

Галицький фаховий коледж імені В'ячеслава Чорновола
відділення комп'ютерних технологій
циклова комісія інформатики та комп'ютерних дисциплін

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач відділення

комп'ютерних технологій

Наталія СТЕФУРАК / _____ /

підпис

«___» _____ 2025 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи

освітньо-професійного ступеня «фаховий молодший бакалавр»

зі спеціальності 122 «Комп'ютерні науки»

на тему: «Ігрова програма в жанрі головоломки»

Студент групи КН-41

Павло ВОЛОШИН

(підпис)

Керівник роботи

Олександра ЧУБЕЙ

(підпис)

Консультанти:

з техніко-економічного

обґрунтування

Любов МЕЛЕНЧУК

(підпис)

нормоконтролер

Надія ГАВРИШКІВ

(підпис)

Галицький фаховий коледж імені В'ячеслава Чорновола
відділення комп'ютерних технологій
циклова комісія інформатики та комп'ютерних дисциплін

ЗАТВЕРДЖУЮ

Завідувач відділення

комп'ютерних технологій

Наталія СТЕФУРАК / _____ /

підпис

«___» _____ 2024 р.

ЗАВДАННЯ

на кваліфікаційну роботу

на здобуття освітньо-професійного ступеня «фаховий молодший бакалавр»

студенту Волошину Павлу Володимировичу

(прізвище, ім'я та по-батькові студента)

1. Тема роботи Ігрова програма в жанрі головоломки
затверджена наказом по коледжу від “25” Листопада 2024 р., №253а-н
2. Термін здачі студентом завершеної роботи “24” Червня 2025 р.
3. Вихідні дані до роботи Результати дослідження ігрових програм, аналіз ринку таких додатків та технологій їхньої реалізації.
4. Перелік питань, які повинні бути розроблені:
 - а) основна частина: аналіз предметної області та постановка завдань.
Проектування системи. Реалізація та тестування системи.
 - б) техніко-економічнеобґрунтування Аналіз ринку. Розрахунок витрат на проектування. Обґрунтування необхідності розробки.
5. Перелік графічного матеріалу: діаграма потоків даних. діаграма IDEF0
схема взаємодії користувача з грою

Розділ	Консультанти	Підпис, дата	
		Завдання видано	Завдання прийнято
З техніко-економічного обґрунтування	Любов МЕЛЕНЧУК	20.04.2025	20.05.2025

КАЛЕНДАРНИЙ ПЛАН
виконання кваліфікаційної роботи

№ п/п	Найменування етапу	Терміни	
		початку	завершення
1	Пошук теми роботи, вивчення вимог для кваліфікаційної роботи	16.11.2024	25.11.2024
2	Аналіз існуючих рішень та визначення їхніх переваг і недоліків	26.11.2024	30.11.2024
3	Вивчення технологій та інструментів реалізації.	01.12.2024	05.12.2024
4	Формування архітектури застосунку, визначення функціональних вимог.	07.12.2024	12.12.2024
5	Підготовка та налаштування програмного середовища	13.12.2024	30.12.2024
6	Проектування структури застосунку, бази даних та інтерфейсу користувача	31.12.2024	23.01.2025
7	Реалізація основної логіки застосунку	24.01.2025	15.04.2025
8	Оптимізація коду та вдосконалення окремих модулів	15.04.2025	23.04.2025
9	Тестування застосунку та усунення помилок	24.04.2025	03.05.2025
10	Аналіз ринку збуту, визначення користувачів і оцінка економічної доцільності	04.05.2025	15.05.2025
11	Оформлення пояснювальної записки.	15.05.2025	11.06.2025
12	Попередній захист кваліфікаційної роботи	13.06.2025	13.06.2025
13	Підготовка до захисту кваліфікаційної роботи	14.06.2025	23.06.2025
14	Захист кваліфікаційної роботи	24.06.2025	24.06.2025

Дата видачі “___” _____ 202_ р. Керівник _____ / Чубей О.О.

Завдання прийняв до виконання _____ / Волошин П.В.

Реферат

Ігрова програма в жанрі головоломки.

Кваліфікаційна робота. Волошин Павло Володимирович. Галицький фаховий коледж імені В'ячеслава Чорновола, відділення комп'ютерних технологій. Спеціальність 122 “Комп'ютерні науки”, 2025. Сторінок – 62, таблиць – 1, рисунків – 30, додатків – 4, джерел – 7.

Об'єкт дослідження – процес розробки ігор-головоломок та сучасні інструменти для їх реалізації, зокрема ігровий рушій Unity.

Метою роботи є створення ігрової програми в жанрі головоломки для ОС Android з використанням ігрового рушія Unity та мови програмування C#. Ігрова програма повинна містити набір рівнів із наростаючою складністю, систему збереження прогресу гравця, а також реалізовувати унікальні ігрові механіки, спрямовані на розвиток логічного мислення та просторової уяви користувача. Крім того, необхідно спроектувати інтуїтивно зрозумілий користувацький інтерфейс та систему налаштувань для забезпечення комфортної гри.

Для реалізації поставленої задачі було використано сучасні технології та інструменти для розробки ігор, зокрема ігровий рушій Unity, мову програмування C# у середовищі Visual Studio Code.

Результатом розробки є повноцінний ігровий застосунок, який є завершеною роботою, готовою до використання та подальшого розповсюдження.

Ключові слова: ГРА-ГОЛОВЛОМКА, UNITY, C#, РОЗРОБКА ІГОР, PUZZLE, ІГРОВА МЕХАНІКА, РІВЕНЬ, VISUAL STUDIO, ANDROID, ЛОГІЧНА ГРА.

Abstract

Game program in the genre of puzzle.

Qualification work. Voloshin Pavlo Volodymyrovych. Galician Vocational College named after Vyacheslav Chornovil, Department of Computer Technologies. Specialty 122 “Computer Science”, 2025. Pages – 62, figures – 30, appendices – 4.

The object of research is the process of developing puzzle games and modern tools for their implementation, in particular the Unity game engine.

The purpose of the work is to create a game program in the genre of puzzle for the Android OS using the Unity game engine and the C# programming language. The game program should contain a set of levels with increasing complexity, a system for saving the player's progress, and also implement unique game mechanics aimed at developing logical thinking and spatial imagination of the user. In addition, it is necessary to design an intuitive user interface and a system of settings to ensure a comfortable game.

To implement the task, modern technologies and tools for game development were used, in particular the Unity game engine, the C# programming language in the Visual Studio Code environment.

The result of the development is a full-fledged game application, which is a completed work, ready for use and further distribution.

Keywords: PUZZLE GAME, UNITY, C#, GAME DEVELOPMENT, PUZZLE, GAME MECHANICS, LEVEL, VISUAL STUDIO, ANDROID, LOGIC GAME.

ЗМІСТ

Скорочення та умовні позначки	7
Вступ	8
1 Аналіз предметної області та постановка завдання	9
1.1 Аналіз предметної області	9
1.2 Аналіз наявних рішень	11
1.3 Аналіз вимог до програмного засобу та постановка завдання	18
2 Проектування системи	20
2.1 Формування цілей і основних завдань	20
2.2 Функціональна структура мобільного додатку	21
2.3 Алгоритм роботи системи.....	21
2.4 Проектування інтерфейсу гри.....	24
2.5 Технології реалізації	27
3 Розробка мобільного додатку	31
3.1 Розробка інтерфейсу гри	31
3.2 Розробка функціонального додатку	34
3.3 Тестування гри.....	49

					КР.КН 25.587.06.000 ПЗ			
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		Волошин П.			Ігрова програма в жанрі головоломки	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Перевір.</i>		Чубей О.					5	62
<i>Рценз</i>		Кульчинська Н.				ГФК. ВКТ. КН-41		
<i>Н.контр</i>		Гавришків Н.						
<i>Зав від</i>		Стефурак Н.						

