

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерної інженерії та управління
(повна назва)

Кафедра _____ електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

Рівень вищої освіти _____ перший (бакалаврський)

Мобільний застосунок 3D-гри на основі рушія Unity

(тема)

Виконав:

здобувач _____ 4 _____ року навчання,

групи _____ КІУКІ-21-6

Костянтин ЗАКІПНИЙ

(власне ім'я, прізвище)

Спеціальність _____

123 «Комп'ютерна інженерія»

(код і повна назва спеціальності)

Тип програми _____ освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма _____

Комп'ютерна інженерія

(повна назва освітньої програми)

Керівник: _____ ас. Юлія АНДРУСЕНКО

(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ЕОМ

(підпис)

Андрій КОВАЛЕНКО

(власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерної інженерії та управління _____

Кафедра _____ електронних обчислювальних машин _____

Рівень вищої освіти _____ перший (бакалаврський) _____

Спеціальність _____ 123 «Комп'ютерна інженерія» _____
(код і повна назва)

Тип програми _____ освітньо-професійна _____
(освітньо-професійна або освітньо-наукова)

Освітня програма _____ Комп'ютерна інженерія _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

“ _____ ” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _____ Закіпному Костянтину Сергійовичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи _____ Мобільний застосунок 3D-гри на основі рушія Unity _____

затверджена наказом по університету від “ 26 ” травня 2025 р. № 424 ст

2. Термін подання здобувачем роботи до екзаменаційної комісії _____ 17 червня 2025 р.

3. Вхідні дані до роботи _____

1) вимоги до мобільної 3D-головоломки з фіксованою камерою;

2) особливості жанру логічних ігор та аналіз популярних аналогів;

3) інструменти для розробки ігрових застосунків на платформі Unity.

4. Перелік питань, що потрібно опрацювати у роботі _____

1) дослідження проблематики та аналіз вже наявних рішень;

2) вибір технології розробки та інструментальних засобів;

3) розробка логіки роботи застосунку та алгоритмів;

4) створення окремих функціональних компонентів програми;

5) тестування та усунення помилок у реалізованих модулях;

6) підбиття підсумків проведеної роботи.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій _____

Слайд-презентація – 11 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

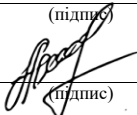
№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Аналіз поставленого завдання та визначення вимог до ігрового застосунку	27.05.25-28.05.25	
2	Розробка ігрового застосунку	29.05.25-07.06.25	
3	Тестування ігрового застосунку	08.06.25-09.06.25	
4	Оформлення матеріалів кваліфікаційної роботи	10.06.25-11.06.25	
5	Подання кваліфікаційної роботи керівникові та її попередній захист	12.06.25-13.06.25	
6	Подання кваліфікаційної роботи на рецензування	14.06.25-16.06.25	

Дата видачі завдання “ 26 ” травня 2025 р.

Здобувач


(підпис)

Керівник роботи


(підпис)

ас. Юлія АНДРУСЕНКО

(посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 53 с., 19 рис., 2 дод., 18 джерел.

UNITY, 3D-ГРА, МОБІЛЬНИЙ ЗАСТОСУНОК, КАМЕРА, ГОЛОВОЛОМКА, ІНВЕНТАР, ІНТЕРАКТИВНІ ОБ'ЄКТИ, UI, ANDROID, C#, КОРИСТУВАЦЬКИЙ ІНТЕРФЕЙС

Метою кваліфікаційної роботи є розробка мобільного застосунку у вигляді тривимірної логічної гри з фіксованою камерою від першої особи, орієнтованої на платформу Android.

У процесі реалізації проєкту було проведено аналіз предметної області та огляд існуючих аналогів у жанрі мобільних головоломок, визначено основні технічні та концептуальні вимоги до застосунку. Обґрунтовано вибір програмного забезпечення для розробки — від 3D-модельовання до реалізації в рушії Unity. Запропоновано алгоритм побудови ігрового процесу, що включає переміщення камери, взаємодію з об'єктами, систему інвентаря, анімації та звукові ефекти.

Розроблено і реалізовано програмні модулі, що відповідають за керування камерою, обробку введення, відтворення анімацій, звуку, а також логіку предметів. У фінальній частині створено користувацький інтерфейс, реалізовано меню гри, екран налаштувань та механізми взаємодії з предметами. Проєкт адаптовано під мобільну платформу з урахуванням обмежень продуктивності.

ABSTRACT

Bachelor's thesis: 53 pages, 19 figures, 2 appendices, 18 sources.

UNITY, 3D GAME, MOBILE APPLICATION, CAMERA, PUZZLE, INVENTORY, INTERACTIVE OBJECTS, UI, ANDROID, C#, USER INTERFACE

The major goal of this thesis is to develop a mobile application in the form of a three-dimensional puzzle game with a fixed first-person camera, targeting the Android platform.

During the development process, an analysis of the subject area was conducted along with a review of existing mobile puzzle games. The main technical and conceptual requirements for the application were defined. The choice of software tools – from 3D modeling to implementation in the Unity engine – was justified. A gameplay algorithm was proposed, including wall-switching mechanics, object interaction, an inventory system, animations, and sound effects.

Several programming modules were developed and implemented to handle camera control, user input, animation playback, audio management, and object logic. In the final stages, a user interface was designed, including the main menu, settings screen, and interaction mechanisms for items. The project was optimized for mobile devices, taking into account performance limitations.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	8
ВСТУП	9
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ	10
1.1 Аналіз предметної області.....	10
1.2 Огляд популярних ігрових застосунків у жанрі 3D-головоломок	13
1.2.1 Hundred Doors	13
1.2.2 Brain Test	15
1.3 Постановка задачі.....	16
2 АНАЛІЗ ТА ВИБІР ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ РОЗРОБКИ ЗАСТОСУНКУ	18
2.1 Blender	18
2.2 Marmoset Toolbag	20
2.3 Adobe Substance 3D Painter.....	20
2.4 Adobe Photoshop	22
2.5 Unity.....	22
2.6 Visual Studio Code	24
3 КОНЦЕПЦІЯ ІГРОВОГО ПРОЦЕСУ	25
4 ПРОГРАМНА РЕАЛІЗАЦІЯ.....	27
4.1 Обробка введення користувача	27
4.2 Відтворення звукових ефектів	28
4.3 Відтворення анімацій для інтерактивних об'єктів.....	29
4.4 Логіка підбирання предметів у грі	30
5 ІНСТРУКЦІЯ КОРИСТУВАЧА	32
5.1 Головне меню	32
5.2 Інтерфейс ігрового вікна	33
ВИСНОВКИ.....	38
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	39

ДОДАТОК А Графічний матеріал кваліфікаційної роботи.....	41
ДОДАТОК Б Код застосунку	46
Б.1 Скрипт ButtonController.cs	46
Б.2 Скрипт ButtonPuzzleController.cs	47
Б.3 Скрипт DoorController.cs.....	48
Б.4 Скрипт LampEmissionController.cs.....	48
Б.5 Скрипт TorchColorChanger.cs	50
Б.7 Скрипт TorchesController.cs	51
Б.8 Скрипт TVcontroller.cs.....	52
Б.9 Скрипт WardrobeController.cs	52

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

АО – оклюзія навколишнього середовища (англ., Ambient Occlusion)

API – програмний інтерфейс додатків (англ., Application Programming Interface)

C# – мова програмування C-sharp (англ., C-sharp)

FPS – кількість кадрів за секунду (англ., Frames Per Second)

GPU – графічний процесор (англ., Graphics Processing Unit)

HDRI – зображення з розширеним динамічним діапазоном (англ., High Dynamic Range Image)

IDE – інтегроване середовище розробки (англ., Integrated Development Environment)

LOD – рівень деталізації (англ., Level of Detail)

PBR – фізично коректний рендеринг (англ., Physically Based Rendering)

SDK – комплект засобів розробника (англ., Software Development Kit)

UI – користувацький інтерфейс (англ., User Interface)

UV – координати текстуровання поверхні (англ., UV Mapping)

UX – досвід користувача (англ., User Experience)

VFX – візуальні ефекти (англ., Visual Effects)

ВСТУП

Сьогодні створення мобільних застосунків на кшталт тривимірних ігор дедалі більше привертає увагу як з боку розробників, так і користувачів. Це не дивно, адже сучасні мобільні пристрої поступово зрівнюються за своїми технічними можливостями з повноцінними комп'ютерами. Водночас зростає й очікування аудиторії щодо якості візуального оформлення, інтерактивності та продуктивності. Саме тому розробка 3D-ігор для мобільних платформ стає надзвичайно актуальним завданням, яке потребує не лише творчого підходу, а й глибокого технічного розуміння.

