

**МІНІСТЕРСТВО ФІНАНСІВ УКРАЇНИ
ДЕРЖАВНИЙ ПОДАТКОВИЙ УНІВЕРСИТЕТ**

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

КВАЛІФІКАЦІЙНА РОБОТА

на тему: Освітній застосунок для розвитку кольорового сприйняття в інтерактивному середовищі Unity

Здобувача вищої освіти першого (бакалаврського) рівня,
IV курсу, групи ЦДБ-22-1, спеціальності 122 «Комп'ютерні науки»,
освітня програма «Технології цифрового дизайну»


(підпис)

Голуба Віталія
Олександровича
(прізвище, ім'я, по батькові)

Науковий керівник


(підпис)

Омельчук Антон
Анатолійович, канд. тех. наук
(прізвище, ім'я, по батькові, науковий ступінь,
вчене звання)

Керівник освітньо-
професійної програми


(підпис)

Ратушняк Тетяна Володимирівна
канд. фіз.-мат. наук, 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 « 12 » травня 2026 № 20

Завідувач кафедри


(підпис)

Філоненко Михайло
Миколайович, канд. фіз.-мат.
наук, доцент
(прізвище, ім'я, по батькові, науковий ступінь,
вчене звання)

Ірпінь – 2026

Реєстрація № 69

29.05.2026

(дата)



(підпис відповідальної особи)

Результати перевірки

до захисту

(до захисту, на доопрацювання)

12.05.2026

(дата)



(підпис)

Результат захисту

5 (відмінно)

(оцінка)

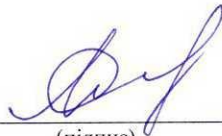
11.06.2026

(дата)



(підпис)

Голова комісії:



(підпис)

О.С. Дюбанов

Члени комісії:



(підпис)

М.М. Філоненко



(підпис)

Т.В. Ратушняк



(підпис)

В.О. Ніжегородцев

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	27		11.06.2026 Омельчук А.А. Дюбанов О.С.
2. Дотримання вимог щодо оформлення	від 0 до 10	8		Філоненко М.М. Ратушняк Т.В.
3. Вчасне виконання роботи на всіх етапах у встановлені кафедрою терміни	від 0 до 5	4		Ніжегородцев В.О. Суховерхня Л.П.
4. Рецензент	від 0 до 5	5		
РАЗОМ БАЛІВ	від 0 до 50	44		

Оцінювання кваліфікаційної роботи науковим керівником – 46 балів.

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

Оцінка за 100-бальною шкалою	Оцінка за шкалою ЄКТС	Оцінка за національною шкалою	Дата	П.І.Б.: голови, членів комісії, секретаря, (підписи)
90	A	5 (відмінно)	11.06.2026	Дюбанов О.С. Філоненко М.М. Ратушняк Т.В. Ніжегородцев В.О. Суховерхня 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	
50-59	E	3 (задовільно)
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.

«01» грудня 2023

Завдання на кваліфікаційну роботу
здобувачу вищої освіти

Голубу Віталію Олександровичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи: Освітній застосунок для розвитку кольорового сприйняття в інтерактивному середовищі Unity.
Затверджена наказом ректора від «28» листопада 2025 № 1526
2. Строк здачі закінченої роботи «11» травня 2026
3. Мета роботи розробка та програмна реалізація освітнього ігрового застосунку для розвитку навичок кольорового сприйняття в інтерактивному середовищі Unity.
Об'єкт дослідження є процес проєктування та розробки спеціалізованого навчального програмного забезпечення.
Предмет дослідження виступають конкретні складові обраної інформаційної технології.
Орієнтовна база дослідження виступає технологічне середовище Unity як платформа для розробки й виконання ігрового програмного коду.
4. Вихідні дані до роботи є технічні та концептуальні параметри розробки освітнього застосунку «Повернення кольорів» у середовищі Unity мовою C#
5. Перелік графічного матеріалу: Рис. 2.1. Початковий стан (перемішані плитки). Рис. 2.2. Фінальний стан (правильний вертикальний градієнт). Рис. 2.3. Стан після випадкового змішування. Рис. 2.4. Відновлений градієнт за кольоровими групами. Рис. 2.5. Початковий стан (Рівень 0) із сірим фоном. Рис. 2.6. Стан після завершення першого рівня з частковим офарбленням. Рис. 3.7. Розроблений логотип.
6. Зміст кваліфікаційної роботи (перелік питань за кожним розділом)
Розділ 1. Теоретико-методологічні основи створення освітніх застосунків для розвитку кольорового сприйняття. Розділ 2. Системний аналіз та проєктування інтерактивного застосунку. Розділ 3. Програмна реалізація та тестування освітнього застосунку в середовищі unity

7. Календарний план виконання роботи

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	
		За планом	Фактично
1	2	3	4
1	Вибір теми, підбір літературних джерел, формування структури кваліфікаційної роботи	20.12.2025	20.12.2025
2	I розділ (розкрити сутність і стан дослідження наукової проблематики роботи, після чого навести її загальну характеристику; огляд літератури за темою і вибір конкретних напрямів досліджень)	26.02.2026	29.02.2026
3	II розділ (аналіз експериментальної частини про методики досліджень)	13.03.2026	28.03.2026
4	III розділ (проведені теоретичні і (або) експериментальні дослідження та проектні рішення)	02.04.2026	11.04.2026
5	Висновки	17.04.2026	17.04.2026
6	Доопрацювання роботи з урахуванням зауважень наукового керівника	25.04.2026	27.04.2026
7	Отримання відгуку керівника	01.05.2026	01.05.2026
8	Отримання зовнішньої рецензії	04.06.2026	04.06.2026

8. Дата видачі завдання «01» грудня 2025

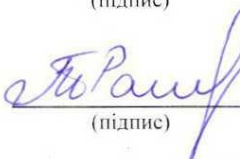
9. Керівник кваліфікаційної роботи

10. Керівник освітньо-професійної програми

11. Завдання до виконання отримав здобувач вищої освіти


(підпис)

Омельчук А.А.
(прізвище, ініціали)


(підпис)

Ратушняк Т.В.
(прізвище, ініціали)


(підпис)

Голуб В.О.
(прізвище, ініціали)

АНОТАЦІЯ

Освітній застосунок для розвитку кольорового сприйняття в інтерактивному середовищі Unity

(назва кваліфікаційної роботи)

Мета роботи розробка та програмна реалізація в Unity освітнього застосунку для розвитку кольорового сприйняття. Завданнями є аналіз гейміфікації навчання, проєктування архітектури, градієнтної логіки та взаємодії з об'єктами, а також тестування системи. Об'єктом дослідження є процес створення інтерактивного навчального ПЗ для розвитку когнітивних здібностей. Предмет дослідження алгоритми формування й верифікації градієнтів у сітці 6×7 . Базою дослідження виступає середовище Unity та тематичні практичні напрацювання.

За результатами роботи сформульовано принципи модульної архітектури інтерфейсу, реалізовано алгоритм змішування Фішера–Єйтса та модель лінійної інтерполяції RGBA-векторів. Верифікація підтвердила стабільність коду та точність логіки перевірки умов перемоги `IsComplete`. Одержані результати можуть застосовуватися в закладах освіти, для підготовки дизайнерів та ІТ-фахівців або у когнітивній реабілітації.

Кваліфікаційна робота містить 60 сторінок, 4 таблиць, 7 рисунків, список використаних джерел із 40 найменувань, 0 додатків.

Ключові слова: UNITY, C#, ГЕЙМІФІКАЦІЯ, КОЛЬОРОВЕ СПРИЙНЯТТЯ, ГРАДІЄНТ, АЛГОРИТМ ФІШЕРА–ЄЙТСА, КОГНІТИВНІ НАВИЧКИ, ПОДІЄВА МОДЕЛЬ.

Рік виконання кваліфікаційної роботи 2026.

Рік захисту роботи 2026.

ABSTRACT

An Educational Application for Developing Color Perception in the Interactive Unity Environment

(title of the qualification work)

The purpose of this qualification work is the development and software implementation of an educational game application for developing color perception skills in the Unity environment. The objectives include researching cognitive learning gamification, justifying development tools, designing architecture and gradient logic, implementing object interaction, and testing algorithms . The object of the study is the design and development of specialized educational software based on interactive multimedia technologies. The subject of the study is the algorithmic software for dynamic formation and verification of color gradients within a 6×7 grid. The base of the study is the Unity technological environment and relevant practical developments .

Based on the research results, the principles of building a modular architecture for game mechanics and the user interface were formulated. An element-shuffling algorithm based on the Fisher-Yates scheme and a mathematical model for the linear interpolation of RGBA color vectors were practically implemented. Comprehensive testing confirmed high code stability and the correctness of the IsComplete win condition logic. The results can be used in educational institutions to develop visual perception, train IT specialists and designers, or serve as a tool for cognitive rehabilitation.

The qualification work contains 60 pages, 4 tables, 7 figures, a list of sources with 40 entries, and 0 appendices.

Keywords: UNITY, C#, GAMIFICATION, COLOR PERCEPTION, GRADIENT, FISHER-YATES ALGORITHM, COGNITIVE SKILLS, EVENT-DRIVEN MODEL.

Year of qualification work completion 2026.

Year of work defense 2026.

ЗМІСТ

ВСТУП.....	3
РОЗДІЛ 1. ТЕОРЕТИКО-МЕТОДОЛОГІЧНІ ОСНОВИ СТВОРЕННЯ ОСВІТНІХ ЗАСТОСУНКІВ ДЛЯ РОЗВИТКУ КОЛЬОРОВОГО СПРИЙНЯТТЯ.....	6
1.1. Аналіз сучасних тенденцій гейміфікації в освіті та застосунків для тренування когнітивних навичок.....	6
1.2. Огляд інтерактивних середовищ та платформ для розробки навчальних ігор.....	13
1.3. Обґрунтування вибору технологій (Unity, C#) та розробка концепції проєкту «Повернення кольорів»	17
РОЗДІЛ 2. СИСТЕМНИЙ АНАЛІЗ ТА ПРОЄКТУВАННЯ ІНТЕРАКТИВНОГО ЗАСТОСУНКУ	21
2.1. Архітектура застосунку та розробка функціональної схеми ігрових рівнів.....	21
2.2. Проєктування алгоритму випадкового змішування плиток та логіки перевірки градієнта.....	25
2.3. Моделювання системи прогресії гравця та динамічної зміни візуального середовища	32
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ОСВІТНЬОГО ЗАСТОСУНКУ В СЕРЕДОВИЩІ UNITY	36
3.1. Програмна реалізація ігрових механік та інтерфейсу користувача засобами C#.....	36
3.2. Тестування працездатності алгоритмів та аналіз результатів розробки.	40
3.3. Перспективи подальшого розвитку та стратегія дистрибуції проєкту	44
ВИСНОВКИ.....	48
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	51

ВСТУП

Актуальність теми. У сучасних умовах цифровізації освіти особливої ваги набувають інноваційні методи навчання, серед яких провідне місце посідає гейміфікація. Використання ігрових механік дозволяє поєднувати освітній процес із розвагою, що значно підвищує залученість користувачів та ефективність засвоєння матеріалу [4]. Розвиток когнітивних навичок, зокрема кольорового сприйняття, є важливим аспектом підготовки фахівців у сферах дизайну, мистецтва та ІТ, проте традиційні методи часто є монотонними [1]. Створення інтерактивного застосунку в середовищі Unity дозволяє розв'язати цю проблему через візуально привабливий та технічно досконалий ігровий процес.

Стан і ступінь розробки проблеми. Питаннями гейміфікації та розробки інтерактивних систем займалися як вітчизняні, так і зарубіжні вчені, зокрема розробники навчальних ІТ-проєктів. Проте, незважаючи на велику кількість ігор, сегмент вузькоспеціалізованих застосунків для тренування сприйняття кольірних градієнтів потребує подальшого вдосконалення в контексті адаптивності та сюжетного наповнення.

Мета дослідження. Метою кваліфікаційної роботи є розробка та програмна реалізація освітнього ігрового застосунку для розвитку навичок кольорового сприйняття в інтерактивному середовищі Unity.

Завдання дослідження:

1. **Дослідити** сучасні тенденції розвитку навчальних ігор та методики тренування когнітивних навичок.
2. **Обґрунтувати** вибір інструментарію розробки (Unity, мова C#) та сформулювати концепцію ігрового сюжету.
3. **Спроекувати** архітектуру застосунку, зокрема логіку генерації пазлів-градієнтів та систему прогресії рівнів.
4. **Реалізувати** програмний код для взаємодії з ігровими об'єктами та динамічної зміни візуального середовища.

5. **Провести** тестування розробленого застосунку на предмет коректності роботи алгоритмів.

Об'єктом дослідження є процес проєктування та розробки спеціалізованого навчального програмного забезпечення, що базується на використанні інтерактивних мультимедійних технологій для стимуляції та розвитку когнітивних здібностей користувача, зокрема сприйняття кольору та логічного мислення. Даний процес охоплює повний життєвий цикл створення продукту від аналізу теоретичних засад кольорознавства до технічної реалізації системи в ігровому середовищі [3].