4 Техніко-економічне обґрунтування	56
4.1 Аналіз ринку збуту продукту.....	56
4.2. Розрахункова частина	57
4.3 Обґрунтування необхідності розробки	59
Висновки.....	61
Перелік джерел посилання	62
Додатки.....	63

					КР.КН 25.587.06.000 ПЗ				
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Ігрова програма в жанрі головоломки	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>	
<i>Розроб.</i>		Волошин П.						6	62
<i>Перевір.</i>		Чубей О.				ГФК. ВКТ. КН-41			
<i>Рценз</i>		Кульчинська Н.							
<i>Н.контр</i>		Гавришків Н..							
<i>Зая від</i>		Стефурак Н.							

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ОС – Операційна система

UX – User Experience

UI – User Interface

ЄСВ – Єдиний соціальний внесок

UE – Unreal Engine

ПК – Персональний Комп'ютер

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

ВСТУП

Кваліфікаційна робота присвячена процесу створення ігрової програми в жанрі головоломки, розробленої для надання гравцям унікального інтелектуального виклику та стимулюючого способу проведення часу. Основна концепція проєкту полягає у створенні середовища, де гравці можуть розвивати логічне та просторове мислення, вирішуючи серію дедалі складніших завдань. Актуальність проєкту зумовлена постійним зростанням ринку казуальних та інді-ігор, а також попитом на продукти, що поєднують розважальний та розвиваючий потенціал.

Гра орієнтована на широке коло користувачів, від казуальних гравців, що шукають цікаву розвагу, до досвідчених шанувальників головоломок, які цінують нетривіальні завдання. Проєкт передбачає поступове введення нових ігрових механік та зростання рівня складності, що має на меті підтримку постійного інтересу та відчуття прогресу в гравця. Ключовий акцент зроблено на розробці чистого та мінімалістичного інтерфейсу, що не відволікає від ігрового процесу та забезпечує легку навігацію.

Створення гри є комплексним завданням, що охоплює ретельне проєктування рівнів, програмування ігрової логіки на C# в середовищі Unity, оптимізацію фізичної взаємодії об'єктів та побудову надійної архітектури застосунку. Дана робота послідовно розкриває кожен етап розробки, починаючи від формування початкових вимог до продукту і закінчуючи його фінальним тестуванням на цільових платформах.

Структура роботи побудована таким чином, щоб надати повне уявлення про життєвий цикл проєкту. В основній частині детально аналізуються використані технології, зокрема ігровий рушій Unity, розглядається процес проєктування архітектури, реалізації ключових ігрових механік, системи збереження прогресу та розробки адаптивного користувацького інтерфейсу.

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1 Аналіз предметної області

Ігрова індустрія в наш час вважається однією з найбільш динамічних та швидкозростаючих сфер цифрових технологій. Головоломки як жанр відеоігор мають широку аудиторію завдяки своїй здатності затягнути користувача у гру, також ігри цього жанру розвивають логічне мислення, креативність та увагу.

Гра є природною частиною розвитку людини з дитинства. З наймолодшого віку діти використовують ігри для пізнання світу, розвитку соціальних навичок і креативного мислення. Настільні ігри, рольові ігри, а згодом і відеоігри стають важливими елементами навчання та розваг. Ігри допомагають розвивати координацію, пам'ять і стратегічне мислення, а також вчать взаємодії з іншими людьми.

Популярність мобільних ігор зростає. Ринок мобільних ігор за останні роки стрімко розвивається. Багато ІТ-компаній зосереджують свої зусилля на створенні мобільних ігор.

Люди люблять грати в ігри з різних причин. Ігри це чудовий спосіб відпочити та зняти стрес, адже граючи, користувач може зануритися у віртуальний світ і тимчасово відволіктися від повсякденних проблем. Також, ігри задовольняють потребу в досягненнях та самореалізації – виконання рівнів, отримання нагород і змагання з іншими гравцями дають відчуття прогресу та успіху. Крім того, багато ігор сприяють розвитку навичок, таких як стратегічне мислення, координація та швидкість реакції. Соціальний аспект також грає важливу роль – можливість грати разом із друзями або змагатися онлайн створює додаткову мотивацію.

Ігри також можуть мати терапевтичний ефект, допомагаючи людям впоратися з тривожністю, депресією та іншими психологічними проблемами. Деякі дослідження показують, що відеоігри можуть сприяти підвищенню

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9

рівня допаміну в мозку, що покращує настрій і загальний стан користувача. Вони також можуть стимулювати креативність, оскільки багато сучасних ігор містять елементи конструювання, модифікації та персоналізації ігрового світу.

Окрім розважального аспекту, головоломки використовуються в освітніх цілях, когнітивних тренуваннях та навіть терапії. Такі ігри сприяють розв'язанню складних завдань, стимулюють мозкову активність і можуть допомагати в навчальному процесі.

Щодо платформ, на яких грають користувачі, варто відзначити дві найпопулярніші операційні системи – Android та iOS.

Таблиця 1.1 – Порівняння операційних систем

Характеристика	Android	iOS
Гнучкість налаштувань	Можливість зміни інтерфейсу, встановлення сторонніх лаунчерів, доступ до файлової системи	Обмежена можливість персоналізації, зміни доступні лише в межах дозволених налаштувань
Безпека	Менш контрольована, можливість встановлення додатків з будь-яких джерел	Високий рівень безпеки, перевірка додатків у App Store
Доступність пристроїв	Великий вибір моделей у різних цінових категоріях	Обмежений вибір дорогих пристроїв
Оптимізація продуктивності	Залежить від виробника та пристрою	Висока, завдяки єдиній екосистемі
Магазин додатків	Google Play, можливість встановлення APK-файлів	Тільки App Store
Підтримка оновлень	Залежить від виробника	Довготривала підтримка пристроїв

Android є відкритою платформою, яка дозволяє широке налаштування системи, підтримує великий вибір пристроїв різних виробників і пропонує велику кількість безкоштовних додатків. Окрім цього, Android дає можливість встановлювати додатки з різних джерел, що надає користувачам більшу свободу, але водночас створює потенційні загрози безпеці.

iOS, у свою чергу, відома своєю стабільністю, оптимізацією продуктивності та високим рівнем безпеки. Завдяки закритій екосистемі Apple, всі додатки проходять ретельну перевірку перед публікацією в App Store, що знижує ймовірність завантаження шкідливого програмного забезпечення. Однак, пристрої Apple є дорогими, а екосистема більш закритою в порівнянні з Android, що може бути недоліком для користувачів, які бажають мати більше контролю над своїми пристроями.

Зважаючи на ці фактори, розробка ігрової програми в жанрі головоломки є актуальною задачею, оскільки поєднує в собі розважальний і освітній потенціал, відповідає сучасним тенденціям розвитку ігрового ринку та має широкі перспективи для комерційного та освітнього застосування.

1.2 Аналіз наявних рішень

Жанр головоломка – це один із найпопулярніших жанрів відеоігор, який зосереджується на вирішенні логічних завдань, просторових головоломок або розпізнаванні закономірностей. Головною метою таких ігор є розвиток когнітивних навичок, уваги, логічного та креативного мислення.

Бувають різні типи головоломок, залежно від механіки гри та способу взаємодії гравця із завданнями.

Основні види головоломок:

- логічні головоломки;
- фізичні головоломки;
- головоломки-лабіринти;
- головоломки на складання слів;
- математичні головоломки;

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

- головоломки три в ряд;
- квестові головоломки;
- Escape Room (головоломки-втечі);

Жанр головоломок постійно розвивається, поєднуючись із іншими жанрами (RPG, платформери, симулятори), що робить його ще цікавішим для гравців різного віку та рівня підготовки.

Color Block Jam - Геймплей гри полягає в тому що потрібно переміщувати кольорові кубики до виходів відповідного кольору. Гра у вільному доступі в Play Market. Кадри з гри зображені на рисунку 1.1.