Загальновизнаним лідером серед засобів для реалізації таких проєктів є рушій Unity, який широко застосовується в навчальних закладах, наукових установах, а також у професійному середовищі. Його потужні можливості, кросплатформеність і підтримка сучасних технологій дають змогу створювати повноцінні 3D-продукти навіть невеликими командами або окремими розробниками.

Метою цієї роботи є розробка оптимізованого, візуально привабливого і водночас технологічно грамотного рішення в сфері мобільних 3D-ігор. У процесі реалізації було поставлено за мету створити застосунок, який не тільки демонструє базові принципи взаємодії з користувачем у тривимірному просторі, а й враховує обмеження мобільних пристроїв.

Результатом цієї роботи є повноцінна реалізація мобільного ігрового застосунку у жанрі 3D-головоломки. На базі сформованої концепції, підбраного інструментарію та проаналізованих прикладів можливим стане подальше розширення функціоналу гри, додавання нових рівнів, механік і сценаріїв взаємодії. Окрему увагу в перспективі буде приділено підвищенню продуктивності на різних мобільних пристроях, а також удосконаленню візуального оформлення з урахуванням технічних обмежень.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз предметної області

Розробка ігрових застосунків, навіть у найвужчому технічному значенні, неможлива без глибокого розуміння тієї ігрової структури, в межах якої буде реалізовуватись ідея. Саме тому аналіз предметної області є вихідною точкою для будь-якого проєкту – незалежно від його масштабу чи платформи.

У сучасних реаліях мобільний геймінг посідає провідне місце серед усіх сегментів ігрової індустрії, як за кількістю користувачів, так і за обсягами прибутків. Згідно з даними звіту компанії Newzoo, у 2024 році мобільні ігри сформували близько 49% від загального доходу глобального ринку відеоігор, що майже вдвічі перевищує прибутки сегментів, пов'язаних із ПК та консолями. Такий високий показник пояснюється доступністю мобільних пристроїв, швидким зростанням ринків, що розвиваються, та широким розповсюдженням моделей монетизації, орієнтованих на мікротранзакції. На рисунку 1.1 представлено діаграму, що ілюструє розподіл доходів ігрового ринку за платформами згідно зі статистикою за 2024 рік [1].

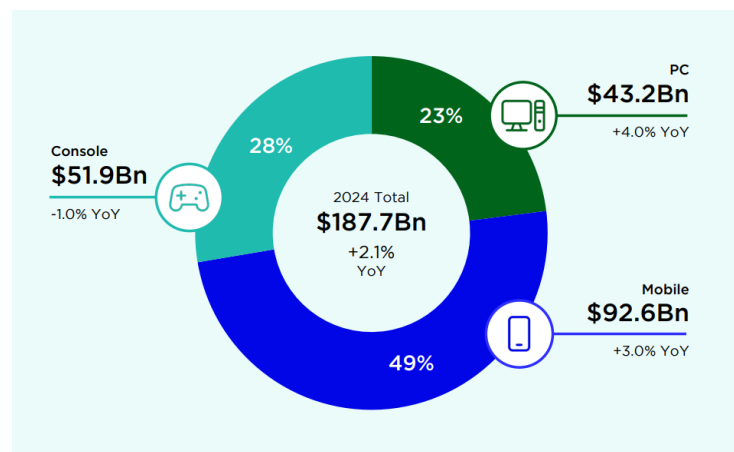


Рисунок 1.1 – Розподіл доходів ігрового ринку

У сучасному мобільному геймінгу спостерігається значне розширення вікової аудиторії, що свідчить про зростаючу популярність мобільних ігор серед різних поколінь. Згідно з даними дослідження, проведеного компанією eMarketer у 2020 році, у Північній Америці найбільше зростання кількості мобільних гравців спостерігалось у віковій групі 45 років і старше — приріст становив 17% порівняно з попереднім роком. Для порівняння, у віковій категорії від 25 до 44 років зростання склало 13%, у групі 2–12 років — 9%, а серед молоді віком 13–24 роки — лише 3% [2]. На рисунку 1.2 продемонстровано відповідну діаграму.

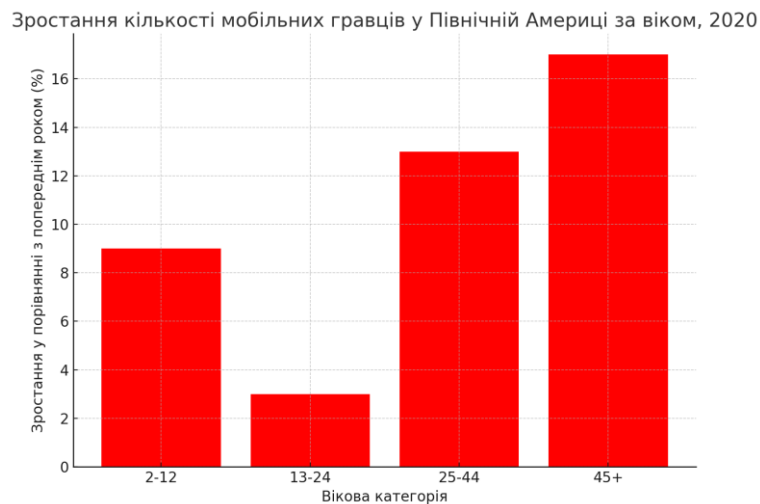


Рисунок 1.2 – Статистика зростання кількості мобільних гравців

Ці дані демонструють, що мобільні ігри стають дедалі привабливішими для старшої аудиторії, яка раніше не була основною цільовою групою для розробників. Зростання інтересу до мобільного геймінгу серед дорослих користувачів відкриває нові можливості для створення ігор, орієнтованих на ширший віковий спектр, з урахуванням різноманітних уподобань та потреб.

Загалом, ігри прийнято класифікувати за жанрами, і кожен із них має свої характерні риси, вимоги до механіки, темпу, аудиторії та навіть стилістики [3]. Серед найбільш поширених можна виокремити шутери, стратегії, симулятори, аркади, пригодницькі ігри, рольові системи (RPG), інді-ігри та, звичайно, головоломки. У деяких випадках відбувається

поєднання жанрів, і тоді межі між ними стають умовними, однак на концептуальному рівні подібна типологія дає змогу краще зрозуміти логіку побудови гри [4].

Так, наприклад, шутери зазвичай орієнтовані на швидкість реакції, точність, динаміку бою. У свою чергу, стратегії потребують стратегічного мислення, планування дій на кілька кроків уперед і здатності до адаптації в умовах змін. Симулятори акцентують увагу на відтворенні реальних або наближених до реальності процесів, з максимальною увагою до деталей. Аркади – це швидкий і часто нескладний геймплей із циклічними діями. Рольові ігри мають у фокусі розвиток персонажа, глибоку історію й складну внутрішню систему прокачування. Головоломки ж, у свою чергу, займають окреме місце, оскільки акцентують увагу гравця не на швидкості чи реакції, а на мисленні – логічному, просторовому та асоціативному [5].

Жанр головоломок вирізняється тим, що вимагає від гравця поступового занурення в систему правил, яка з кожним кроком ускладнюється або відкривається під новим кутом [6]. Саме ця властивість робить його ідеальним для застосування в мобільних іграх: короткі сесії, чіткі завдання, мінімум зовнішнього шуму. Особливо цінною є здатність подібних ігор не лише розважати, але й активізовувати когнітивні процеси – увагу, пам'ять, аналітичне мислення [7].

Саме до цього жанру – жанру логічних 3D-головоломок – належить гра, що стала предметом кваліфікаційної роботи. У межах проєкту основний акцент було зроблено не на динаміці, а на побудові логічної структури, де гравець має вирішити послідовність задач, щоб перейти на наступний рівень. При цьому важливим елементом стала не тільки складність окремих загадок, а й спосіб подачі – через візуальні підказки, анімації, звуки, зміну обстановки. Це дозволяє створити ефект «живого простору», в якому кожна деталь має значення. Завдяки цьому гра орієнтована на поступове занурення, де користувач вчиться розуміти логіку рівнів у процесі взаємодії з середовищем.

Таким чином, аналіз предметної області показує, що обраний жанр – головоломка – цілком відповідає як технічним можливостям мобільних платформ, так і сучасним запитам користувачів. Це той тип гри, який не потребує складного управління, але здатен викликати інтерес, напругу і бажання пройти ще хоча б один рівень. Урахування специфіки жанру лягло в основу постановки задач, формування механік, вибору інструментів і побудови геймплейної логіки.