Предметом дослідження виступають конкретні складові обраної інформаційної технології, а саме алгоритмічне забезпечення для динамічного формування та верифікації колірних градієнтів у межах ігрової сітки 6×7 , що забезпечує коректність навчального процесу. Окрім цього, досліджуються механіки ігрової прогресії, які відповідають за адаптивність інтерфейсу та сюжетний розвиток, зокрема зміну колірної гами меню відповідно до успіхів гравця. Важливою складовою є також технічний інструментарій середовища Unity та об'єктно-орієнтована мова програмування C#, що використовуються як основні засоби реалізації інтерактивної взаємодії.

Методи дослідження включають загальнонаукові підходи аналізу та синтезу, які були застосовані для опрацювання наукової літератури, порівняння існуючих рішень і формування концепції «світу без кольору». Також використано методи об'єктно-орієнтованого програмування при розробці програмних скриптів мовою C# для керування логікою пазлів, перемішування елементів та перевірки стану ігрового поля. Додатково застосовано методи візуального моделювання для створення графічного інтерфейсу користувача, розробки функціональних схем рівнів і налаштування візуальних ефектів у середовищі Unity.

База дослідження охоплює технологічне середовище, інформаційні джерела та практичні напрацювання, на яких ґрунтується виконана робота. Зокрема, основною платформою виступає Unity як середовище розробки та

виконання ігрового програмного коду. Також враховано досвід сучасних платформ цифрової дистрибуції шляхом аналізу існуючих проєктів, зокрема мобільних логічних ігор, що дозволило визначити вимоги до якості кінцевого продукту. Додатково використано стандарти ергономіки та дизайну, включаючи теоретичні засади UX/UI, необхідні для створення зручного та ефективного освітнього застосунку.

Наукова та практична значимість роботи полягає у поєднанні інноваційних підходів до навчання з сучасними технологіями розробки програмного забезпечення. Наукова новизна дослідження полягає в удосконаленні методів навчання шляхом інтеграції алгоритмів колірної аналізу в сюжетно-ігрову оболонку, що забезпечує безперервний зворотний зв'язок із користувачем. Практичне значення полягає у створенні готового до впровадження програмного продукту, який може бути використаний у закладах освіти для розвитку зорового сприйняття або як інструмент реабілітації [16]. Крім того, розроблений алгоритм перевірки градієнтів у сітці 6×7 має високий рівень масштабованості, що дозволяє адаптувати його для створення складніших рівнів і розширення функціоналу в інших освітніх або логічних ІТ-рішеннях.

РОЗДІЛ 1. ТЕОРЕТИКО-МЕТОДОЛОГІЧНІ ОСНОВИ СТВОРЕННЯ ОСВІТНІХ ЗАСТОСУНКІВ ДЛЯ РОЗВИТКУ КОЛЬОРОВОГО СПРИЙНЯТТЯ

1.1. Аналіз сучасних тенденцій гейміфікації в освіті та застосунків для тренування когнітивних навичок

Сучасний етап розвитку інформаційних технологій в освіті характеризується активним і цілеспрямованим впровадженням концепції гейміфікації, яка розглядається як один із найбільш ефективних інструментів підвищення якості навчального процесу. Під гейміфікацією розуміють використання ігрових елементів, стратегій, механік та принципів дизайну у неігровому контексті, зокрема в освітньому середовищі, з метою суттєвого підвищення рівня залученості, зацікавленості, лояльності та внутрішньої мотивації користувачів [18]. Такий підхід дозволяє змінити традиційне сприйняття навчання як обов'язкового та часто рутинного процесу, трансформуючи його у захопливу та емоційно насичену діяльність. Важливо підкреслити, що гейміфікація не обмежується поверхневим додаванням ігрових атрибутів, таких як бали, рівні чи нагороди. Вона являє собою комплексну методологію проєктування освітнього середовища, яка ґрунтується на глибокому розумінні психологічних особливостей людини, її мотиваційних механізмів та поведінкових моделей. Зокрема, враховуються такі чинники, як прагнення до досягнення цілей, потреба у визнанні, ефект змагання, а також задоволення від поступового прогресу та саморозвитку. Саме завдяки цьому гейміфікація здатна забезпечити не лише короточасне зацікавлення, але й довготривале утримання уваги користувача [40]. Основним стратегічним завданням гейміфікації є глибинна трансформація традиційного навчального процесу, який часто характеризується монотонністю, пасивністю та низьким рівнем інтерактивності, у динамічне, адаптивне та високоінтерактивне середовище. У такому середовищі користувач виступає не пасивним споживачем інформації, а активним

учасником навчального процесу, який взаємодіє з контентом, приймає рішення, експериментує та отримує миттєвий зворотний зв'язок щодо своїх дій. Це, у свою чергу, сприяє кращому засвоєнню матеріалу, розвитку критичного мислення та формуванню практичних навичок. Реалізація гейміфікаційних підходів також дозволяє ефективно впровадити принцип «edutainment» органічного поєднання освітньої діяльності з елементами розваги [31]. Такий підхід сприяє зниженню рівня когнітивного навантаження та мінімізує психологічний бар'єр, який часто виникає перед складними або абстрактними навчальними завданнями. Завдяки використанню сюжетних ліній, візуальних ефектів, інтерактивних сценаріїв і поступового ускладнення завдань створюється сприятливе середовище для навчання, у якому користувач не лише здобуває знання, але й отримує позитивний емоційний досвід. Таким чином, гейміфікація виступає не лише допоміжним інструментом, а повноцінною педагогічною технологією, яка має потенціал суттєво підвищити ефективність освітніх процесів, зробити їх більш персоналізованими, адаптивними та орієнтованими на потреби сучасного користувача.

У контексті підготовки фахівців цифрової галузі розвиток когнітивних навичок, зокрема тонкого зорового сприйняття, аналітичного мислення, логічної дедукції та здатності до швидкої обробки візуальної інформації, постає як один із ключових і стратегічно важливих науково-практичних напрямів, оскільки саме ці компетенції формують інтелектуальну та операційну основу професійної діяльності у сфері інформаційних технологій і безпосередньо визначають ефективність виконання складних прикладних завдань, пов'язаних із розробкою програмного забезпечення, комп'ютерною графікою, дизайном інтерфейсів та аналітикою даних, зокрема, високий рівень розвитку зорового сприйняття забезпечує здатність до точної ідентифікації кольорів, відтінків і візуальних патернів, що є критично важливим при створенні якісного графічного контенту та забезпеченні візуальної узгодженості інтерфейсів, тоді як аналітичне мислення та логічна дедукція

дозволяють ефективно структурувати інформацію, виявляти закономірності та приймати обґрунтовані рішення у процесі проєктування та оптимізації цифрових систем, у цьому контексті інтеграція ігрових механік у процес розвитку когнітивних здібностей відкриває принципово нові можливості для підвищення результативності навчання, оскільки дозволяє не лише підвищити рівень залученості користувача, але й активізувати складні нейрокогнітивні процеси, пов'язані з навчанням через досвід, повторення та емоційне підкріплення, забезпечуючи формування стійких навичок на рівні автоматизованих дій, важливою перевагою такого підходу є реалізація механізму миттєвого зворотного зв'язку, який забезпечує безперервний контроль правильності виконання завдань і дозволяє користувачу оперативно коригувати власні дії, що значно підвищує ефективність засвоєння матеріалу порівняно з традиційними методами навчання, де оцінювання часто має відкладений характер, застосування системи ігрової прогресії, яка включає рівні складності, систему досягнень, рейтинги та візуалізацію результатів, зокрема через зміну параметрів ігрового середовища, кольорової гами чи доступних функціональних елементів відповідно до успіхів користувача, сприяє формуванню чіткого причинно-наслідкового зв'язку між витраченими зусиллями та отриманим результатом, що, у свою чергу, стимулює внутрішню мотивацію, підвищує рівень самозалученості та формує позитивний досвід взаємодії з навчальним середовищем, додатково слід зазначити, що гейміфікований підхід дозволяє ефективно реалізувати принцип адаптивності навчання, відповідно до якого складність завдань, швидкість подачі матеріалу та рівень підтримки динамічно змінюються залежно від індивідуальних характеристик користувача, його поточного рівня підготовки та темпу засвоєння інформації, що забезпечує оптимальний баланс між складністю завдань і можливістю їх успішного виконання, запобігаючи як перевантаженню, так і втраті інтересу, це набуває особливої актуальності у вузькоспеціалізованих напрямках, таких як розвиток здатності до точної кольорової диференціації, просторового мислення та роботи з візуальними

структурами, де навчальний процес вимагає високої концентрації уваги, точності сприйняття та багаторазового повторення однотипних операцій, що в умовах традиційного підходу часто призводить до когнітивної втоми, зниження продуктивності та поступової втрати мотивації, натомість використання інтерактивного ігрового середовища дозволяє мінімізувати ці негативні ефекти за рахунок поєднання навчальної діяльності з елементами змагання, досягнення, відкриття нового контенту та емоційного підкріплення, підтримуючи стабільно високий рівень когнітивної активності, зацікавленості та ефективності засвоєння знань протягом тривалого часу, що в підсумку сприяє формуванню висококваліфікованих фахівців, здатних успішно адаптуватися до вимог сучасного цифрового середовища та вирішувати складні професійні задачі [23].

Тенденції останніх років у сфері інформаційних технологій та цифрової освіти свідчать про поступове, але виразне зміщення акценту від створення виключно розважальних програмних продуктів до розробки так званих «серйозних ігор» (serious games), які інтегрують у собі ігрову динаміку з чітко визначеними освітніми, тренувальними, науковими або реабілітаційними цілями, що дозволяє використовувати їх як повноцінний інструмент формування професійних компетенцій і розвитку когнітивних здібностей, при цьому створення таких систем дедалі частіше здійснюється на базі сучасних професійних ігрових рушіїв, зокрема Unity, який забезпечує високий рівень графічної візуалізації, інтерактивності, масштабованості та технічної гнучкості, що дає змогу реалізовувати складні алгоритмічні моделі, адаптивні сценарії поведінки системи та багаторівневі механіки взаємодії користувача із цифровим середовищем, на відміну від традиційних освітніх підходів, де користувач часто виступає пасивним споживачем інформації, serious games орієнтовані на активну участь, експериментування та навчання через досвід, що досягається шляхом моделювання умовних або наближених до реальності ситуацій, у яких користувач приймає рішення, аналізує результати власних дій і поступово вдосконалює свої навички, зокрема, інтерактивні застосунки,

спрямовані на розвиток кольорового сприйняття, дозволяють не лише здійснювати комплексну діагностику поточного рівня сформованості перцептивних навичок користувача, включаючи точність розпізнавання відтінків, швидкість реакції та здатність до диференціації кольорових градієнтів, але й реалізовувати механізми інтелектуальної адаптації, за яких складність завдань, кількість елементів, варіативність кольорових комбінацій і темп подачі інформації динамічно змінюються відповідно до індивідуального прогресу користувача, забезпечуючи оптимальні умови для розвитку когнітивних функцій без перевантаження або втрати інтересу, додатково важливим елементом таких систем є впровадження продуманих ігрових наративів, зокрема сюжетних моделей, що базуються на ідеї відновлення втраченого кольору у віртуальному світі, які не лише виконують декоративну функцію, але й формують смисловий каркас навчального процесу, надаючи кожному завданню контекстуального значення, підсилюючи емоційне залучення користувача та стимулюючи внутрішню мотивацію через відчуття прогресу, досягнення мети та впливу на віртуальне середовище, важливо також зазначити, що використання таких підходів сприяє розвитку не лише вузькоспеціалізованих навичок, але й загальних когнітивних здібностей, включаючи увагу, пам'ять, швидкість обробки інформації та здатність до прийняття рішень у умовах обмеженого часу, у підсумку гейміфікація в межах концепції *serious games* постає не лише як засіб підвищення залученості чи інтересу до навчання, а як комплексна міждисциплінарна методологія, що поєднує досягнення інформаційних технологій, когнітивної психології та педагогіки, забезпечуючи створення адаптивних, інтерактивних і ефективних освітніх середовищ, які відповідають вимогам сучасного цифрового суспільства, сприяють підвищенню якості підготовки фахівців і формуванню їх здатності до безперервного професійного розвитку в умовах швидкої технологічної еволюції [25].

Порівняльний аналіз із існуючими рішеннями («I Love Hue» та «Blendoku»). Аналіз сучасного ринку ігрових застосунків дозволяє

виокремити кілька ключових проєктів, що спеціалізуються на роботі з колірною гармонією, серед яких найбільш значущими є «I Love Hue» та «Blendoku». Проєкт «I Love Hue» характеризується вираженням медитативним підходом, де основний фокус зосереджено на сортуванні хаотично розміщених колірних плиток для відновлення візуального порядку. Попри високу естетичну цінність, цей застосунок часто позбавлений чіткої сюжетної лінії, що може призводити до поступового зниження залученості користувача через відсутність зовнішньої мотивації. На відміну від нього, «Blendoku» використовує логіку, наближену до класичних головоломок типу sudoku, де гравцеві необхідно вибудовувати складні ланцюжки колірних переходів.

У порівнянні з цими аналогами, розроблювана концепція освітнього застосунку в Unity пропонує значно вищий рівень інтерактивності завдяки впровадженню ігрового наративу. Сюжетна складова про світ, що втратив кольори, створює додаткову психологічну цінність кожної перемоги. Використання специфічної сітки 6x7 є оптимальним балансом між складністю та когнітивним навантаженням, а механіка динамічного офарблення головного меню відповідно до ігрової прогресії створює ефект «живого» середовища, якого не вистачає комерційним аналогам. Це дозволяє не лише тренувати навички, а й підтримувати високий рівень інтересу за рахунок видимих результатів діяльності гравця.

