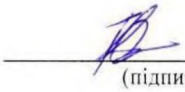
МІНІСТЕРСТВО ФІНАНСІВ УКРАЇНИ
ДЕРЖАВНИЙ ПОДАТКОВИЙ УНІВЕРСИТЕТ

ФАКУЛЬТЕТ ФІНАНСІВ ТА ЦИФРОВИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І СИСТЕМ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: Розробка комп'ютерної гри на платформі Unity

Здобувача вищої освіти першого (бакалаврського) рівня,
IV курсу, групи ЦДБ-21-1, спеціальності 122 «Комп'ютерні науки»,
освітня програма «Технології цифрового дизайну»


(підпис)

Торгонського Олексія Сергійовича
(прізвище, ім'я, по батькові)

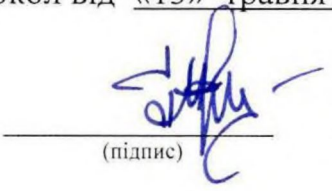
Науковий керівник 
(підпис)

Ратушняк Тетяна Володимирівна, к.ф.-м.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Керівник освітньо-професійної програми 
(підпис)

Ратушняк Тетяна Володимирівна, к.ф.-м.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Допущено до захисту за рішенням кафедри комп'ютерних та інформаційних технологій і систем, протокол від «13» травня 2025 № 18

Завідувач кафедри 
(підпис)

Філоненко Михайло Миколайович, к.ф.-м.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Ірпінь – 2025

Реєстрація № 69 29.05.2025 _____
(дата) (підпис відповідальної особи)

Результати перевірки до захисту 29.05.2025 _____
(до захисту, на доопрацювання) (дата) (підпис)

Результат захисту 4 (добре) 11.06.2025 _____
(оцінка) (дата) (підпис)

Голова комісії: _____ В.М. Франчук
(підпис)

Члени комісії: _____ М.М. Філоненко
(підпис)

_____ Т.В. Ратушняк
(підпис)

_____ В.О. Ніжегородцев
(підпис)

БЛАНК ОЦІНЮВАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Об'єкти контролю	Діапазон оцінки в балах	Фактична кількість балів	Примітки	Дата, П.І.Б.: керівника, голови комісії членів комісії,секретаря, (підписи)
ОЦІНЮВАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ КОМІСІЄЮ ПО ЗАХИСТУ				
1. Зміст випускної роботи (якість теоретичного матеріалу, його зв'язок з практичним, вміння представляти матеріал в логічній послідовності, наявність висновків та пропозицій)	від 0 до 30	25		Ратушняк Т.В. Франчук В.М. Філоненко М.М. Ратушняк Т.В. Ніжегородцев В.О. Суховецька Л.П.
2. Дотримання вимог щодо оформлення	від 0 до 10	8		
3. Вчасне виконання роботи на всіх етапах у встановлені кафедрою терміни	від 0 до 5	4		
4. Рецензент	від 0 до 5	4		
РАЗОМ БАЛІВ	від 0 до 50	41		

Оцінювання кваліфікаційної роботи науковим керівником - 40 балів.

ПЕРЕВЕДЕННЯ ОЦІНКИ ЗА 100-БАЛЬНОЮ ШКАЛОЮ ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

Оцінка за 100-бальною шкалою	Оцінка за шкалоюЄКТС	Оцінка за національною шкалою	Дата	П.І.Б.: голови, членів комісії, секретаря, (підпис)
81	B	4 (добре)	11.06.2025	Франчук В.М. Філоненко М.М. Ратушняк Т.В. Ніжегородцев В.О. Суховецька Л.П.

ПРАВИЛА ПЕРЕВЕДЕННЯ ОЦІНОК ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

Оцінка за 100-бальною шкалою	Оцінка за шкалою ЄКТС	Оцінка за національною шкалою
90-100	A	5 (відмінно)
80-89	B	4 (добре)
70-79	C	
60-69	D	3 (задовільно)
50-59	E	
35-49	FX	2 (незадовільно)

Факультет фінансів та цифрових технологій
Кафедра комп'ютерних та інформаційних технологій і систем
Галузь знань: 12 «Інформаційні технології»
Спеціальність: 122 «Комп'ютерні науки»
Освітня програма: «Технології цифрового дизайну»

Затверджую
Завідувач кафедри _____ Філоненко М.М.
«14» листопада 2024

Завдання на кваліфікаційну роботу
здобувачу вищої освіти
Торгонському Олексію Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи
Розробка комп'ютерної гри на платформі Unity
Затверджена наказом ректора від «13» листопада 2024 № 1717
2. Строк здачі закінченої роботи «12 » травня 2025
3. Мета роботи: розробити комп'ютерну гру з 2D-графікою на платформі Unity.
Об'єкт дослідження: інформаційні технології для розробки комп'ютерної гри з двовимірною графікою, цифровий інструмент розробки і рушій відеоігор UNITY.
Предмет дослідження: методи розробки, етапи розробки, прийоми оптимізації комп'ютерної гри на платформі Unity.
Орієнтовна база дослідження: ТОВ «АВТОРІТЕЙЛ АСИСТАНС»
4. Вихідні дані до роботи: програмне забезпечення (редактор графіки Adobe Photoshop для 2D-анімацій та спрайтів, IDE Microsoft Visual Studio, ігрова платформа Unity); наукові публікації, референси відеоігор з двовимірною графікою.
5. Перелік графічного матеріалу: схеми анімації, скріншоти інтерфейсів програми, етапи створення 2D-анімацій та 2D-спрайтів, програмний код мовою C# в середовищі Visual Studio, налаштування в Unity, приклади роботи анімацій, фінальний вид гри.
6. Зміст кваліфікаційної роботи (перелік питань за кожним розділом)
ВСТУП
1. РОЗРОБКА КОНЦЕПЦІЇ КОМП'ЮТЕРНОЇ ГРИ
1.1. Огляд популярних платформ для розробки ігор
1.2. Огляд референсів комп'ютерної гри та технологій для розробки гри
1.3. Ідея, жанр та сюжет гри. Опис персонажів, локацій та об'єктів
1.4. Розробка сценарію та рівнів
2. ТЕХНОЛОГІЧНА РЕАЛІЗАЦІЯ ПРОЄКТУ
2.1. Створення ігрового дизайну (UI/UX-дизайн)
2.2. Створення графічних елементів: спрайти, анімації, фони
2.3. Структура проєкту в середовищі Unity
2.4. Реалізація основних механік гри
2.5. Тестування функціоналу гри
3. ОЦІНКА ЕФЕКТИВНОСТІ ТА ПЕРСПЕКТИВИ РОЗВИТКУ
3.1. Верифікація результатів проєкту
3.2. Можливості масштабування гри
3.3. Рекомендації щодо публікації гри
ВИСНОВКИ; СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

7. Календарний план виконання роботи

№ п/п	Назва етапів кваліфікаційноїроботи	Строк виконання етапів роботи	
		За планом	Фактично
1	2	3	4
1	Вибір теми, підбір літературних джерел, формування структури кваліфікаційної роботи	До 30.10.2024	До 30.10.2024
2	**І розділ** (розкрити сутність і стан дослідження наукової проблематики роботи, після чого навести її загальну характеристику; огляд літератури за темою і вибір конкретних напрямів досліджень)	25.12.2024	25.12.2024
3	**ІІ розділ** (аналіз експериментальної частини про методики досліджень)	15.02.2025	15.02.2025
4	**ІІІ розділ** (проведені теоретичні і (або) експериментальні дослідження та проектні рішення)	14.04.2025	14.04.2025
5	**Висновки**	01.05.2025	01.05.2025
6	Доопрацювання роботи з урахуванням зауважень наукового керівника	09.05.2025	09.05.2025
7	Отримання відгуку керівника	03.06.2025	03.06.2025
8	Отримання зовнішньої рецензії	05.06.2025	05.06.2025

8. Дата видачі завдання «14» листопада 2024

9. Керівник кваліфікаційної роботи _____ (підпис) Ратушняк Т. (прізвище, ініціали

10. Керівник освітньо-професійної програми _____ (підпис) Ратушняк Т. (прізвище, ініціали

11. Завдання до виконання отримав здобувач вищої освіти _____ (підпис) Торгонський (прізвище, ініціали

АНОТАЦІЯ

Розробка комп'ютерної гри на платформі Unity

Мета роботи: розробити комп'ютерну гру з 2D-графікою на платформі Unity.

Об'єкт дослідження: інформаційні технології розробки комп'ютерної гри з двовимірною графікою, цифровий інструмент розробки і рушій відеоігор UNITY.

Завдання дослідження: аналіз теоретичних основ розробки комп'ютерної гри з двовимірною графікою та інструментів для її реалізації; розробка сюжету та персонажів гри; розробка UI/UX-дизайну гри; створення графічних елементів (спрайти, анімації, фони); реалізація функціональної частини гри засобами мови програмування C#; проведення тестування та оцінки ефективності створеного програмного продукту.

Об'єкт дослідження: інформаційні технології для розробки комп'ютерної гри з двовимірною графікою.

Предмет дослідження: методи розробки, етапи розробки, прийоми оптимізації комп'ютерної гри на платформі Unity

Методи дослідження: аналіз літературних джерел – для вивчення теоретичних основ розробки комп'ютерної гри з 2D-графікою, порівняльний аналіз – для оцінки функціональних можливостей різних інструментів розробки комп'ютерної гри з двовимірною графікою, практичний метод – для розробки відеогри, метод тестування – для перевірки функціоналу гри.

База дослідження: ТОВ «АВТОРІТЕЙЛ АСИСТАНС».

Одержані результати можуть бути використані у навчальних та індивідуальних проєктах як приклад розробки гри на платформі Unity.

Кваліфікаційна робота містить 50 сторінок, 11 рисунків, список використаних джерел з 41 найменувань.

Ключові слова: цифрове проєктування, відеогра, UX/UI-дизайн, двовимірна графіка, Unity.

Рік виконання кваліфікаційної роботи 2025.

Рік захисту роботи 2025.

ABSTRACT

Development of a computer game on the Unity platform

Purpose of the work: to develop a computer game with 2D graphics on the Unity platform.

Object of research: information technologies for developing a computer game with two-dimensional graphics, a digital development tool and a video game engine UNITY.

Research tasks: analysis of the theoretical foundations of developing a computer game with two-dimensional graphics and tools for its implementation; development of the plot and characters of the game; development of the game's UI/UX design; creation of graphic elements (sprites, animations, backgrounds); implementation of the functional part of the game using the C# programming language; testing and evaluation of the effectiveness of the created software product.

Object of research: information technologies for developing a computer game with two-dimensional graphics.

Subject of research: development methods, development stages, optimization techniques for a computer game on the Unity platform

Research methods: analysis of literary sources - to study the theoretical foundations of developing a computer game with 2D graphics, comparative analysis - to assess the functionality of various tools for developing a computer game with two-dimensional graphics, practical method - for developing a video game, testing method - to verify the functionality of the game.

The results obtained can be used in educational and individual projects as an example of developing a game on the Unity platform.

The qualification work contains 50 pages, 11 figures, a list of used sources with 41 names.

Keywords: digital design, video game, UX/UI design, two-dimensional graphics, Unity.

