок у майбутньому, суттєвого уповільнення темпів написання логіки взаємодії об'єктів, виникнення критичних збоїв у роботі фізичного рушія, а також до повної відсутності адекватної оптимізації готового застосунку під сучасні операційні системи [3, 8]. Якщо озирнутися на історичні етапи розвитку комп'ютерної графіки та інтерактивного софту, то можна помітити, що раніше інженерам доводилося створювати графічні конвеєри, промальовувати кожен піксель та прораховувати вектори зіткнення абсолютно вручну [3]. Це вимагало колосальних часових та інтелектуальних ресурсів. Проте в умовах сьогодення цей процес зазнав докорінної трансформації та автоматизації завдяки появі та стрімкому впровадженню спеціалізованих ігрових рушіїв, хмарних маркетплейсів та інноваційних сервісів на базі штучного інтелекту [1, 3].

У процесі детального дослідження існуючих засобів розробки було виявлено цілий спектр програмних рішень, які умовно можна розділити на кілька фундаментальних категорій [8]. Якщо розглядати рішення, орієнтовані на створення простих, переважно двовимірних ігор або спрощених казуальних застосунків, то серед розробників-початківців часто мають попит різноманітні візуальні конструктори [8]. До таких утиліт можна впевнено віднести такі середовища, як Construct 2, Stencyl або ClickteamFusion [8]. Головна архітектурна особливість та концептуальна ідея цих програм полягає в тому, що вони пропонують користувачеві роботу з так званими табличними логічними схемами подій та технологією перетягування елементів мишкою (Drag'n'Drop) [8]. Це дозволяє збирати перші демонстраційні прототипи без необхідності написання класичних текстових рядків програмного коду [8, 9].

До цієї ж групи засобів автоматизації логіки можна віднести такі інструменти, як GameEditor, а також спеціалізований навчальний програмний комплекс KoduGameLab, розроблений корпорацією Microsoft з метою наочного моделювання базових тривимірних світів у межах освітніх програм [8]. Сюди ж відноситься і конструктор 3D Rad, який має у своєму арсеналі набір заздалегідь інтегрованих фізичних пресетів для симуляції руху автомобілів або найпростіших геометричних об'єктів [8]. Проте, попри свою уявну простоту та високу швидкість освоєння, всі вищеперераховані інструменти та утиліти мають ряд критичних, неусувних обмежень [8]. Вони є абсолютно нездатними функціонувати в умовах обробки складної тривимірної геометрії високої полігональності, не підтримують сучасні технології динамічного затінення, реалістичного рендерингу та кастомних фізичних матеріалів для взаємодії твердих тіл [8, 9]. Це робить їх використання в межах кваліфікаційної бакалаврської роботи абсолютно неможливим, недоцільним та невідповідним сучасним технічним стандартам вищої освіти [3, 9].

Зовсім інший клас програмного забезпечення представляють професійні графоорієнтовані рушії високого рівня, які використовуються у світовій практиці для створення комерційних 3D-ігор з відкритими або закритими послідовними

світами [3, 8]. Беззаперечними лідерами в цьому сегменті на сьогоднішній день є такі програмні платформи, як Unity, CryEngine та UnrealEngine 5 [3, 7, 8].

Наприклад, ігровий движок Unity характеризується надзвичайно високим рівнем гнучкості архітектури, і саме на його основі було створено чимало успішних логічних проєктів світового рівня (зокрема, відома гра-головоломка Superliminal) [3, 7]. Цей рушій використовує класичну мову програмування високого рівня C# та компонентну модель побудови об'єктів [3]. Однак у контексті виконання індивідуального студентського проєкту його застосування має суттєвий недолік у вигляді величезної кількості базових систем доводиться налаштовувати або прописувати вручну з нуля [3, 9]. В умовах обмеженого часу це значно підвищує ймовірність виникнення синтаксичних помилок і затягує процес розробки [9].