1.2 Огляд популярних ігрових застосунків у жанрі 3D-головоломок

Жодна гра не створюється у повному відриві від уже існуючих рішень. Тому для обґрунтованого вибору механік та візуальних рішень у власному проєкті доцільно звернутися до досвіду популярних продуктів. Серед таких ігор вирізняються не лише багаторівневі проєкти з розгалуженим геймплеєм, як-от The Room або Monument Valley, а й більш легкі та доступні у реалізації формати, орієнтовані на широку аудиторію. До останніх належать Hundred Doors та Brain Test – два мобільні застосунки, які стали прикладом успішного поєднання простоти, логічного навантаження та привабливого візуального стилю. Саме ці ігри будуть розглянуті далі як приклади для порівняльного аналізу.

1.2.1 Hundred Doors

Одним із найбільш відомих представників жанру мобільних головоломок є гра «Hundred Doors», яка свого часу здобула значну популярність завдяки простоту, але водночас затягуючому геймплею. Основна ідея полягає в тому, що на кожному рівні гравець опиняється перед зачиненими дверима і має знайти спосіб їх відкрити, взаємодіючи з різними об'єктами на сцені. Попри привабливу концепцію, ця гра має низку помітних недоліків, які суттєво впливають на комфорт гри. Насамперед йдеться про

геймплей, який часто зводиться не до розв'язання логічної задачі, а до хаотичного натискання на всі можливі елементи інтерфейсу в пошуках хоч якоїсь реакції. На рисунку 1.3 продемонстрований один із рівнів даної гри.



Рисунок 1.3 – Інтерфейс гри «Hundred Doors»

Через велику кількість дрібних предметів на екрані й відсутність чітких підказок гравець нерідко змушений просто клацати по всьому, сподіваючись випадково натрапити на правильну дію. З часом такий підхід викликає втому і втрату інтересу, адже відчуття логічної побудови задачі поступається випадковості.

Іншим суттєвим недоліком є залежність гри від інтернет-з'єднання. Навіть у випадках, коли немає об'єктивної потреби у використанні інтернету, користувач стикається з обмеженнями у функціональності без доступу до мережі. Крім того, після завершення кожного рівня демонструється реклама, що значно уповільнює ігровий темп і може викликати роздратування.

Разом із тим, не можна не відзначити сильні сторони "Hundred Doors". Насамперед, це зручний і зрозумілий інтерфейс: всі елементи керування логічно розташовані, інвентар відкривається інтуїтивно, а перехід між рівнями не викликає труднощів навіть у новачка. Також заслуговує уваги загальний візуальний стиль гри – дизайн екранів приємний, кольори

збалансовані, а візуальні елементи добре пропрацьовані. Це створює певну атмосферу й робить гру візуально привабливою, попри спрощену структуру рівнів.

У підсумку, "Hundred Doors" можна вважати класичним прикладом мобільної головоломки, яка має потенціал, але вимагає ретельнішого підходу до побудови ігрових задач і зменшення зовнішніх обмежень, таких як реклама чи інтернет-залежність.

1.2.2 Brain Test

Ще одним цікавим представником жанру мобільних головоломок є гра «Brain Test», яка вирізняється нестандартним підходом до побудови задач. Її основна привабливість полягає у простих, але часто неочікуваних загадках, де ключ до розв'язання криється в нестандартному мисленні або навіть грі слів. Це робить процес проходження легким і водночас цікавим, адже кожен рівень супроводжується певним елементом гумору або абсурду, що викликає усмішку.

Особливу атмосферу гри формує її приємна, м'яка графіка – стилізована під ручну рисовку, без перевантаження деталями. Вона виглядає доступною для гравців різного віку й створює враження легкої, неформальної гри, до якої можна повертатися у будь-який момент.

Варто також відзначити звукове оформлення: фонову музику та прості звукові ефекти не відволікають, але додають настрою, роблячи гру цілісною у сприйнятті.

Загалом, «Brain Test» є прикладом того, як простота реалізації у поєднанні з якісною візуальною подачею та творчим підходом до дизайну рівнів може забезпечити стабільний інтерес аудиторії. Незважаючи на 2D-формат, ця гра демонструє, що головоломки не обов'язково мають бути складними – головне, щоб вони були живими, несподіваними й приємними у проходженні. На рисунку 1.4 зображений один із рівнів даної гри.

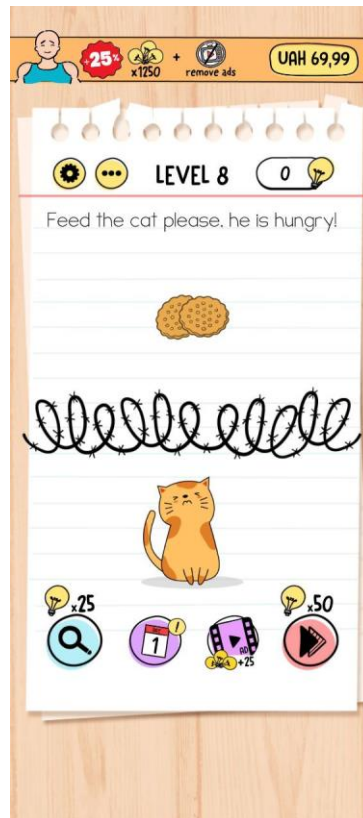


Рисунок 1.4 – Інтерфейс гри «Brain Test»

1.3 Постановка задачі

Після аналізу аналогічних мобільних застосунків були виявлені як вдалі рішення, так і типові недоліки, пов'язані з надмірною випадковістю, перевантаженням інтерфейсу, інтернет-залежністю та агресивною рекламою. Враховуючи це, основною метою проєкту стало створення логічної 3D-гри з камерою від першої особи, яка зберігає фокус на послідовному розв'язанні задач у межах обмеженого простору, без відволікаючих факторів.

Для досягнення цієї мети було сформульовано такі підзадачі:

- сформувати геймплейну концепцію, що забезпечує послідовне логічне проходження без випадкових дій;
- розробити візуальне та звукове оформлення, яке сприяє глибокому зануренню в атмосферу гри;
- забезпечити високу продуктивність гри на мобільних пристроях шляхом оптимізації графіки, освітлення та ресурсів.

Для реалізації першої підзадачі було визначено принципи, на яких базуватиметься ігрова логіка. Головним завданням є побудова структури рівнів так, щоб гравець мав змогу крок за кроком вирішувати задачі на основі логіки, аналізу та послідовного спостереження, а не методом випадкового натискання. Передбачено, що кожна взаємодія з об'єктом повинна мати чітке обґрунтування в контексті сцени. Завдяки цьому забезпечується не лише зрозумілість проходження, а й відчуття прогресу, що мотивує рухатися далі.

Для реалізації другої підзадачі планується створити візуальну стилістику, яка не перевантажує гравця, але при цьому підтримує атмосферу гри. Кольорова палітра, інтерфейс, а також анімації об'єктів будуть підібрані таким чином, щоб посилювати ефект присутності, а не відволікати. Звукове оформлення виконуватиме роль підсилювача ігрового досвіду – звуки натискань, перемикань, відкриття об'єктів тощо допоможуть зробити середовище «живим», не перевантажуючи при цьому аудіальний канал. Всі елементи подачі мають працювати на одне – створення спокійної, глибоко занурюючої атмосфери.

Для реалізації третьої підзадачі передбачено низку технічних рішень, спрямованих на зменшення навантаження на систему. Зокрема, графіка буде адаптована під мобільні умови: використовуватимуться текстури з невисокою роздільністю, об'єкти матимуть обмежену кількість полігонів, а матеріали – спрощені мобільні шейдери без зайвих обчислень. Освітлення реалізується через попередньо запечене освітлення, що дозволяє повністю уникнути реального рендерингу тіней під час гри. Це дає змогу підтримувати стабільний FPS, що є критичним показником для комфортного ігрового досвіду на мобільних пристроях. Завдяки такому підходу гра працюватиме стабільно навіть на пристроях з базовими характеристиками, не втрачаючи при цьому цілісного візуального враження.

2 АНАЛІЗ ТА ВИБІР ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ РОЗРОБКИ ЗАСТОСУНКУ

Перш ніж переходити безпосередньо до реалізації ідеї, надзвичайно важливо зосередити увагу на виборі інструментарію, який ляже в основу всієї подальшої роботи. Як показує практика, саме на цьому етапі часто визначається не лише технічна складність проєкту, а й його подальші перспективи розвитку та підтримки.

Правильно підібране програмне забезпечення здатне суттєво спростити реалізацію багатьох завдань, тоді як поспішне або невдале рішення може згодом призвести до численних обмежень. У випадку створення мобільного застосунку з тривимірною графікою особливо важливо враховувати такі аспекти, як підтримка платформ, продуктивність рушія, зручність роботи з 3D-об'єктами, а також можливості оптимізації.