Year of completion: 2025

Year of defense: 2025

ЗМІСТ

ВСТУП 3

1. РОЗРОБКА КОНЦЕПЦІЇ КОМП'ЮТЕРНОЇ ГРИ 5

1.1. Огляд популярних платформ для створення ігор 5

1.2. Огляд аналогів комп'ютерної гри, ігрових механік та технологій для реалізації гри 8

1.3. Ідея, жанр та сюжет гри. Опис персонажів, локацій та об'єктів 11

1.4. Розробка сценарію та рівнів 14

2. ТЕХНОЛОГІЧНА РЕАЛІЗАЦІЯ ПРОЄКТУ 17

2.1. Створення ігрового дизайну (UI/UX-дизайн) 17

2.2. Створення графічних елементів: спрайти, анімації, фони 18

2.3. Структура проєкту в середовищі Unity 21

2.4. Реалізація основних механік гри 24

2.5. Тестування функціоналу гри 36

3. ОЦІНКА ЕФЕКТИВНОСТІ ТА ПЕРСПЕКТИВИ РОЗВИТКУ 39

3.1. Верифікація результатів проєкту 39

3.2. Можливості масштабування гри 40

3.3. Рекомендації щодо публікації гри на платформа 42

ВИСНОВКИ 44

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 46

ВСТУП

Індустрія комп’ютерних ігор є однією з найдинамічніших сфер цифрових технологій. З кожним роком вона демонструє стабільне зростання, залучаючи мільйони користувачів по всьому світу. Сучасні вимоги до якості ігор охоплюють не лише графічне оформлення, а й розробку цікавого сюжету, зручного інтерфейсу користувача, продуманої ігрової механіки та оптимізованого функціоналу. У зв’язку з цим, актуальним є використання потужних інструментів для розробки ігор, таких як Unity — кросплатформеного ігрового рушія, що надає широкі можливості для створення 2D та 3D ігор різного рівня складності.

Особливий інтерес викликає розробка ігор для персонального комп’ютера з використанням інструментів візуального дизайну, сценаріїв програмування, об’єктно-орієнтованої моделі та анімації. Unity забезпечує ефективне поєднання художнього підходу з технічною реалізацією, що дозволяє студентам і початківцям розробляти повноцінні ігрові проєкти. Саме тому тема створення комп’ютерної гри за допомогою Unity є актуальною в контексті підготовки фахівців з цифрового дизайну та програмування.

Історія комп’ютерних ігор бере свій початок ще у 50–60-х роках XX століття, коли комп’ютери лише почали з’являтись у наукових і військових установах. Перші прототипи ігор не мали комерційного значення і були розроблені переважно ентузіастами для демонстрації можливостей обчислювальних машин. Однією з перших таких ігор вважається "Tennis for Two" (1958), створена Вільямом Гіґінботемом на осцилографі. У 1962 році в Массачусетському технологічному інституті була створена інша знаменита гра — "Spacewar!", яка вже мала просту механіку бою космічних кораблів.

Поява мікропроцесорів і розвиток персональних комп’ютерів у 70-х роках дали потужний поштовх до розширення індустрії. У 1972 році компанія Atari випустила "Pong", що стала першою успішною аркадною грою. Вона започаткувала еру ігрових автоматів, які стали надзвичайно популярними у 70–

80-х роках. Саме в цей період з’явились класичні ігри, що стали знаковими: "Space Invaders" (1978), "Pac-Man" (1980), "Donkey Kong" (1981).

1980-ті роки характеризуються стрімким розвитком домашніх ігрових консолей та комп’ютерів. Вихід Nintendo Entertainment System (NES) у 1983 році ознаменував нову еру, в якій японські компанії зайняли провідне місце на ринку. З’явились перші гри з персонажами, які стали іконами ігрової культури — Super Mario Bros., The Legend of Zelda, Metroid тощо. У цей же період комп’ютери на зразок Commodore 64 і ZX Spectrum дозволили користувачам не тільки грати, а й програмувати власні ігри.

У 90-х роках відбувся перехід до 3D-графіки, який докорінно змінив підхід до геймдизайну. Ігри стали більш візуально привабливими, динамічними та складними. Вихід Wolfenstein 3D (1992), Doom (1993), Quake (1996) започаткував жанр шутерів від першої особи, а компанії на кшталт id Software, Blizzard та Electronic Arts почали формувати ринок ігрової індустрії як великий бізнес. У 1994 році вийшла Sony PlayStation, яка зробила 3D-ігри масовими завдяки своїм технічним можливостям і маркетинговій стратегії.

Початок 2000-х ознаменувався розвитком онлайн-ігор, зокрема MMORPG (наприклад, World of Warcraft), а також появою незалежних (indie) проєктів. Розвиток інтернету сприяв новим формам розповсюдження ігор — через цифрові платформи як Steam. У 2010-х роках мобільні ігри на платформах Android та iOS стали новим масовим явищем, відкривши геймплей для мільйонів користувачів.

Сьогодні комп’ютерні ігри є потужною індустрією, яка поєднує мистецтво, програмування, музику, анімацію, наратив і комунікацію. Вони використовуються не тільки для розваг, а й у навчанні, симуляції, терапії, культурній спадщині. Штучний інтелект, доповнена та віртуальна реальність, кросплатформність і хмарні технології — все це формує майбутнє цифрових ігор. Історія комп’ютерних ігор, що розпочалась з простої точки на екрані осцилографа, сьогодні є одним з найважливіших проявів сучасної цифрової культури.

1. РОЗРОБКА КОНЦЕПЦІЇ КОМП'ЮТЕРНОЇ ГРИ

1.1. Огляд популярних платформ для створення ігор

У сучасному світі створення комп'ютерних ігор стало значно доступнішим завдяки розробці потужних, гнучких та інтуїтивно зрозумілих ігрових рушіїв. Такі платформи забезпечують розробників усіма необхідними інструментами — від візуального редагування сцен до складного кодування логіки гри. Існує багато рушіїв, однак кілька з них набули світової популярності завдяки своїй функціональності, підтримці різних платформ і активній спільноті.

Unity — один з найпопулярніших ігрових рушіїв у світі, особливо серед незалежних розробників. Він підтримує 2D та 3D-графіку, дозволяє експортувати ігри на десятки платформ, включно з Windows, macOS, Android, iOS, WebGL, PlayStation, Xbox, Nintendo Switch тощо. Рушій базується на мові програмування C#, що робить його досить зручним для широкого кола розробників. Unity має потужний інтерфейс редактора сцен, фізичний рушій, інструменти для анімації, а також велику бібліотеку готових рішень у магазині Unity Asset Store. Ігри на Unity варіюються від простих мобільних аркад до серйозних 3D-проєктів.

Unreal Engine — ще один провідний ігровий рушій, відомий завдяки своїй високій якості графіки. Він часто використовується у створенні тривимірних ігор преміум-класу (AAA). Unreal Engine підтримує розробку за допомогою мови C++, а також пропонує Blueprints — візуальну систему програмування, яка дозволяє створювати логіку гри без написання коду. Завдяки потужному рендеринговому модулю Unreal Engine часто використовується не лише у геймдеві, а й у візуалізації архітектури, фільмах, симуляціях. У 2020-х роках рушій активно розвивається як інструмент для роботи з віртуальною та доповненою реальністю.

Godot Engine — безкоштовний рушій з відкритим кодом, який набирає популярності серед інді-розробників. Godot підтримує 2D та 3D-графіку, має власну мову програмування **GDScript**, подібну до Python. Перевагою є легкість

у навчанні, швидкість розробки та гнучка структура сцен. Godot чудово підходить для невеликих ігор, мобільних додатків і прототипів. Завдяки відкритому коду, рушій можна адаптувати до власних потреб, що робить його привабливим для експериментальних проєктів.

GameMaker Studio — платформа, орієнтована на створення 2D-ігор. Вона має простий інтерфейс і дозволяє як використовувати візуальне програмування, так і писати код на власній мові GML (GameMaker Language). GameMaker популярний серед розробників мобільних та інді-ігор, завдяки швидкому запуску проєктів і гнучкій роботі з анімацією та об’єктами.

Крім того, варто згадати **Construct**, **RPG Maker**, **CryEngine**, **Defold**, які теж мають свою нішу і активно використовуються у відповідних жанрах та форматах.

У підсумку, вибір платформи для створення ігор залежить від цілей проєкту, рівня досвіду розробника, вимог до графіки, платформи релізу та інших факторів. Unity і Unreal Engine залишаються лідерами завдяки універсальності та потужному інструментарію, однак альтернативні рушії також пропонують цікаві можливості, особливо для нішевих та інді-проєктів.

Аналіз середовища Unity: можливості та переваги

Розглянемо характеристики середовища ігрового рушія Unity:

1. Багатоплатформність

Однією з ключових переваг Unity є його здатність експортувати проєкти на різні платформи, включаючи Windows, macOS, Linux, iOS, Android, а також ігрові консолі, такі як PlayStation та Xbox. Це дозволяє розробникам охоплювати широку аудиторію без необхідності створювати окремі версії гри для кожної платформи. Крім того, Unity підтримує WebGL, що дає змогу запускати ігри безпосередньо в браузері без додаткових плагінів.

2. Інтегрований редактор та зручний інтерфейс

Unity надає інтуїтивно зрозумілий інтерфейс користувача, який включає в себе візуальний редактор сцен, інспектор властивостей, ієрархію об'єктів та інші інструменти, що спрощують процес розробки. Це дозволяє розробникам швидко

створювати та налаштовувати ігрові об'єкти, а також бачити результати змін у реальному часі.

3. Підтримка C# та об'єктно-орієнтованого програмування

Unity використовує мову програмування C#, яка є популярною та потужною мовою для розробки ігор. Це забезпечує розробникам доступ до широкого спектру функціональних можливостей, а також дозволяє використовувати об'єктно-орієнтований підхід у створенні ігрової логіки [2].

4. Велика спільнота та доступність ресурсів

Unity має велику та активну спільноту розробників, що сприяє обміну знаннями та досвідом. Існує безліч навчальних матеріалів, форумів, відеоуроків та документації, які допомагають як новачкам, так і досвідченим розробникам у вирішенні різноманітних завдань.

5. Магазин Asset Store

Unity Asset Store — це онлайн-магазин, де розробники можуть купувати або безкоштовно завантажувати різноманітні ресурси для своїх проєктів, такі як моделі, текстури, скрипти, анімації та інші. Це значно прискорює процес розробки та дозволяє зосередитися на унікальних аспектах гри .

6. Підтримка 2D та 3D графіки

Unity надає потужні інструменти для створення як 2D, так і 3D ігор. Це включає в себе системи анімації, фізики, освітлення та інші функції, що дозволяють створювати візуально привабливі та динамічні ігрові світи [3-5, 14].

7. Гнучкість та масштабованість

Unity підходить як для невеликих інді-проєктів, так і для великих комерційних ігор. Його архітектура дозволяє легко масштабувати проєкти та додавати нові функціональні можливості в міру зростання потреб розробників.

8. Інтеграція з іншими інструментами та сервісами

Unity підтримує інтеграцію з різноманітними сторонніми сервісами та інструментами, такими як аналітика, реклама, хмарні сервіси та інші. Це дозволяє розробникам розширювати функціональність своїх ігор та забезпечувати кращий користувацький досвід.

9. Безперервний розвиток та оновлення

Unity Technologies постійно оновлює та вдосконалює свій рушій, додаючи нові функції, покращуючи продуктивність та виправляючи помилки. Це забезпечує розробникам доступ до найсучасніших технологій та інструментів для створення високоякісних ігор.

1.2. Огляд аналогів комп'ютерної гри, ігрових механік та технологій для реалізації гри

На першому етапі розробки комп'ютерної гри важливо провести детальний огляд існуючих аналогів, які вже представлені на ринку, а також дослідити найпоширеніші та ефективні ігрові механіки. Такий підхід дозволяє не тільки уникнути повторення чужих помилок, але й надихнутись успішними рішеннями, покращити власну ідею та зробити гру більш цікавою й конкурентоспроможною.

Огляд аналогів комп'ютерної гри

Якщо розробляється 2D-платформер, доцільно розглянути такі популярні ігри як *Hollow Knight, Celeste, Dead Cells* або *Shovel Knight*. Ці проєкти відзначаються глибоким геймплеєм, цікавим візуальним стилем та вдалою побудовою рівнів. Варто звернути увагу на такі аспекти:

- Графічний стиль — у більшості успішних платформерів використовується унікальний художній стиль (піксель-арт, мультяшна графіка або темна готична естетика);

- Ігрова фізика — плавність рухів, реакція персонажа на дії гравця, відчуття ваги та інерції є критично важливими для створення задоволення від геймплею;

- Рівень складності — ігри часто будуються на принципі «легко навчитись, важко опанувати», що дозволяє залучити широку аудиторію;

- Наявність прогресії — гравець має можливість покращувати навички персонажа, відкривати нові рівні чи здібності, що мотивує проходити гру далі.