Психологічні аспекти сприйняття градієнтів та когнітивне значення. З точки зору когнітивної психології та нейрофізіології, сприйняття градієнтів є складною операцією, що вимагає злагодженої роботи зорової кори та префронтальних зон мозку. Людське око сприймає колірні переходи через фоторецептори сітківки, проте саме мозок інтерпретує ці сигнали, вибудовуючи логічні зв'язки між відтінками на основі їхньої насиченості та яскравості. Процес відновлення градієнта змушує користувача постійно виконувати операції порівняння та аналізу, що стимулює нейропластичність та покращує здатність до диференціації тонких візуальних нюансів [5].

Систематичне виконання таких вправ є критично важливим для розвитку зорової грамотності, оскільки воно привчає мозок звертати увагу на деталі, які зазвичай ігноруються під час швидкого огляду об'єктів. Окрім суто навчального ефекту, робота з гармонійними колірними спектрами сприяє зниженню рівня стресу та входженню користувача у «стан потоку», що підвищує загальну працездатність та концентрацію уваги. Таким чином, ігровий процес стає не просто розвагою, а дієвим інструментом для покращення когнітивної гнучкості та підготовки фахівців до роботи з професійними графічними та аналітичними інтерфейсами [27].

Таблиця 1.1

**Порівняльна характеристика традиційного та гейміфікованого
підходів до розвитку когнітивних навичок**

Критерій порівняння	Традиційний підхід	Гейміфікований підхід (проект «Повернення кольорів»)
Характер мотивації	Зовнішня (виконання вказівок викладача, оцінювання)	Внутрішня (ігровий сюжет, бажання відновити світ)
Зворотний зв'язок	Відстрочений (після перевірки завдання фахівцем)	Миттєвий (візуальна реакція гри на правильність градієнта)
Рівень залученості	Низький або середній (через монотонність вправ)	Високий (через інтерактивність та механіку «drag-and-drop»)
Візуалізація прогресу	Статична (записи у журналах, оцінки)	Динамічна (поступове офарблення ігрового меню та світу)
Стан користувача	Висока ймовірність швидкої втомлюваності	Стан «потоку», емоційне задоволення від гри

Джерело: розроблено автором

1.2. Огляд інтерактивних середовищ та платформ для розробки навчальних ігор

Сучасний ринок програмного забезпечення для розробки ігрових та освітніх застосунків пропонує широкий спектр інтерактивних середовищ, кожне з яких має свої специфічні особливості, переваги та обмеження. Вибір платформи для створення проєкту «Повернення кольорів» потребує детального порівняльного аналізу, оскільки обраний інструментарій повинен забезпечувати не лише високу якість візуалізації колірних градієнтів, а й гнучкість у налаштуванні користувацького інтерфейсу (UI) та логіки взаємодії з ігровими об'єктами [10]. Крім того, важливим аспектом є продуктивність середовища розробки та його здатність ефективно працювати з графічними ресурсами, особливо у випадках, коли застосунок передбачає плавні переходи кольорів, анімації та інтерактивні ефекти. Не менш значущою є підтримка різних платформ настільних, мобільних або веб що дозволяє розширити аудиторію користувачів і забезпечити доступність продукту для різних категорій гравців.

Особливу увагу слід приділити також інструментам для роботи з кольором, зокрема можливості точного налаштування палітр, використання шейдерів та обробки освітлення, що безпосередньо впливає на візуальне сприйняття гри. У контексті освітнього спрямування проєкту важливо, щоб середовище підтримувало реалізацію навчальних механік, таких як підказки, адаптивна складність та зворотний зв'язок для користувача. Додатковим критерієм вибору є зручність розробки: наявність зрозумілої документації, активної спільноти, готових бібліотек і модулів, що можуть суттєво пришвидшити процес створення гри. Інтеграція з іншими інструментами, такими як графічні редактори або системи контролю версій, також відіграє важливу роль у забезпеченні ефективного робочого процесу.

Таким чином, обрання оптимальної платформи для реалізації проєкту «Повернення кольорів» є комплексним завданням, що потребує врахування технічних, візуальних та педагогічних вимог. Лише збалансований підхід до

оцінки можливостей різних середовищ дозволить створити якісний, привабливий та функціональний продукт, який відповідатиме поставленим цілям і очікуванням користувачів.

Серед найбільш затребуваних професійних рішень для розробки інтерактивного контенту виділяють такі платформи, як Unreal Engine, Godot Engine та Unity. Кожна з них має власну філософію розробки, набір інструментів та цільову аудиторію, що безпосередньо впливає на доцільність їх використання в межах конкретного проєкту [8].

Unreal Engine характеризується надзвичайно потужними можливостями для створення високореалістичної графіки, що досягається завдяки використанню сучасних технологій рендерингу, таких як динамічне глобальне освітлення, фізично коректні матеріали та підтримка трасування променів у реальному часі. Рушій надає розробникам доступ до інструментів кінематографічного рівня, які широко застосовуються не лише в ігровій індустрії, але й у сфері кіно та візуалізації. Важливою перевагою є система візуального програмування Blueprints, яка дозволяє реалізовувати складну ігрову логіку шляхом побудови графічних схем, що значно спрощує розробку для дизайнерів і початківців, зменшуючи потребу у глибокому знанні мов програмування [11]. Разом із цим, високий рівень функціональності Unreal Engine супроводжується значною складністю архітектури, що вимагає більше часу на освоєння та налаштування середовища. Для проєктів, орієнтованих на 2D-графіку, зокрема освітніх застосунків із відносно простою механікою, більшість можливостей рушія залишаються невикористаними, що робить його застосування нераціональним. Крім того, Unreal Engine має підвищені вимоги до апаратного забезпечення як під час розробки, так і під час виконання готового продукту, що може обмежити доступність застосунку для користувачів із менш продуктивними пристроями. Додатковим недоліком є збільшений розмір збірки, що є критичним фактором для мобільних платформ, де важливу роль відіграють обмеження пам'яті та швидкість завантаження. У сукупності ці фактори свідчать про те, що, незважаючи на свої потужні

можливості, Unreal Engine є надлишковим і менш доцільним вибором для реалізації проєкту «Повернення кольорів».

Godot Engine є відкритим (open-source) середовищем розробки, яке активно розвивається та набуває популярності завдяки своїй легкій і гнучкій архітектурі. Рушій не потребує складного встановлення, має невеликий розмір і швидко запускається, що значно спрощує початок роботи над проєктом. Однією з ключових переваг є використання власної мови програмування GDScript, синтаксис якої подібний до Python, що робить її інтуїтивно зрозумілою навіть для початківців. Крім того, Godot підтримує також інші мови, зокрема C#, що розширює можливості для більш досвідчених розробників. Вбудована система сцен і вузлів (nodes) забезпечує зручну організацію проєкту та дозволяє ефективно керувати ігровими об'єктами й їх поведінкою. Завдяки низьким системним вимогам Godot добре підходить для створення 2D-проєктів, включаючи навчальні застосунки та прості ігри, а також для швидкого прототипування і тестування ідей. Водночас, попри активний розвиток, екосистема рушія залишається менш зрілою порівняно з провідними комерційними рішеннями. Зокрема, обмежена кількість готових ассетів, плагінів і бібліотек може вимагати додаткового часу на самостійну реалізацію базових функцій. Менша спільнота розробників означає також меншу кількість навчальних матеріалів, прикладів і швидких рішень типових проблем. Окрему складність може становити створення складних і масштабованих користувацьких інтерфейсів, оскільки інструменти для роботи з UI у Godot є менш розвиненими, ніж у деяких інших рушіях. У підсумку, хоча Godot є перспективним і доступним інструментом, його використання може бути менш ефективним у випадках, коли важливими є швидкість розробки, наявність готових рішень і підтримка великої екосистеми [26].

Unity на сьогодні є одним із найбільш збалансованих середовищ для створення кросплатформних застосунків, особливо у сфері мобільних і 2D-ігор. Основними перевагами Unity є розвинена система роботи з 2D-графікою (включаючи спрайти, анімації та сортування шарів), потужний інструментарій

UI Canvas для побудови складних та адаптивних інтерфейсів, а також використання мови програмування C#, яка є галузевим стандартом об'єктно-орієнтованого програмування [2]. Крім того, важливою перевагою є наявність великого магазину ресурсів (Asset Store), що дозволяє значно прискорити розробку завдяки використанню готових рішень, бібліотек і шаблонів.

З урахуванням поставлених вимог до проєкту «Повернення кольорів», таких як якісна робота з кольоровими градієнтами, зручний інтерфейс, підтримка мобільних платформ і ефективна реалізація освітніх механік, саме Unity виглядає найбільш доцільним вибором. Він поєднує в собі достатню продуктивність, гнучкість та доступність інструментів, що дозволяє оптимально реалізувати як технічні, так і педагогічні аспекти застосунку. Особливе значення для реалізації проєкту має система Unity UI, яка дозволяє легко маніпулювати спрайтами (PNG-файлами) та створювати динамічні сітки об'єктів. Для гри, що базується на сітці 6x7, Unity пропонує готові компоненти (наприклад, `GridLayoutGroup`), що значно спрощує верстку ігрового поля навіть для розробників, які не мають глибокого досвіду у низькорівневому програмуванні графіки.

Інтеграція об'єктів у середовищі Unity відбувається через компонентну модель, де кожен елемент гри є об'єктом, до якого можна прикріпити відповідні скрипти поведінки. Це дозволяє розділити візуальну частину (дизайн рівнів, кольори меню) та програмну логіку (алгоритм змішування та перевірки градієнта), що відповідає принципам чистої архітектури програмного забезпечення [22]. Такий підхід також спрощує масштабування проєкту, дозволяючи без значних змін у коді додавати нові механіки, рівні або елементи інтерфейсу. Крім того, компонентна структура сприяє повторному використанню коду та підвищує зручність тестування окремих функціональних модулів. Таким чином, обґрунтований вибір середовища Unity як базової платформи дозволяє зосередитися на розробці освітньої складової та якісній реалізації механік когнітивного розвитку користувача [13].

1.3. Обґрунтування вибору технологій (Unity, C#) та розробка концепції проєкту «Повернення кольорів»

Вибір програмного інструментарію для реалізації освітнього застосунку ґрунтується на необхідності створення гнучкої, кросплатформної та продуктивної системи, здатної забезпечити високу якість візуалізації та точність ігрових механік. У сучасних умовах розробки важливо не лише досягти коректної роботи функціоналу, але й забезпечити стабільність застосунку на різних типах пристроїв від мобільних телефонів до персональних комп'ютерів. Саме тому обраний інструмент повинен поєднувати в собі ефективність використання ресурсів, гнучкість налаштування та зручність розробки. Основним середовищем розробки було обрано інтерактивну платформу Unity, що зумовлено її широкими можливостями у сфері створення 2D-застосунків та потужною системою рендерингу графічного контенту. Даний рушій забезпечує стабільну роботу з графікою, включаючи підтримку спрайтів, шарів, анімацій та ефектів, що є критично важливим для коректного відображення кольорових градієнтів і плавних переходів між ними. Крім того, Unity дозволяє оптимізувати продуктивність застосунку завдяки гнучким налаштуванням рендерингу та керуванню ресурсами, що особливо актуально для мобільних платформ із обмеженими обчислювальними можливостями. Unity надає розробнику компонентно-орієнтовану архітектуру, яка дозволяє ефективно розділяти візуальне представлення об'єктів та їхню внутрішню логіку. Кожен ігровий об'єкт може містити набір компонентів, що відповідають за окремі аспекти його функціонування від відображення на екрані до обробки взаємодії з користувачем. Такий підхід значно спрощує структурування проєкту, підвищує читабельність коду та дозволяє легко масштабувати систему шляхом додавання нових функціональних модулів без суттєвого втручання в уже існуючу логіку. Особливого значення для даного проєкту має вбудована система Unity UI (Canvas), яка дозволяє реалізувати складні інтерфейси, зокрема ігрову сітку 6×7, із забезпеченням адаптивності під різні роздільні

здатності екранів. Використання механізмів якорів (anchors), масштабування та автоматичного вирівнювання елементів інтерфейсу дає змогу зберігати коректне відображення гри незалежно від формату екрану пристрою. Це є важливим аспектом для забезпечення комфортного користувацького досвіду. Окрім цього, використання даної платформи дозволяє зосередитися на проектуванні освітнього досвіду, використовуючи готові рішення для обробки подій введення, таких як перетягування об'єктів (Drag-and-Drop), натискання та взаємодія з елементами інтерфейсу. Це є критично важливим для реалізації механіки пазлів-градієнтів, де користувач безпосередньо взаємодіє з кольоровими елементами, змінюючи їх положення. Наявність вбудованих інструментів обробки подій значно скорочує час розробки та дозволяє приділити більше уваги педагогічним аспектам застосунку, таким як зворотний зв'язок, поступове ускладнення завдань і формування навичок сприйняття кольору.