Беручи до уваги всі вищезазначені фактори, аргументи та технічні вимоги, найбільш оптимальним, збалансованим та виправданим вибором для успішної програмної реалізації тривимірної головоломки став ігровий рушій UnrealEngine 5 [3, 8]. Головна унікальна особливість та перевага даного рушія полягає в наявності потужної інтегрованої підсистеми візуального кодування під назвою Blueprints [8]. Ця система дозволяє розробнику створювати об'єктно-орієнтовану логіку будь-якої складності шляхом простого поєднання функціональних вузлів та інформаційних потоків у спеціальному графічному редакторі, повністю виключаючи потребу в ручному написанні текстового коду [8]. Крім того, UnrealEngine 5 володіє інноваційним фізичним ядром Chaos, яке забезпечує феноменальну точність обробки зіткнень твердих тіл у реальному часі, що є критично важливим для нашої механіки взаємодії з кубами [3, 6, 8].

Для створення візуального ряду, текстурування та наповнення віртуальних кімнат унікальними тривимірними ассетами було організовано ефективну взаємодію між, офіційним глобальним маркетплейсом Fab та професійним пакетом тривимірного моделювання Blender [3, 8], це дозволило додати базові тривимірні об'єкти інтер'єру з уже готовими картами текстур, що значно прискорило етап препродакшну [3]. Нова хмарна платформа Fab від EpicGames об'єднала провідні світові бібліотеки тривимірного контенту, ставши надійним джерелом сумісних

матеріалів [8]. Натомість безкоштовний пакет Blender виступив як головний інструмент для фінального доопрацювання геометрії, проведення ре топології, розгортки текстур та створення спрощених меж зіткнень перед безпосереднім завантаженням моделей в UnrealEngine 5 [8].

3.2 Порівняння аналогів технологій

Для обґрунтування вибору UnrealEngine 5 як головного засобу реалізації дипломного проекту необхідно провести порівняльний аналіз ключових технологій за основними параметрами розробки [8, 9].

Таблиця 3.1

Порівняння різних рушіїв та конструкторів для створення 2D-3D ігр

Параметр порівняння	Конструктори	Unity	UnrealEngine 5
Основний тип графіки	Переважно 2D, найпростіше 3D	Універсальне 2D та 3D	Високотехнологічне 3D
Необхідність знання програмування	Не потрібно	Потрібно	Не обов'язково
Робота з тривимірним світлом та фізикою	Дуже обмежена, базові функції	Хороша, потребує додаткових налаштувань	Максимальна якість вбудованих інструментів
Швидкість створення 3D-сцен	Низька через брак інструментів	Середня, вимагає ручної збірки компонентів	Висока (Є повна інтеграція з бібліотеками)

Детально аналізуючи результати, відображені у таблиці, стає цілком очевидним, що категорія простих конструкторів програмного забезпечення повністю відкидається ще на етапі первинного відбору [8]. Вони технічно нездатні

розгорнути стабільне тривимірне середовище та забезпечити коректну симуляцію фізики [8]. З іншого боку, професійний рушій Unity є дуже сильним та гідним конкурентом, проте його використання в межах даної роботи суттєво збільшило б терміни написання коду взаємодії [3, 7]. Створення системи променів LineTrace та утримання об'єктів у руках вимагало б написання довгих, громіздких текстових скриптів мовою C# та постійного ручного прорахунку математичних векторів камери гравця, що підвищує ризик виникнення прихованих багів [3, 7, 9].

Таким чином, підбиваючи підсумок порівняльного аналізу, можна стверджувати, що вибір UnrealEngine 5 є єдино правильним, технічно обґрунтованим та логічним рішенням для реалізації даного бакалаврського проєкту [3]. Застосування його прогресивного візуального інструментарію разом із тонкою оптимізацією моделей у середовищі Blender дозволило успішно втілити в життя архітектуру 3D-головоломки [3, 8]. Це забезпечило найвищий рівень стабільності обробки логічних тригерів, оптимізувало навантаження на систему та дозволило повністю вкластися у жорсткі часові рамки, відведені на написання та захист випускної кваліфікаційної роботи [3, 8, 9].