Огляд ігрових механік

Ігрові механіки — це базові правила та взаємодії, які визначають, як гравець контролює персонажа, виконує завдання, долає перешкоди тощо.

Найбільш поширені механіки включають:

- Пересування — стрибки, біг, лазіння, ковзання тощо. Ці дії повинні бути інтуїтивно зрозумілими та мати чіткий зворотний зв’язок;

- Бойова система — може варіюватися від простої атаки в один дотик до складної комбінаційної системи;

- Взаємодія з об’єктами — відкривання дверей, збирання предметів, активація важелів — усе це створює відчуття живого світу;

- Штучний інтелект ворогів — його варіативність дозволяє зробити гру динамічнішою та непередбачуваною;

- Система життя / шкоди / збереження прогресу — часто застосовуються серця або шкала здоров’я, чекпоїнти, які дозволяють не починати гру з початку після кожної помилки.

Після огляду аналогів і механік розробник може обрати ті, що найкраще підходять до задуму гри, та адаптувати їх відповідно до власної концепції. Наприклад, можна поєднати платформерну механіку з елементами головоломки або інтерактивного середовища, де гравець не лише долає ворогів, а й повинен вирішувати логічні завдання, щоб просуватись далі.

Таким чином, огляд референсів є важливим етапом у процесі розробки нової гри.

Обґрунтування вибору технологій та інструментів для розробки гри

При розробці комп’ютерної гри важливим етапом є вибір відповідних технологій — інструментів, мов програмування, середовищ розробки та ресурсів, які дозволять ефективно реалізувати ігровий задум.

Технології для реалізації гри:

1. Ігровий рушій — Unity

Оскільки у процесі було використано практичний гайд на базі ігрового рушія Unity, усі технічні рішення були обґрунтованими, актуальними та перевіреними на практиці. Основою проекту став рушій Unity, який є одним з найпопулярніших середовищ для створення 2D та 3D-ігор. Його переваги:

- Міжплатформеність — Unity дозволяє створювати ігри для Windows, Android, iOS, WebGL та інших платформ;

- Потужний редактор сцен — дає змогу зручно розміщувати об’єкти, налаштовувати компоненти, працювати з анімацією та фізикою;

- Розвинена спільнота — велика база знань, документації, навчальних курсів і форумів спрощує процес навчання та вирішення проблем;

- Гнучка система компонентів — об’єкти в грі складаються з компонентів, що легко змінюються або додаються без потреби переписувати код;

- Магазин ресурсів (Asset Store) — дозволяє завантажувати безкоштовні або платні графічні, звукові, анімаційні та скриптові ресурси для прискорення розробки.

2. Мова програмування — C#

Для написання логіки гри було використано C#, офіційну мову програмування для Unity. Її переваги:

- Об’єктно-орієнтована структура — полегшує організацію коду, забезпечує повторне використання класів і методів;

- Висока читабельність — синтаксис зрозумілий навіть початківцям;

- Багата стандартна бібліотека — містить все необхідне для роботи з математикою, графікою, подіями тощо;

- Простота взаємодії з елементами Unity — C# має повну інтеграцію з Unity API, що дозволяє взаємодіяти з об’єктами сцени, трансформами, анімаціями, колайдерами та іншими елементами гри.

3. Графічні та звукові ресурси

Для візуального оформлення гри використовувалися:

- Плоска 2D-графіка (спрайти) — найкраще підходить для простих платформерів і дозволяє створити легкий стиль з невеликим обсягом пам’яті;

- Sprite sheets та анімації — дозволяють реалізувати плавні рухи персонажа, ефекти зіткнень, стрибків, атак тощо;

- Аудіофайли — звукові ефекти для стрибків, атак, поразок, а також фонове звучання гри.

Усі ресурси можна було завантажити з Unity Asset Store або створити вручну за допомогою програм типу Photoshop, Aseprite, або Audacity для звуку.

4. Інструменти розробника

У процесі реалізації також використовувалися такі інструменти:

- Visual Studio — основне середовище для написання коду C#, що має підтримку автозаповнення, налагодження, підказок;

- Git (опціонально) — система контролю версій, що дозволяє відстежувати зміни в проєкті та працювати в команді;

- Unity Hub — менеджер проєктів, який спрощує запуск різних версій Unity та організовує середовище розробки.

1.3. Ідея, жанр та сюжет гри. Опис персонажів, локацій та об’єктів

Наступним етапом у розробці будь-якої комп’ютерної гри є формування загальної ідеї, визначення жанру та створення базового сюжету. Ці складові слугують фундаментом для всієї подальшої реалізації — від візуального стилю до програмної архітектури та механік геймплею.

Ідея гри

Основна ідея полягає у створенні простої 2D-платформерної гри, що поєднує динамічний геймплей зі зручним керуванням і приємним візуальним стилем. Гравець керує персонажем, який долає перешкоди, стрибає через платформи, уникає ворогів та збирає ігрові об’єкти. Мета — досягти кінця рівня, залишаючись неушкодженим.

Ця концепція є класичною для жанру платформерів, але завжди залишається актуальною завдяки своїй простоті, динамічності та можливості додавання нових механік у майбутньому.

Жанр гри

Гра відноситься до жанру 2D платформер (англ. 2D platformer), який характеризується:

- управлінням персонажем у двовимірному просторі;

- платформами, по яких гравець пересувається;

- наявністю перешкод, ворогів, колекторних елементів;

- стрибками, атаками та іншими базовими діями.

Цей жанр добре підходить для реалізації в середовищі Unity, адже рушій має вбудовані інструменти для роботи з фізикою, колізіями, анімацією, що значно спрощує реалізацію платформи та рухів персонажа.

Сюжет гри

Сюжет гри є лаконічним і слугує радше фоном для ігрового процесу, ніж повноцінною історією. Гравець виступає в ролі хороброго героя, який подорожує крізь небезпечні території, аби зібрати цінні артефакти (або інші об’єкти, наприклад, монети) та дістатися фінішу кожного рівня.

На його шляху трапляються вороги, ями, пастки та інші перешкоди, які потрібно обійти або подолати. Головна мета — вижити та пройти всі рівні, удосконалюючи власні навички гри. Така структура дозволяє зосередитись на розробці якісного геймплею з потенціалом для розширення (наприклад, додавання нових рівнів, босів, апгрейдів тощо).

Гравець і цільова аудиторія

Гра орієнтована на широку аудиторію, включаючи як початківців, так і досвідчених гравців, які цінують аркадну динаміку та класичну механіку платформерів. Завдяки простому управлінню й поступовому ускладненню рівнів, гра підходить для всіх вікових категорій.

Мотивація та реіграбельність

Мотивацією для проходження гри виступає бажання подолати нові рівні, уникати помилок, вдосконалювати свої навички та відкривати нові ігрові елементи. У перспективі гру можна доповнити системою досягнень, збереженнями прогресу, різними скінами персонажа — що забезпечить повторний інтерес до проходження гри.

Опис персонажів, локацій та об’єктів

У будь-якій комп’ютерній грі важливу роль відіграють персонажі, локації та ігрові об’єкти. Вони формують візуальний стиль гри, створюють атмосферу та

впливають на геймплей. Їхній ретельний дизайн забезпечує гравцеві приємне та захопливе ігрове враження.

Головний персонаж

Основним ігровим персонажем є невеликий герой, зовнішній вигляд якого виконано у стилі піксель-арт або мультяшної графіки. Він має просту анімацію: рух праворуч/ліворуч, стрибок, зупинка. У майбутньому до персонажа можуть бути додані нові дії — наприклад, атака, присідання або використання предметів.

Особливості персонажа:

- рух у двох напрямках (вліво, вправо);
- стрибки на різну висоту в залежності від тривалості натискання клавіші;
- зіткнення з ворогами призводить до втрати життя або завершення гри;
- взаємодія з колекторними об’єктами (монетами, бонусами).

Противники

На поточному етапі розробки в грі передбачено базових ворогів. Їхні дії обмежені патрулюванням обмеженої зони або простим переміщенням по платформі. При зіткненні з головним героєм вони завдають шкоди або вбивають його.

Можливості для розширення:

- додавання різних типів ворогів (літаючі, стрибучі, стріляючі);
- введення складніших алгоритмів поведінки (погоня, уникання, засідка).

Локації

Ігровий світ складається з окремих рівнів, виконаних у стилі класичного платформера. Кожна локація має унікальне розташування платформ, перешкод, ворогів і об’єктів для збирання. Візуально рівні оформлені в єдиному стилі, але з можливістю варіацій — наприклад, рівень у лісі, в підземеллі або на пустелі.

Елементи локацій:

- перешкоди (ями, шипи, рухомі платформи);
- декоративні елементи (дерева, хмари, фони).

Ігрові об’єкти

У грі присутні кілька типів об’єктів, з якими може взаємодіяти гравець:

- Монети / артефакти — об’єкти, які гравець збирає під час проходження рівня. Вони можуть впливати на загальний рахунок або служити умовою для відкриття нового рівня.

- Життя / здоров’я — об’єкти, які поновлюють здоров’я героя (якщо така механіка реалізована).

- Фінішний портал / прапорець — точка завершення рівня. При досягненні цього об’єкта гра зберігає прогрес і переходить до наступного рівня.

Інтерфейсні об’єкти

Окрім об’єктів на рівнях, гра містить елементи інтерфейсу:

- індикатор життя;

- лічильник монет;

- меню паузи;

- кнопка «повторити» або «вийти».

Усі ці компоненти формують завершену ігрову сцену, забезпечуючи зручність і зрозумілість користувачеві.

1.4. Розробка сценарію та рівнів

Розробка сценарію та структури рівнів є ключовим етапом у створенні комп’ютерної гри. Від грамотно побудованого сценарію залежить глибина занурення гравця у гру, а від якісного дизайну рівнів — баланс між цікавістю та складністю проходження.

Сценарій гри

У запропонованій грі жанру платформер сценарій є лаконічним, проте достатнім для заданого типу проєкту. Гравець керує головним героєм, який мандрує через серію небезпечних рівнів з метою зібрати всі монети та досягти кінцевої точки. Кожен рівень — це окремий виклик, що поступово ускладнюється.

Головна сюжетна лінія (опціонально): Головний герой вирушає у подорож, щоб знайти втрачені артефакти свого народу. На шляху йому доведеться подолати різноманітні перешкоди, ворогів і пастки.

Хоча гра не має складної історії чи діалогів, навіть проста мета надає гравцеві мотивацію проходити рівні й досягати успіху. За бажанням у наступних версіях гри сценарій може бути розширений за допомогою кат-сцен, текстових повідомлень, внутрішньоігрових діалогів.

Проєктування рівнів гри

Гра складається з декількох послідовних рівнів. Кожен рівень являє собою окрему карту з унікальним розміщенням платформ, перешкод і ворогів. З кожним новим рівнем зростає складність — з’являються нові типи пасток, ворогів або складніші секції для стрибків.

Основні принципи побудови рівнів:

- Прогресія складності — на першому рівні гравець ознайомлюється з базовими механіками (рух, стрибок, збір монет), а на наступних — з новими викликами;

- Збалансованість — гравець має відчувати виклик, але не фрустрацію;

- Нагородження дослідження — розміщення секретних ділянок із бонусами або додатковими монетами;

- Ігрова логіка — використання рухомих платформ, ворогів, механізмів, які треба активувати тощо.