В якості мови програмування обрано C#, яка є базовою для середовища Unity та відповідає сучасним стандартам об'єктно-орієнтованого проектування [12]. Вибір C# обумовлений високим рівнем безпеки коду, наявністю розгалужених бібліотек для роботи з масивами та колекціями, що необхідно для реалізації алгоритмів випадкового змішування плиток та подальшої перевірки правильності сформованого користувачем градієнта. Завдяки інтеграції з Unity API, мова C# дозволяє оперативно керувати властивостями матеріалів та кольорів ігрових об'єктів у режимі реального часу, забезпечуючи плавність візуальних переходів та динамічну зміну стану ігрового середовища залежно від успіхів гравця. Крім того, C# підтримує принципи модульності та повторного використання коду, що дозволяє ефективно організувати структуру проєкту та спростити подальшу підтримку і розширення функціоналу. Важливою перевагою є також автоматичне керування пам'яттю (garbage collection), яке зменшує ризик витоків ресурсів і підвищує стабільність роботи застосунку [32]. Розвинена екосистема інструментів, зокрема середовища розробки та налагодження (наприклад, Visual Studio),

забезпечує зручність написання, тестування та оптимізації коду. Окрім цього, використання строгої типізації сприяє зменшенню кількості помилок на етапі компіляції та підвищує надійність програмного продукту [37]. У сукупності ці особливості роблять C# ефективним інструментом для реалізації складної логіки освітнього застосунку та забезпечення його стабільної роботи.

Концепція проєкту «Повернення кольорів» базується на синергії наративного дизайну та когнітивного тренування, що дозволяє поєднати розважальний та освітній аспекти в єдиному інтерактивному досвіді. Ідея гри розгортається у вигаданому ігровому всесвіті, який за сюжетом втратив свої кольори, перетворившись на монохромне середовище, що символізує втрату емоційної глибини та сприйняття навколишнього світу. Такий художній прийом не лише формує атмосферу гри, але й підсилює мотивацію гравця до відновлення гармонії та «оживлення» віртуального простору. Головним завданням гравця є повернення хроматичної насиченості світу через проходження трьох послідовних рівнів, кожен з яких представляє собою логічну головоломку на основі колірних переходів. З кожним новим рівнем складність зростає за рахунок менш очевидних градієнтів, більшої кількості схожих відтінків та зменшення візуальних підказок, що стимулює поступовий розвиток когнітивних навичок користувача. Така прогресія складності сприяє формуванню ефекту навчання через практику та підтримує інтерес гравця протягом усього ігрового процесу. Основна механіка гри полягає у маніпулюванні плитками на сітці розмірністю 6×7 , що створює оптимальний баланс між складністю та зручністю взаємодії. На початку кожного рівня плитки, що утворюють певний градієнт, перемішуються випадковим чином за допомогою розробленого алгоритму, який гарантує збереження логічної розв'язності задачі. Гравець повинен відновити правильну послідовність кольорів, орієнтуючись на ледь помітні відмінності у відтінках, що стимулює розвиток зорової диференціації та уваги до деталей. Додатково, така механіка сприяє тренуванню концентрації, просторового мислення та здатності до аналізу візуальної інформації. Важливим аспектом є також створення

інтуїтивно зрозумілого інтерфейсу, який не перевантажує користувача зайвими елементами та дозволяє зосередитися безпосередньо на вирішенні завдання. У результаті, поєднання наративу та ігрових механік формує цілісний досвід, що не лише розважає, але й має виражений розвивальний потенціал.

Важливою особливістю концепції є система динамічного зворотного зв'язку, реалізована через трансформацію ігрового меню. Прогресія гравця напряму впливає на візуальний стан інтерфейсу: успішне завершення першого рівня ініціює часткове повернення кольорів у головне меню, створюючи відчуття реального впливу дій користувача на ігровий світ. Повне проходження всіх етапів завершується остаточним відновленням кольорової гами всесвіту та фіналізацією ігрової сесії. Такий підхід дозволяє перетворити процес тестування колірного сприйняття на захопливу подорож, де кожен технічний аспект від розміру сітки до алгоритму перевірки працює на досягнення загальної освітньої мети проєкту. Таким чином, обраний технологічний стек та розроблена концепція створюють цілісний фундамент для подальшої програмної реалізації застосунку.

РОЗДІЛ 2. СИСТЕМНИЙ АНАЛІЗ ТА ПРОЄКТУВАННЯ ІНТЕРАКТИВНОГО ЗАСТОСУНКУ

2.1. Архітектура застосунку та розробка функціональної схеми ігрових рівнів

Архітектура освітнього застосунку побудована за модульним принципом, що передбачає чітке розділення системи на незалежні функціональні блоки, кожен із яких виконує окрему роль у загальній структурі програми. Центральним елементом такої архітектури є сцена логічно завершений модуль, що відповідає за конкретний етап взаємодії користувача із застосунком (головне меню, меню вибору рівнів, ігрові рівні, фінальна сцена) [15]. Такий підхід забезпечує зручність навігації, спрощує процес тестування та дозволяє легко масштабувати проєкт шляхом додавання нових сцен або модифікації існуючих без порушення цілісності системи [21].

Основу архітектури ігрової механіки складає система префабів (заздалегідь налаштованих шаблонів об'єктів), яка є важливим інструментом середовища Unity. Використання префабів дозволяє стандартизувати створення ігрових елементів, забезпечуючи їхню однакову поведінку та структуру, а також значно спрощує процес генерації ігрового поля під час завантаження сцени. Завдяки цьому ігрова сітка формується динамічно, що дає змогу легко змінювати її параметри (розмір, конфігурацію, складність) без необхідності ручного редагування кожного елемента.

Функціональна одиниця ігрового поля плитка є складним об'єктом, що складається з декількох взаємопов'язаних архітектурних компонентів, кожен із яких виконує спеціалізовану функцію:

- **Колайдер**, який забезпечує реєстрацію взаємодій з курсором миші або сенсорним введенням, дозволяючи визначати вибір, натискання та переміщення об'єкта

- **Скрипт параметрів**, що відповідає за збереження та оновлення стану об'єкта, включаючи початкову та поточну позиції, статус вибору для переміщення, а також параметри анімації та взаємодії
- **Візуальний скрипт**, що реалізує механізми зворотного зв'язку через зміну розміру, кольору або інших візуальних характеристик об'єкта при наведенні або виборі, що підвищує інтуїтивність взаємодії для користувача.

Додатково, така модульна організація дозволяє легко розширювати функціональність кожної плитки, наприклад, шляхом додавання нових компонентів для реалізації ефектів, звукового супроводу або більш складної логіки взаємодії. Взаємодія між компонентами реалізується через чітко визначені інтерфейси, що зменшує залежність між частинами системи та підвищує її гнучкість. У результаті, обрана архітектура забезпечує не лише стабільну роботу застосунку, але й створює основу для його подальшого розвитку та вдосконалення.

Функціональна схема ігрового рівня реалізується через послідовність автоматизованих етапів, які забезпечують цілісність ігрового процесу та коректну взаємодію між користувачем і системою. Така поетапна організація дозволяє чітко структурувати логіку роботи застосунку та спростити її подальше розширення й оптимізацію.

1. **Ініціалізація:** на цьому етапі відбувається завантаження графічних ресурсів (спрайтів), налаштування параметрів сцени та інстанціювання префабів плиток у межах сітки розмірністю 6×7 . Для ефективного маніпулювання даними всі об'єкти плиток зберігаються у двовимірному масиві, що дозволяє швидко отримувати доступ до їхніх позицій і властивостей. Додатково виконується початкове налаштування камери, освітлення та базових параметрів інтерфейсу, що забезпечує коректне відображення ігрового поля на різних пристроях.
2. **Формування вихідного стану:** після генерації ігрового поля активується алгоритм випадкового перемішування елементів, який

змінює початковий порядок плиток, зберігаючи при цьому можливість логічного відновлення правильного градієнта. Частина плиток залишається фіксованою і виконує роль візуальних орієнтирів (підказок), що допомагають гравцю визначити напрямок правильного розташування кольорів. Такий підхід забезпечує баланс між складністю та доступністю, запобігаючи надмірній фрустрації користувача.

3. **Ігровий цикл (Gameplay loop):** на цьому етапі реалізується безпосередня взаємодія гравця з інтерфейсом через механіку обміну позиціями плиток. Пересування здійснюється шляхом послідовного вибору двох об'єктів, після чого система виконує їх обмін, оновлюючи координати як у візуальному просторі, так і в керуючому масиві даних. Паралельно можуть запускатися анімації переміщення та візуальні ефекти, що забезпечують зворотний зв'язок і покращують сприйняття дій користувача. Ігровий цикл працює безперервно до моменту досягнення цільового стану або виходу гравця з рівня.
4. **Верифікація стану:** після кожної дії користувача викликається метод перевірки цілісності градієнта, який визначає, чи правильно сформовано послідовність кольорів. Залежно від складності рівня (перпендикулярні, діагональні або лінійні градієнти), система застосовує різні алгоритми перевірки, зокрема порівняння координат за осями або аналіз належності плиток до певних колірних груп. У разі успішного завершення рівня активується відповідний сценарій переходу до наступного етапу гри.

Загальна архітектура застосунку також передбачає ієрархічну структуру меню, яка забезпечує логічну організацію навігації між різними функціональними модулями. Головне меню надає доступ до налаштувань звуку через глобальний аудіоміксер, а також до переходів між сценами гри. Система прогресії рівнів реалізована через механізм блокування контенту: доступ до наступної головоломки відкривається лише після успішного

завершення поточної, що стимулює послідовне проходження та підтримує мотивацію користувача.

Архітектурною особливістю застосунку є динамічна зміна візуального оформлення поступовий перехід фону від ахроматичного (сірого) до насиченого кольорового стану в міру просування гравця. Такий підхід виконує не лише естетичну, але й концептуальну функцію, візуалізуючи прогрес користувача та підсилюючи основну ідею проєкту «повернення кольорів». Це створює додатковий емоційний зв'язок між гравцем і ігровим світом, підвищуючи загальну залученість у процес навчання.

Таблиця 2.1

Склад ігрових сцен застосунку та їх функціональне призначення

Назва сцени	Основні компоненти та об'єкти	Функціональне призначення
Main Menu	Canvas, Main Mixer, UI Buttons, Анімований заголовок	Забезпечення навігації, доступ до налаштувань звуку та вихід із гри.
Level Selection	UI Grid, Level Buttons, Logic Progress Manager	Вибір доступних рівнів та візуалізація ігрового прогресу через зміну фону.
Game Level (1, 2, 3)	Grid Generator, Tile Prefabs, Game Logic Manager	Генерація ігрового поля 6x7, обробка механіки пересування плиток та перевірка градієнта.
Final Scene	UI Text, Exit Button	Відображення результатів проходження та завершення роботи із застосунком.

Джерело: розроблено автором

2.2. Проєктування алгоритму випадкового змішування плиток та логіки перевірки градієнта

Математичний опис представлення кольору та механізми формування градієнта у проєкті базуються на використанні векторної моделі RGBA, яка застосовується в середовищі Unity. У цій моделі кожен колір задається як набір із чотирьох чисел червоного, зеленого, синього та прозорості [14]. Іншими словами, кожна плитка має власний “набір параметрів кольору”, який можна уявити як чотири значення: скільки в ній червоного, зеленого, синього і наскільки вона прозора. Усі ці значення знаходяться в межах від 0 до 1, де 0 означає повну відсутність компонента, а 1 його максимальну інтенсивність. Такий спосіб представлення є зручним, оскільки дозволяє точно керувати кольорами та плавно їх змінювати під час роботи програми. Завдяки такій векторній моделі система може виконувати точні обчислення кольору для кожної плитки окремо. Це особливо важливо для відображення градієнтів, де різниця між сусідніми відтінками може бути дуже незначною. Кожна плитка фактично “знає” свій точний колір, що дозволяє не лише правильно відображати її на екрані, але й використовувати ці дані для перевірки правильності розташування.

Ключовим механізмом створення градієнта є лінійна інтерполяція, тобто поступовий перехід від одного кольору до іншого [19]. Простими словами, система бере два крайні кольори початковий і кінцевий і обчислює проміжні значення між ними. Параметр t визначає, наскільки близько конкретна плитка знаходиться до початку або до кінця градієнта: якщо t дорівнює 0 це початковий колір, якщо 1 кінцевий, а всі значення між ними створюють плавний перехід. Таким чином, кожна плитка отримує свій унікальний відтінок залежно від її позиції в сітці. У межах розробленого проєкту цей підхід застосовується для поділу градієнта на окремі частини відповідно до структури ігрового поля 6×7 [24]. Кожна плитка отримує свій унікальний “кольоровий ідентифікатор”, який визначає її правильне місце у загальному переході кольорів. Навіть після випадкового перемішування всі плитки

зберігають ці значення, що дозволяє системі точно відстежувати їх правильне розташування.

Саме порівняння цих кольорових значень або їхніх позицій у структурі даних лежить в основі алгоритму перевірки. Коли гравець відновлює правильний порядок, система визначає, що всі плитки знову відповідають своїм місцям у градієнті. Такий підхід забезпечує високу точність перевірки та робить ігровий процес логічним і зрозумілим, оскільки базується безпосередньо на властивостях кольору, які сприймає користувач.