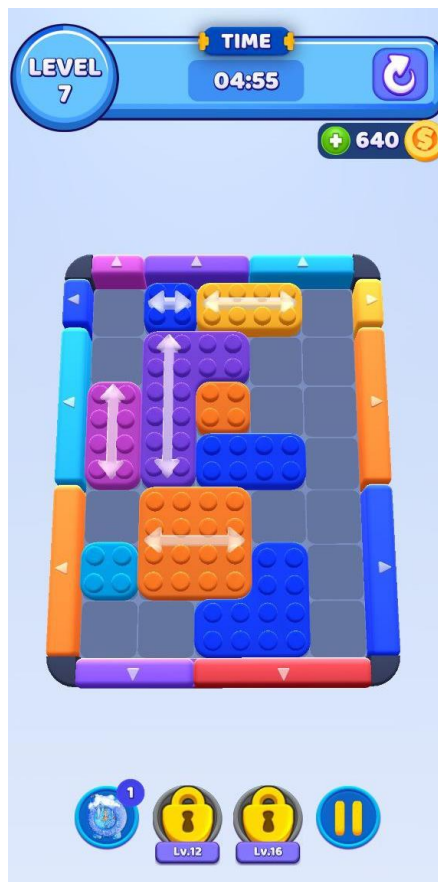


Рисунок 1.1 - Кадри з гри Color Block Jam

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

Перевагами гри є:

- Яскравий візуальний стиль – використання насичених кольорів робить гру привабливою та легкою для сприйняття.
- Простий, але захопливий геймплей – гра зрозуміла з першого рівня, але поступово стає складнішою.
- Безліч рівнів – доступна велика кількість рівнів, що забезпечує довготривалу гру.

Недоліками гри є:

- Монотонність – основний геймплей полягає лише в тому, щоб пересувати кубики до стінок відповідного кольору. Це цікаво на перших рівнях, але з часом може набридати через відсутність нових механік або змін у процесі гри.
- Реклама та донат – після проходження рівня або невдалої спроби часто з'являються рекламні вставки, що заважають насолоджуватися геймплеєм.
- Через ці моменти гра може здатися більше спрямованою на заробіток, ніж на цікаве проведення часу.

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

Roll the Ball - Гра в якій потрібно допомогти кульці попасти до фінішу викладаючи блоки з доріжками в правильному порядку. Гра у вільному доступі в Play Market та App Store. Кадри з гри зображено на рисунку 1.2.

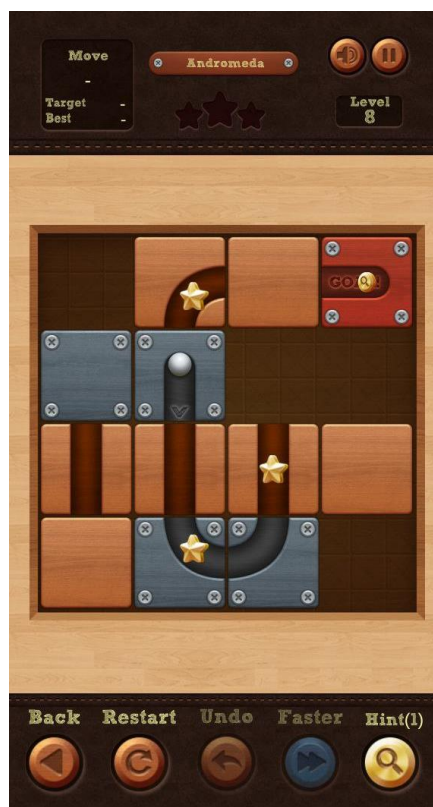


Рисунок 1.2 - Кадри з гри Roll the Ball

Перевагами гри є:

- Логічна складова – гра сприяє розвитку просторового мислення та навичок вирішення головоломок.

- Приємна дерев'яна стилізація – візуальне оформлення нагадує класичні настільні головоломки, що додає шарму.

- Плавне зростання складності – початкові рівні прості, але гра поступово ускладнюється.

- Недоліками гри є:

- Геймплей не змінюється – хоча в грі є безліч рівнів, кожен з них має одну і ту ж задачу переміщувати блоки з доріжками так, щоб кулька докотилася до фінішу. Це швидко стає одноманітним і не дає відчуття розвитку.

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

- Відсутність нових механік – гра не додає нових елементів або змін у геймплей, що робить її передбачуваною і менш цікавою з часом.
- Реклама та платний контент – у грі присутня реклама, що може відволікати.
- Деякі функції, як-от підказки, можна отримати лише за внутрішню валюту або купити за реальні гроші, що створює відчуття нав'язливого донату.
- Простий візуальний стиль – хоча гра має приємний дерев'яний дизайн, анімації пересування блоків не дуже плавні.

Одноманітний геймплей гри та нав'язлива реклама можуть знизити інтерес у довготривалій перспективі. Якщо додати більше варіативності в механіки, гра стала б ще цікавішою.

Flow Free - Гравець повинен з'єднати дві точки одного кольору, заповнюючи всю сітку, щоб завершити рівень. При цьому лінії не можуть перетинатися. Гра в безкоштовному доступі в Play Market. Кадри з гри зображені на рисунку 1.3.

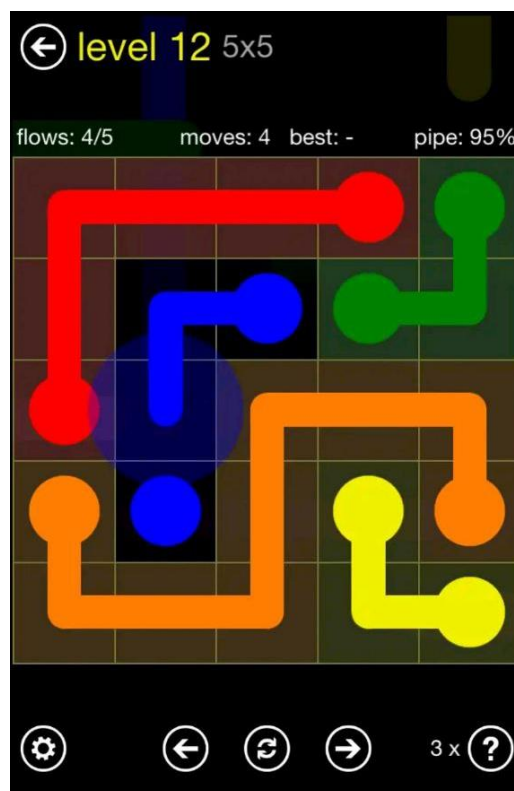


Рисунок 1.3 - Кадри з гри Flow Free

Перевагами гри є:

– Простота правил – зрозумілий ігровий процес, який легко освоїти новачкам.

– Різноманітність рівнів – велика кількість варіантів складності та розмірів сітки для гри.

– Медитативний ефект – плавний ігровий процес без таймера дозволяє який дозволяє розслабитися.

– Недоліками гри є:

– Одноманітний геймплей – кожен рівень має ту ж саму задачу - з'єднувати точки одного кольору. Відсутність змін у механіках робить гру передбачуваною.

– Відсутність розвитку – складність рівнів зростає лише за рахунок більшої кількості точок і перешкод, але не додаються нові елементи геймплею, які могли б урізноманітнити гру.

– Немає мотивації для проходження – оскільки рівні майже не відрізняються, немає відчуття досягнень чи прогресу.

Якщо б у гру додали нові механіки, таймери, бонуси чи інші варіанти проходження, вона стала б набагато цікавішою.

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

Block Puzzle & Conquer - В цій грі є пуста ділянка яку потрібно заповнювати різними фігурами так щоб не залишалось вільного простору. Гра в безкоштовному в Play Market. Кадри гри зображені на рисунку 1.4.

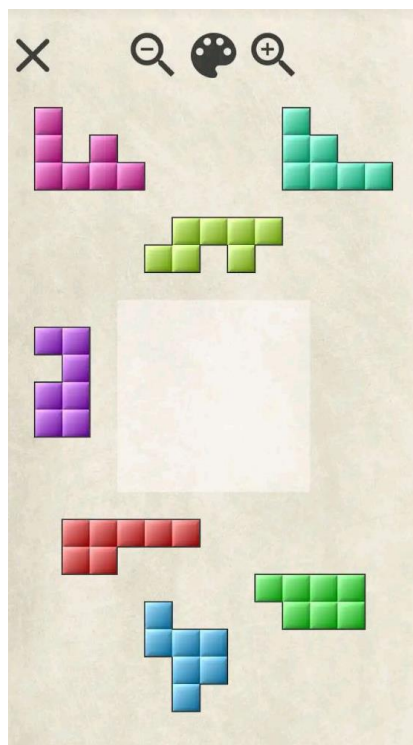


Рисунок 1.4 – Кадри з гри Block Puzzle & Conquer

Перевагами гри є:

- Класичний геймплей - знайомий стиль головоломок із фігурами нагадує тетріс, що робить гру зрозумілою та ностальгічною.
- Розвиток логічного мислення - гра змушує продумувати розміщення фігур наперед.