Інструменти для створення рівнів

Unity дозволяє зручно створювати рівні за допомогою компонента Tilemap, який спрощує побудову двовимірного світу. Кожен рівень проєктується в окремій сцені Unity, де розміщуються всі потрібні об’єкти: платформи, колектори, вороги, точки старту та завершення.

Також можливо створити рівні вручну або за допомогою зовнішніх редакторів карт (наприклад, Tiled), які потім імпортуються в проєкт.

Можливості для розширення:

- створення тематичних рівнів (пустеля, ліс, печера);

- додавання нових механік (рухомі блоки, важелі, зникаючі платформи);

- генерація рівнів процедурним способом;

- система збереження прогресу.

Таким чином, сценарій та рівні гри утворюють логічно цілісну структуру, що дозволяє гравцеві поступово занурюватись у світ гри, розвиваючи навички керування персонажем і відкриваючи нові можливості ігрового процесу.

2. ТЕХНОЛОГІЧНА РЕАЛІЗАЦІЯ ПРОЄКТУ

2.1. Створення ігрового дизайну (UI/UX-дизайн)

Ігровий дизайн інтерфейсу користувача (UI — User Interface) та досвіду користувача (UX — User Experience) є важливою складовою загального враження від гри. Саме ці елементи забезпечують комфортне, інтуїтивно зрозуміле управління та взаємодію користувача з ігровим середовищем. У платформері, що був реалізований у даній роботі, було враховано базові принципи UI/UX-дизайну, які дозволяють зробити гру доступною для широкої аудиторії гравців.

Інтерфейс користувача (UI)

Інтерфейс гри розроблено таким чином, щоб бути мінімалістичним, але інформативним. Основні елементи UI включають:

- Лічильник монет — розташований у верхній частині екрану, він показує кількість зібраних монет у поточному рівні. Це допомагає гравцю орієнтуватися у власному прогресі;

- Головне меню — просте і зрозуміле, з кнопками «Почати гру», «Продовжити», «Налаштування» та «Вихід»;

- Екран завершення рівня — після досягнення фінішу, гравець бачить результат: кількість зібраних монет, час проходження і кнопку для переходу до наступного рівня;

- Кнопка паузи — дозволяє поставити гру на паузу та отримати доступ до меню.

Дизайн інтерфейсу був виконаний у простому стилі, відповідному до загального візуального оформлення гри. Використовувались зрозумілі іконки, великі кнопки для зручного натискання та контрастні кольори, щоб важливі елементи були добре видимими на фоні гри.

Досвід користувача (UX)

UX-дизайн зосереджувався на забезпеченні логічного та інтуїтивного взаємодії гравця з грою. Основні UX-принципи, реалізовані в проєкті:

- Простота керування — гравець використовує базові клавіші (наприклад, стрілки або WASD для руху, пробіл для стрибка), що не вимагають складного навчання;

- Швидкий зворотний зв'язок — при натисканні кнопок, зборі монет або зіткненні з ворогами гравець одразу отримує візуальну й аудіоіндикацію;

- Послідовне зростання складності — перші рівні виконують функцію «навчального режиму», де гравець знайомиться з базовими механіками без надмірного навантаження;

- Відсутність перевантаження — на екрані відображається лише найнеобхідніша інформація, щоб не відволікати користувача.

Інструменти реалізації

Для реалізації UI/UX-дизайну у Unity використовувалася система Canvas, яка дозволяє створювати графічні елементи інтерфейсу — текстові поля, кнопки, зображення. Кожен елемент був налаштований у відповідності до роздільної здатності екрана, що дозволяє зберігати коректне відображення інтерфейсу на різних пристроях.

Також застосовувалися стандартні компоненти Unity: Button, Text (TMP), Image, а також EventSystem для обробки натискань.

Плани на розвиток дизайну

У майбутньому інтерфейс може бути доповнений:

- анімаціями при переходах між екранами;
- меню з вибором рівня;
- підказками для новачків;
- системою досягнень;
- адаптацією для мобільних пристроїв.

2.2. Створення графічних елементів: спрайти, анімації, фони

Графічне оформлення відіграє ключову роль у створенні привабливої та зрозумілої гри. У процесі розробки гри на платформі Unity було створено набір візуальних елементів, які включають спрайти персонажів, ворогів, об’єктів

середовища, а також анімовані ефекти та фонові зображення. Головною метою було створити яскравий, стилістично цілісний візуальний стиль, який би відповідав жанру 2D-платформера.

Спрайти персонажів і об’єктів

Спрайти — це зображення, що використовуються для візуального представлення об'єктів у грі. У цьому проєкті спрайти були або взяті з відкритих джерел (free asset packs), або створені вручну в графічних редакторах, таких як Photoshop або Aseprite. Основні спрайти включали: головного персонажа, ворогів, монети, платформи, фінішні прапори, пастки тощо. Усі графічні елементи зберігалися у форматі PNG для збереження прозорого фону, що є стандартом у 2D-розробці.



Рис. 2.1. Імпорт 2D-спрайтів у Unity через вікно Project

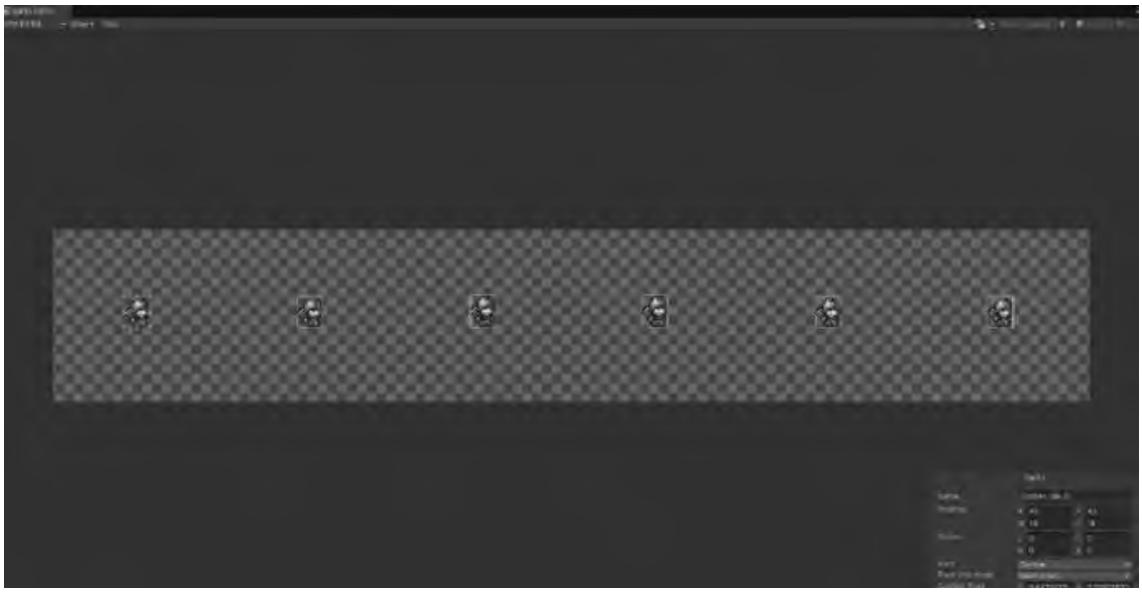


Рис. 2.2. Sprite Editor – поділ спрайт-листа на окремі кадри

Анімації. Анімація в Unity реалізується через систему Animator. Для кожного активного об'єкта було створено анімаційний контролер. Наприклад, для гравця було реалізовано такі стани: idle (стояння), run (біг), jump (стрибок), fall (падіння) та death (смерть). Ці анімації створювались з кількох послідовних спрайтів — фреймів, які циклічно змінюються в Animator.

Анімації керуються тригерами та параметрами, що задаються у скриптах на C#. Наприклад, зміна анімації з "біг" на "стрибок" відбувається при натисканні відповідної клавіші та за умови, що персонаж знаходиться на землі.

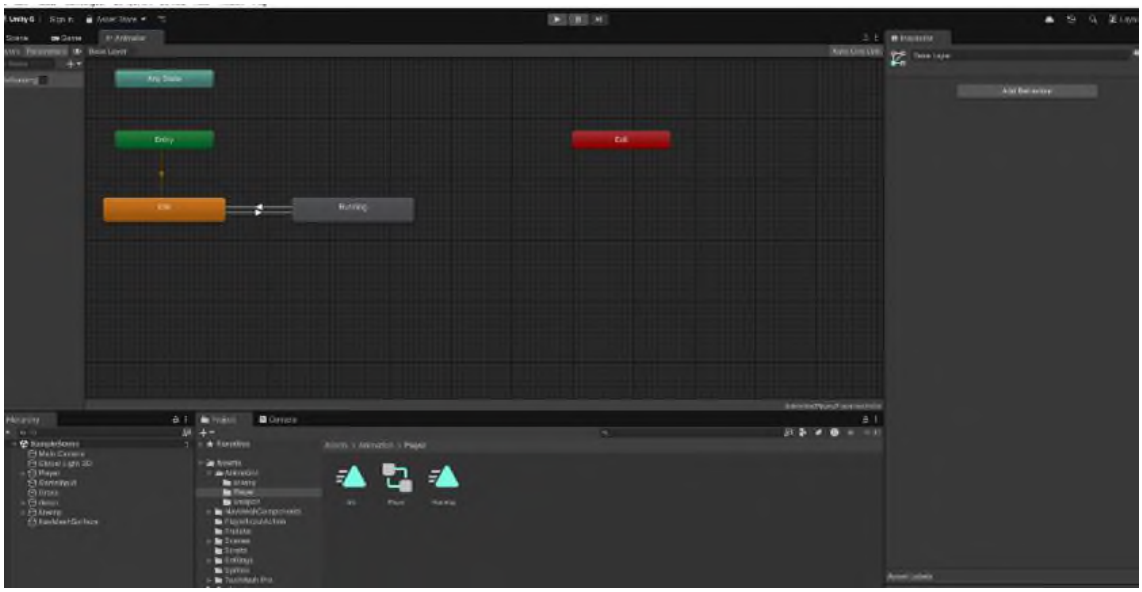


Рис.2.3. Анімація руху гравця у вікні Animation (Animator Controller)

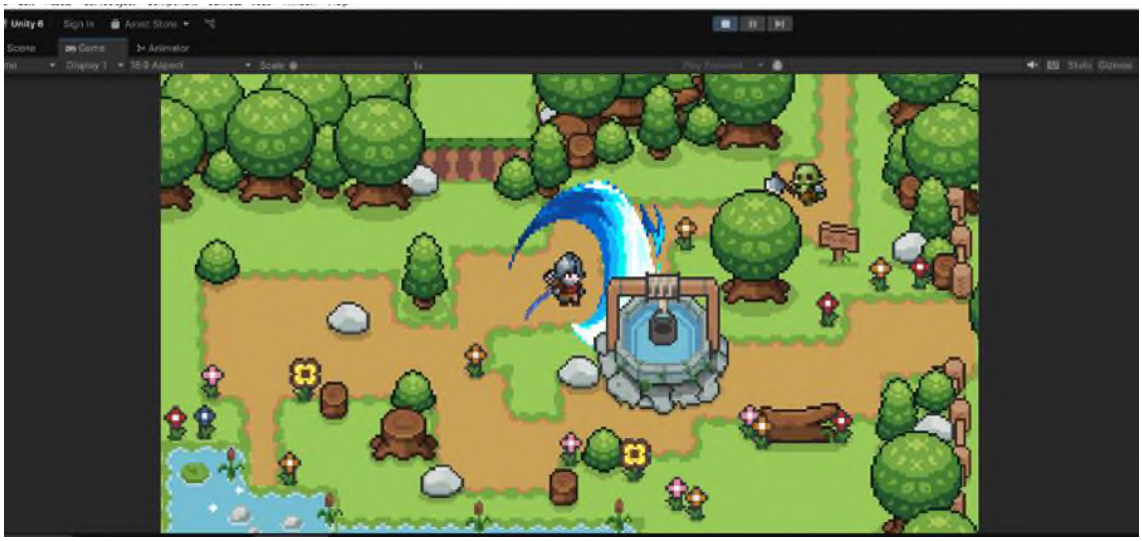


Рис. 2.4. Візуалізація головного персонажа на сцені
із застосованою анімацією

Фони та паралакс-ефект

Фонові зображення задають загальний настрій гри. Для кожного рівня були підібрані фонові картинки, які імітують денний пейзаж. Додатково було реалізовано паралакс-ефект — коли фонові шари рухаються повільніше за передній план, що створює ілюзію глибини. Цей ефект досягається завдяки різній швидкості руху шарів фону щодо позиції камери.