4 АНАЛІЗ РЕЗУЛЬТАТІВ ТА ПЕРСПЕКТИВИ ПОДАЛЬШОГО РОЗВИТКУ

4.1 Аналіз результатів

У результаті виконання цього дипломного проекту було успішно розроблено та зібрано функціонуючу програмну систему інтерактивної 3D-головоломки на базі рушія UnrealEngine 5 [3, 8]. Створений продукт повністю відповідає поставленим функціональним вимогам [9]. У грі успішно реалізовано механіку переміщення від першої особи, систему точного виявлення та захоплення фізичних об'єктів перед поглядом гравця, а також логіку роботи просторових тригерів [4, 6]. Завдяки використанню передових інструментів обраного ігрового рушія, вдалося досягти високої якості тривимірного зображення та стабільного прорахунку фізичних колізій [8]. Інтеграція графічних об'єктів, зібраних у бібліотеці сучасної платформи Fab, дозволила наповнити ігрові кімнати різноманітними моделями без втрати загальної продуктивності системи [3]. Проект показав свою працездатність як програмний продукт, готовий до запуску на комп'ютерах під керуванням операційної системи Windows [8, 9].

4.2 Результати тестування системи

Після завершення фази активного виробництва було проведено детальне тестування розробленої гри [3]. Тестування виконувалося методом перевірки основних ігрових сценаріїв та аналізу стабільності ігрової логіки [9]. Результати перевірки показали, що модуль керування персонажем працює коректно, рухи є плавними, а затримки введення команд відсутні [4]. Функція захоплення та перенесення об'єктів у просторі працює, основні сюжетні предмети не провалилися крізь стіни чи підлогу та лишаються на своїх місцях очікую взаємодії з гравцем, завдяки правильному налаштуванню фізичних колізій у рушії [7, 8]. Модуль логічних тригерів успішно зчитує взаємодію об'єктів: двері між кімнатами

відкриваються лише за умови повного виконання завдання, як це й було закладено в проектній документації [5, 6]. Під час тестування інтерфейсу користувача не було виявлено збоїв у роботі головного меню чи відображенні системних підказок [9]. Загальна частота кадрів залишалася стабільною, що підтверджує успішне виконання нефункціональних вимог щодо оптимізації та надійності програмного продукту [3, 9].

4.3 Перспективи подальшого розвитку системи

Створена архітектура програмного комплексу має високий потенціал для подальшого розширення та модернізації завдяки модульному принципу побудови коду в UnrealEngine 5 [3, 8, 9]. Першим напрямком розвитку є розширення ігрового контенту: додавання нових тематичних кімнат із ускладненими логічними завданнями та новими типами інтерактивних об'єктів [1]. Другим напрямком може стати впровадження складніших механік зміни простору, наприклад, створення оптичних ілюзій або систем зміни перспективи, подібних до гри Superliminal [4, 7]. З технічного боку перспективним є впровадження підтримки систем віртуальної реальності, що дозволить гравцеві ще глибше зануритися у тривимірний світ головоломки та взаємодіяти з предметами безпосередньо руками [2]. Також розвиток системи може включати інтеграцію більш глибоких алгоритмів штучного інтелекту для динамічної зміни складності загадок залежно від швидкості проходження рівнів конкретним користувачем, що зробить гру адаптивнішою [3, 9].

4.4 Практична та економічна цінність розробленої системи

Практична цінність розробленої інтерактивної 3D-головоломки полягає у створенні готового програмного шаблону для проектування логічних завдань у тривимірному просторі, який може бути використаний іншими розробниками як база для створення власних ігор [3, 8]. Також проєкт має високу освітню цінність, демонструючи успішне застосування сучасних візуальних інструментів Blueprints та інтеграцію з Fab для прискорення виробництва контенту [3, 8]. Дана програма може

використовуватися як допоміжний інструмент для розвитку просторового мислення та логіки у користувачів різного віку [1]. Економічна цінність системи зумовлена низькою собівартістю її розробки, оскільки використання безкоштовних версій рушія UnrealEngine 5 та допоміжних сайтів з модельками дозволяє мінімізувати фінансові витрати на етапі виробництва [3, 8]. Готовий продукт може бути повністю оптимізований для розміщення на комерційних майданчиках, таких як Steam [5, 6, 7]. Завдяки стабільному інтересу масової аудиторії до казуальних та просторових логічних ігор, цей проект має хороші перспективи для успішного виходу на ринок інді-ігор та потенційного отримання комерційного прибутку при мінімальних початкових вкладеннях [1, 9].