2.1 Blender

Blender – це безкоштовне програмне забезпечення з відкритим кодом для створення тривимірної комп'ютерної графіки. Воно підтримує моделювання, анімацію, рендеринг, скульптинг, відеомонтаж, симуляцію фізики, текстуровання, а також розробку інтерактивних 3D-додатків і ігор. Завдяки своїй багатофункціональності та активній спільноті, Blender широко використовується як професіоналами, так і початківцями в галузях анімації, візуальних ефектів, ігрового дизайну та 3D-друку. Його кросплатформенність і постійне оновлення функціоналу роблять його привабливим вибором для широкого кола користувачів [8].

Цей безкоштовний, але водночас професійний пакет тривимірного моделювання вже давно зарекомендував себе як надійний інструмент для створення ігрових моделей. Однією з його сильних сторін є можливість

точного контролю над топологією сітки, що критично важливо для забезпечення оптимальної продуктивності в мобільному середовищі. До того ж, у Blender реалізовані зручні механізми UV-розгортки, які суттєво полегшують подальшу роботу з текстуруванням [9].

Ще однією перевагою є інтеграція скелетної анімації, яка дозволяє одразу підготувати об'єкти до експорту в ігровий рушій. Завдяки цьому зменшується кількість проміжних етапів і виключається потреба у додатковому програмному забезпеченні для підготовки рігінгу. Крім того, варто зауважити, що Blender підтримує експорт у форматах, які легко сприймаються Unity, тож уся сцена або окремі моделі можуть бути безпосередньо імпортовані без втрати даних [10].

Таким чином, використання Blender у рамках розробки даного застосунку є не лише логічним, а й цілком обґрунтованим рішенням, яке дозволить досягти балансу між якістю моделей і вимогами до їхньої оптимізації. На рисунку 2.1 представлений приклад моделювання ігрового об'єкта.

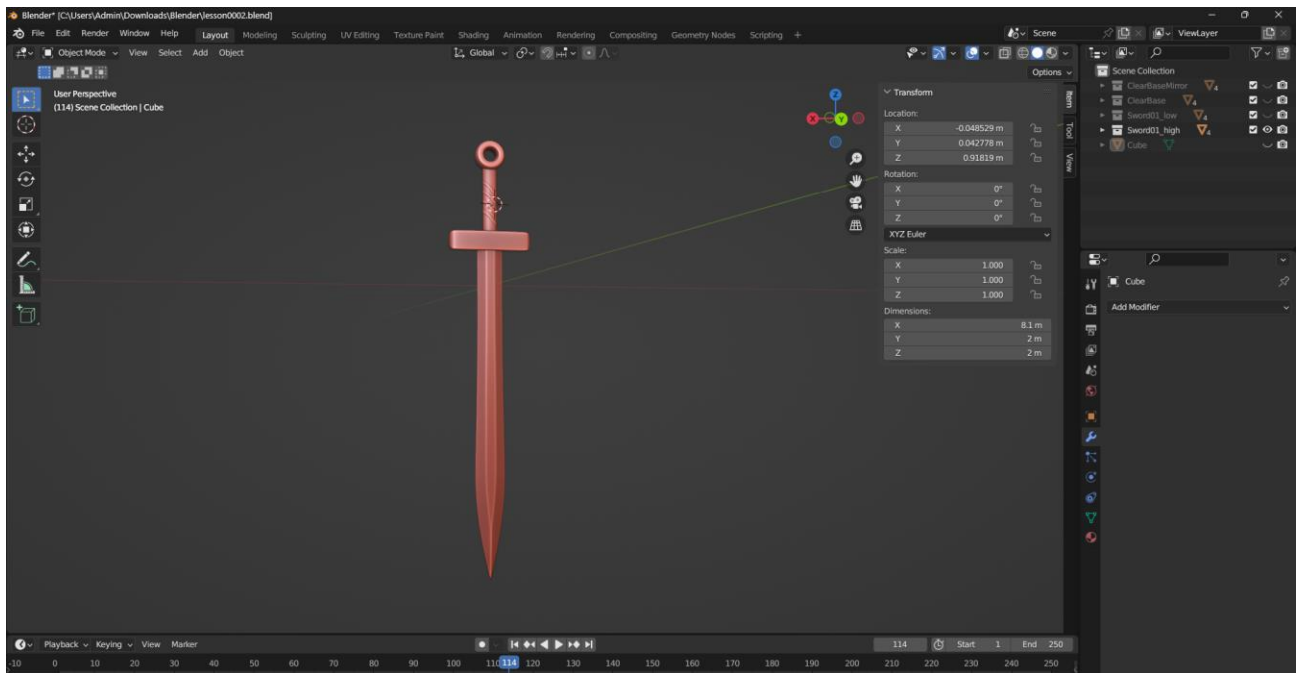


Рисунок 2.1 – Моделювання ігрового об'єкта

2.2 Marmoset Toolbag

Marmoset Toolbag – це спеціалізоване програмне забезпечення для візуалізації тривимірних моделей у режимі реального часу. Програма широко використовується у сфері 3D-графіки для попереднього рендерингу, презентації моделей, перевірки матеріалів і освітлення, а також як інструмент фінального перегляду перед інтеграцією в ігровий рушій. Завдяки підтримці сучасних технологій відображення та гнучкому інтерфейсу, Marmoset Toolbag став популярним серед 3D-художників, які працюють у геймдеві, анімації та візуалізації продуктів.

Серед переваг цієї програми варто відзначити, передусім, можливість реалістичного відображення моделей у різних умовах освітлення, що дозволяє заздалегідь оцінити, як виглядатиме об'єкт у грі. Це особливо важливо при роботі з нормальми, швами UV-розгортки і текстурами — будь-які дефекти можна виявити ще до переходу в основний рушій. Marmoset також дозволяє легко перемикаєти HDRI-карти освітлення, застосовувати PBR-матеріали, налаштовувати камеру та глибину різкості, що дає змогу створити максимально наближене до кінцевого вигляду зображення.

Окрему увагу заслуговує інструментарій для попереднього перегляду запечених текстур. Саме перед початком роботи в Adobe Substance 3D Painter часто доцільно протестувати модель у Marmoset Toolbag, аби переконатися в правильності геометрії та якості мап, зокрема нормалей і Ambient Occlusion. У такий спосіб забезпечується послідовний, контрольований робочий процес із мінімізацією потенційних помилок у фінальній версії активу.

2.3 Adobe Substance 3D Painter

Adobe Substance 3D Painter – це професійне програмне забезпечення для текстурювання 3D-моделей, яке широко використовується в індустріях відеоігор, анімації, візуальних ефектів та дизайну. Програма надає

художникам можливість наносити текстури безпосередньо на 3D-об'єкти в режимі реального часу, використовуючи фізично коректне рендеринг-середовище (PBR), що забезпечує високий рівень фотореалізму [11].

Головна перевага Adobe Substance 3D Painter полягає в тому, що він дозволяє працювати з PBR-матеріалами, які надалі коректно інтерпретуються ігровим рушієм Unity. Це забезпечує точне відображення поверхонь у різних умовах освітлення, що, у свою чергу, підвищує реалістичність ігрового середовища навіть за обмежених ресурсів мобільних пристроїв [12].

Крім того, Adobe Substance 3D Painter дає змогу працювати з багатьма матеріалами одночасно, використовувати маски, фільтри та смарт-матеріали, які реагують на геометрію об'єкта. Такий підхід не лише пришвидшує сам процес, але й гарантує вищу точність результату. Важливо й те, що програма дозволяє одразу бачити результат в реальному часі на тривимірній моделі, що виключає потребу в постійній перевірці в інших середовищах.

У межах проєкту саме за допомогою Adobe Substance 3D Painter буде створено всі основні текстури – від базового кольору до нормалей, roughness і metallic-карт. Це дасть змогу досягти чіткої візуальної цілісності між різними об'єктами, зберігаючи при цьому оптимальний баланс між якістю й продуктивністю. На рисунку 2.2 зображено приклад текстурування ігрового об'єкта.