Отже, можна відзначити, що графіка в геймплеї є дуже важливою. Візуальні елементи не лише покращують зовнішній вигляд, але й допомагають гравцеві орієнтуватися в грі. Наприклад, відмінний колір ворогів сигналізує про небезпеку, тоді як блискучі монети привертають увагу. Таким чином, графіка виконує не тільки естетичну, а й функціональну роль у дизайні рівнів.

Технічні аспекти реалізації

Усі спрайти були імпортовані в Unity з відповідними налаштуваннями: тип зображення — Sprite (2D and UI), Pixel Per Unit встановлено відповідно до роздільної здатності спрайта, а Sprite Mode — Multiple для спрайт-листів. Для анімацій використовувався Animation Window, де створювалися кліпи, які потім об’єднувалися в Animator Controller з переходами між станами.

Рисунок 3.3: Код взаємодії з об’єктами (наприклад, підбір предметів).

2.3. Структура проєкту в середовищі Unity

Організація структури проєкту в Unity має велике значення для ефективної розробки, тестування та масштабування гри. Чітка структура полегшує навігацію по файлах, роботу в команді та забезпечує логічне розділення ресурсів. У даному проєкті, реалізованому в середовищі Unity, структура була сформована відповідно до кращих практик та порад з навчальних матеріалів.

Головні директорії проєкту

Усі елементи проєкту зберігаються в папці Assets, яка є основною робочою директорією Unity. Всередині неї були створені окремі папки для поділу файлів за типом:

- Animations — містить усі анімаційні кліпи та Animator Controller'и, які керують станами персонажів та об'єктів;

- Audio — зберігає аудіофайли, такі як звуки стрибків, збору монет, фонова музика тощо;

- Fonts — використані шрифти для елементів UI;

- Materials — матеріали, які можуть бути застосовані до об'єктів, наприклад, для ефектів;

- Prefabs — шаблони об'єктів, які часто використовуються в грі: платформи, вороги, монети, персонажі. Prefab дозволяє створити копії об'єкта з однаковими параметрами.

- Scenes — усі ігрові рівні, збережені як сцени (файли з розширенням .unity). Наприклад, MenuScene, Level1, GameOverScene.

- Scripts — усі скрипти на C#, які реалізують логіку гри: управління персонажем, обробка зіткнень, підрахунок очок тощо.

- Sprites — графічні елементи: спрайти персонажів, ворогів, об’єктів середовища, фони тощо.

- UI — елементи інтерфейсу користувача: кнопки, панелі, текстові об’єкти.

Такий розподіл дозволяє швидко знаходити потрібні файли, не плутатися під час імпорту нових ресурсів, а також забезпечує гнучкість при зміні чи оновленні компонентів. Наприклад, щоб змінити зовнішній вигляд монети, достатньо відкрити папку Sprites і замінити відповідне PNG-зображення, після чого всі prefab-об’єкти автоматично оновляться.

Крім Assets, Unity автоматично створює папки Library, Logs, Packages та ProjectSettings, які зберігають службову інформацію про конфігурацію проєкту, версію рушія, кеш, імпортовані пакунки тощо. Ручного втручання в ці папки зазвичай не потрібно, але вони є важливими для цілісності проєкту.

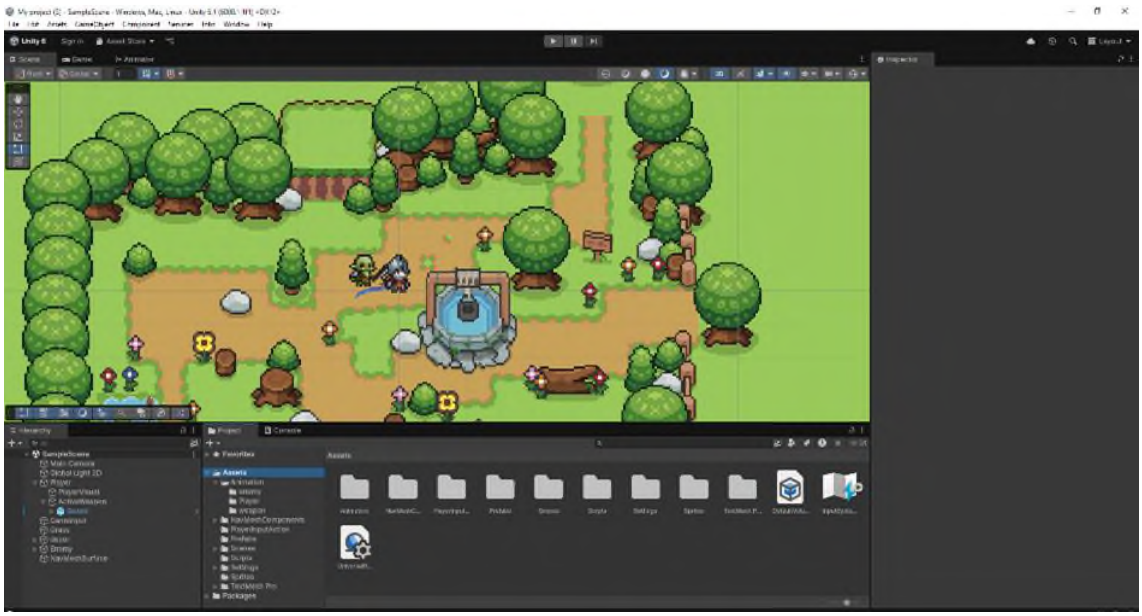


Рис. 2.5. Загальний вигляд сцени в Unity Editor



Рис. 2.6. Ієрархія об’єктів у сцені

Підтримка масштабованості проєкту

Організація структури з урахуванням принципу модульності дозволяє легко додавати нові сцени, об’єкти або функціонал, не порушуючи логіку проєкту. Це особливо корисно при роботі над більшими іграми або в командних проєктах, де кілька розробників працюють над різними аспектами.

2.4. Реалізація основних механік гри

Реалізація основних механік гри є фундаментальним етапом у процесі створення комп’ютерної гри. Саме ігрові механіки визначають, як гравець взаємодіє з віртуальним світом, які можливості він має, як розвивається сюжет та який досвід отримує користувач під час гри. Завдяки цьому компоненту формується не лише технічна структура гри, а й її емоційне наповнення та ігрова логіка. У рамках цього етапу відбувається впровадження руху персонажа, взаємодії з об’єктами, системи здоров’я, атак, збору предметів, інтерфейсу користувача, а також поведінки ворогів і навколишнього середовища.

Першочергово реалізовується управління головним персонажем. Гравець повинен мати змогу керувати героєм — переміщуватись у межах ігрової сцени, стрибати, уникати небезпеки або долати перешкоди. Важливо забезпечити плавність руху та швидку реакцію на команди. Це створює відчуття контролю та занурення у гру. Також налаштовується камера, яка динамічно слідкує за рухами гравця, не втрачаючи його з поля зору, що є важливою умовою комфортного проходження.

Окрему увагу приділено взаємодії персонажа з навколишніми об’єктами. Це можуть бути як предмети, що підбираються, так і механізми, які активуються або відкривають нові ділянки рівня. Така взаємодія має бути зрозумілою та інтуїтивною. Вона стимулює дослідницький інтерес гравця та сприяє поступовому розкриттю потенціалу гри.

Також реалізується система здоров’я персонажа, яка відображає його стан у грі. Вона дозволяє визначити, коли гравець зазнає пошкодження, втрачає життя або гине. Це важлива механіка, яка створює виклик та змушує бути обережним.

```
public class Player : MonoBehaviour
{
    [SerializeField] private float movingspeed = 10f;
    public static Player Instanse { get; private set; }
    private Rigidbody2D rb;
    private float minMovingSpeed = 0.1f;
    private bool isRunning = false;

  private void Awake()
    {
        Instanse = this;
        rb = GetComponent<Rigidbody2D>();
    }
    //private void Update()
    //{
    //}
    private void Start()
    {
        Gameinput.instance.OnPlayerAttack += GameInput_OnPlayerAttack;
    }

  private void GameInput_OnPlayerAttack(object sender, System.EventArgs e)
    {
      ActiveWeapon.Instance.GetActiveWeapon().Attack();
    }

 private void FixedUpdate() {
        HandleMovement();
    }
```

```
        private void HandleMovement()
        {
            Vector2 inputVector = Gameinput.instance.GetMovementVector();
            inputVector = inputVector.normalized;
            rb.MovePosition(rb.position + inputVector * movingspeed * Time.fixedDeltaTime);
            if (Mathf.Abs(inputVector.x) > minMovingSpeed || Mathf.Abs(inputVector.y) > minMovingSpeed)
            {
                isRunning = true;
            } else
            {
                isRunning = false;
            }
        }
            public bool IsRunning()
            {
                return isRunning;
            }

       public Vector3 GetPlayerScreenPosition()
        {
            Vector3 playerScreenPosition = Camera.main.WorldToScreenPoint(transform.position);
            return playerScreenPosition;
        }
        }
```

Поряд із цим, у грі можуть з’являтися вороги, з якими гравець змушений боротися або уникати їх. Поведінка ворогів моделюється так, щоб створювати напругу та підвищувати динаміку гри.

```
using System.Runtime.CompilerServices;
using Unity.VisualScripting;
using UnityEngine;
using UnityEngine.AI;
using Diplom.Utils;
public class EnemyAI : MonoBehaviour
{
    [SerializeField] private State startingState;
    [SerializeField] private float roamingDistanceMax = 7f;
    [SerializeField] private float roamingDistanceMin = 3f;
    [SerializeField] private float roamingTimerMax = 2f;
    private float roamingTime;
    private Vector3 roamPosition;
    private Vector3 startingPosition;
    private NavMeshAgent navMeshAgent;
    private State state;

    private enum State
    {
        Roaming
    }

    //private void Start()
    //{
    //    startingPosition = transform.position;
    //}
```

```
    private void Awake()
    {
        navMeshAgent = GetComponent<NavMeshAgent>();
        navMeshAgent.updateRotation = false;
        navMeshAgent.updateUpAxis = false;
        state = startingState;
    }

    private void Update()
    {
        switch (state) {
            default:
            case State.Roaming:
                roamingTime -= Time.deltaTime;
                if (roamingTime < 0)
                {
                    Roaming();
                    roamingTime = roamingTimerMax;
                }
                break;
        }   }

    private void Roaming()
    {
        startingPosition = transform.position;
        roamPosition = GetRoamingPosition();
        ChangeFacingDirection(startingPosition, roamPosition);
        navMeshAgent.SetDestination(roamPosition);
    }
```

```
        private Vector3 GetRoamingPosition()
        {
            return startingPosition + Utils.GetRandomDir() * UnityEngine.Random.Range(roamingDistanceMin, roamingDistanceMax);
        }

        private void ChangeFacingDirection(Vector3 soursePosition, Vector3 targetPosition)
        {
            if (soursePosition.x > targetPosition.x)
            {
                transform.rotation = Quaternion.Euler(0, -180, 0);
            }
            else
            {
                transform.rotation = Quaternion.Euler(0, 0, 0);
            }
        }
    }
```

Ще одним важливим елементом є реалізація ігрового інтерфейсу. Він надає гравцеві інформацію про поточний стан справ — кількість зібраних предметів, рівень здоров’я, набрані очки тощо. Завдяки цьому гравець орієнтується у прогресі та розуміє наслідки своїх дій. Меню гри та системи управління, такі як пауза або збереження, дозволяють контролювати ігровий процес у зручний спосіб.