Математична модель розробленої головоломки базується на матричному представленні ігрового поля, де кожен елемент (плитка) описується як об'єкт у двовимірному масиві розмірністю 6 на 7. Такий підхід дозволяє зручно організувати зберігання даних і швидко виконувати операції доступу до будь-якої плитки за її координатами. Кожна плитка в системі має два набори координат: початкові (цільові), які визначають її правильне положення у градієнті, та поточні, що змінюються під час гри внаслідок дій користувача. Це дає змогу постійно порівнювати фактичний стан ігрового поля з еталонним і визначати прогрес гравця. Процес підготовки ігрового рівня починається з алгоритму випадкового змішування плиток. Його логіка полягає в послідовному обході всіх елементів масиву, де для кожної нефіксованої плитки обирається нова випадкова позиція. При цьому важливо, що частина плиток залишається незмінною вони виконують роль опорних точок, які допомагають гравцеві орієнтуватися у правильному напрямку градієнта. З математичної точки зору це спрощує задачу, оскільки зменшує кількість можливих варіантів розташування і створює базову структуру для відновлення послідовності кольорів. Зміна позицій реалізується через механіку обміну: коли гравець обирає дві плитки, система просто міняє їх місцями як у масиві даних, так і на екрані. Це гарантує узгодженість між логічною моделлю та візуальним відображенням гри.

Особливої уваги потребує модель перевірки умови перемоги. Для першого рівня, де градієнт має просту вертикальну структуру,

використовується перевірка по вертикалі: система аналізує, чи знаходиться кожна плитка саме в тому рядку, де вона повинна бути. Простими словами, для кожної плитки порівнюється її правильна позиція з поточною, і якщо всі плитки стоять на своїх місцях, рівень вважається завершеним. Тобто умова перемоги виконується лише тоді, коли жодна плитка не відрізняється від свого правильного положення.

Візуалізація стану ігрового поля для першого рівня:

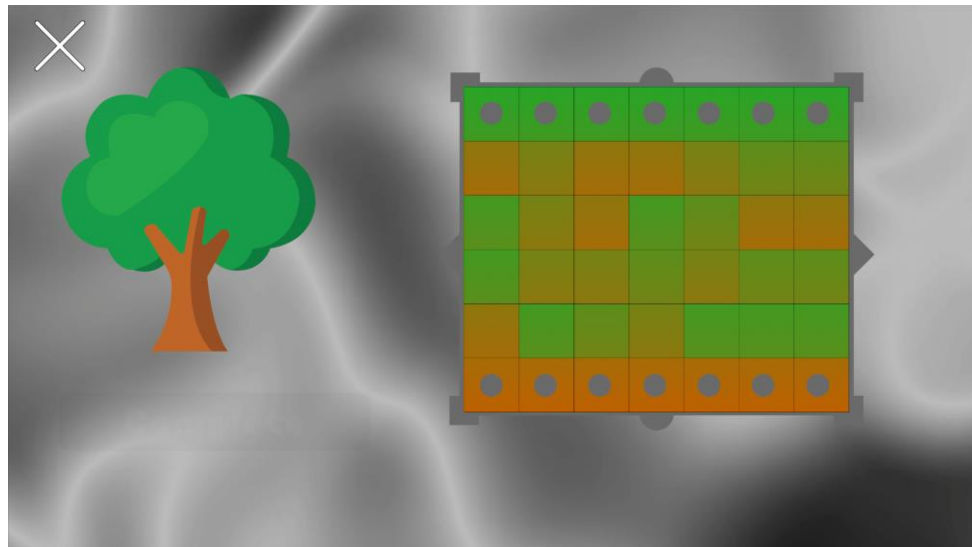


Рис. 2.1 Початковий стан (перемішані плитки)

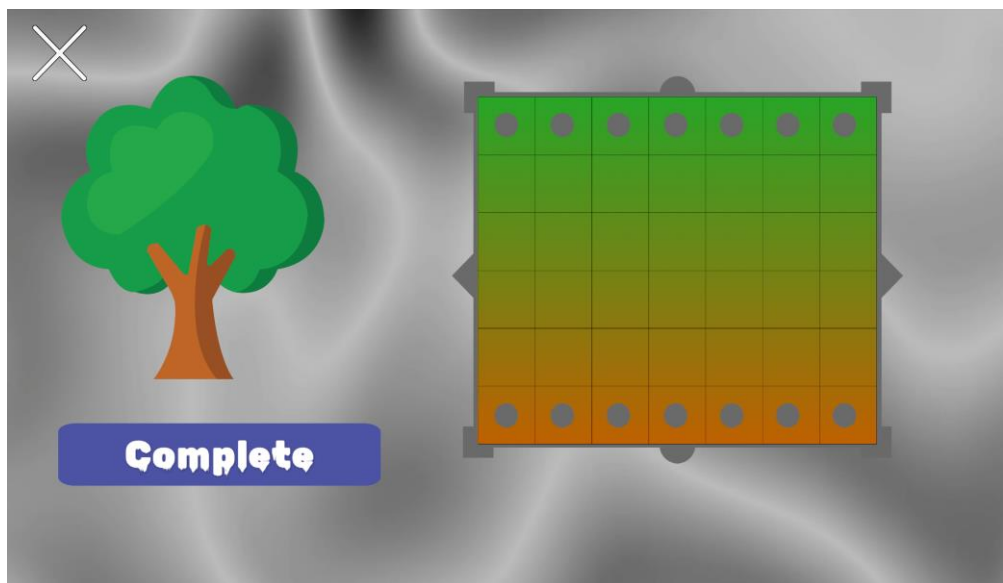


Рис. 2.2 Фінальний стан (правильний вертикальний градієнт)

Для другого та третього рівнів, де градієнти мають складнішу структуру (наприклад, діагональну або комбіновану), застосовується інший підхід групування плиток за кольором. У цьому випадку кожна плитка належить до певної “кольорової групи”, і система перевіряє, чи знаходиться вона в межах правильної зони, а не на конкретній позиції. Це означає, що плитки з однаковим або дуже схожим кольором можуть змінюватися місцями між собою без порушення правил. Такий підхід робить гру більш гнучкою і справедливою, оскільки враховує візуальну подібність елементів і не змушує гравця до надмірно точних перестановок там, де це не має суттєвого значення.

Реалізація складного градієнта для другого рівня:

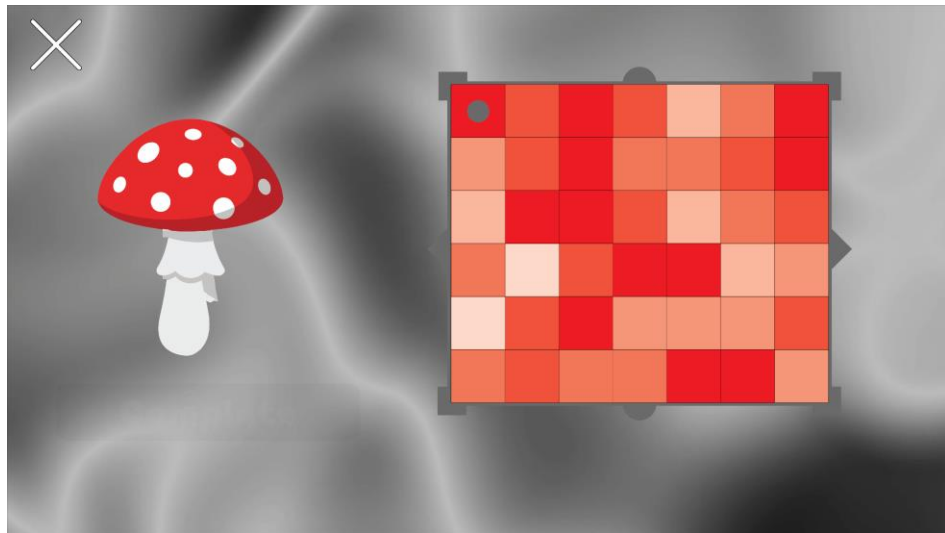


Рис. 2.3 Стан після випадкового змішування

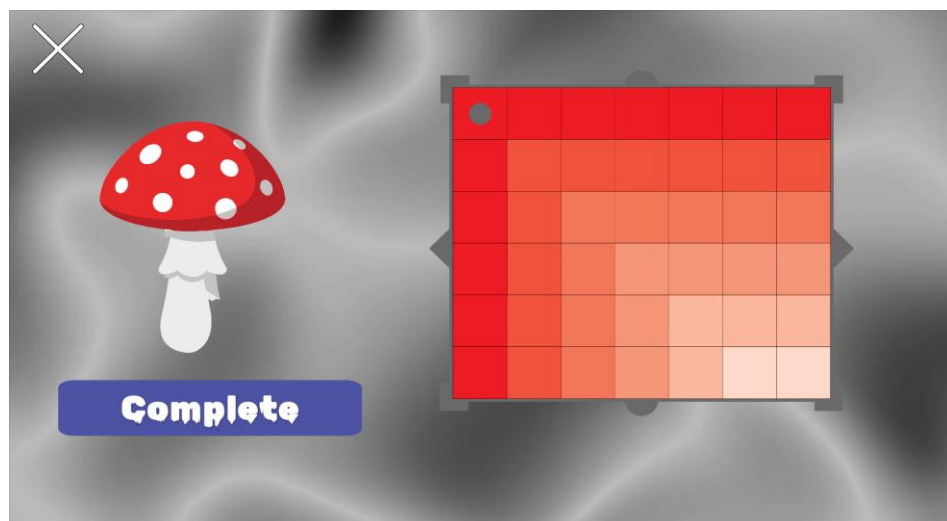


Рис. 2.4 Відновлений градієнт за кольоровими групами

У результаті, запропонована математична модель забезпечує баланс між строгою логікою перевірки та зручністю ігрового процесу. Вона гарантує коректність визначення результату незалежно від способу розв'язання, обраного гравцем, і водночас підтримує навчальну мету застосунку розвиток здатності розрізняти кольори та аналізувати їхні переходи.

Для детального опису технічної реалізації механіки взаємодії користувача з ігровим середовищем необхідно розглянути алгоритм обміну об'єктів (Swap), який є фундаментом усього ігрового процесу. Саме ця механіка визначає, яким чином гравець впливає на стан ігрового поля та досягає поставленої мети. Пересування плиток реалізовано через логіку послідовного вибору двох елементів сітки, після чого відбувається синхронна зміна їхніх позицій як у структурі даних (двовимірному масиві), так і у візуальному просторі сцени. Такий підхід забезпечує узгодженість між внутрішнім станом системи та тим, що бачить користувач на екрані. Процес зміни положення ігрових об'єктів ініціюється під час взаємодії гравця з плитками через колайдери за допомогою курсора миші або сенсорного введення. Алгоритм працює за подієвою моделлю, тобто реагує на дії користувача, і складається з кількох послідовних етапів. Спочатку відбувається ідентифікація та вибір об'єктів: при першому натисканні система запам'ятовує обрану плитку (Плитку А), зберігає її позицію в масиві та активує візуальний ефект виділення. При другому натисканні на іншу плитку (Плитку Б) система фіксує другий об'єкт і готується до виконання обміну.

Далі виконується логічний обмін у структурі даних. Це означає, що система спочатку міняє місцями не самі об'єкти на екрані, а їхні позиції у внутрішньому масиві, який зберігає інформацію про розташування плиток. Для цього використовується простий і надійний підхід із тимчасовою змінною: значення однієї позиції тимчасово зберігається, щоб уникнути втрати даних, після чого обидві плитки міняються місцями. Такий метод гарантує коректність операції навіть при багаторазових повтореннях. Після оновлення даних відбувається синхронізація візуального стану. На цьому етапі плитки

плавно переміщуються до нових позицій у просторі сцени відповідно до оновлених координат у масиві. Зазвичай це супроводжується анімацією руху, що робить процес більш зрозумілим і приємним для користувача, а також забезпечує чіткий зворотний зв'язок щодо виконаної дії.

З математичної точки зору, цей алгоритм гарантує, що кількість об'єктів у системі завжди залишається сталою, а їхній стан повністю визначеним у кожний момент часу. Іншими словами, кожна плитка має лише одне конкретне місце в структурі даних, і після обміну всі зв'язки залишаються коректними. Такий підхід дозволяє чітко відокремити логіку обчислень від візуального відображення, що є важливим принципом побудови надійних програмних систем [20]. Після кожного завершеного обміну система автоматично запускає перевірку стану ігрового поля, визначаючи, чи досягнуто правильного розташування плиток і чи виконано умову перемоги.

Важливою складовою надійності будь-якого програмного продукту, а особливо ігрового застосунку з інтерактивною логікою, є обробка виняткових ситуацій (так званих «Edge Cases»). Для забезпечення стабільності ігрового процесу та запобігання логічним помилкам у структурі даних масиву, було реалізовано декілька рівнів програмних обмежень. Першим і ключовим рівнем захисту є логічна перевірка статусу плитки перед початком будь-якої маніпуляції. Оскільки деякі плитки за архітектурою рівня залишаються фіксованими для орієнтації гравця у градієнті, до їхнього об'єкта було додано властивість `isFixed`. У коді це реалізовано через умовну конструкцію `if (isFixed)`, яка блокує можливість вибору або переміщення таких елементів. Якщо гравець намагається взаємодіяти з опорною плиткою, алгоритм ігнорує запит, що дозволяє зберегти математичну структуру цільового градієнта незмінною протягом всієї сесії. Наступним аспектом безпеки є контроль за послідовністю дій користувача при виборі об'єктів. Виникає ситуація, коли гравець може випадково або навмисно натиснути на одну й ту саму плитку двічі. Без належної обробки це могло б призвести до спроби алгоритму виконати обмін плитки «самої з собою», що потенційно створює програмні

конфлікти або непотрібні виклики методів оновлення UI. Для запобігання цьому було впроваджено логіку обмеження вибору: якщо ідентифікатор другої обраної плитки збігається з ідентифікатором першої, система автоматично скидає вибір (обнуляє посилання на об'єкт) або просто ігнорує друге натискання. Це забезпечує коректну роботу механіки «swar» лише для двох різних елементів сітки, підтримуючи цілісність логічних операцій у масиві.