ВИСНОВКИ

У ході виконання цього дипломного проекту було детально досліджено предметну область тривимірних інтерактивних ігор-головоломок та розроблено робочий програмний продукт на базі сучасного ігрового рушія UnrealEngine 5 [3, 8]. Аналіз історичного розвитку жанру показав, що за останні сорок років подібні комп'ютерні ігри пройшли величезний шлях - від плоскої двовимірної механіки «Тетрісу» та ранніх плиточних ігор до складних просторових симуляцій, що змінюють сприйняття реальності [1, 4].

Сучасні комерційні проєкти, представлені на платформі Steam, такі як TheWitness, TheTalosPrinciple та Superliminal, прямо демонструють, що тривимірний простір відкриває унікальні можливості для проектування рівнів [5, 6, 7]. Рішення логічного завдання тепер інтегрується в саму геометрію оточення, змушуючи гравця взаємодіяти з кутами огляду, світлом, тінями та фізичними властивостями предметів [4, 5, 6].

Під час проектування системи було чітко сформульовано функціональні та нефункціональні вимоги [9]. Було реалізовано модульну архітектуру проєкту, де кожен елемент працює як незалежний програмний блок [4, 6, 8]. Такий підхід дозволив створити гнучкий код, який легко тестувати та розширювати [3, 9]. Порівняльний аналіз сучасних технологій створення ігор підтвердив, що для реалізації якісного тривимірного простору з реалістичним освітленням та стабільною фізикою прості конструктори не підходять [8]. Водночас рушія UnrealEngine 5, завдяки передовій системі візуального програмування Blueprints, дозволив налаштувати важку логіку взаємодії об'єктів без написання складного текстового коду, що суттєво прискорило розробку [8].

Важливим практичним досягненням роботи стало використання інноваційних допоміжних інструментів, а саме платформи зі збіркою різних моделей Fab[3]. Це дозволило автоматизувати рутинний процес тривимірного моделювання меблів та декорацій на етапі активного виробництва, значно знизивши витрати часу без втрати

якості графіки [3]. Проведене тестування готової збірки під операційну систему Windows підтвердило стабільність програми: персонаж рухається плавно, предмети переносяться без збоїв у фізичних колізіях, тригери вчасно відкривають двері між кімнатами, а інтерфейс меню працює коректно [4, 7, 8, 9]. Гра демонструє стабільну частоту кадрів на комп'ютерах середньої потужності, що повністю задовольняє вимоги щодо оптимізації [3, 9].

Розроблена система має серйозні перспективи для подальшого розвитку. Створений програмний каркас можна використовувати як готовий шаблон для створення масштабніших 3D-головоломок [8]. У майбутньому проєкт можна розширити новими логічними кімнатами, додати складні механіки зміни перспективи за прикладом Superliminal, інтегрувати підтримку пристроїв віртуальної реальності або додати ІІІ-скрипти для динамічної зміни складності загадок [2, 4, 7, 9]. З економічної та практичної точки зору проєкт має високу цінність: використання безкоштовних версій UnrealEngine 5 та інструментів генерації контенту мінімізує витрати на розробку, а стабільний інтерес аудиторії до просторових логічних ігор відкриває можливості для успішного виходу продукту на ринок незалежних інді-ігор у Steam [1, 5, 7, 8]. Таким чином, поставлені на початку дипломної роботи цілі та завдання виконано в повному обсязі [9].