Таким чином, реалізація основних механік гри вимагає цілісного підходу, де враховуються як технічні аспекти, так і враження гравця. Всі елементи повинні бути узгодженими, гармонійно взаємодіяти між собою, створюючи

захопливий ігровий досвід. Саме ці механіки формують основу геймплею, від якої залежить якість, популярність та успішність фінального продукту.

Обробка подій, зіткнень та взаємодії з об’єктами

Одним із важливих аспектів у розробці комп’ютерних ігор є створення механізмів для взаємодії об’єктів у світі гри, які включають обробку подій, зіткнень та різноманітних фізичних взаємодій. У сучасних ігрових движках, таких як Unity, ці процеси реалізовані через складні системи фізики та управління подіями, що дозволяють створювати реалістичні та інтерактивні ігрові світи.

У контексті платформи Unity, кожен об'єкт, який існує в сцені, може мати взаємодії з іншими об’єктами або навколишнім середовищем. Ці взаємодії можуть бути як фізичними, наприклад, коли два об’єкти стикаються, так і логічними, коли один об’єкт активує або змінює інший за допомогою подій, наприклад, кнопок чи тригерів. Для досягнення такого ефекту у Unity використовуються коллайдери, скрипти та інтерфейси для обробки подій.



Рис. 2.7. Компоненти об’єкта (панель Inspector) — приклад: гравець або ворог

Обробка подій

Події у Unity — це зміни в стані об’єкта або системи, на які гра може реагувати. Це можуть бути події, ініційовані гравцем, такі як натискання кнопок або рух миші, або зміни, що відбуваються всередині гри, наприклад, коли об'єкт досягає певної точки або змінює стан. Події можуть бути оброблені через різні механізми, найпоширенішим з яких є використання методів, що реагують на ці зміни.

Наприклад, для обробки подій введення в Unity використовуються класи, такі як Input, що дозволяє зчитувати дані з клавіатури, миші або сенсорних екранів. Це дозволяє обробляти дії гравця в реальному часі, реагувати на

натискання кнопок або рухи миші. Система обробки подій в Unity дозволяє організувати зручне та швидке реагування на дії користувача, забезпечуючи інтерактивність гри.

Для зручності роботи з подіями також використовуються делегати та події, що дають змогу відправляти повідомлення різним об'єктам гри. Наприклад, коли один об'єкт активує певну дію (наприклад, натискання на кнопку), він може відправити подію, яку інші об'єкти прослуховують і відповідно реагують на неї. Такий підхід дозволяє забезпечити чітку організацію взаємодії різних елементів гри.

Обробка зіткнень

Зіткнення є важливою частиною фізики ігор. В Unity, для управління зіткненнями між об’єктами, використовуються спеціальні компоненти — коллайдери. Коллайдери є невидимими елементами, які визначають межі об’єкта для фізичних взаємодій. Важливою особливістю Unity є те, що обробка зіткнень відбувається за допомогою фізичних матеріалів і фізичних двигунів, що дає змогу досягти реалістичних результатів.

Unity використовує фізичний движок, який автоматично обробляє зіткнення об'єктів з різними коллайдерами. Однак для розширення функціональності можна використовувати скрипти для додаткової обробки реакцій при зіткненнях. Наприклад, при зіткненні двох об'єктів можна визначити силу пошкодження, спричинену зіткненням, або змінити траєкторію об’єкта.

Взаємодія з об’єктами

Взаємодія між об'єктами є ще одним важливим аспектом ігрової механіки, який може охоплювати різні елементи: від активування механізмів до збору предметів або виконання спеціальних дій. Наприклад, гравець може взаємодіяти з іншими об'єктами гри, натискаючи на кнопки, підбираючи предмети або взаємодіючи з персонажами.

У Unity для таких взаємодій використовуються тригери, які дозволяють виконувати певні дії, коли об'єкт потрапляє в зону взаємодії. Тригери можуть бути налаштовані так, щоб не викликати фізичних зіткнень, але спрацьовувати

для запуску певної логіки, наприклад, коли гравець підходить до дверей і вони автоматично відкриваються. Це забезпечує гнучкість у створенні ігрових механік.

Крім того, важливою складовою є фізична взаємодія об'єктів, яка відбувається через різні рухи або застосування сил. Unity дозволяє створювати реалістичні реакції при взаємодії об'єктів через систему фізики, де кожен об'єкт може бути оснащений масою, швидкістю і моментом інерції, що дозволяє змоделювати реалістичні рухи.

Налаштування анімацій і фізики

Анімація та фізика відіграють важливу роль у створенні динамічності, реалістичності та привабливості геймплею. У грі, розробленій на базі рушія Unity, реалізація цих компонентів відбувалася з використанням вбудованих засобів — Animator, Rigidbody2D, Collider2D та інших елементів.

Для персонажа гравця та деяких об’єктів було створено кілька основних анімаційних станів: Idle (спокій), Run (біг), Jump (стрибок), Fall (падіння), Death (смерть). Анімації формувалися на основі послідовності спрайтів (Sprite Sheet), імпортованих у Unity. Ці зображення були розділені на окремі кадри за допомогою Sprite Editor.

Після цього за допомогою інструменту Animator було створено контролер анімацій (Animator Controller), який дозволяє керувати переходами між станами. Для кожного стану було створено правило переходу (Transition), наприклад, при натисканні клавіші руху вліво або вправо персонаж переходить з Idle у Run.

Логіка зміни анімацій реалізується через скрипт, який опрацьовує введення гравця та встановлює відповідні параметри в Animator, наприклад:

```
csharp
КопироватьРедактировать
animator.SetBool("isRunning", true);
animator.SetBool("isJumping", false);
```

Таким чином, гра реагує на дії гравця, забезпечуючи плавність і злагодженість рухів персонажа.

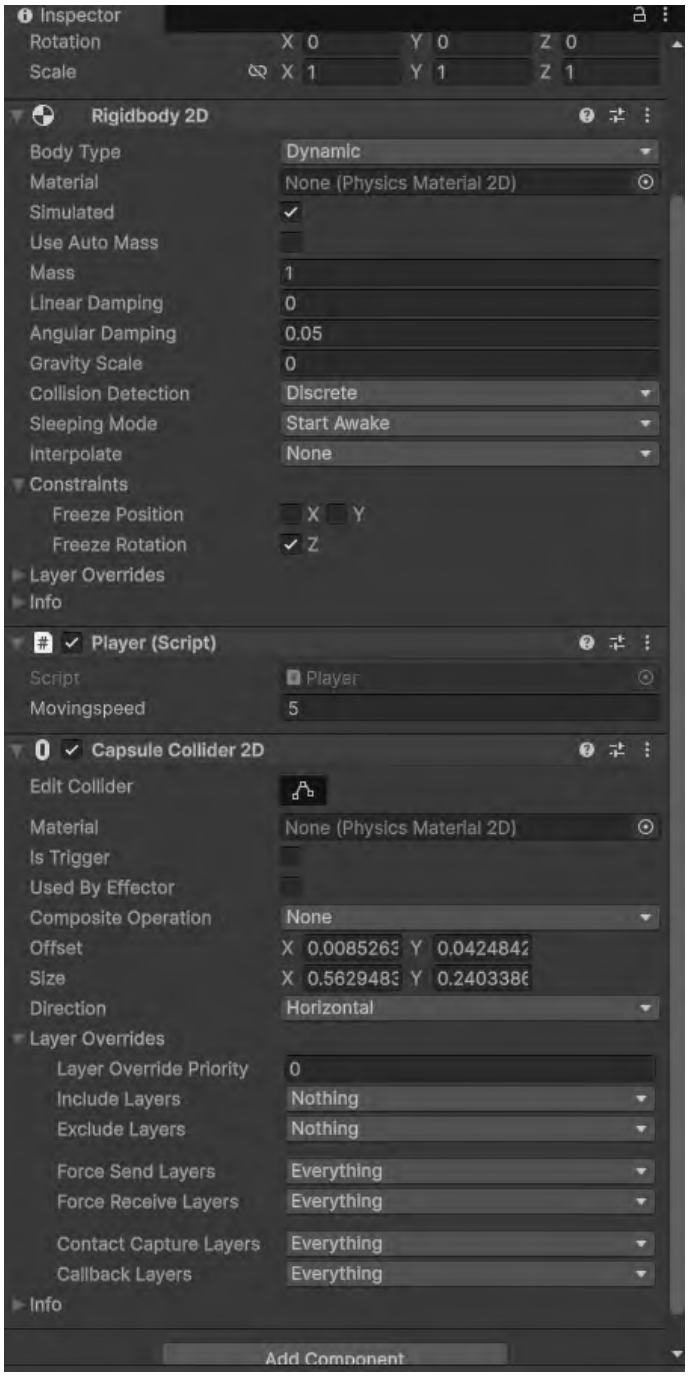


Рис. 2.8. Компоненти об’єкта (панель Inspector) — Налаштування Capsule Collider 2D та RigidBody гравця



Рис. 2.9. Використання Capsule Collider, гравець захований за дерево

Фізика у 2D-середовищі

Unity надає зручний інструментарій для симуляції фізичних процесів у 2D-іграх. Для реалізації фізики було використано компоненти Rigidbody2D — для надання об’єктам маси, сили гравітації та інерції, а також BoxCollider2D, CircleCollider2D або PolygonCollider2D — для визначення контурів зіткнень.

Головний персонаж має Rigidbody2D, що дозволяє йому підпорядковуватися законам фізики: стрибати, падати з висоти, реагувати на платформи. Сили до нього прикладаються через метод AddForce() або пряме редагування швидкості velocity. Наприклад, стрибок реалізується таким чином:

```
csharp
КопироватьРедактировать
rb.velocity = new Vector2(rb.velocity.x, jumpForce);
```

Щоб уникнути "слизького" ковзання чи небажаного відскоку, були налаштовані фізичні матеріали (Physics 2D Material), які контролюють рівень тертя та відскоку між об’єктами.

Взаємодія об’єктів через фізику

Колізії між об’єктами, наприклад, між персонажем та ворогом або монетою, обробляються через події OnCollisionEnter2D та OnTriggerEnter2D. Це дозволяє реалізувати такі механіки, як втрата життя при дотику до ворога або збір монет при їх торканні.

Оптимізація

Щоб уникнути надмірного навантаження на рушій, було обмежено використання зайвих компонентів фізики лише для об’єктів, що дійсно беруть участь у фізичній взаємодії. Наприклад, елементи фону або UI не потребують колайдерів або Rigidbody.

Отже, завдяки інструментам Unity налаштування анімацій та фізики відбувається швидко та інтуїтивно. Це дозволяє зосередитися на вдосконаленні геймплею, створюючи привабливий і реалістичний ігровий досвід навіть без складного програмування.

2.5. Тестування функціоналу гри

Тестування є важливим етапом розробки комп’ютерної гри, що дозволяє виявити й усунути помилки в логіці, анімації, управлінні, фізиці та взаємодії з інтерфейсом. У процесі створення гри на Unity було проведено декілька рівнів тестування: модульне, інтеграційне та ручне тестування всієї гри в цілому.

Модульне тестування

На першому етапі перевірялася коректність роботи окремих компонентів. Було протестовано логіку руху персонажа, стрибків, реакції на клавіші керування. За допомогою функцій Debug.Log() перевірялися внутрішні значення змінних у реальному часі (наприклад, швидкість руху або факт дотику до землі). Це дало змогу переконатися, що умови переходу між станами (наприклад, “біг” чи “стрибок”) відпрацьовують правильно.

Також перевірялися спрацьовування колайдерів: чи відбувається взаємодія з монетами, ворогами, платформами, чи належно працюють тригери OnTriggerEnter2D і OnCollisionEnter2D.