Недоліками гри є:

- Одноманітний геймплей – у кожному рівні потрібно лише заповнювати область фігурами, без змін у механіках чи додаткових викликів. Складність підвищується лише за рахунок нових форм фігур, але гра не пропонує нових елементів чи режимів.
- Простий дизайн – графіка мінімалістична, без особливих візуальних ефектів або анімацій, що робить гру менш привабливою.

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

– Нав'язлива реклама – практично після кожного рівня з'являється реклама, це може дратувати та псувати враження від гри.

Якби додали більше варіацій у геймплей, покращили графіку та зменшили кількість реклами, гра стала б значно цікавішою.

Отже, зважаючи на вищезазначений матеріал, основними недоліками цих ігор є:

– Надокучлива реклама та донат – після кожного рівня або навіть під час гри з'являється реклама, яка псує враження від процесу. Часто є платний контент або можливість придбати підписку, що нав'язливо пропонується гравцю.

– Монотонність геймплею – усі рівні побудовані за одним і тим же принципом, що швидко набридає. Відсутність нових механік або цікавих змін ускладнює довготривале проходження.

– Недолугий дизайн – мінімалістична, але застаріла графіка без приємних анімацій та візуальних ефектів. Виглядає дешево та не дуже привабливо.

А з плюсів можна зазначити наступні елементи:

– Цікаві механіки – в цих іграх реалізовані прості та зрозумілі механіки, які роблять гру доступною для всіх.

– Логічні завдання змушують мислити та шукати правильні рішення, головоломки такого типу добре тренують уважність і просторове мислення. Це чудовий формат для любителів спокійних ігор.

1.3 Аналіз вимог до програмного засобу та постановка завдання

Проектування та розробка android гри-головоломки вимагає створення захоплюючого та інтелектуально стимулюючого продукту, що буде цікавим для широкої аудиторії.

Основною цільовою аудиторією гри є шанувальники логічних завдань та казуальних ігор, а також усі, хто бажає розвинути просторове мислення та навички вирішення проблем в інтерактивному форматі.

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		18

До функціональних вимог належить реалізація системи рівнів-головоломок із наростаючою складністю, можливість керування ігровим об'єктом у реальному часі, а також взаємодія з інтерактивними елементами, такими як кнопки-перемикачі, що змінюють геометрію рівня. Застосунок повинен автоматично зберігати прогрес гравця, візуально відобразити доступні та заблоковані рівні й надавати можливість перезапуску рівня або повернення в головне меню через меню паузи.

Нефункціональні вимоги охоплюють високу швидкодію та стабільність роботи гри, а також адаптивність її інтерфейсу до різних розмірів екранів мобільних пристроїв та ПК. Інтерфейс має бути інтуїтивно зрозумілим, візуально цілісним та виконаним у мінімалістичному стилі, що сприяє концентрації на вирішенні головоломок. Керування повинно бути чутливим та забезпечувати миттєвий відгук на дії гравця.

Завданням розробки є створення повноцінного ігрового застосунку для операційної системи Android, що об'єднує цікавий дизайн рівнів, унікальні ігрові механіки та прозору систему прогресії. Особливу увагу в процесі розробки було приділено логічній вивіреності головоломок, інтуїтивності керування та створенню плавного ігрового досвіду, що мотивує гравця рухатися далі.

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

2 ПРОЄКТУВАННЯ СИСТЕМИ

2.1 Формування цілей і основних завдань

Розробка мобільної гри-головоломки включає кілька ключових етапів, які впливають на її успіх.

Перший етап – це аналіз вимог і стратегічне планування. На цьому етапі необхідно визначити головну ідею гри, її цільову аудиторію та ключові ігрові механіки. Грамотно сформульовані вимоги ще на початковій стадії допоможуть уникнути небажаних змін у процесі розробки, що дозволить зекономити час і ресурси.

Згідно з ігровими механіками та розробленою логікою головоломок формується ігровий процес, який враховує можливості мобільного додатку, технічні ресурси та особливості гравця, під час проходження щоб отримати доступ до наступного рівня потрібно пройти попередній. Прогресія проходження рівнів зберігається і буде можливість її скинути.

Наступним етапом є проєктування, яке визначає загальну структуру, візуальне оформлення та вибір технологій для реалізації гри.

Після цього розпочинається етап розробки, що включає написання програмного коду, впровадження ігрової логіки, створення інтерфейсу користувача. Цей етап потребує уважності та відповідального підходу, щоб забезпечити коректну роботу гри відповідно до всіх вимог.

Фінальним етапом проводиться тестування та оптимізація, під час яких виявляються та виправляються можливі помилки, перевіряється відповідність гри заявленим характеристикам, а також усуваються недоліки перед її публікацією.

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

2.2 Функціональна структура мобільного додатку

Функціональна структура гри визначає ключові аспекти її розробки, взаємодію між різними елементами та потенціал для подальшого вдосконалення й розширення функціоналу.

Пересування та проходження рівнів – гравець зможе керувати персонажем

або взаємодіяти з об'єктами, поступово долаючи рівні з різною складністю.

Збереження прогресії проходження рівнів, де користувач зможе переглядати пройдені рівні.

Налаштування гри – можливість змінювати параметри графіки (наприклад, якість текстур або частоту кадрів).

2.3 Алгоритм роботи системи

На рисунку 2.1 зображена схема, яка показує механіку взаємодії користувача з грою.

Вона відображає процес обробки жестів користувача на пристрої Android, визначення змін у грі та виконання відповідних дій. Потік даних проходить через код додатка, що інтерпретує введення і формує команди для геймплею, що зрештою призводить до отримання результату.

Простими словами ця схема показує, як Android-додаток перетворює жест користувача на дію. Спочатку система розпізнає сам жест (наприклад, свайп). Потім, за допомогою коду додатка, вона інтерпретує цей жест у конкретну команду, яка одразу ж виконується. У результаті програма реагує на дію користувача, показуючи відповідну зміну на екрані.