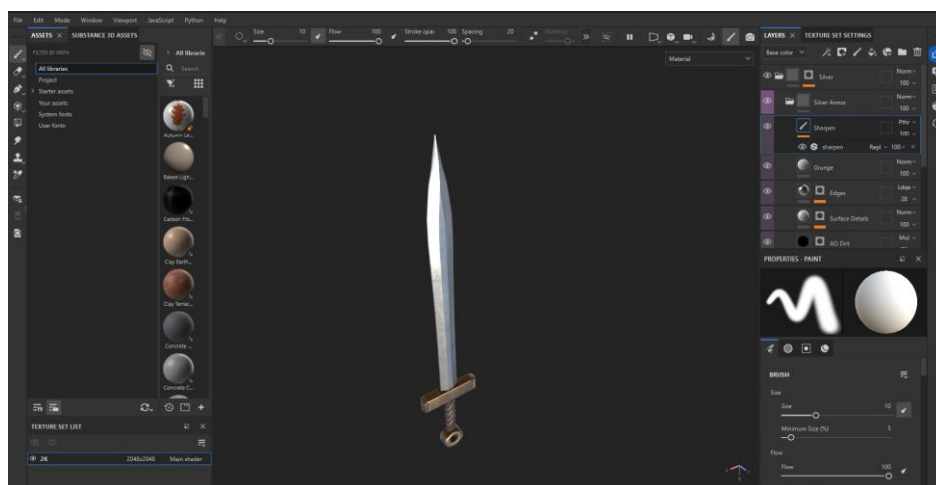


Рисунок 2.2 – Моделювання ігрового об'єкта

2.4 Adobe Photoshop

Adobe Photoshop – це потужний графічний редактор, який уже багато років вважається галузевим стандартом у сфері цифрового зображення. Однією з головних переваг Photoshop є його гнучкість: він дозволяє працювати як із растровою графікою, так і з текстом, шарами, фільтрами, масками та кольоровими профілями. Також, Photoshop активно використовується також як допоміжний інструмент у створенні текстур для 3D-моделей [13].

Завдяки широкому набору фільтрів і можливості тонкої роботи з шарами, Adobe Photoshop дозволяє вносити зміни в текстури з точністю до пікселя. Наприклад, у випадках, коли автоматичне розміщення деталей у Substance 3D Painter не дає бажаного результату, зображення можна експортувати до Photoshop для ручного редагування.

Також Photoshop може бути використаний для створення спеціалізованих масок, які згодом імпортувались назад у Substance Painter і застосовувались до конкретних матеріалів або деталей моделі.

Окремо варто згадати про гнучкість кольорокорекції, яка реалізована в Adobe Photoshop. Саме тут можна було швидко підлаштувати відтінки, контрастність або яскравість певних елементів текстури без необхідності повертатися до попереднього етапу.

2.5 Unity

Обрання саме Unity було зумовлене низкою практичних міркувань. По-перше, він має повноцінну підтримку Android – основної платформи, для якої створювався застосунок. По-друге, інтерфейс рушія і його архітектура дозволяють швидко інтегрувати 3D-моделі, налаштовувати матеріали, камери, освітлення і навігацію без втрати контролю над внутрішніми процесами [14]. На рисунку 2.3 зображений інтерфейс даного ігрового рушія.

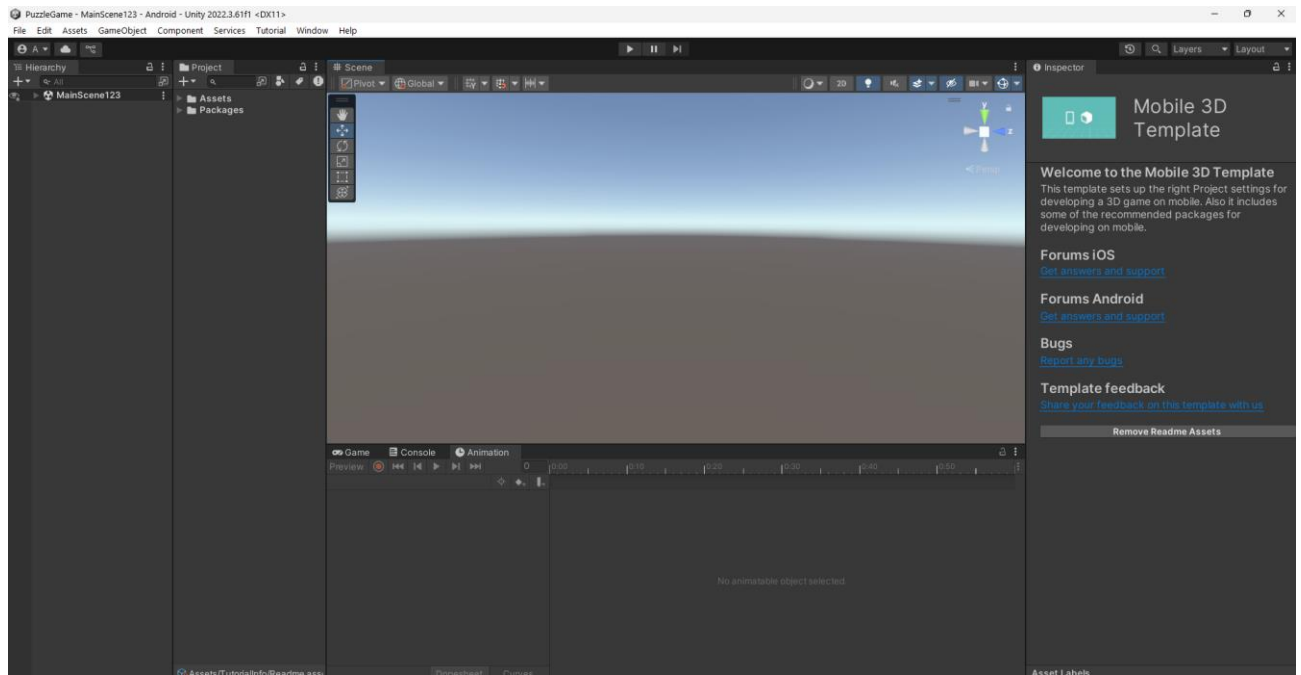


Рисунок 2.3 – Інтерфейс ігрового рушія Unity

Особливо важливо те, що Unity чудово працює з PBR-матеріалами, які були попередньо створені в Adobe Substance 3D Painter. Під час імпорту текстур, таких як metallic, normal map чи ambient occlusion, рушій автоматично адаптує їх до свого стандартного шейдера, що значно спрощує процес [15].

У межах реалізації гри рушій забезпечить можливість створення інтерактивного середовища, включаючи обробку натискань на екрані, анімацію, перемикання сцен та базову ігрову логіку. Значна увага приділялась також оптимізації продуктивності – через обмеженість мобільних ресурсів довелося працювати зі зменшеними полігональними сітками, кастомізованими LOD-рівнями, зменшеними роздільними здатностями текстур та стисненням аудіофайлів [16].

Окрім того, Unity має вбудовані засоби профілювання, які дозволять протестувати гру на різних пристроях і виявити проблемні місця [17]. Саме це дасть змогу підтримати стабільну частоту кадрів без істотних втрат якості візуалізації.

2.6 Visual Studio Code

Visual Studio Code – це легкий, але потужний редактор вихідного коду, розроблений компанією Microsoft. Він підтримує велику кількість мов програмування, включаючи C#, JavaScript, Python, а також мови розмітки на кшталт HTML і CSS. Основною перевагою редактора є його розширюваність: за допомогою тисяч безкоштовних розширень користувач може додати підтримку нових мов, інструментів дебагінгу, інтеграцій із системами контролю версій, середовищами розробки та платформами для розгортання [18].

Ще однією перевагою стала підтримка одночасної роботи з кількома файлами, що дало змогу швидко перемикатися між компонентами гри — наприклад, від скриптів, що відповідають за камеру, до обробників взаємодії чи UI-елементів.

Простота, швидкість і адаптивність цього середовища стали суттєвими перевагами, особливо в умовах розробки мобільного проєкту, де цінується кожна хвилина часу. На рисунку 2.4 зображений інтерфейс даного середовища розробки.