Окрему увагу приділено синхронізації візуальних ефектів із внутрішньою логікою програми. Оскільки пересування плиток супроводжується анімацією зміни позиції, виникає ризик виникнення десинхронізації, якщо гравець ініціює новий обмін до завершення попереднього. Для вирішення цієї проблеми було реалізовано систему анімаційних затримок (input lock). Допоки скрипт анімації виконує переміщення об'єктів у просторі сцени, встановлюється логічний прапор «IsBusy», який тимчасово блокує реєстрацію нових подій натискання. Це гарантує, що наступний цикл обміну та перевірки градієнта почнеться лише тоді, коли всі плитки займуть свої фактичні позиції, а двовимірний масив буде повністю оновлений. Такий підхід не лише робить ігровий процес плавнішим, а й забезпечує стовідсоткову відповідність між тим, що бачить користувач, і тим, як ці дані обробляються алгоритмом перевірки умови перемоги. Додатково це дозволяє уникнути логічних помилок, пов'язаних із накладанням кількох дій одночасно, що могло б призвести до некоректного стану ігрового поля. Використання такого механізму також позитивно впливає на стабільність роботи застосунку, особливо на пристроях із різною продуктивністю, де швидкість виконання анімацій може відрізнятися. У результаті система зберігає передбачуваність поведінки, що є критично важливим для забезпечення якісного користувацького досвіду.

2.3. Моделювання системи прогресії гравця та динамічної зміни візуального середовища

Система прогресії в розробленому застосунку побудована на моделі послідовного розблокування контенту, що забезпечує кероване навчання користувача та підтримку ігрового інтересу протягом усього ігрового процесу [33]. Такий підхід дозволяє уникнути перевантаження гравця складними завданнями на початкових етапах і поступово вводити нові механіки та рівні складності. Логіка прогресії інтегрована безпосередньо в архітектуру меню вибору рівнів, яке складається з трьох інтерактивних елементів (кнопок), кожен з яких відповідає окремому етапу проходження гри.

На початковому етапі ігрового циклу доступним для обрання є лише перший рівень, тоді як наступні два залишаються заблокованими для взаємодії. Це створює чітку послідовність проходження та стимулює користувача завершити поточне завдання для відкриття нових можливостей. Технічно така поведінка реалізується через систему логічних прапорів, які фіксують факт успішного завершення рівня. Після проходження першої головоломки система автоматично оновлює стан інтерфейсу, відкриваючи доступ до наступного рівня, і аналогічна логіка застосовується для переходу до третього етапу. Це дозволяє не лише контролювати прогрес, але й зберігати його між сесіями гри, забезпечуючи безперервність користувацького досвіду.

Технічно це реалізовано через перевірку прапора успішного завершення попередньої головоломки. Як продемонстровано на рис. 2.5, рис. 2.6, стан інтерфейсу та ігрового фону змінюється динамічно: на початковому етапі (рис. 2.5) світ представлений у монохромному вигляді, що підкреслює відсутність кольорів у ігровому середовищі, тоді як після успішного проходження першого рівня (рис. 2.6) система ініціює часткове повернення кольорової палітри. Це не лише відображає технічну зміну стану сцени, але й слугує наочним індикатором прогресу гравця, підсилюючи візуальний зв'язок між виконаними діями та їх результатом у межах ігрового світу.

Візуалізація динамічної зміни візуального середовища в меню вибору рівнів:



Рис. 2.5 Початковий стан (Рівень 0) із сірим фоном



Рис. 2.6 Стан після завершення першого рівня з частковим офарбленням.

Така структура забезпечує поступове зростання складності когнітивних завдань: від простих лінійних градієнтів до складніших конфігурацій, включаючи діагональні та комбіновані переходи кольорів. Завдяки цьому гравець поступово адаптується до механіки гри, розвиваючи навички аналізу кольорів і просторового мислення. Важливо, що складність підвищується не

різко, а поетапно, що позитивно впливає на мотивацію та зменшує ймовірність втрати інтересу.

Центральною особливістю візуального моделювання є динамічна трансформація ігрового середовища залежно від прогресу користувача. Відповідно до концепції «світу, що втратив кольори», початковий стан інтерфейсу та фону є ахроматичним (сірим), що створює відчуття порожнечі та незавершеності. У міру проходження рівнів система поступово змінює кольорову палітру, додаючи насиченість і яскравість, що візуально відображає досягнення гравця. Така трансформація виконує дві важливі функції: з одного боку, вона підсилює наратив, демонструючи відновлення світу, а з іншого створює психологічний ефект винагороди, підкріплюючи відчуття прогресу та значущості виконаних дій. Додатковими елементами візуальної динаміки є анімовані компоненти головного меню, які роблять інтерфейс більш «живим» і привабливим. Наприклад, букви назви гри здійснюють плавні коливання по вертикалі, що створює відчуття руху та динаміки навіть у статичних сценах. Це досягається за рахунок використання періодичних функцій, які задають циклічний характер руху. Такі деталі, хоча й не впливають безпосередньо на геймплей, суттєво покращують загальне сприйняття продукту [30].

Користувацький досвід також доповнюється аудіо-супроводом, який відіграє важливу роль у формуванні атмосфери гри. Кожен рівень супроводжується унікальною фоновою музикою, що відповідає його настрою та складності, а взаємодія з елементами інтерфейсу підкріплюється звуковими ефектами. Керування звуковими потоками здійснюється через головний аудіоміксер, який дозволяє гравцеві налаштовувати гучність відповідно до власних уподобань у меню налаштувань. Це забезпечує додаткову гнучкість та комфорт використання застосунку. Завершення системи прогресії відбувається після проходження останньої головоломки, коли гравець автоматично переходить до фінальної сцени. У цьому модулі відображається підсумкове повідомлення, яке символізує повне відновлення кольорів у світі гри та досягнення основної мети. Такий фінал виконує роль логічного

завершення як ігрового, так і освітнього процесу. Після цього користувач отримує можливість завершити сеанс або повернутися до гри для повторного проходження.

Таким чином, змодельована система прогресії забезпечує цілісний і продуманий ігровий досвід, у якому кожен елемент від логіки розблокування рівнів до візуальних і звукових ефектів працює на підтримку мотивації гравця, його залученості та ефективного засвоєння навчального матеріалу. Взаємозв'язок між ігровими механіками та освітньою складовою дозволяє користувачу не лише виконувати завдання, але й поступово розвивати когнітивні навички у природному, інтерактивному середовищі [7]. Крім того, система стимулює повторне проходження рівнів, що сприяє закріпленню отриманих знань і підвищенню точності сприйняття кольорів. Важливим аспектом є також поступове ускладнення завдань, яке підтримує баланс між викликом і доступністю, не допускаючи перевантаження користувача. Інтеграція візуальних і аудіо-ефектів підсилює емоційне сприйняття гри, формуючи позитивний досвід взаємодії та підвищуючи рівень занурення. Система прогресії також виконує роль орієнтира для користувача, чітко демонструючи його досягнення та стимулюючи рухатися далі. У результаті формується стійка мотивація до завершення всіх рівнів, що є важливим як з точки зору геймдизайну, так і з позиції досягнення освітніх цілей застосунку.

РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ОСВІТНЬОГО ЗАСТОСУНКУ В СЕРЕДОВИЩІ UNITY

3.1. Програмна реалізація ігрових механік та інтерфейсу користувача засобами C#

Практична реалізація освітнього застосунку «Повернення кольорів» здійснювалася в інтерактивному середовищі Unity з використанням мови програмування C#. Програмна архітектура проєкту базується на об'єктно-орієнтованому підході, де кожен ігровий аспект керується відповідним класом-скриптом. Така організація коду дозволяє чітко розподілити відповідальність між різними компонентами системи та спрощує процес підтримки й розширення функціоналу. Центральним модулем системи виступає клас GridManager, який відповідає за генерацію ігрового поля, маніпуляцію об'єктами плиток та перевірку завершеності завдання. GridManager координує взаємодію між іншими компонентами, забезпечуючи узгоджену роботу всієї системи. Наприклад, він контролює процес ініціалізації рівня, запуск алгоритмів перемішування та обробку дій користувача. Такий підхід дозволяє зосередити ключову логіку в одному місці, що підвищує читабельність і керованість коду. Окрім цього, використання окремих скриптів для різних функцій (анімації, взаємодії, перевірки) сприяє модульності системи. У результаті розробник отримує гнучку архітектуру, яку легко адаптувати під нові вимоги або доповнювати новими можливостями без значних змін у вже реалізованій логіці.

Реалізація основної ігрової механіки сітки розмірністю 6x7 починається з динамічного завантаження графічних ресурсів (спрайтів) із папки Resources залежно від поточної сцени [36]. Метод InitTiles здійснює інстанціювання префабів плиток, присвоюючи кожній з них відповідний спрайт та координати у двовимірному масиві tiles[width, height]. Після генерації поля спрацьовує алгоритм перемішування ShuffleTiles, реалізований на основі алгоритму Фішера-Єйтса. Програмно забезпечено ігнорування плиток, назва яких

містить ідентифікатор «fixed», що дозволяє зберігати опорні точки градієнта незмінними для підказки гравцеві.

Взаємодія користувача з плитками реалізована через обробку подій миші. Клас `Tile` містить логіку візуального відгуку: при наведенні курсора спрацьовує метод `OnMouseEnter`, який плавно збільшує масштаб об'єкта за допомогою функції `Vector3.Lerp`, забезпечуючи плавність анімації. Механіка обміну позицій (`SwapTiles`) працює шляхом збереження посилання на першу обрану плитку та подальшого переписування її індексів у масиві при обранні другої плитки. Після кожного успішного обміну викликається метод `IsComplete`, який порівнює поточний стан масиву із цільовим градієнтом, враховуючи специфіку кожного рівня: вертикальну перевірку по осі `Y` для першого рівня або порівняння за кольірними групами для наступних етапів.

Програмна реалізація інтерфейсу користувача та системи прогресії базується на використанні класів `MenuManager` та `LevelSelectManager`. Система збереження стану гри реалізована за допомогою класу `PlayerPrefs`, який фіксує індекс останнього пройденого рівня (`LastLevel`). На основі цих даних клас `LevelSelectManager` динамічно оновлює стан кнопок вибору рівнів та керує візуальними ефектами офарблення фону, активуючи відповідні об'єкти `background1Of3`, `background2Of3` або `backgroundFull`. Анімація текстових елементів у головному меню реалізована в класі `FloatingLetter`, де вертикальне переміщення символів розраховується через тригонометричну функцію `Mathf.Sin` з додаванням випадкового зсуву фази (`randomOffset`) для створення ефекту хвилі.

Налаштування аудіопараметрів реалізовано через клас `AudioSettings`, який взаємодіє з `AudioMixer` середовища `Unity`. Програмно забезпечено логарифмічне регулювання гучності для музики та звукових ефектів, результати якого також зберігаються в `PlayerPrefs` для підтримки налаштувань при наступних запусках застосунку [28]. Така комплексна програмна реалізація забезпечує стабільне функціонування всіх заявлених механік та високий рівень інтерактивності освітнього продукту.

Для детального опису програмно-апаратної взаємодії всередині ігрового об'єкта необхідно розглянути механізми взаємодії авторського скрипта Tile із вбудованими компонентами середовища Unity. Ця взаємодія забезпечує як візуальне відображення ігрових елементів, так і обробку користувацького введення.

Взаємодія скрипта Tile з компонентами SpriteRenderer та BoxCollider2D. Ефективне функціонування ігрової плитки як самостійної одиниці базується на тісній інтеграції скрипта Tile з компонентом SpriteRenderer. У процесі життєвого циклу об'єкта, під час виклику методу Awake(), скрипт отримує посилання на цей компонент за допомогою методу GetComponent<SpriteRenderer>(). Це дозволяє програмно маніпулювати візуальними властивостями об'єкта безпосередньо під час виконання програми. Зокрема, скрипт звертається до властивості sprite.name, щоб ідентифікувати тип плитки: якщо назва містить маркер «fixed», скрипт блокує можливість вибору та переміщення даного елемента, що є критичним для збереження опорних точок градієнта. Взаємодія з компонентом BoxCollider2D реалізована через подієву модель Unity. Хоча скрипт не звертається до нього безпосередньо, його наявність є необхідною для реєстрації взаємодії з курсором. Використовуються вбудовані методи OnMouseDown(), OnMouseEnter() та OnMouseExit(), які викликаються лише тоді, коли курсор (raycast) перетинає межі коллайдера.