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		21

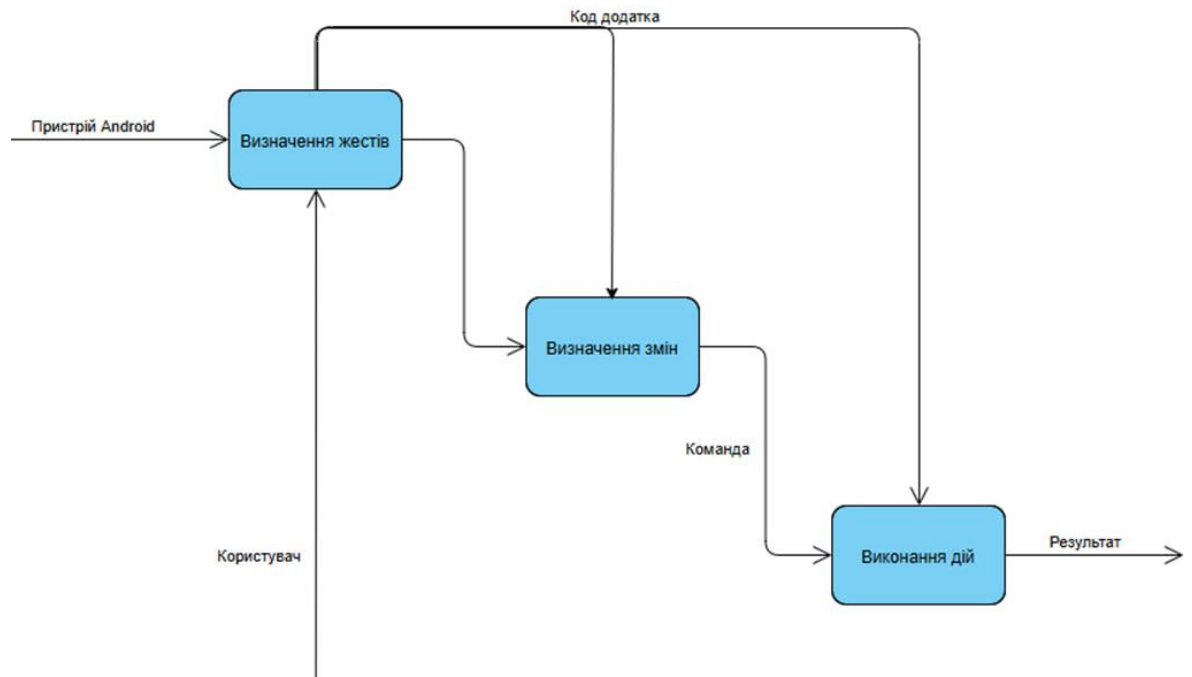


Рисунок 2.1 – Схема взаємодії користувача з грою

Діаграма яка зображена на рисунку 2.2 відображає загальну логіку роботи гри та є діаграмою потоків даних, яка показує взаємодію між користувачем та основними модулями гри. Усі команди користувача спочатку обробляються в модулі “Обробка меню”, де він може обрати потрібну дію, наприклад, почати гру або переглянути рівні. Після вибору гра переходить до модуля “Стан рівня”, який визначає хід гри та передає дані на “Екран гравця”, де безпосередньо відбувається ігровий процес. Після завершення гри інформація передається до модуля “Обробка завершення”, який зберігає або оновлює рекорди, після чого користувач може повернутися до головного меню.

Таким чином, діаграма ілюструє послідовність дій у грі та рух інформації між її основними компонентами.

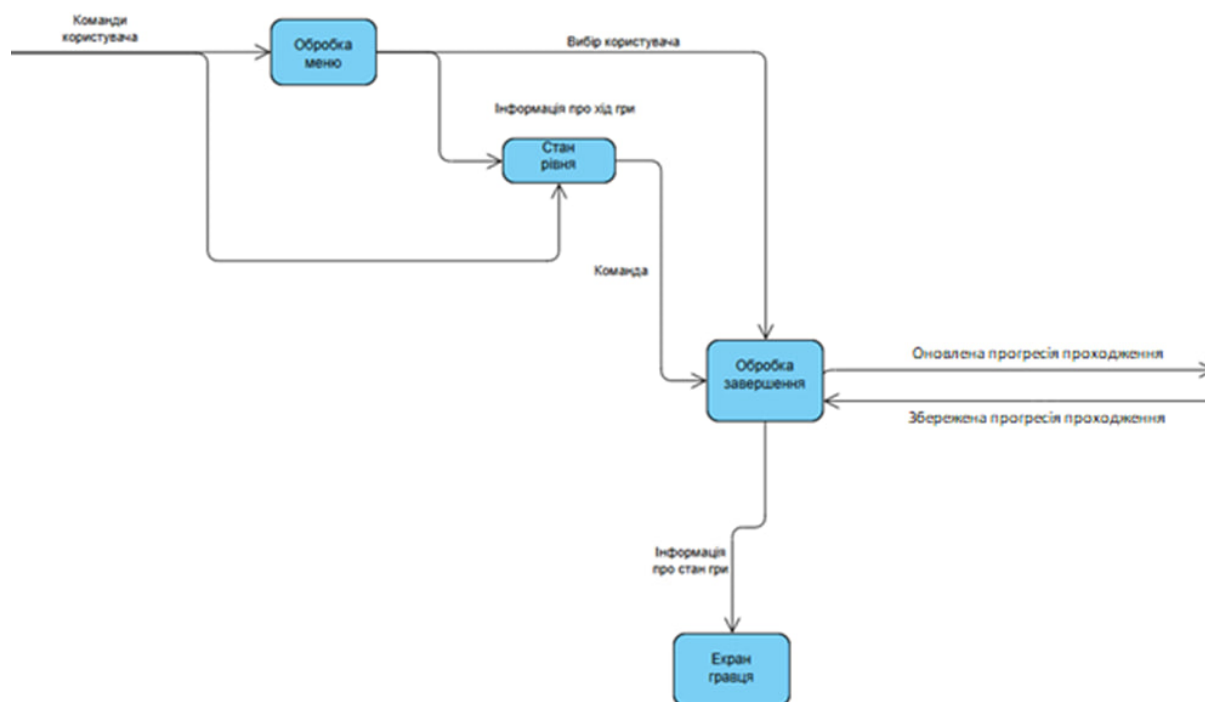


Рисунок 2.2 – Діаграма потоків даних

На діаграмі IDEF0 (Рисунок 2.3) зображено контекстну модель роботи ігрового додатка, яка показує його як єдиний цілісний процес та описує його взаємодію із зовнішнім середовищем. Центральний блок, “Ігровий додаток”, представляє собою основну функцію. Вхідними даними для цього процесу є “Взаємодія з екраном” — тобто торкання, свайпи та інші дії, які гра має обробити. Результатом роботи цього процесу є вихідний продукт — певна “Дія” в грі.

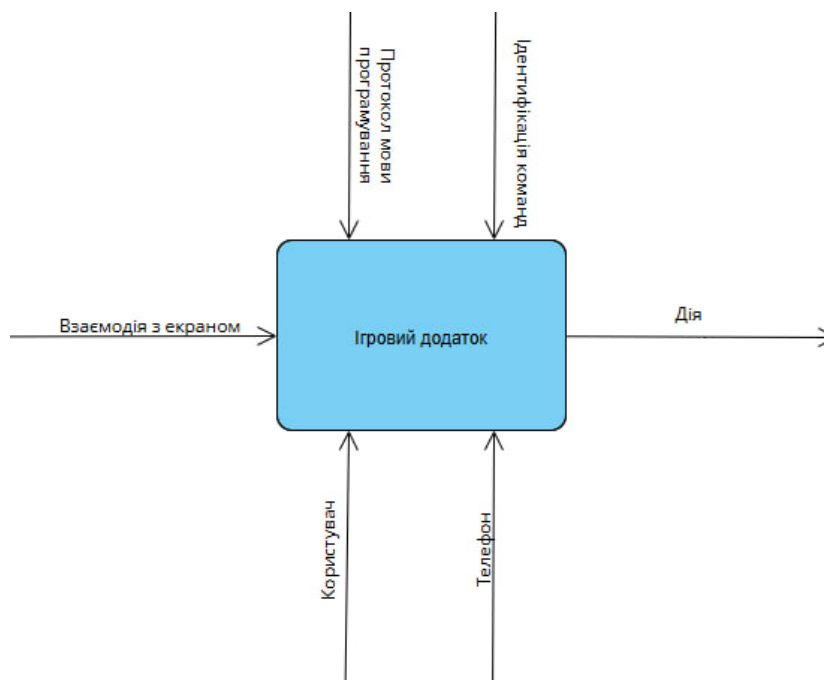


Рисунок 2.3 – Діаграма IDEF0

2.4 Проєктування інтерфейсу гри

Інтерфейс користувача (UI) у мобільній грі-головоломці відіграє вирішальну роль у забезпеченні зручності керування та приємного користувацького досвіду. Для створення інтерфейсу використовується фреймворк Figma, який дозволяє моделювати зовнішній вигляд основних екранів гри та взаємозв'язки між ними. Дизайн побудовано на основі принципів матеріального стилю Android, що забезпечує сучасний вигляд і зрозумілу навігацію.

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		24

На рисунку 2.4 розміщено макет головного меню яке буде з'являтися першим після запуску гри. На головному меню розташовано лише одну кнопку, вона розташована по центру для зручності взаємодії/ Дизайн гри вирізняється мінімалістичним та сучасним стилем, що базується на простих геометричних формах та яскравій, гармонійній кольоровій палітрі.