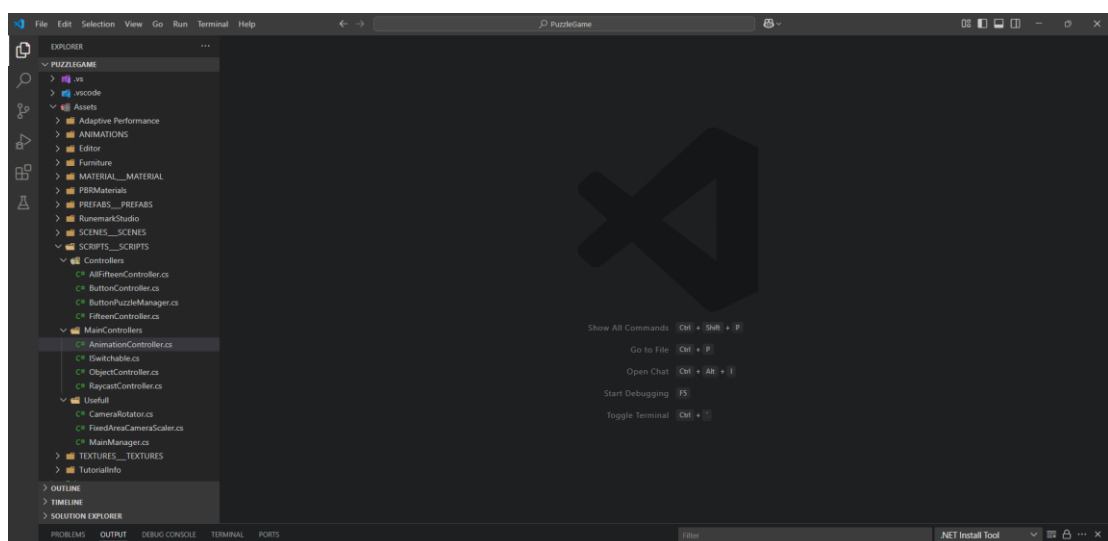


Рисунок 2.4 – Інтерфейс Visual Studio Code

3 КОНЦЕПЦІЯ ІГРОВОГО ПРОЦЕСУ

Гра належить до жанру логічних 3D-головоломок з камерою від першої особи, яка, втім, має обмеження на пересування гравця. Такий підхід дозволяє зосередити увагу саме на розв'язанні задач, а не на навігації простором. Гравець перебуває у замкненому приміщенні, яке умовно поділене на чотири функціональні стіни. Камера жорстко прив'язана до центра кімнати, і користувач має можливість лише обертати її вліво або вправо, переключаючись між стінами за допомогою зручних стрілок, розміщених на екрані.

Кожна зі стін містить унікальні інтерактивні об'єкти, які взаємопов'язані між собою через систему головоломок. Вони вимагають від гравця логічного мислення, уважності до деталей та здатності до поетапного аналізу. У просторі розміщені такі предмети, як шафи, важелі, кнопки, панелі з кодами, механізми зі звуковим або візуальним зворотним зв'язком. Всі вони не лише мають анімацію, яка супроводжує дію, але й звукові ефекти.

Для кращого розуміння послідовності ігрових дій у грі, на рисунку 3.1 нижче представлено спрощену схему алгоритму проходження рівня.



Рисунок 3.1 – Алгоритм проходження рівня

Особливу роль у проходженні відіграють предмети, які можна підібрати. Вони автоматично додаються до інвентарю, де залишаються до моменту використання. Це можуть бути ключі, частини кодів, інструменти чи інші артефакти, потрібні для розв'язання наступних задач. Сам інвентар реалізований максимально просто і зрозуміло – з іконками та можливістю вибору об'єкта для подальшої взаємодії. Завдяки цьому користувачеві не потрібно відволікатися від головного – логіки самої гри.

Після успішного розв'язання всіх загадок на рівні, гравець отримує доступ до дверей, які, як правило, закриті на початку. Їх відкриття символізує завершення поточної секції гри. Далі передбачено перехід до наступного рівня – нового приміщення з новим набором викликів. Кожен наступний рівень поступово ускладнює логіку, вводить нові типи завдань і вимагає глибшого розуміння ігрових механік, які вже були засвоєні раніше.

Важливо, що геймплей тримається на постійній взаємодії з оточенням. Немає елементів випадковості або таймерів, що дозволяє гравцеві зосередитися виключно на розв'язанні задач у комфортному темпі. Така побудова гри орієнтована не стільки на динаміку, скільки на глибоке занурення в процес розгадування, що робить проєкт схожим на віртуальну квест-кімнату – лише в цифровому просторі.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ

4.1 Обробка введення користувача

У проєкті обробкою взаємодії користувача з екраном займається скрипт `RaycastController`, реалізований як сінглтон (лістинг 4.1). Такий підхід дозволяє звертатися до нього з будь-якої частини проєкту без необхідності створювати додаткові екземпляри. Весь контроль за дотиком до екрана реалізовано через мобільні методи введення, що охоплюють натискання, утримання та відпускання.

Лістинг 4.1 – Зчитування введення на Android

```
public class RaycastController : MonoBehaviour
{
    ...
    if (Input.touchCount > 0 && Input.touches[0].phase ==
    TouchPhase.Began)
    {
        inputPressed = true;
        inputPosition = Input.touches[0].position;
    }

    inputHeld = Input.touchCount > 0 &&
                (Input.GetTouch(0).phase == TouchPhase.Began ||
                 Input.GetTouch(0).phase == TouchPhase.Moved ||
                 Input.GetTouch(0).phase ==
    TouchPhase.Stationary);

    if (inputHeld)
        inputPosition = Input.GetTouch(0).position;

    if (Input.touchCount > 0 && Input.touches[0].phase ==
    TouchPhase.Ended)
    {
        inputReleased = true;
        inputPosition = Input.touches[0].position;
    }
    ...
}
```

Цей фрагмент визначає момент початку, утримання та завершення

торкання, зберігаючи координати взаємодії в змінній `inputPosition`. Таким чином, інші скрипти можуть у будь-який момент дізнатись, чи натиснуто зараз на екран, чи торкання вже завершилось, і де саме воно відбулося. Це дає змогу точно реагувати на взаємодію користувача з об'єктами гри або елементами інтерфейсу.

4.2 Відтворення звукових ефектів

Щоб забезпечити озвучення інтерактивних подій у грі – таких як відкриття дверей, натискання кнопок або перемикання важелів – реалізовано окремий компонент `AudioManager`. Він дозволяє централізовано керувати всіма аудіокліпами та забезпечує їх відтворення за текстовим ідентифікатором. `AudioManager` створюється у вигляді сінглтона, що виключає дублювання об'єкта між сценами. Нижче наведено метод `Play`, що відповідає за програвання звуку (лістинг 4.2).

Лістинг 4.2 – Зчитування введення на Android

```
public void Play(string name)
{
    if (audioMap.TryGetValue(name, out AudioClip clip))
    {
        audioSource.PlayOneShot(clip);
    }
    else
    {
        Debug.LogWarning($"Sound with the name \"{name}\" was
not found in AudioManager");
    }
}
```

Цей метод приймає назву звукового ефекту як рядок і, якщо відповідний аудіокліп знайдено у словнику `audioMap`, програє його один раз через `AudioSource`. У разі відсутності – виводиться попередження в консоль. У такій формі цей метод можна викликати з будь-якого іншого скрипта. Наприклад, коли двері відкриваються, достатньо викликати

`AudioManager.Instance.Play("door_open")`, і відповідний звук буде відтворено без потреби прямого посилання на аудіофайл.

4.3 Відтворення анімацій для інтерактивних об'єктів

У грі передбачено наявність об'єктів, які можуть змінювати свій стан у відповідь на взаємодію користувача – наприклад, двері, шафи чи важелі. Для реалізації цього використовується компонент `AnimationController`, що забезпечує програвання заздалегідь підготовлених анімацій. Сценарій працює як з автоматичною взаємодією, так і через використання інвентарного предмета (інтерфейс `ItemUsable`).

У методі `Start` збираються всі доступні анімації, прив'язані до об'єкта, і зберігаються в окремому списку. Цей підхід дозволяє зручно організовувати послідовність станів, уникаючи жорстко заданих назв (лістинг 4.3).

Лістинг 4.3 – Метод `Start()`

```
void Start()
{
    animat = GetComponent<Animation>();
    foreach (AnimationState animState in animat)
    {
        animations.Add(animState.name);
    }
}
```

Після ініціалізації в методі `Update` (лістинг 4.4) перевіряється, чи об'єкт не статичний (`isStatic == false`), а також чи не виконується вже анімація. Якщо користувач натиснув на об'єкт, і жодна анімація не відтворюється в даний момент, запускається поточна анімація за індексом `State`. Це забезпечує одноразову взаємодію при натисканні.

Лістинг 4.4 – Метод `Update()`

```
void Update() {
    if (!isStatic) {
```

```

        isPlaying = animat.isPlaying;
        if (RaycastController.Instance.isPressed &&
RaycastController.Instance.hit.collider.gameObject ==
gameObject){
            if (!isPlaying)
                animat.Play(animations[State]);
        }
    }
}

```

4.4 Логіка підбирання предметів у грі

Для реалізації механіки підбору інтерактивних предметів, які можуть використовуватись гравцем у процесі проходження рівня, у грі застосовується скрипт `ItemPickupController` (лістинг 4.5). Його призначення – відслідковувати взаємодію гравця з об'єктом у сцені, ініціювати додавання предмета до інвентаря, а також зареєструвати зв'язок між предметом і цільовим об'єктом, на який він може впливати.