ПЕРЕЛІК ПОСИЛАНЬ

1. Lennelluc A. A history of puzzle games. *Medium*. 2018. URL: <https://medium.com/@AlenisLennelluc/a-history-of-puzzle-games-f92c750b1485> (дата звернення 26.04.2026)
2. The first 3D puzzle game and its history : Google Search. URL: <https://www.google.com/search?q=The+first+3D+puzzle+game+and+its+history> (дата звернення 26.04.2026)
3. The first 3D puzzle computer game and its history : Google Search. URL: <https://www.google.com/search?q=The+first+3D+puzzle+computer+game+and+its+history> (дата звернення 31.04.2026)
4. Програми для створення ігор. *DAAD Україна*. URL: <https://daad.org.ua/4741-programi-dlya-stvorenniya-igor.html> (дата звернення 10.05.2026)
5. Методичні вказівки до виконання дипломних робіт. *Науковий вісник Національного університету «Дніпровська політехніка»*. URL: <https://ir.nmu.org.ua/server/api/core/bitstreams/35234abf-4c0a-4e3b-ba81-ff7c1ed6dd9e/content> (дата звернення 12.05.2026)
6. How are video games made. *Meshy AI Blog*. 2024. URL: <https://www.meshy.ai/uk/blog/how-are-video-games-made> (дата звернення 12.05.2026)
7. The Witness (App ID 210970). *SteamDB*. URL: <https://steamdb.info/app/210970/> (дата звернення 12.05.2026)
8. The Talos Principle (App ID 257510). *SteamDB*. URL: <https://steamdb.info/app/257510/> (дата звернення 12.05.2026)
9. Superliminal (App ID 3478700). *SteamDB*. URL: <https://steamdb.info/app/3478700/> (дата звернення 12.05.2026)
10. An analysis of the mechanics and narrative of Superliminal. *Game Developer*. 2020. URL: <https://www.gamedeveloper.com/design/an-analysis-of-the-mechanics-and-narrative-of-superliminal> (дата звернення 18.05.2026)

ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-
КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ



НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ

КАФЕДРА ТЕХНОЛОГІЙ ЦИФРОВОГО РОЗВИТКУ



Інтерактивна 3D-головоломка на базі ігрового рушія Unreal
Engine 5

Виконала студентка 4 курсу
групи ТЦР-42

Огнева Софія Євгенівна

Керівник роботи

Ст. викладач кафедри ТЦР Олейніков Іван Анатолійович

Київ-2026

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

Мета роботи: підвищення якості ігрового враження та залучення гравців шляхом розробки, інтерактивної 3D головоломки з використанням сучасних технологій візуалізації та просторової взаємодії

Об'єкт дослідження: процес розробки 3D-головоломки та проектування інтерактивного ігрового оточення

Предметом дослідження: методи, алгоритми та інструменталні засоби створення ігрових продуктів, а саме 3D головоломок із механіками просторової взаємодії на базі ігрового рушія Unreal Engine 5 та візуального програмування Blueprints

ЗАДАЧІ КВАЛІФІКАЦІЙНОЇ РОБОТИ

1. Проаналізувати сучасний стану жанру 3D-головоломок, а також виконано порівняльну характеристику ігрових рушіїв, що дозволило обґрунтовано обрати Unreal Engine 5 та Blueprints як основний технологічний стек.
2. Обґрунтувати гнучку архітектуру проєкту, яка ефективно вирішує проблеми реалізації особливостей просторових механік та фізики тривимірного єредовища
3. Реалізовувати ключові ігрові системи: інтелектуальне керування персонажем, взаємодія в реальному часі та взаємодія з об'єктом
4. Створити набір взаємопов'язаних головоломок та ігрових тригерів, що взаємодіють між собою
5. Провести комплексне тестування на роботу логіки

3

АНАЛІЗ АНАЛОГІВ

Назва гри	Головна механіка	Плюси <u>проєкту</u>	Мінуси
<u>The Witness</u>	Малювання ліній на терміналах за підказками з оточення.	Великий відкритий світ, де рішення інтегровані в природу та архітектуру	Відкритий світ важко оптимізувати, він споживає забагато ресурсів ПК
<u>The Talos Principle</u>	Керування лазерами, дезактиваторами та копіюванням дій у часі.	Чітка послідовна логіка рівнів та глибокий сюжет про роботів.	Технічно занадто складно реалізувати систему запису «привидів» гравця за короткий час.
<u>Superliminal</u>	Зміна розміру предметів через оптичні ілюзії та	Дуже оригінальний <u>геймплей</u> та спокійна атмосфера без	Математичний прорахунок масштабів об'єктів викликає <u>баги</u>