Результатом цієї взаємодії є динамічний візуальний відгук, що реалізується через зміну параметрів компонента Transform. При спрацюванні події OnMouseEnter() скрипт обчислює новий масштаб, збільшуючи початковий розмір на певний коефіцієнт. У методі Update() ці зміни плавно застосовуються за допомогою лінійної інтерполяції, створюючи ефект «оживання» плитки під курсором. Таким чином, BoxCollider2D виконує роль сенсора введення, SpriteRenderer відповідає за візуалізацію, а скрипт Tile об'єднує їх у єдину інтерактивну систему.

Таблиця 3.1

Функціональне призначення основних програмних модулів (C#-скриптів) застосунку

Назва класу (скрипта)	Основні функції та задачі	Використані компоненти/API Unity
GridManager	Генерація сітки 6x7, випадкове перемішування плиток, обмін об'єктів місцями та перевірка умови перемоги (Win Condition).	Resources, Instantiate, SceneManager, List, Array.
Tile	Керування станом окремої плитки: обробка натискань, зміна масштабу при наведенні та візуалізація вибору.	OnMouseDown, OnMouseEnter, Vector3.Lerp, BoxCollider2D.
LevelSelectManager	Логіка розблокування рівнів на основі прогресу та динамічна зміна візуального фону (від сірого до кольорового).	PlayerPrefs, Animator, SetActive, Sprite.
MenuManager	Глобальне керування сценами (запуск, вихід, завершення рівня) та збереження прогресу гравця.	SceneManager.LoadScene, PlayerPrefs.
AudioSettings	Інтерфейс налаштувань звуку: зв'язок між UI-слайдерами та міксером, логарифмічне регулювання гучності.	AudioMixer, Slider, Mathf.Log10.
FloatingLetter	Програмна анімація символів назви гри за допомогою синусоїдальної функції для створення ефекту «хвилі».	Mathf.Sin, Time.time, transform.localPosition.
SoundManager	Забезпечення звукового супроводу взаємодії з інтерфейсо та підтримка цілісності об'єкта між сценами.	AudioSource.PlayOneShot, DontDestroyOnLoad.

Джерело: розроблено автором

3.2. Тестування працездатності алгоритмів та аналіз результатів розробки.

Метою тестування розробленого застосунку «Повернення кольорів» є перевірка відповідності фактичних результатів роботи програми поставленим завданням, а також виявлення та усунення потенційних помилок у логіці ігрових механік. Комплексний підхід до тестування дозволяє оцінити не лише коректність виконання окремих функцій, але й загальну стабільність і зручність використання застосунку. Процес верифікації охоплював функціональне тестування алгоритмів, тестування користувацького інтерфейсу (UI), перевірку системи збереження прогресу, а також аналіз поведінки застосунку в різних сценаріях взаємодії користувача.

Основним об'єктом функціонального тестування став алгоритм перевірки умови перемоги `IsComplete`, інтегрований у клас `GridManager`. Особлива увага приділялася коректності його роботи після кожної дії гравця, щоб гарантувати своєчасне та точне визначення завершення рівня. Перевірка здійснювалася для кожного з трьох рівнів окремо, з урахуванням різної логіки формування та аналізу градієнтів.

На першому рівні підтверджено коректність порівняння поточних і початкових позицій плиток по вертикалі, тобто система правильно визначає, чи знаходиться кожна плитка у своєму рядку. Було встановлено, що навіть при великій кількості перестановок алгоритм стабільно фіксує момент досягнення правильного розташування. На другому та третьому рівнях протестовано логіку групування за кольорами, де перевірка здійснюється не за точною позицією, а за належністю плитки до відповідної колірної зони. Це дозволило гравцеві успішно завершувати рівень навіть у випадках, коли плитки з однаковими або дуже схожими відтінками змінені місцями всередині однієї групи.

Додатково було перевірено реакцію системи на граничні випадки, зокрема швидке багаторазове натискання, неповні дії або спроби взаємодії під

час анімацій. Результати показали, що реалізовані механізми синхронізації та блокування введення працюють коректно і запобігають виникненню логічних помилок. Таким чином, проведене тестування підтвердило надійність ключових алгоритмів та їх відповідність поставленим вимогам до функціонування освітнього застосунку [35].

Для систематизації результатів було проведено серію тест-кейсів, результати яких наведено у таблиці 3.2

Таблиця 3.2

Результати функціонального тестування ключових механік застосунку

Об'єкт тестування	Очікуваний результат	Фактичний результат	Статус
Генерація сітки 6x7	Створення 42 об'єктів плиток із відповідними спрайтами.	Поле генерується згідно з поточною сценою.	Пройдено
Алгоритм Shuffle	Випадкове змішування плиток, окрім фіксованих (isFixed).	Плитки з назвою "fixed" залишаються на місцях.	Пройдено
Механіка Swap	Обмін позицій двох обраних плиток у масиві та на екрані.	Позиції в масиві та Transform синхронізовані.	Пройдено
Система прогресії	Розблокування Рівня 2 після проходження Рівня 1.	Значення LastLevel у PlayerPrefs оновлюється коректно.	Пройдено
Аудіо-налаштування	Зміна гучності через слайдери в меню Settings.	AudioMixer плавно змінює рівень сигналу.	Пройдено

Джерело: розроблено автором

Аналіз результатів розробки свідчить про високу стабільність застосунку. Завдяки використанню анімаційних затримок (input lock) та перевірок на повторний вибір однієї плитки, вдалося уникнути критичних помилок десинхронізації даних. Тестування UI підтвердило, що візуальний відгук (збільшення масштабу плиток при наведенні) працює коректно завдяки скрипту Tile та компоненту BoxCollider2D, забезпечуючи інтуїтивно зрозуміле керування [29].

Окремо було проаналізовано систему динамічної зміни середовища. Тестування підтвердило, що активація об'єктів фону (background1Of3, backgroundFull) відбувається миттєво після повернення гравця в меню вибору рівнів, що створює цілісний наративний ефект відновлення кольорів світу. Таким чином, розроблений застосунок повністю відповідає технічним вимогам та готовий до впровадження як освітній інструмент для розвитку кольорового сприйняття.



Рис. 3.7 Розроблений логотип

Візуальна ідентифікація та символіка логотипу проєкту. Дизайн логотипу освітнього застосунку «Повернення кольорів» (рис. 3.7) розроблений як візуальна метафора центральної ігрової механіки та сюжетної лінії. Логотип поєднує в собі елементи класичного художнього інструментарію та сучасні

принципи цифрового дизайну, що підкреслює творчий характер процесу навчання.

Ключові компоненти композиції та їхнє значення:

Ахроматичний фон: Основою логотипу є коло сірого кольору, що символізує початковий стан ігрового всесвіту монохромний світ, позбавлений кольорів. Це створює необхідний візуальний контраст для центральних елементів.

Стилізований пензель: Зображення пензля, що розташоване по діагоналі, символізує роль гравця як активного суб'єкта, який своїми діями (вирішенням головоломок) відновлює колірну цілісність оточення. Чорно-біле виконання руків'я пензля підкреслює перехід від безбарвності до насиченості.

Градiєнтний мазок: Найважливішим елементом є вигнутий колірний слід, що виходить з-під пензля [39]. Він демонструє плавний перехід від холодного блакитного через жовтий та помаранчевий до насиченого червоного кольору. Цей мазок є прямою відсилкою до механіки ігрових рівнів, де гравець маніпулює плитками для створення ідентичних градієнтів у сітці 6x7.

Технічні та естетичні особливості. Логотип виконаний у мінімалістичному стилі з використанням чітких геометричних форм, що забезпечує його впізнаваність при масштабуванні в ігровому інтерфейсі. Використання градієнта в центрі композиції не лише привертає увагу, а й слугує технічною демонстрацією колірного простору, з яким працює користувач під час проходження рівнів. Динамічна форма мазка створює відчуття руху, що корелює з програмною анімацією літер та елементів меню, реалізованою в середовищі Unity.

Таким чином, логотип є не просто декоративним елементом, а візуальним вираженням концепції «оживлення» світу через розвиток когнітивних навичок кольорового сприйняття.

3.3. Перспективи подальшого розвитку та стратегія дистрибуції проєкту

Завершальним етапом розробки освітнього застосунку є визначення стратегії його виходу на ринок та формування плану подальшої модернізації, що дозволяє не лише представити продукт користувачам, але й забезпечити його довготривалий розвиток. Важливим аспектом цього етапу є вибір ефективних каналів дистрибуції, які забезпечать доступ до цільової аудиторії та сприятимуть популяризації застосунку. Враховуючи освітню спрямованість та ігрову природу проєкту «Повернення кольорів», доцільним є розміщення продукту на спеціалізованих платформах цифрової дистрибуції, таких як Steam, Itch.io та Epic Games Store.

Кожна з цих платформ має свої переваги: Steam забезпечує доступ до великої аудиторії та розвиненої системи просування, Itch.io є зручною платформою для інді-розробників і освітніх проєктів, а Epic Games Store пропонує вигідні умови розміщення та підтримку розробників. Розміщення на декількох платформах одночасно дозволяє охопити ширший сегмент користувачів та підвищити впізнаваність продукту.

Окрім дистрибуції, важливою складовою є план подальшої модернізації застосунку. Він може включати розширення кількості рівнів, впровадження нових типів головоломок, додавання адаптивної складності, а також інтеграцію додаткових освітніх елементів. Також перспективним є вдосконалення користувацького інтерфейсу, оптимізація продуктивності та адаптація під різні платформи, зокрема мобільні пристрої. Варто врахувати можливість збору зворотного зв'язку від користувачів після релізу, що дозволить виявити слабкі місця та визначити напрями подальшого вдосконалення. Таким чином, поєднання продуманої стратегії розповсюдження та системного підходу до модернізації створює основу для успішного розвитку освітнього застосунку та його ефективного використання в навчальному середовищі.

Для забезпечення максимального охоплення цільової аудиторії та отримання якісного зворотного зв'язку від спільноти, на початковому етапі доцільно дотримуватися стратегії безкоштовного розповсюдження (Free-to-Play) або встановлення мінімальної вартості продукту. Такий підхід знижує бар'єр входу для нових користувачів і дозволяє швидко сформувати базу гравців, що особливо важливо для освітніх проєктів. Залучення більшої кількості користувачів сприятиме накопиченню відгуків, які можна використовувати для аналізу поведінки гравців та виявлення недоліків у геймплеї або інтерфейсі. Отримані аналітичні дані, зокрема інформація про складність рівнів, частоту помилок та час проходження, можуть стати основою для подальшого вдосконалення ігрових алгоритмів і балансування механік. Це дозволяє поступово адаптувати продукт під реальні потреби аудиторії, підвищуючи його ефективність як освітнього інструменту.

Важливою складовою успіху є також активне просування проєкту на тематичних форумах, ігрових спільнотах та соціальних платформах, таких як Reddit, Discord або YouTube. Це сприяє формуванню активної спільноти навколо гри, де користувачі можуть обмінюватися досвідом, пропонувати ідеї та брати участь у розвитку проєкту. Взаємодія з аудиторією дозволяє підвищити рівень довіри до продукту та створити відчуття причетності до його розвитку. У довгостроковій перспективі це може позитивно вплинути на популярність застосунку та його подальшу монетизацію. Таким чином, поєднання доступної цінової стратегії та активної комунікації з користувачами створює сприятливі умови для успішного розвитку проєкту «Повернення кольорів».

Масштабування проєкту передбачає розширення його функціональних можливостей за кількома ключовими напрямками, що дозволить перетворити застосунок із базового прототипу на комплексний освітньо-ігровий продукт. Такий розвиток спрямований не лише на збільшення контенту, але й на поглиблення взаємодії користувача з ігровим середовищем, підвищення мотивації та розширення аудиторії.

Першим важливим напрямом є наративний та візуальний розвиток. Впровадження інтерактивної карти світу дозволить гравцеві самостійно обирати різні локації для відновлення кольорів, що значно урізноманітнить ігровий процес. Замість однієї фонові сцени проєкт може включати різноманітні біоми пустелі, міста, океани або гірські регіони, кожен із яких матиме власну кольорову палітру та унікальні типи градієнтів. Це не лише підвищить естетичну привабливість гри, але й розширить навчальний ефект за рахунок роботи з різними спектрами кольорів.

Другим напрямом є геймплейні інновації. Урізноманітнення механік можливе через додавання нових типів головоломок, які виходять за межі класичної перестановки плиток. Наприклад, можна впровадити рівні з обмеженням за часом або кількістю ходів, що змусить гравця приймати більш обдумані рішення. Також перспективним є введення комбінованих механік, де необхідно не лише правильно розташувати плитки, але й виконати додаткові умови, що підвищить складність і зробить процес більш захопливим.

Третім напрямом є розвиток соціальної взаємодії та аналітики. Впровадження системи післяігрової статистики дозволить користувачам аналізувати власні результати, зокрема кількість ходів або час проходження, та порівнювати їх із середніми показниками інших гравців. Це створює елемент змагальності та підвищує повторну зацікавленість у проходженні рівнів. Додатково можна реалізувати таблиці лідерів або систему досягнень, що стимулюватиме гравців до вдосконалення своїх навичок.

Четвертим напрямом є технічна експансія та локалізація. Завдяки кросплатформним можливостям Unity, застосунок може бути адаптований для роботи на різних операційних системах, включаючи Linux та macOS, а також на мобільних пристроях і портативних консолях, таких як Steam Deck. Розширення доступності продукту дозволить залучити ширшу аудиторію. Важливим кроком також є локалізація інтерфейсу та контенту на різні мови, що сприятиме виходу на міжнародний ринок.