Лістинг 4.5 – Клас `ItemPickupController`

```

public class ItemPickupController : MonoBehaviour
{
    public string itemId;
    public Sprite itemIcon;
    public GameObject targetObject;
    public bool isHaveTarget = true;
    public bool isReusable = false;

    private void Start()
    {
        ItemTargetRegistry.Instance.RegisterTarget(itemId,
targetObject);
    }

    private void OnMouseDown()
    {
        ItemUIController[] items =
FindObjectOfType<ItemUIController>();
        foreach (var slot in items)
        {
            if (!slot.HasItem)
            {
                slot.SetItem(itemIcon, itemId, isReusable,
isHaveTarget);
                gameObject.SetActive(false);
                AudioManager.Instance.Play("ItemPickUpSFX");
                return;
            }
        }
    }
}

```

```

    }
}
}

```

Ключові параметри скрипта задаються через інспектор: `itemId` – унікальний ідентифікатор предмета, `itemIcon` – спрайт, що відображається в інвентарі, `targetObject` – об’єкт, до якого прив’язаний предмет, `isReusable` – чи можна використати предмет повторно, та `isHaveTarget` – чи прив’язано предмет до якогось цільового елементу.

Подальша взаємодія реалізується у методі `OnMouseDown()`, який викликається при натисканні на об’єкт. Система перевіряє всі слоти інвентаря в сцені та знаходить перший вільний. Як тільки знайдено доступне місце, предмет додається до інвентаря з усіма супутніми параметрами, сам об’єкт деактивується, а також запускається звуковий ефект.

5 ІНСТРУКЦІЯ КОРИСТУВАЧА

5.1 Головне меню

Головне меню має мінімалістичну структуру: зліва розміщено логотип гри, а справа – вертикально чотири основні кнопки. Перша кнопка адаптується до прогресу гравця: якщо проходження ще не розпочато, на ній відображається напис «Почати гру»; якщо ж існує збереження – змінюється на «Продовжити». Друга кнопка веде до вікна вибору рівнів, а третя відкриває вікно статистики. На рисунку 5.1 наведено зовнішній вигляд головного меню гри.



Рисунок 5.1 – Головне меню

У вікні вибору рівня гравцеві представлено горизонтальний список доступних етапів. Кожен рівень супроводжується мініатюрою зображення, яка відображає загальний вигляд кімнати або ключовий елемент головоломки, що дозволяє користувачеві зорієнтуватися перед початком проходження. На екрані одночасно відображаються лише три рівні, а для перегляду інших передбачено горизонтальний скролінг, завдяки якому гравець може зручно перегортати список і обрати потрібний етап. На рисунку 5.2 зображено інтерфейс вікна вибору рівня.

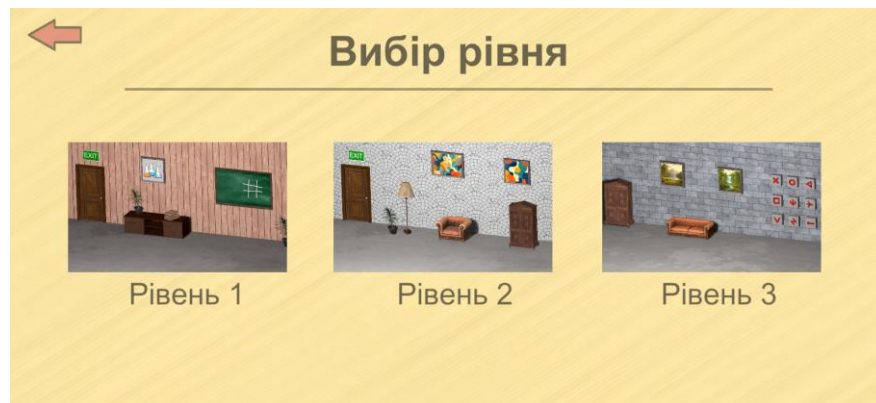


Рисунок 5.2 – Інтерфейс вікна вибору рівня

Вікно статистики (рисунок 5.3) надає гравцеві змогу переглянути свій прогрес у грі. У цьому розділі відображається вертикальний список рівнів, кожен з яких супроводжується вказівкою часу, витраченого на його проходження.

Рівень	-	Час
Рівень 1	-	00:01:23
Рівень 2	-	00:03:41
Рівень 3	-	00:04:13
Рівень 4	-	00:03:09

Рисунок 5.3 – Інтерфейс вікна статистики

5.2 Інтерфейс ігрового вікна

Під час проходження рівнів гравець взаємодіє з інтерфейсом основного ігрового вікна (рисунок 5.4), де з правого боку розміщено вертикальну сіру панель інвентарю. Вона призначена для зберігання предметів, які гравець може підібрати у процесі гри. Усі предмети умовно поділяються на два типи: інформаційні (наприклад, записки з підказками) та використовувані (ключі, важелі, інструменти тощо), з якими можна взаємодіяти.

Використовувані предмети призначені для подальшої взаємодії з об'єктами в ігровому середовищі. Їх можна або обрати кліком, після чого предмет підсвічується, і тоді він буде активним до моменту використання, або перетягнути на відповідний об'єкт у сцені – у такому разі взаємодія відбудеться одразу.



Рисунок 5.4 – Інтерфейс ігрового вікна

Натиснувши на інформаційний предмет, гравець відкриває його у збільшеному вигляді в центрі екрану для детального ознайомлення з вмістом. Така взаємодія продемонстрована на рисунку 5.5.

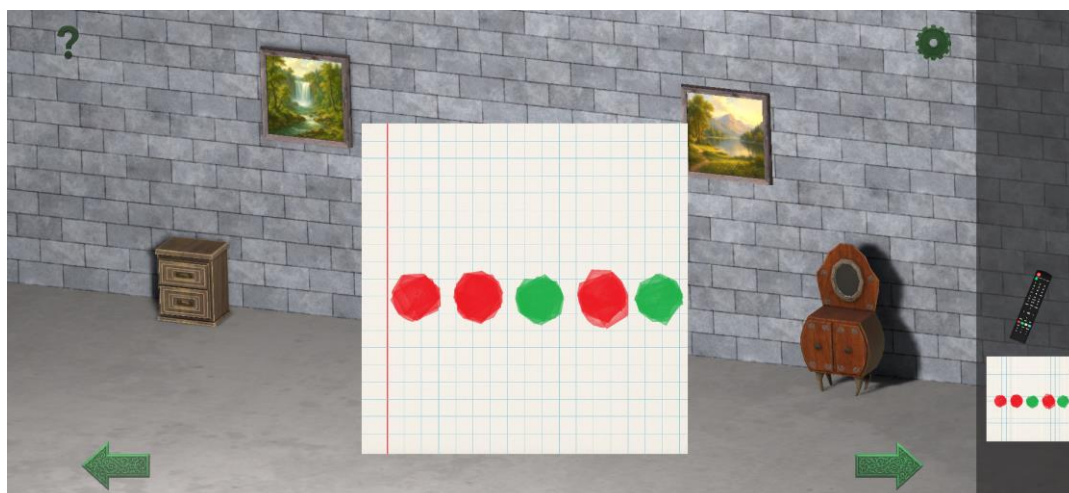


Рисунок 5.5 – Перегляд інформаційного предмета

Для просування в грі користувачеві необхідно взаємодіяти з об'єктами в середовищі. Так, наприклад, на рисунку 5.6 показано рівень із дошкою для гри в хрестики-нулики: щоб пройти його, гравець має перемогти в партії проти штучного інтелекту, по черзі обираючи клітинки та стратегічно розміщуючи символи.



Рисунок 5.6 – Приклад рівню гри

Крім того, на початкових рівнях передбачені підказки, який допомагає гравцю освоїти основні принципи геймплею. Для цього використовується мигаюча іконка у вигляді вказівного пальця, що поетапно демонструє необхідні дії. На рисунку 5.7 зображено приклад стартового рівня з такими підказками.



Рисунок 5.7 – Приклад стартового рівня

На деяких рівнях для повороту камери використовуються дві стрілки, розміщені в нижній частині екрану. Вони дозволяють гравцеві переключатися між стінами кімнати вліво або вправо, забезпечуючи навігацію в межах ігрового простору без необхідності вільного переміщення. Зовнішній вигляд елементів керування продемонстровано на рисунку 5.8.