Інтеграційне тестування

Після перевірки окремих скриптів тестувалися взаємозв’язки між ними. Особлива увага приділялася збереженню станів персонажа, плавності переходів між рівнями, відображенню інтерфейсу, оновленню очок та життів. Наприклад, при зборі монети мало не лише зникати зображення монети, але й оновлюватися значення рахунку в UI. Важливим етапом стало тестування життєвого циклу гравця: втрата життя, респаун або завершення гри при зіткненні з ворогом. Також проводилась перевірка роботи кнопок у головному меню: «Грати», «Вийти», «Повторити» тощо.

Ручне тестування геймплею

Завершальний етап передбачав багаторазове проходження гри на різних рівнях складності. У процесі ручного тестування враховувалася зручність керування, логіка розміщення платформ, складність рівнів та зручність взаємодії з UI. Перевірялося, чи працюють усі анімації відповідно до дій гравця. Тестувалися також критичні моменти: чи не може персонаж застрягнути між платформами, чи не падає за межі рівня, чи коректно обробляються помилки (наприклад, втрата всіх життів).



Рис. 2.10. Гра в режимі Play Mode: гравець, об’єкти та логіка руху

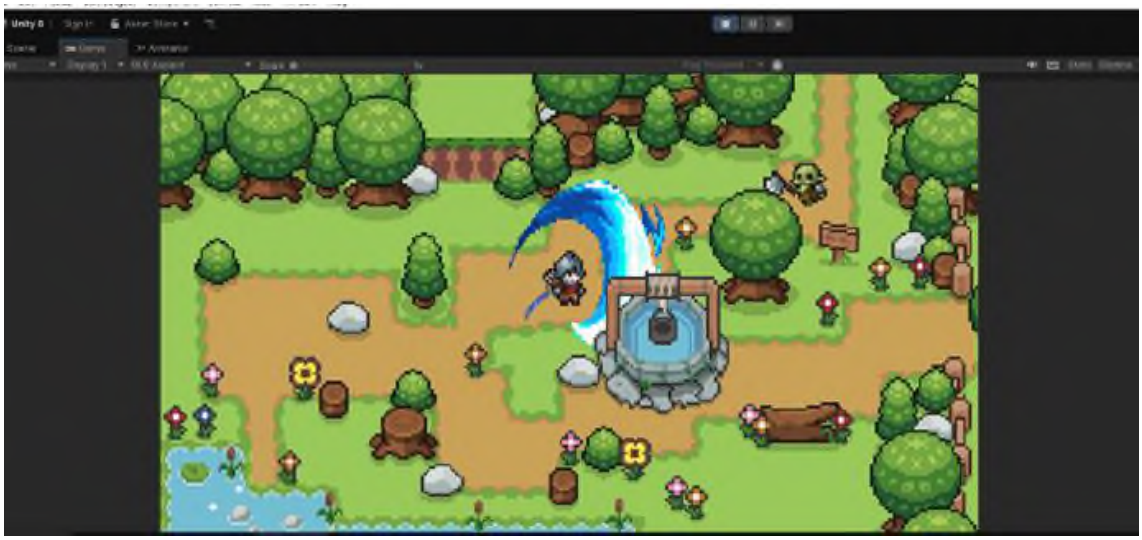


Рис. 2.11. Скриншот ігрового процесу з реалізованою анімацією

Інструменти, використані для тестування

Усі тести проводилися в редакторі Unity у режимі Play. Для виведення діагностичної інформації застосовувалися вікна Console та Inspector. У деяких випадках використовувалися Gizmos для візуалізації колайдерів та зон взаємодії.

3. ОЦІНКА ЕФЕКТИВНОСТІ ТА ПЕРСПЕКТИВИ РОЗВИТКУ

3.1. Верифікація результатів проєкту

Верифікація результатів — це процес підтвердження відповідності створеного продукту початковим цілям, технічним вимогам і очікуванням користувачів. У контексті даного проєкту, спрямованого на створення двовимірної комп'ютерної гри в середовищі Unity, верифікація дозволяє оцінити, наскільки успішно реалізовані функціональні можливості гри, її дизайн, стабільність роботи та загальна якість.

Порівняння з початковими цілями

На початку проєкту були поставлені такі цілі:

Створити функціональну гру в жанрі платформера з базовим сюжетом;

- Реалізувати базову механіку управління персонажем (рух, стрибки, взаємодія з об'єктами);

- Забезпечити наявність системи очок, життів та умов переходу між рівнями;

- Створити зручний користувацький інтерфейс (UI);

- Забезпечити коректну роботу гри без критичних помилок.

У процесі реалізації всі ці пункти були виконані. Гра дозволяє гравцю керувати персонажем, збирати предмети, уникати ворогів, а також переходити між етапами гри. Присутній інтерфейс з відображенням поточних очок і життів, меню, кнопки «Грати» та «Вийти». Отже, технічні цілі проєкту були досягнуті повністю.

Тестування відповідності функціоналу

Для верифікації було проведено серію ручних тестів у середовищі Unity, які підтвердили:

- Всі кнопки інтерфейсу працюють відповідно до очікувань;

- Управління персонажем інтуїтивно зрозуміле та відгукується на натискання клавіш без затримок;

- Всі ігрові механіки працюють стабільно (збирання монет, взаємодія з ворогами, переходи між рівнями);

- Гра коректно завершується у випадку втрати всіх життів або при досягненні фінального етапу.

Оцінка якості реалізації графіки та UI

Графічні елементи гри — спрайти, фони, анімації — відповідають обраному стилю гри та не перевантажують екран. Інтерфейс чітко відображає основні дані (очки, життя), кнопки розташовані зручно та читаються добре. Анімації плавні, без ривків, забезпечуючи приємний візуальний досвід.

Перевірка на відповідність технічним вимогам

Гра успішно компілюється для цільової платформи (Windows), не містить критичних багів, має мінімальні вимоги до системних ресурсів, що дозволяє її запуск на більшості сучасних комп'ютерів. Застосовані методи оптимізації, такі як обмеження кількості активних об'єктів на сцені та правильне використання компонентів Unity, сприяли стабільній роботі гри.

3.2. Можливості масштабування гри

Масштабування гри — це процес розширення її функціоналу, збільшення контенту, адаптації до різних платформ та вдосконалення механік без повного переписування початкового коду. Якісна структура проєкту, реалізованого в середовищі Unity, дозволяє легко здійснювати масштабування як у технічному, так і в ігровому плані.

Розширення контенту

Одним з найпростіших способів масштабування гри є додавання нових рівнів. Оскільки кожен рівень у грі реалізовано окремою сценою, можна безперешкодно створювати нові сцени з унікальними об'єктами, перешкодами, фонами та противниками. При цьому базова механіка персонажа, UI та логіка гри залишаються незмінними, що значно спрощує процес.

Також є можливість додавати нові типи ворогів із різною поведінкою, бонуси, перешкоди, а також спеціальні об'єкти, які вносять різноманітність у геймплей (наприклад, портали, пастки, тимчасові посилення).

Покращення геймплею

Гра може бути масштабована за рахунок впровадження складніших ігрових механік, зокрема:

- системи збереження прогресу;
- діалогів або сюжетних вставок;
- вибору персонажів з різними характеристиками;
- квестових завдань чи місій;
- багатокористувацького режиму (локального або онлайн).

Ці покращення не потребують повного переписування наявної структури, а лише інтегруються в існуючий код.

Графічне та аудіо масштабування

Оскільки Unity підтримує імпорт якісних спрайтів, 3D-моделей і звукових файлів, гру можна масштабувати візуально: покращити фони, створити нові анімації для персонажів, урізноманітнити звуковий супровід та фонову музику. Це значно покращить загальне враження користувача та зробить проєкт більш конкурентоспроможним.

Кросплатформенність

Однією з основних переваг Unity є підтримка багатьох платформ. Створену гру можна з мінімальними змінами експортувати для:

- Windows, macOS, Linux;
- Android, iOS;
- WebGL (браузерна версія гри);
- ігрових консолей (за відповідної ліцензії).

Це відкриває великі можливості для розширення цільової аудиторії.

Підключення сторонніх сервісів

Для масштабування гри можна інтегрувати зовнішні сервіси: аналітику (наприклад, Unity Analytics або Firebase), хмарне збереження, рекламу або мікротранзакції. Це робить гру придатною для комерційного використання або розміщення на платформах, таких як Google Play чи App Store.

3.3. Рекомендації щодо публікації гри на платформа

Публікація комп’ютерної гри є важливим завершальним етапом розробки, що дозволяє донести продукт до гравців, отримати зворотний зв’язок та потенційно монетизувати проєкт. Платформа Unity надає широкі можливості для експорту гри на різноманітні платформи, зокрема ПК, мобільні пристрої, веб-браузери та навіть ігрові консолі. Для успішної публікації необхідно дотримуватись технічних вимог обраної платформи, а також враховувати маркетингові аспекти.

Вибір платформи публікації

Залежно від цільової аудиторії, технічних характеристик гри та бюджету, можна обрати одну або декілька платформ:

- ПК (Windows, Linux, macOS) — ідеально підходить для перших тестових релізів. Платформи на зразок itch.io, Game Jolt або Steam дозволяють швидко опублікувати гру, часто без серйозної модерації;

- Мобільні пристрої (Android, iOS) — потребують детальнішої підготовки, зокрема оптимізації під різні екрани, налаштування сенсорного керування та відповідності політикам Google Play або App Store;

- WebGL (браузерна версія гри) — зручна для швидкого доступу через сайт або платформи, такі як itch.io, без потреби завантаження гри.

Підготовка до публікації

Перед публікацією слід перевірити:

- Оптимізацію продуктивності — гра має працювати без зависань і помилок на більшості пристроїв;

Коректність UI/UX — інтерфейс повинен бути зручним, адаптованим до різних роздільностей екранів;

Відсутність багів — необхідно протестувати всі рівні, анімації, взаємодії з об’єктами.

Також потрібно створити іконку гри, знімки екрана, трейлер (за можливості) та оформити опис гри, який буде розміщений на сторінці магазину.

Особливості публікації на Google Play (Android):

- Зареєструвати Google Play Console (разовий платіж — 25$);

- Створити APK або AAB файл гри в Unity;

- Завантажити гру, заповнити інформацію про неї: опис, жанр, скріншоти, значок;

- Пройти модерацію (триває від кількох годин до декількох днів);

- Отримати гру в публічному доступі з можливістю оновлень.

Публікація на itch.io (WebGL або ПК):

- Зареєструвати акаунт на сайті;

- Створити сторінку гри, додати опис, логотип, скріншоти;

- Завантажити файл гри (наприклад, WebGL-білд або .zip-архів для Windows);

- Налаштувати доступ: відкритий, прихований або за запрошенням.

Маркетингові поради:

- Просування через соціальні мережі, тематичні форуми та YouTube;

- Збір відгуків для покращення наступних версій гри;

- Регулярні оновлення після публікації підвищують інтерес та підтримують активність гравців.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи на тему «Розробка комп’ютерної гри на платформі Unity» було проведено повноцінний цикл розробки ігрового проєкту, що охоплює аналіз сучасних інструментів, проєктування структури гри, створення ігрових механік, графічного контенту, інтерфейсу користувача, тестування, а також підготовку рекомендацій щодо розширення і публікації гри.

Розробка гри базувалася на рушії Unity — одному з найпопулярніших середовищ для створення 2D- та 3D-ігор. У процесі дослідження було проаналізовано основні можливості цієї платформи, зокрема її модульність, потужний компонентний підхід, розвинену систему анімацій, фізики та зручну інтеграцію з мовою програмування C#. Unity показав себе як зручний інструмент не лише для досвідчених розробників, а й для початківців, що дозволило ефективно реалізувати ігрову ідею з мінімальними ресурсами.