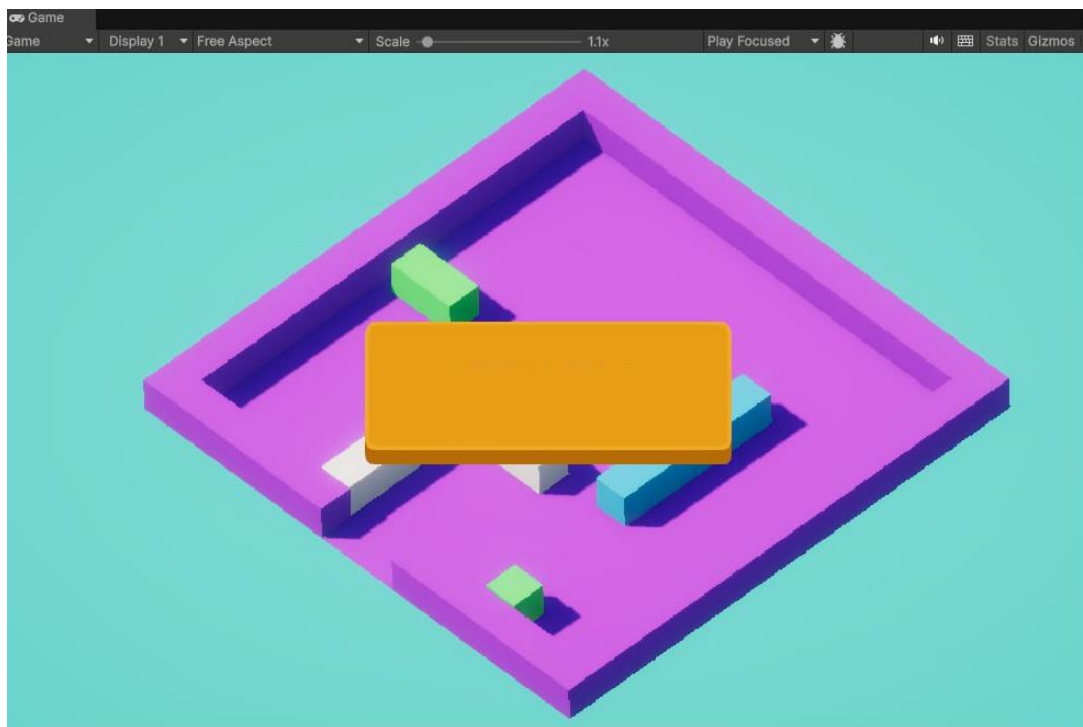


Рисунок 2.4 – Макет головного меню

Ключовою особливістю головного меню є те, що фоном буде слугувати безпосередньо сам ігровий рівень. Гравця одразу занурюють в ізометричну 3D-сцену, що складається з фіолетової квадратної основи та кількох кольорових прямокутних блоків. Поверх цієї інтерактивної сцени розміщено велику, контрастну помаранчеву кнопку “ГРАТИ”, яка слугує чітким та інтуїтивно зрозумілим закликком до дії.

Після натискання кнопки на головному меню буде з'являтися наступне не менш важливе меню – це меню вибору рівнів. Це меню дозволяє гравцеві обирати потрібний рівень серед доступних. Для зручності додано сітку з нумерованими плитками рівнів, а також індикація пройдених та заблокованих

рівнів. При натисканні на певний рівень відбувається перехід безпосередньо до ігрового екрана. Макет меню вибору рівнів зображено на рисунку 2.5.

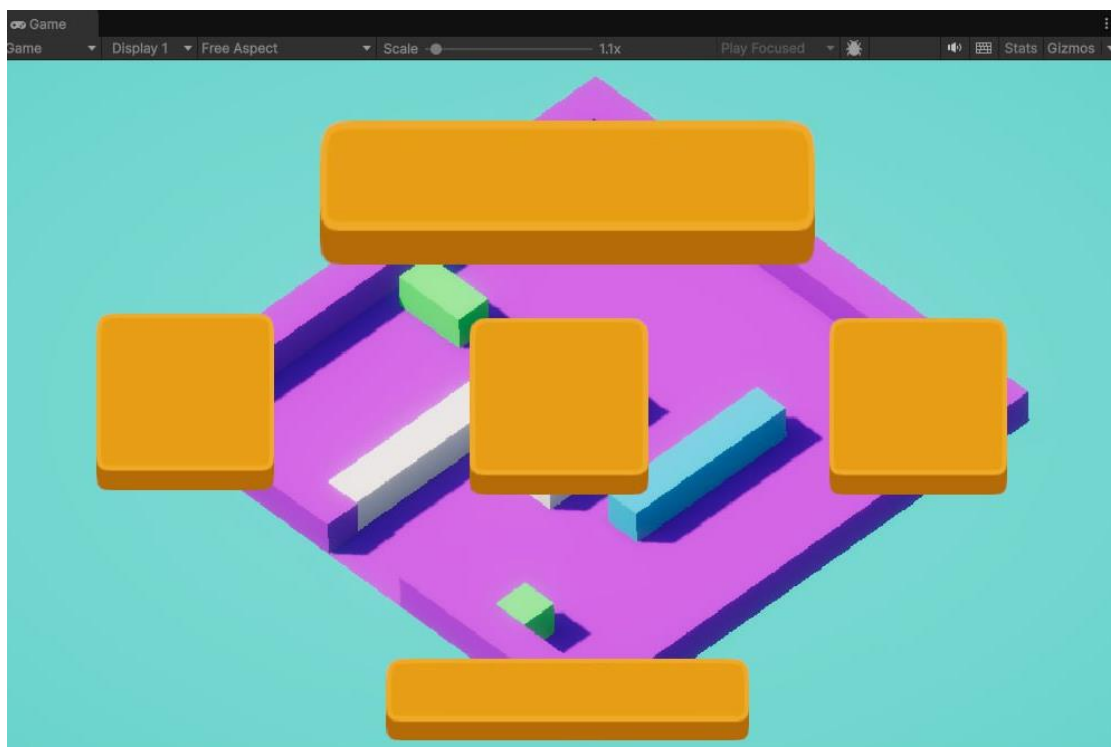


Рисунок 2.5 – Макет меню вибору рівнів

Меню вибору рівнів продовжує мінімалістичний та стильний дизайн, що був заданий у попередніх екранах гри. На звичному бірюзовому фоні з ізометричним ігровим полем розташовані основні елементи інтерфейсу.

У верхній частині екрана знаходиться напівпрозора плашка з написом “ВИБІР РІВНЯ”, а в центрі розміщені три великі помаранчеві кнопки-квадрати з номерами рівнів. Внизу екрана знаходиться додаткова кнопка “СКИНУТИ” для обнулення прогресу пройдених рівнів. Дизайн є інтуїтивно зрозумілим та візуально узгодженим з іншими частинами гри, використовуючи ту саму кольорову палітру, шрифти та прості геометричні форми.

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26

Інтерфейс гри взаємопов'язаний логічними переходами, що відпрацьовуються під час моделювання. Такий підхід забезпечує повноцінний інтерактивний макет, готовий для реалізації на етапі розробки програмного забезпечення. Орієнтовні візуалізації інтерфейсу наведено на рисунках.

2.5 Технології реалізації

Unreal Engine (UE), розроблений компанією Epic Games, є одним із найпотужніших і технологічно просунутих ігрових рушіїв у світі. Він відомий своєю здатністю створювати фотореалістичну графіку найвищого рівня, що робить його стандартом для багатьох високобюджетних (AAA) проєктів. На відміну від Unity, де основною мовою програмування є C#, в Unreal Engine використовується C++, що надає розробникам глибший контроль над оптимізацією та системними ресурсами.



Рисунок 2.6 – Unreal Engine

Ключовою особливістю Unreal Engine є система візуального скриптингу Blueprints. Вона дозволяє створювати складну ігрову логіку, не написавши жодного рядка коду, що є потужним інструментом для дизайнерів та художників. Рушій також пропонує передові інструменти для роботи зі світлом (Lumen), геометрією (Nanite) та анімацією, що дозволяє досягати кінематографічної якості зображення.

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		27



Рисунок 2.7 – Інтерфейс Unreal Engine

На Unreal Engine створено такі відомі ігри, як Fortnite, The Matrix Awakens, Cyberpunk 2077 (з версії Phantom Liberty), Star Wars Jedi: Survivor та серія Gears of War. Русій має власний магазин асетів — Unreal Engine Marketplace, який також пропонує величезну кількість готових ресурсів. Вибір на користь Unity, а не Unreal Engine, для даної роботи був зумовлений нижчим порогом входження, гнучкістю для мобільних платформ та більшою простотою C# порівняно з C++.