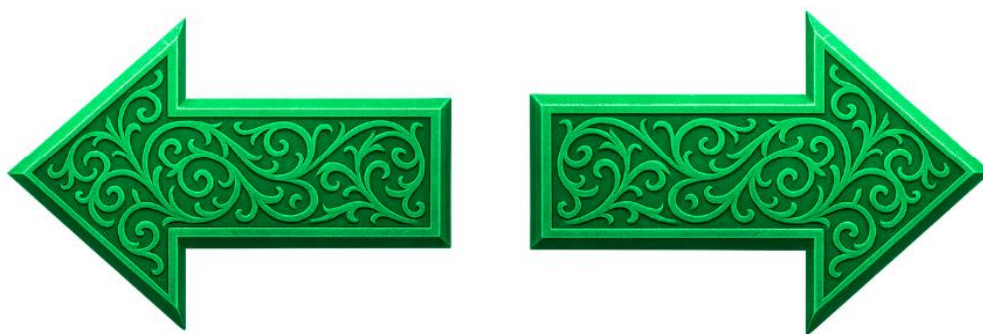


Рисунок 5.8 – Елементи інтерфейсу для повороту камери

Після успішного завершення рівня на екрані з'являється спеціальна панель із короткою статистикою проходження (рисунок 5.9). Крім цього, панель містить кнопку, яка дозволяє одразу перейти до наступного рівня без повернення в головне меню. Такий підхід сприяє безперервному ігровому процесу й зручності користувача.

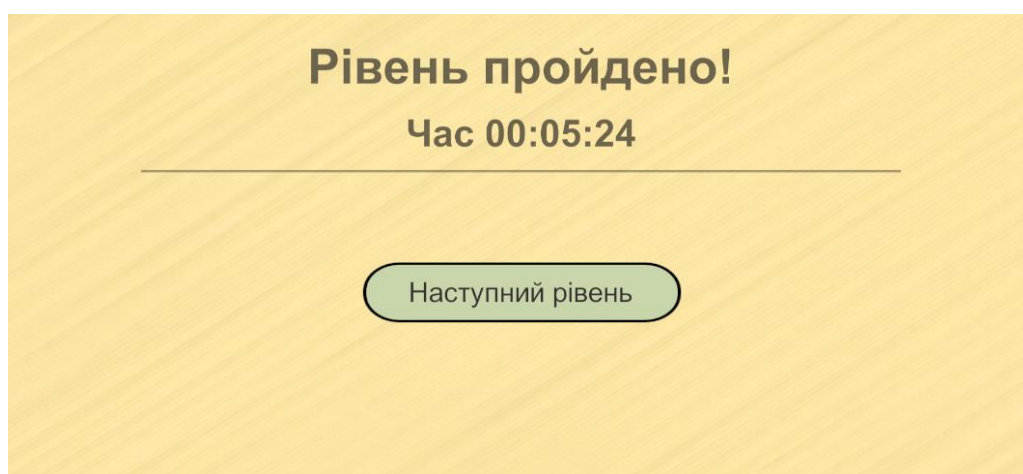


Рисунок 5.9 – Статистика проходження рівня

Також, інтерфейс ігрового вікна містить іконку у вигляді шестерні, що відкриває меню налаштувань. У цьому меню користувач має доступ до основних параметрів, які впливають на комфорт проходження гри. Передбачено окремі кнопки для вмикання та вимикання звуків і фонові музики. Стан кожної з них позначається кольором: зелений відтінок означає, що параметр увімкнено, червоний – що вимкнено.

Також у меню розміщено кнопку перемикання мови між українською та англійською. Вибрана мова застосовується до всього інтерфейсу гри.

Крім того, присутня кнопка для повторного запуску поточного рівня — вона дає змогу почати його заново з початкового стану. Ще одна кнопка відповідає за вихід у головне меню без завершення гри. Остання, шоста кнопка, відкриває сторінку гри в Google Play для її оцінювання. На рисунку 5.10 продемонстровано інтерфейс меню налаштувань.



Рисунок 5.10 – Меню налаштувань

ВИСНОВКИ

У межах виконання кваліфікаційної роботи було реалізовано повноцінний мобільний застосунок у жанрі тривимірної головоломки з камерою від першої особи. Увагу зосереджено на забезпеченні логічної послідовності завдань, відсутності елементів випадковості, зручному управлінні та глибокому зануренні в ігровий процес без зайвих відволікань на рекламу чи механіки типу таймерів.

У ході реалізації було підібрано оптимальне програмне забезпечення. Імплементацию ігрової логіки здійснено в середовищі Unity з використанням мови програмування C#. Код написано таким чином, щоб забезпечити модульність, розширюваність та стабільну роботу на мобільних пристроях. Взаємодія з користувачем реалізована через інтуїтивно зрозумілий інтерфейс, з можливістю вибору мови, перемикання між рівнями, налаштуваннями звуку та інвентарем.

Ключовими реалізованими функціями стали фіксована камера з перемиканням між стінами, система підбору та застосування предметів, анімації, інтерактивні головоломки з візуальними підказками, а також інтеграція звукового супроводу. Окрему увагу приділено оптимізації: зменшено навантаження на процесор і графіку, враховано обмеження текстур, освітлення та скриптів для збереження плавності геймплею.

Таким чином, у результаті роботи вдалося не лише реалізувати поставлену мету, а й створити функціональний базис для подальшого розширення гри – додавання нових рівнів, сценаріїв, інтерактивних елементів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Newzoo. Newzoo's Global Games Market Report 2024 [Електронний ресурс]. – Режим доступу: <https://newzoo.com/resources/trend-reports/newzoos-global-games-market-report-2024-free-version>. – Дата звернення: 02.05.2025.
2. Lebow S. More people over the age of 45 started mobile gaming in 2020 than any other age group [Електронний ресурс] / S. Lebow. – eMarketer, 2021. – 5 березня. – Режим доступу: <https://www.emarketer.com/content/more-people-over-age-of-45-started-mobile-gaming-2020-than-any-other-age-group>. – Дата звернення: 02.05.2025.
3. Chris Crawford. The Art of Computer Game Design. – 1-st ed. – Berkeley: McGraw-Hill/Osborne Media, 1984. – 113 с. – ISBN 0-88134-117-7.
4. Wendy Despain (ed.). Writing for Video Game Genres: From FPS to RPG. – 1-st ed. – A K Peters/CRC Press, 2009. – 305 с. – ISBN 978-1568814179.
5. Tracy Fullerton. Game Design Workshop: A Play-Centric Approach to Creating Innovative Games. – 4-th ed. – New York: A K Peters/CRC Press, 2023. – 556 с. – ISBN 978-1-138-09877-0.
6. Levitin A., Levitin M. Algorithmic Puzzles. – 1st ed. – Oxford: Oxford University Press, 2011. – 192 с. – ISBN 9780199740449.
7. Hearn R. A., Demaine E. D. Games, Puzzles, and Computation. – 1-st ed. – Natick: A K Peters/CRC Press, 2009. – 464 с. – ISBN 9781568813226.
8. Blain J. M. The Complete Guide to Blender Graphics: Computer Modeling & Animation. – 6th ed. – Boca Raton: CRC Press, 2022. – 576 с. – ISBN 9780367741365.
9. Hill T. The Essential Beginners Guide to Blender: 2023 Edition. – Fairfield: Quail Ridge Books, 2023. – 146 с. – ISBN 979-8393406950.

10. Renderfield M. Mastering Blender 4.2: Comprehensive Step-by-Step Guide to Create Stunning 3D Arts, Animations & Transforming Your Imagination into Reality. – [S.l.]: Independently Published, 2024. – 200 c. – ISBN 979-8328955355.

11. Arevalo-Tovar R., Li M. Creating Games with Unity, Substance Painter & Maya: Models, Textures, Animation & Blueprint. – Boca Raton: CRC Press, 2021. – 320 c. – ISBN 9780367506017.

12. Shah Z. J. Realistic Asset Creation with Adobe Substance 3D: Create materials, textures, filters, and 3D models using Substance 3D Painter, Designer, and Stager. – 1-st ed. – Birmingham: Packt Publishing Ltd, 2022. – 486 c. – ISBN 9781803233406.

13. Faulkner A., Chavez C. Adobe Photoshop Classroom in a Book (2024 Release). – San Francisco: Adobe Press, 2023. – 416 c. – ISBN 9780138262525.

14. Hocking J. Unity in Action: Multiplatform Game Development in C# with Unity. – 2nd ed. – Shelter Island: Manning Publications, 2018. – 400 c. – ISBN 9781617294969.

15. Goldstone W. Unity 2021 By Example: Learn about game and virtual reality development by creating five engaging projects. – Birmingham: Packt Publishing, 2021. – 820 c. – ISBN 9781801079714.

16. Davis A., Baptiste T. Unity 3D Game Development: Designed for Passionate Game Developers. – Birmingham: Packt Publishing, 2021. – 370 c. – ISBN 9781801076140.

17. Cameron S. H. Unity 2022 by Example: A project-based guide to building 2D and 3D games, enhanced for AR, VR, and MR experiences. – 1st ed. – Birmingham: Packt Publishing, 2024. – 596 c. – ISBN 9781803234595.

18. Hussain K., Hussain F. Mastering Visual Studio Code: Navigating the Future of Development. – [S.l.]: [s.n.], 2024. – 232 c. – ISBN 9798877664975.