4

ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Функціональні вимоги:

1. Реалізація точного 3D-керування (переміщення, стрибки, взаємодія з об'єктом, огляд на 360 градусів).
2. Створення системи 3D головоломок, заснованих на взаємодії елементів середовища (книги, плити, двері)
3. Зміна кімнати під час взаємодії з об'єктом у реальному часі
4. Можливість вивчати кімнату, та взаємодіяти з підказками для подальшого проходження

Нефункціональні вимоги:

1. Стабільна продуктивність не нижче 60 FPS на середньостатистичному ПК.
2. Розмір після стиснення не більше 1.5-2 гб
3. Зручне та інтуїтивне керування для гравця.
4. Висока стабільність роботи.
5. Мінімалістичний, не нав'язливий інтерфейс користувача

5

Актуальність роботи

Ринок інді-ігор активно розвивається, а 3D головоломки досі демонструють високу популярність завдяки оригінальності та низькій вартості розробки.

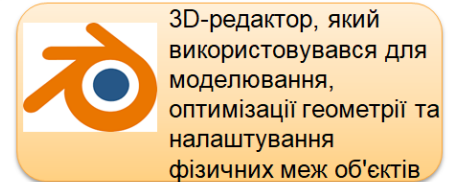
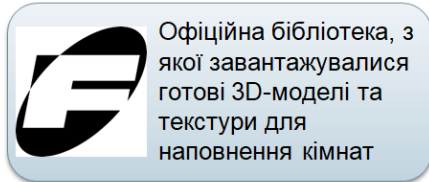
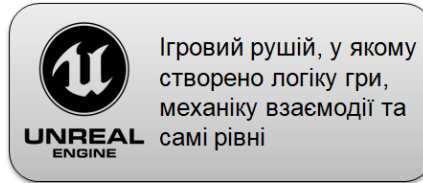
Дані ігри також займають високе місце в топах більшості гравців, що підтверджує її актуальність

Застосування новітніх технологій Unreal Engine 5 дозволяє підняти візуальну якість без втрати потужності

Завдання роботи:
Створити механіку з поєднанням різного типу головоломок, що дозволить любителям жанру зануритися у світ гри. Проста графіка, з незвичними загадками та взаємодією з різними об'єктами, що тренує просторове бачення гравця через взаємодію з довкіллям у кімнаті.

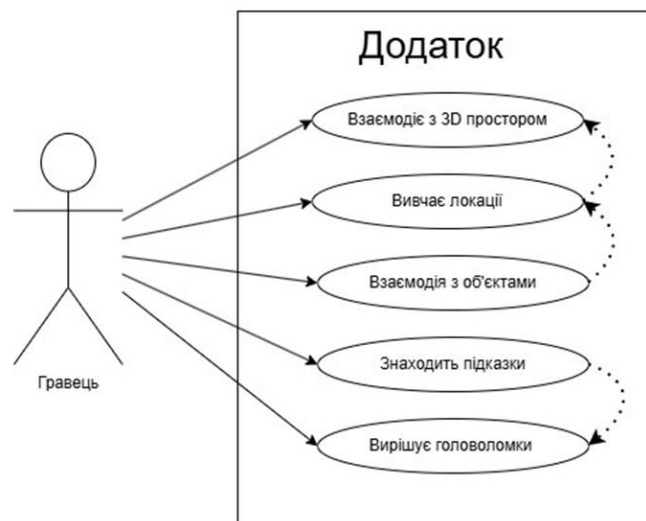
6

ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ



7

Алгоритм взаємодії гравця з 3D середовищем



8

ЕКРАННІ ФОРМИ

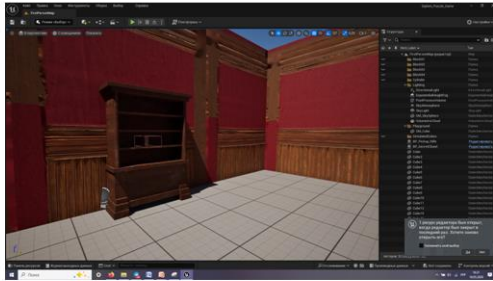


Рис. 9.1. Початкова кімната (до облаштування)