Для реалізації поставлених завдань було обрано сучасні та ефективні програмні інструменти. Основним середовищем розробки виступив ігровий рушій Unity 3D, який є однією з провідних платформ для створення інтерактивних 2D та 3D проєктів.

Unity — це потужна кросплатформна система для розробки ігор та інших інтерактивних застосунків, що надає розробникам комплексний набір інструментів для створення, анімації, візуалізації та монетизації. Гнучкість рушія дозволяє розробляти проєкти для більш ніж 25 платформ, включаючи персональні комп'ютери (Windows, macOS, Linux), мобільні пристрої (iOS, Android), ігрові консолі (PlayStation, Xbox, Nintendo Switch) та пристрої віртуальної (VR) і доповненої (AR) реальності. Програмування ігрової логіки та складних механік здійснюється переважно мовою C#.

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		28

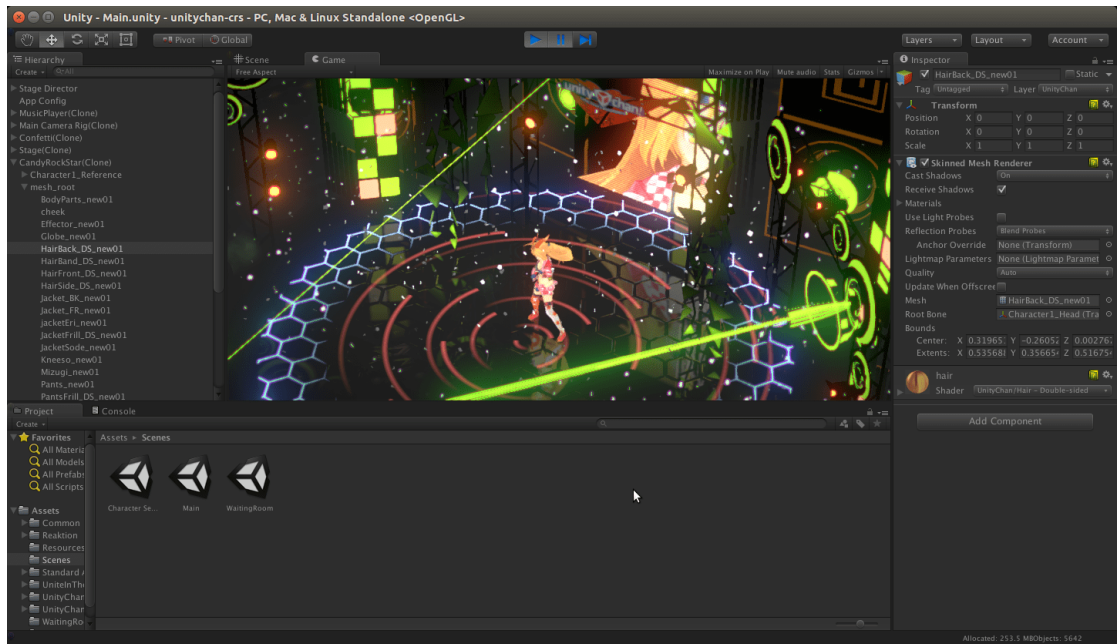


Рисунок 2.8 – Інтерфейс Unity 3D

Надійність та функціональність Unity підтверджується безліччю всесвітньо відомих і комерційно успішних ігор, створених на його основі. Серед найяскравіших прикладів можна назвати рольову гру з відкритим світом Genshin Impact, визнану критиками інді-гру Hollow Knight, унікальну за своїм візуальним стилем Cuphead та культурний феномен Among Us. Ці проекти демонструють широкі можливості рушія у створенні ігор абсолютно різних жанрів та масштабів.

Важливою частиною екосистеми Unity є Unity Asset Store — вбудований онлайн-магазин, де розробники можуть знаходити, купувати та продавати готові ресурси. Цей маркетплейс пропонує величезний вибір 3D-моделей, анімацій, аудіоефектів, скриптів та візуальних ефектів. Використання готових асетів з Asset Store дозволило значно прискорити процес розробки даного проекту, уникнути необхідності створювати кожен елемент з нуля та зосередитись на реалізації ключових механік. Зокрема, для цієї роботи були залучені моделі оточення та деякі скрипти, що допомогло підвищити якість візуальної складової та оптимізувати загальний час, витрачений на розробку.

									Арк.
									29
Змн.	Арк.	№ докум.	Підпис	Дата					

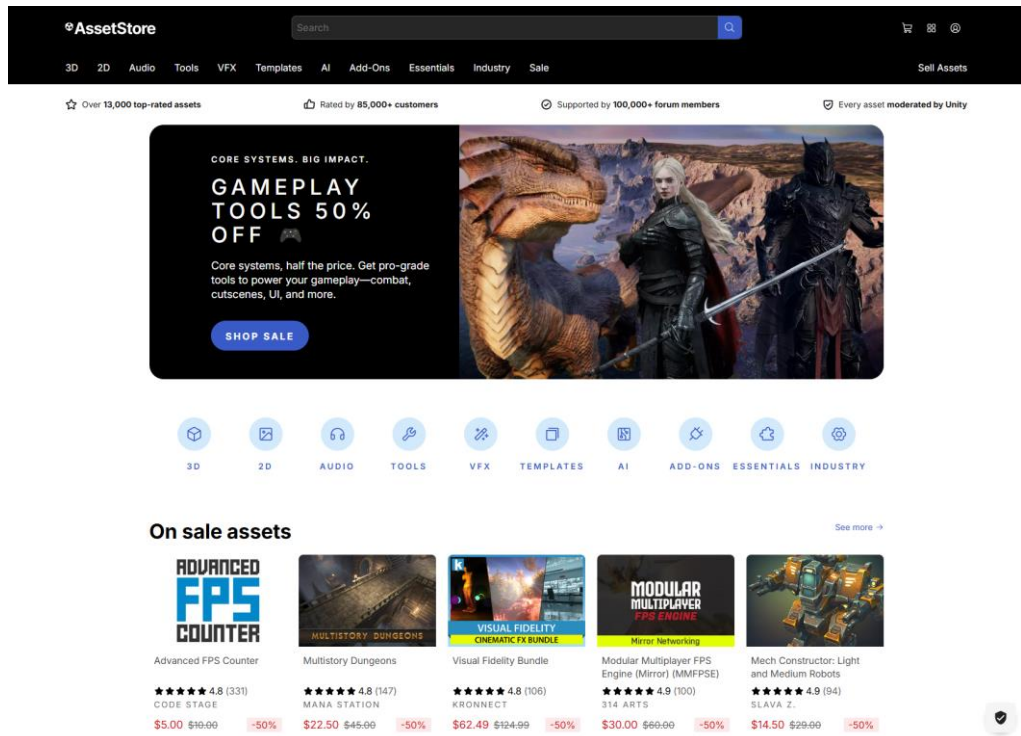


Рисунок 2.9 – Unity AssetStore

Отже, проаналізувавши можливості провідних ігрових рушіїв, для реалізації даного проєкту було свідомо обрано Unity, а не його головного конкурента Unreal Engine. Хоча Unreal Engine славиться своєю передовою графікою, перевага була надана Unity через значно вищу швидкість ітеративної розробки та нижчий поріг входження. Інтуїтивний редактор, гнучка система префабів та відносна простота мови C# дозволили швидше прототипувати та втілювати ігрові механіки, що є критично важливим в умовах обмеженого часу дипломної роботи. На відміну від цього, робота з C++ в Unreal Engine вимагає глибших технічних знань та більших часових витрат. Крім того, величезна спільнота та обсяг документації Unity є більш вагомим перевагою для вирішення поточних завдань, ніж спеціалізовані, хоч і потужніші, інструменти Unreal Engine. Таким чином, вибір Unity став прагматичним рішенням, де пріоритет надавався гнучкості, швидкості розробки та доступності ресурсів.