Поглиблення історії ігрового світу (лору) дозволить створити більш емоційний зв'язок між гравцем і застосунком, що позитивно вплине на рівень залученості. Розширення сюжетної складової може включати введення нових персонажів, пояснення причин втрати кольорів та поступове розкриття історії світу через ігровий процес або короткі інтерактивні сцени. Це сприятиме формуванню цілісного сприйняття ігрового всесвіту та підсилить мотивацію гравця до проходження рівнів. Крім того, наявність глибшого наративу дозволяє інтегрувати освітній контент більш природно, поєднуючи навчання з емоційним досвідом.

У результаті реалізація зазначених напрямів розвитку дозволить перетворити проєкт «Повернення кольорів» із прототипу в повноцінний освітній продукт, здатний конкурувати на міжнародному ринку інтерактивного контенту. Такий підхід забезпечить баланс між розважальною та навчальною функціями, підвищуючи цінність застосунку для широкого кола користувачів. Додатково це відкриває можливості для подальшого масштабування, включаючи розширення контенту, адаптацію під різні платформи та інтеграцію нових технологій у майбутньому.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було розроблено та програмно реалізовано освітній застосунок «Повернення кольорів», спрямований на розвиток когнітивних навичок та кольорового сприйняття користувачів. Проєкт поєднує ігрові механіки з навчальними елементами, що дозволяє ефективно залучати користувача до процесу навчання в інтерактивній формі. Під час розробки було спроектовано архітектуру застосунку, реалізовано основні ігрові механіки, систему прогресії та алгоритми перевірки стану рівнів. На основі проведеного дослідження та практичної реалізації можна зробити наступні висновки:

Аналіз та вибір технологій передбачав проведення порівняльного дослідження сучасних ігрових рушіїв, зокрема Unreal Engine, Godot та Unity. Під час аналізу враховувалися такі критерії, як продуктивність, зручність розробки, підтримка 2D-графіки, кросплатформеність та наявність інструментів для створення користувацького інтерфейсу. За результатами порівняння було обґрунтовано вибір середовища Unity та мови програмування C# як найбільш оптимальних для реалізації проєкту. Даний технологічний стек дозволив ефективно реалізувати компонентно-орієнтовану архітектуру, що забезпечує модульність та масштабованість системи. Крім того, використання Unity і C# забезпечило кросплатформеність застосунку та спростило процес його подальшого розширення і підтримки.

Проєктування архітектури та механік передбачало створення унікальної ігрової системи, побудованої на основі сітки розмірністю 6x7. Така структура дозволила організувати чітке представлення ігрового поля та забезпечити зручну взаємодію користувача з елементами головоломки. Розроблено математичну модель гри, яка базується на формуванні колірних градієнтів за допомогою лінійної інтерполяції векторів \$RGBA\$, що забезпечує плавні переходи між кольорами. Для генерації початкового стану рівнів реалізовано алгоритм випадкового змішування плиток на основі модифікованого алгоритму Фішера–Єйтса. При цьому збереження опорних

фіксованих елементів дозволяє гравцеві орієнтуватися у структурі градієнта та логічно відновлювати правильну послідовність.

Програмна реалізація проєкту включає створення комплексу взаємопов'язаних програмних модулів, які забезпечують повноцінне функціонування ігрової системи. Центральну роль у керуванні ігровим процесом виконує клас GridManager, що відповідає за генерацію рівнів, перевірку стану гри та координацію взаємодії між елементами сітки [38]. Клас Tile реалізує логіку інтерактивності окремих об'єктів, включаючи обробку дій користувача та візуальні реакції на наведення і вибір. Окремий модуль LevelSelectManager відповідає за систему прогресії, зокрема за розблокування рівнів і контроль доступу до контенту. Особливу увагу приділено візуальному фідбеку, де реалізовано плавне масштабування елементів інтерфейсу та синусоїдальну анімацію літер, що підвищує загальну привабливість застосунку.

Візуальна айдентика та прогресія проєкту базуються на авторській концепції «світу без кольорів», яка формує унікальне художнє та наративне підґрунтя застосунку. У межах цієї концепції ігровий прогрес користувача візуалізується через поступове «офарблення» середовища, яке трансформується від повністю монохромного стану до насиченого повноколірного світу. Така динамічна зміна візуального оформлення безпосередньо відображає успіхи гравця та підсилює відчуття досягнення результату. Окремо було розроблено логотип проєкту, який символічно передає ідею відновлення колірної гармонії за допомогою художнього пензля як ключового образу. У результаті поєднання візуальної стилістики та системи прогресії створюється цілісна естетична концепція, що підсилює занурення користувача в ігровий світ.

Тестування та верифікація розробленого застосунку включали комплексну перевірку всіх ключових функціональних модулів системи. Особливу увагу було приділено функціональному тестуванню алгоритмів перевірки умови перемоги IsComplete, які реалізовано для трьох рівнів

складності. У результаті тестування підтверджено коректність роботи кожного з варіантів перевірки, враховуючи різні математичні принципи визначення правильного стану ігрового поля. Додатково було перевірено систему збереження даних через Unity PlayerPrefs, яка забезпечує стабільне збереження прогресу користувача між ігровими сесіями. Також підтверджено, що механізм збереження коректно підтримує послідовне розблокування рівнів і зберігання налаштувань користувача без втрати даних.

Перспективи розвитку проєкту передбачають подальше розширення його функціональних можливостей та вдосконалення ігрового досвіду. Одним із ключових напрямів є створення інтерактивної карти ігрового світу, яка дозволить користувачеві вільно обирати різні локації та рівні для проходження. Також планується впровадження додаткових режимів гри, зокрема з обмеженням за часом або кількістю кроків, що підвищить складність та варіативність ігрового процесу. Важливим етапом розвитку є адаптація застосунку для мобільних платформ, що забезпечить доступність гри для ширшої аудиторії користувачів. Окремо розглядається можливість портування проєкту на портативні ігрові пристрої, зокрема консолі типу Steam Deck, що сприятиме розширенню ринку використання застосунку.

Мета роботи досягнута, завдання виконані в повному обсязі. Створений продукт є функціональним освітнім інструментом, що поєднує технічну складність ігрових алгоритмів із високим рівнем візуальної та емоційної залученості користувача.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Биков В. Ю. Моделі середовища відкритої освіти : монографія. Київ : Атіка, 2008. 484 с.(дата звернення: 07.04.2026).
2. Гуржій А. М., Поворознюк Н. І., Самсонов В. В. Інформатика та інформаційні технології : підручник. Харків : ТОВ «Сміт», 2007. 352 с. (дата звернення: 08.04.2026).
3. Державний стандарт України. Інформаційна технологія. Розроблення програмних засобів (ДСТУ ISO/IEC 12207:2014). Київ : Держспоживстандарт України, 2015. 102 с. (дата звернення: 09.04.2026).
4. Жалдак М. І. Комп'ютерно-орієнтовані системи навчання : становлення і розвиток. Комп'ютер у школі та сім'ї. 2013. № 2. С. 3–11. (дата звернення: 10.04.2026).
5. Коновал О. А. Психологічні особливості сприйняття кольору в цифровому середовищі. Актуальні проблеми психології. 2019. Т. 7, вип. 48. С. 112–125. (дата звернення: 11.04.2026).
6. Морзе Н. В. Методика навчання інформатики : навч. посіб. у 3-х ч. Київ : Навчальна книга, 2004. Ч. 1. 256 с. (дата звернення: 12.04.2026).
7. Семеріков С. О. Теорія і методика навчання фундаментальних дисциплін у вищій школі : монографія. Кривий Ріг : Видавничий відділ НМетАУ, 2009. 407 с. (дата звернення: 13.04.2026).
8. Спірін О. М. Моделі навчання об'єктно-орієнтованого програмування в середовищі сучасних ігрових рушіїв. Інформаційні технології і засоби навчання. 2021. Т. 82, № 2. С. 45–60. (дата звернення: 14.04.2026).
9. Триус Ю. В. Комп'ютерно-орієнтовані методичні системи навчання : монографія. Черкаси : Брама-Україна, 2005. 400 с. (дата звернення: 15.04.2026).
10. Шаров С. В. Аналіз ігрових рушіїв для створення освітніх ресурсів. Науковий вісник Мелітопольського державного педагогічного університету. 2018. Вип. 21. С. 134–140. (дата звернення: 16.04.2026).

11. Adams E. Fundamentals of Game Design. 3rd ed. New Riders, 2014. 576 p. (дата звернення: 16.04.2026).
12. Albahari J., Albahari B. C# 10 in a Nutshell: The Definitive Reference. O'Reilly Media, 2022. 1060 p. (дата звернення: 07.04.2026).
13. Alessi S. M., Trollip S. R. Multimedia for Learning: Methods and Development. 3rd ed. Allyn & Bacon, 2001. 592 p. (дата звернення: 08.04.2026).
14. Berns R. S. Billmeyer and Saltzman's Principles of Color Technology. 4th ed. Wiley, 2019. 288 p. (дата звернення: 09.04.2026).
15. Bond J. G. Introduction to Game Design, Prototyping, and Development: From Concept to Playable Game with Unity and C#. 3rd ed. Addison-Wesley, 2022. 992 p. (дата звернення: 10.04.2026).
16. Clark R. C., Mayer R. E. E-Learning and the Science of Instruction. 4th ed. Wiley, 2016. 528 p. (дата звернення: 11.04.2026).
17. Csikszentmihalyi M. Flow: The Psychology of Optimal Experience. Harper Perennial Modern Classics, 2008. 336 p. (дата звернення: 12.04.2026).
18. Deterding S., Dixon D., Khaled R., Nacke L. From Game Design Elements to Gamefulness: Defining "Gamification". *Proceedings of the 15th International Academic MindTrek Conference*. Tampere, 2011. P. 9–15. (дата звернення: 13.04.2026).
19. Fairchild M. D. Color Appearance Models. 3rd ed. Wiley, 2013. 472 p. (дата звернення: 14.04.2026).
20. Freeman E., Robson E. Head First Design Patterns: Building Extensible and Maintainable Object-Oriented Software. 2nd ed. O'Reilly Media, 2020. 672 p. (дата звернення: 15.04.2026).
21. Fullerton T. Game Design Workshop: A Playcentric Approach to Creating Innovative Games. 4th ed. CRC Press, 2018. 556 p. (дата звернення: 16.04.2026).
22. Gamma E., Helm R., Johnson R., Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, 1994. 395 p. (дата звернення: 07.04.2026).

23. Gee J. P. What Video Games Have to Teach Us About Learning and Literacy. Revised ed. Palgrave Macmillan, 2007. 256 p. (дата звернення: 08.04.2026).
24. Hunt R. W. G., Pointer M. R. The Reproduction of Colour. 6th ed. Wiley, 2004. 724 p. (дата звернення: 09.04.2026).
25. Kapp K. M. The Gamification of Learning and Instruction: Game-based Methods and Strategies for Training and Education. Pfeiffer, 2012. 336 p. (дата звернення: 10.04.2026).
26. Koster R. A Theory of Fun for Game Design. 2nd ed. O'Reilly Media, 2013. 304 p. (дата звернення: 11.04.2026).
27. Livingstone M. Vision and Art: The Biology of Seeing. Revised ed. Abrams, 2014. 256 p. (дата звернення: 12.04.2026).
28. Microsoft Docs. C# Language Reference. URL: <https://docs.microsoft.com/en-us/dotnet/csharp/> (дата звернення: 13.04.2026).
29. Nielsen J. Usability Engineering. Morgan Kaufmann, 1993. 368 p. (дата звернення: 14.04.2026).
30. Norman D. The Design of Everyday Things. Revised ed. Basic Books, 2013. 368 p. (дата звернення: 15.04.2026).
31. Prensky M. Digital Game-Based Learning. Paragon House, 2007. 464 p. (дата звернення: 16.04.2026).
32. Richter J. CLR via C#. 4th ed. Microsoft Press, 2012. 896 p. (дата звернення: 07.04.2026).
33. Rogers S. Level Up! The Guide to Great Video Game Design. 2nd ed. Wiley, 2014. 552 p. (дата звернення: 08.04.2026).
34. Schell J. The Art of Game Design: A Book of Lenses. 3rd ed. A K Peters/CRC Press, 2019. 650 p. (дата звернення: 09.04.2026).
35. Shneiderman B., Plaisant C., Cohen M., Jacobs S. Designing the User Interface: Strategies for Effective Human-Computer Interaction. 6th ed. Pearson, 2016. 624 p. (дата звернення: 10.04.2026).

36. Thorn A. Pro Unity Game Development with C#. 2nd ed. Apress, 2020. 450 p. (дата звернення: 11.04.2026).

37. Troelsen A., Japikse P. Pro C# 10 with .NET 6: Foundational Principles and Practices in Programming. 11th ed. Apress, 2022. 1200 p. (дата звернення: 12.04.2026).

38. Unity Technologies. Unity Manual. URL: <https://docs.unity3d.com/Manual/> (дата звернення: 13.04.2026).

39. Wyszecki G., Stiles W. S. Color Science: Concepts and Methods, Quantitative Data and Formulae. 2nd ed. Wiley, 2000. 968 p. (дата звернення: 13.04.2026).

40. Zichermann G., Cunningham C. Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps. O'Reilly Media, 2011. 208 p. (дата звернення: 14.04.2026).