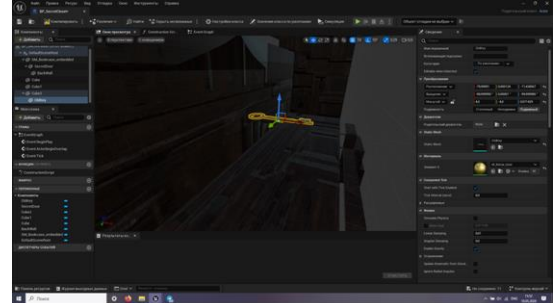


Рис. 9.2. Відображення інтерактивного об'єкту

9

ЕКРАННІ ФОРМИ

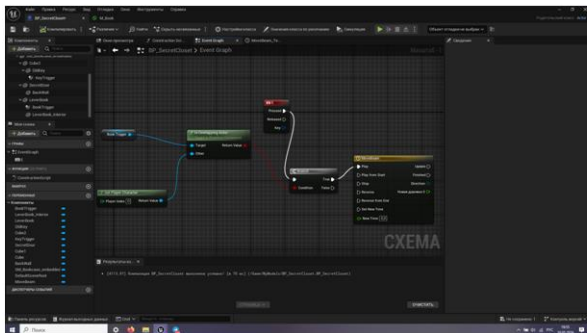


Рис. 10.1. Поєднання логіки об'єктів

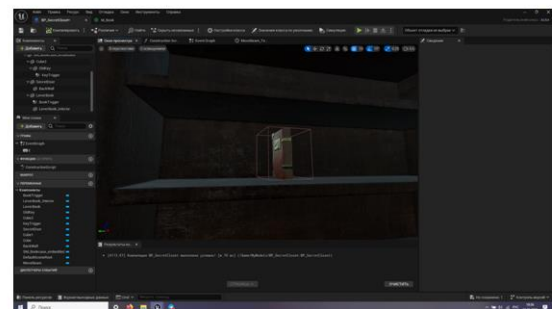


Рис. 10.2. Основний об'єкт (ричаг)

10

АПРОБАЦІЯ РЕЗУЛЬТАТІВ

Огнева С.Є. Інтерактивна 3D-головоломка на базі ігрового рушія Unreal Engine 5 // VII Всеукраїнській науково-технічній конференції «Застосування програмного забезпечення в інформаційно-комунікаційних технологіях». Збірник тез. К.: ДУІКТ, 2026 (подано до друку).

Огнева С.Є. Методи організації об'єктної взаємодії в інтерактивних середовищах на базі Unreal Engine 5// Всеукраїнській науково-технічній конференції «Технології цифрового розвитку: програмні системи та децентралізація». Збірник тез. К.: ДУІКТ, 2026. (подано до друку).

11

ВИСНОВКИ

1. Проаналізовано сучасний стану жанру 3D-головоломок, а також виконано порівняльну характеристику ігрових рушіїв, що дозволило обґрунтовано обрати Unreal Engine 5 та Blueprints як основний технологічний стек.
2. Обґрунтовано гнучку архітектуру проєкту, яка ефективно вирішує проблеми реалізації особливостей просторових механік та фізики тривимірного середовища
3. Реалізовано ключові ігрові системи: інтелектуальне керування персонажем, взаємодія в реальному часі та взаємодія з об'єктом
4. Створено набір взаємопов'язаних головоломок та ігрових тригерів
5. Проведено комплексне тестування

12

ДОДАТОК Б. ЛІСТИНГИ ПРОГРАМНИХ МОДУЛІВ