У рамках роботи було створено гру з базовим сюжетом, ігровими рівнями, інтерактивними об’єктами, персонажами, системою зіткнень, анімацією, UI-елементами та екраном завершення гри. Графічна частина була реалізована за допомогою власноруч створених та адаптованих спрайтів, використано анімації для живості геймплею, а також дотримано сучасних принципів UI/UX дизайну для забезпечення зручності користувача.

Особливу увагу приділено структурі проєкту, організації сцен та підключенню скриптів, що контролюють логіку гри. В процесі розробки використовувались різні аспекти роботи з Unity, такі як Physics2D, Animator, AudioSource, Tilemap, Canvas, EventSystem тощо. Створено кілька ігрових рівнів, сценарії взаємодії з NPC та об’єктами, а також реалізовано завершення гри за умовами виграшу або поразки.

Також було проведено базове тестування функціональності гри, що дозволило усунути помилки логіки та переконатися у стабільності проєкту. На завершення було проаналізовано можливості масштабування: додавання нових

рівнів, складніших механік, підтримки мобільних платформ, збереження прогресу, та покращення графічної частини.

В результаті виконання поставлених завдань сформоване повне уявлення про процес створення 2D-ігор, починаючи від вибору рушія до завершальної підготовки гри до релізу. Отримані знання та практичний досвід можуть бути використані в подальших проєктах або вдосконаленні створеної гри.

Таким чином, реалізація цього проєкту засвідчила доцільність використання Unity для створення повноцінних навчальних та комерційних ігор, а також сприяла розвитку професійних навичок у сфері ігрової розробки, зокрема: програмування, дизайну, проєктування інтерфейсу, тестування та управління проєктом. Кваліфікаційна робота також демонструє, як навіть базові інструменти можуть бути використані для створення цікавої, функціональної та технічно стабільної гри, що має потенціал для подальшого розвитку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Brackeys: How to make a 2D Game in Unity. URL: https://www.youtube.com/playlist?list=PLPV2KyIb3jR4u5jX8za5iU1eF4k4Br3d4

2. C# Language Documentation. URL: https://learn.microsoft.com/en-us/dotnet/csharp/

3. Create 2D and 3D games in Unity. URL: https://unity.com/how-to/difference-between-2D-and-3Dgames.

4. David B. Hands-On Game Development Patterns with Unity. Unity Technologies, 2019. 116 с.

5. GameDev.net — Unity Game Programming Resources. URL: https://gamedev.net/

6. Github. Unity Samples and Open Source Projects. URL: https://github.com/Unity-Technologies

7. IGDA (International Game Developers Association). URL: https://igda.org/

8. Jason Weimann. Unity Game Dev Tutorials [YouTube Channel].URL: https://www.youtube.com/channel/UCX_b3NNQN5bzExm-22-NVVg/videos

9. Lanzinger, F. (2020). 2D Game Development with Unity (1st ed.). CRC Press. https://doi.org/10.1201/9780429328664

10. Packt Publishing: Mastering Unity 2D Game Development — 2021. URL: https://salo.li/2DDB8b5

11. Pluralsight: Unity 2D Fundamentals. URL: https://www.pluralsight.com/courses/unity-2d-fundamentals-character-interactions

12. Prometheus: Основи створення комп’ютерних ігор. URL: https://prometheus.org.ua

13. UI/UX дизайн. URL: https://abitap.com/category/ui-ux-dyzajn/

14. Unity (ігровий рушій) — Вікіпедія. URL: https://uk.wikipedia.org/wiki/Unity_(%D1%96%D0%B3%D1%80%D0%BE%D0%B2%D0%B8%D0%B9_%D1%80%D1%83%D1%88%D1%96%D0%B9)

15. Unity Forums — Community Discussions. URL: https://discussions.unity.com/

16. Unity GDC Talks — конференції. URL: https://unity.com/events-hub

17. Unity Learn. Create with Code. URL: https://learn.unity.com

18. Unity Technologies. Unity Documentation. URL: https://docs.unity3d.com

19. Unity Technologies. Офіційна документація. URL: https://docs.unity3d.com/ru

20. Алісова Ю.В Використання UNITY для проєктування 2d ігрових рівнів: досвід оптимізації // Комп'ютерні ігри та мультимедіа як інноваційний підхід до комунікації - 2024 / Матеріали IV Всеукраїнської науково-технічної конференції молодих вчених, аспірантів і студентів, Одеса, 26-27 вересня 2024 р. - Одеса, Видавництво ОНТУ, 2024 р. – 400 с.

21. Вишемірська Я.С., Ратушняк Т.В., Семенова Н.В. Використання інтерактивних технологій в умовах дистанційного навчання як засіб активізації навчальної діяльності здобувачів вищої освіти // Практичні питання функціонування і відновлення об’єктів муніципальної інфраструктури та промисловості України в сучасних умовах: збірник матеріалів всеукраїнської науково-практичної конференції (м. Київ, 16 – 17 листопада 2023 року) / упоряд. Гуйда О.Г. К: ТНУ імені В.І. Вернадського, 2023. 259 с. – С. 39-42. https://drive.google.com/file/d/15pV2ntgc89ZzHMHgB4MIknhekV_f6A-S/view

22. Галас А.Я. Аналіз методів розробки програмного додатку-гри в жанрі стратегічного симулятора з використанням ігрового рушія UNITY // Комп'ютерні ігри та мультимедіа як інноваційний підхід до комунікації - 2024 / Матеріали IV Всеукраїнської науково-технічної конференції молодих вчених, аспірантів і студентів, Одеса, 26-27 вересня 2024 р. - Одеса, Видавництво ОНТУ, 2024 р. – 400 с.

23. Гладченко О. Особливості формування інформаційної культури майбутніх фахівців цифрового дизайну (освітньо-професійна програма “Технології цифрового дизайну”) / [О. Гладченко, Т. Ратушняк, А. Омельчук, В. Ніжегородцев, Я. Вишемірська, О. Беспарточна] // Теорія і практика управління

соціальними системами, 2023. - № 4, С. 85-103. DOI: https://doi.org/10.20998/2078-7782.2023.4.08

24. Глеєл Т. Власний Game Engine: як і навіщо створювати ігровий рушій. URL: https://gamedev.dou.ua/blogs/how-and-why-to-create-a-game-engine.

25. Документація MonoBehaviour Unity API: URL: https://docs.unity3d.com/ScriptReference/MonoBehaviour.html

26. Комп'ютерні ігри та мультимедіа як інноваційний підхід до комунікації - 2024 / Матеріали IV Всеукраїнської науково-технічної конференції молодих вчених, аспірантів і студентів, Одеса, 26-27 вересня 2024 р. - Одеса, Видавництво ОНТУ, 2024 р. – 400 с. URL: https://ontu.edu.ua/download/konfi/2024/Abstracts_Computer_games_and_multimedia_innovative_approach_electronic_communication_2024.pdf

27. Куріцин Д.К. UNITY як лідер серед платформ для розробки ігор та додатків: переваги, приклади та перспективи // Комп'ютерні ігри та мультимедіа як інноваційний підхід до комунікації - 2024 / Матеріали IV Всеукраїнської науково-технічної конференції молодих вчених, аспірантів і студентів, Одеса, 26-27 вересня 2024 р. - Одеса, Видавництво ОНТУ, 2024 р. – 400 с.

28. Кучук Г.А., Матвєєв М.І (2022), «Мобільна гра в середовищі Unity проблеми інформатизації», Тези доповідей десятої міжнародної науково-технічної конференції: «Проблеми інформатизації», 24 – 25 листопада 2022 року, Т. 2, С. 116. URL: https://er.chdtu.edu.ua/handle/ChSTU/4231.

29. Кучук Г.А., Матвєєв М.І (2022), «Розробка двомірного ігрового застосунку», Матеріали IV Міжнарод. наук.-техн. конфер.: «Автоматизація, електроніка, інформаційно-вимірювальні технології: освіта, наука, практика», 01- 02 грудня 2022 р., С. 19-20/ URL: https://repository.kpi.kharkov.ua/handle/KhPI-Press/60095

30. Матвєєв М. І., Кучук Г. А. Оптимізація мобільної гри в середовищі UNITY // Системи управління, навігації та зв'язку. 2023. № 2. С. 131-134. URL: DOI: 10.26906/SUNZ.2023.2.131

31. Матеріали на тему «GameDev Camp» | DOUGitHub Ukraine: відкриті Unity-проєкти. URL: https://gamedev.dou.ua/tags/GameDev%20Camp/

32. Мельник П.П. Технологічні аспекти використання рушія UNITY для розробки гри-квесту //Сучасні цифрові технології та інноваційні методики навчання: досвід, тенденції, перспективи : матеріали XIII Міжнародної науково-практичної інтернет-конференції, м. Тернопіль, 5 квітня, 2024 р. Тернопіль : ТНПУ ім. В. Гнатюка, 2024. 260 с. С. 169-172. URL: https://lib.iitta.gov.ua/id/eprint/740555/1/%D0%97%D0%B1%D1%96%D1%80%D0%BD%D0%B8%D0%BA_%D0%A2%D0%B5%D1%80%D0%BD%D0%BE%D0%BF%D1%96%D0%BB%D1%8C_2024.pdf

33. Москаленко І. А. Розробка 2D гри на Unity : дипломна бакалаврська робота за спеціальністю 122 Комп'ютерні науки / І. А. Москаленко ; наук. кер. Т. І. Демківська; рец. В. І. Чупринка. – Київ : КНУТД, 2023. – 77 с. URL: https://er.knutd.edu.ua/handle/123456789/23807

34. Онлайн-курс «Розробка мобільних ігор на Unity». URL: https://salo.li/09Bf4aC

35. Освітній портал ITVDN — курс "Unity Game Developer"

36. Ратушняк Т.В. Технологія створення віртуальної сферичної 3d-панорами для соціальної мережі Facebook [Т.В. Ратушняк, О.В. Гладченко, А.А. Омельчук, Я.С. Вишемірська] // Прикладні питання математичного моделювання, 2023. - № 1, Том 6. - С. 124-133. DOI: https://doi.org/10.32782/mathematical-modelling/2023-6-1-14

37. Ратушняк Т.В., Заврін Д.Р. Практичне застосування анімаційних технологій у соціальних проєктах // Мультимедійні технології в освіті та інших сферах діяльності: збірник тез науково–практичної конференції з міжнародною участю [Електронний ресурс] - Київ: Національний авіаційний університет, 10 листопада 2022. - С. 53-56. https://drive.google.com/file/d/1lvoUkhb3qs3HvLQ9FAwqjD2zcj3LOnPy/view http://kmmt.nau.edu.ua/kmmt-page/kmmt-conference/

38. Ресурси з розробки ігор. URL: https://unity.com/ru/resources

39. Суліма Ю.Ю. Розробка гри в жанрі платформер з використанням фізики програмного рушія UNITY // Комп'ютерні ігри та мультимедіа як інноваційний підхід до комунікації - 2024 / Матеріали IV Всеукраїнської науково-технічної конференції молодих вчених, аспірантів і студентів, Одеса, 26-27 вересня 2024 р. - Одеса, Видавництво ОНТУ, 2024 р. – 400 с.

40. Тарасенко М. А. Розробка 2D гри-платформера на Unity / М.А. Тарасенко; наук. керівник ст. викл. Н. С. Калайда // Радіоелектроніка та молодь у XXI столітті : матеріали 28-го Міжнар. молодіж. форуму, 16–18 квітня 2024 р. – Харків : ХНУРЕ, 2024. – Т. 6 – С. 734-735. URL: https://openarchive.nure.ua/handle/document/28822

41. Уніфіковані освітні ресурси з проєктування UI/UX: UX Design in Games — Interaction Design Foundation. URL: https://www.interaction-design.org/?srsltid=AfmBOorUF-18WjFtNUZYOi_IWxa26pNHxColpgTX7OJ28W5nJ1PMGiRS