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		30

3 РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ

3.1 Розробка інтерфейсу гри

Однією з найважливіших частин розробки гри-головоломки є створення зручного та інтуїтивно зрозумілого користувацького інтерфейсу. З метою забезпечення максимального комфорту та концентрації гравців під час вирішення завдань, інтерфейс повинен бути чітко й ергономічно спроектованим.

Процес розробки користувацького інтерфейсу для гри складався з кількох етапів. На початкових стадіях проектування була обрана мінімалістична концепція, що найкраще відповідає специфіці жанру головоломок. З огляду на те, що гра вимагає від гравця зосередженості, дизайн додатку мав бути простим та зрозумілим, а весь функціонал — досяжним за мінімум кроків, не відволікаючи від основного ігрового процесу.

Для розробки інтерфейсу були застосовані підходи та принципи, властиві сучасним головоломкам. Простота та інтуїтивність взаємодії були ключовими вимогами. Щоб прискорити процес розробки та забезпечити якісний вигляд, деякі готові елементи інтерфейсу, такі як іконки та стилі кнопок, були взяті з Unity Asset Store та адаптовані під загальний візуальний стиль гри. Інструментарій взаємодії з грою був спроектований таким чином, щоб забезпечити зручність переходу між меню та рівнями, а також комфортне керування безпосередньо в грі.

При проектуванні користувацького інтерфейсу активно використовувалась Figma — потужний інструмент для розробки дизайну. Завдяки Figma була успішно реалізована можливість створення єдиного та зручного стилю для всіх екранів гри.

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		31

Figma дозволила ефективно розпланувати розміщення та взаємодію елементів інтерфейсу, зокрема кнопок керування, меню вибору рівнів та вікон для відображення ігрового процесу, що в кінцевому результаті дозволило створити приємний візуальний досвід для гравця.

Наочним прикладом цього етапу роботи є скріншот з робочого простору Unity 3D (рисунок 3.1), де зображено процес компоновки екрану вибору рівня. На сітці для вирівнювання розташовані ключові елементи інтерфейсу: плашка з заголовком “ВИБІР РІВНЯ”, кнопки з номерами рівнів, а також функціональні кнопки “СКИНУТИ” та “ГРАТИ”.

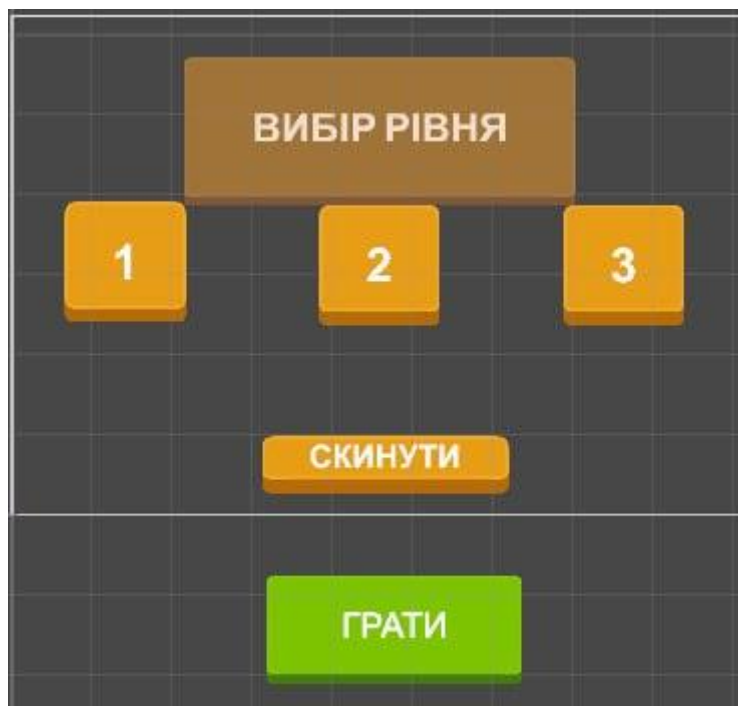


Рисунок 3.1 – Робочий простір Unity3D

Окрему увагу було приділено розробці інтерфейсу безпосередньо в процесі гри. Як видно на макеті, ігровий екран є максимально лаконічним, щоб не відволікати гравця. У верхній частині інтерфейсу розташовані лише найнеобхідніші елементи: зліва – індикація поточного рівня (LVL 1), а справа – іконка паузи. (рисунок 3.2)

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		32

При натисканні на паузу з'являється центральне модальне вікно з відповідним заголовком “ПАУЗА”. Воно містить ключові для цього стану гри опції: "ПРОДОВЖИТИ", “РЕСТАРТ” та “ГОЛОВНЕ МЕНЮ”.

Кожна кнопка має свій колір для швидкої візуальної ідентифікації. Таке рішення дозволяє гравцеві миттєво призупинити гру та обрати подальшу дію, не виходячи з ігрового контексту.

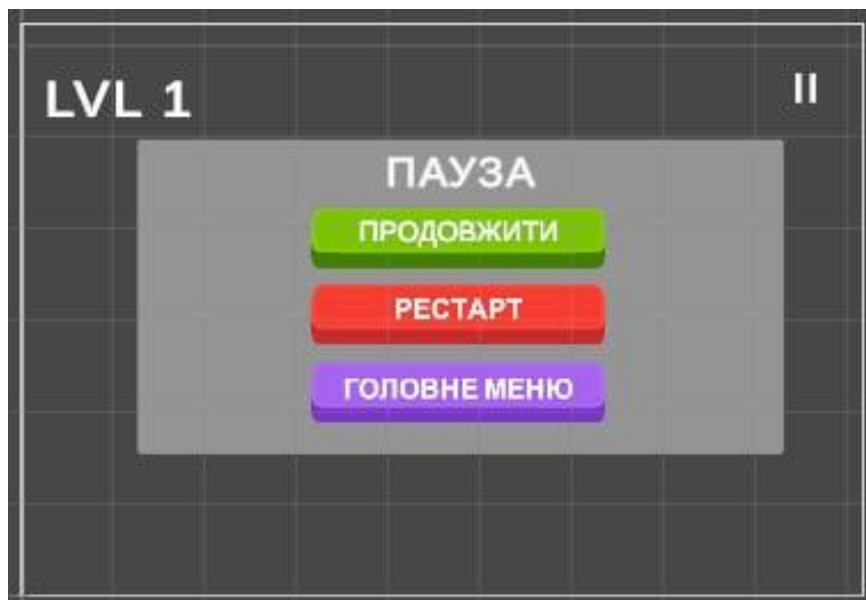


Рисунок 3.2 – Меню паузи

Для створення впізнаваного образу гри було прийнято рішення не розробляти окремий складний логотип. Натомість, в якості логотипу та іконки додатку було використано стилізоване зображення ігрової арени. Цей підхід дозволив миттєво донести до гравця суть гри ще до її запуску, підкресливши її мінімалістичний стиль та основну механіку, що створює цілісний та запам'ятовуючий візуальний образ.

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		33

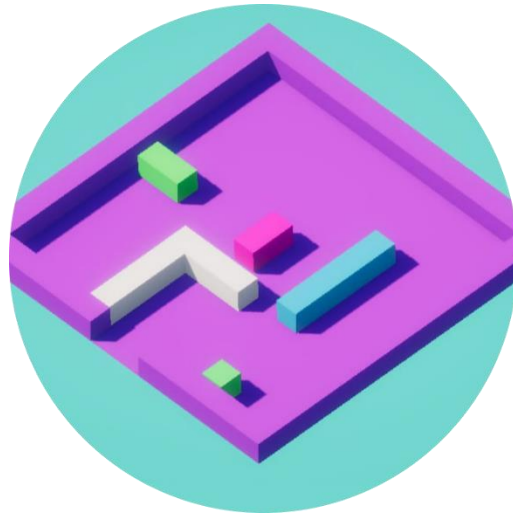


Рисунок 3.3 – Логотип гри

3.2 Розробка функціонального додатку

На початковому етапі розробки тривимірної гри-головоломки було створено базову ігрову сцену яка зображена на рисунку 3.4, в середовищі Unity 3D, що закладає фундамент для подальшої імплементації ігрових механік. На представленому зображенні продемонстровано результат першого кроку — моделювання основного ігрового простору.

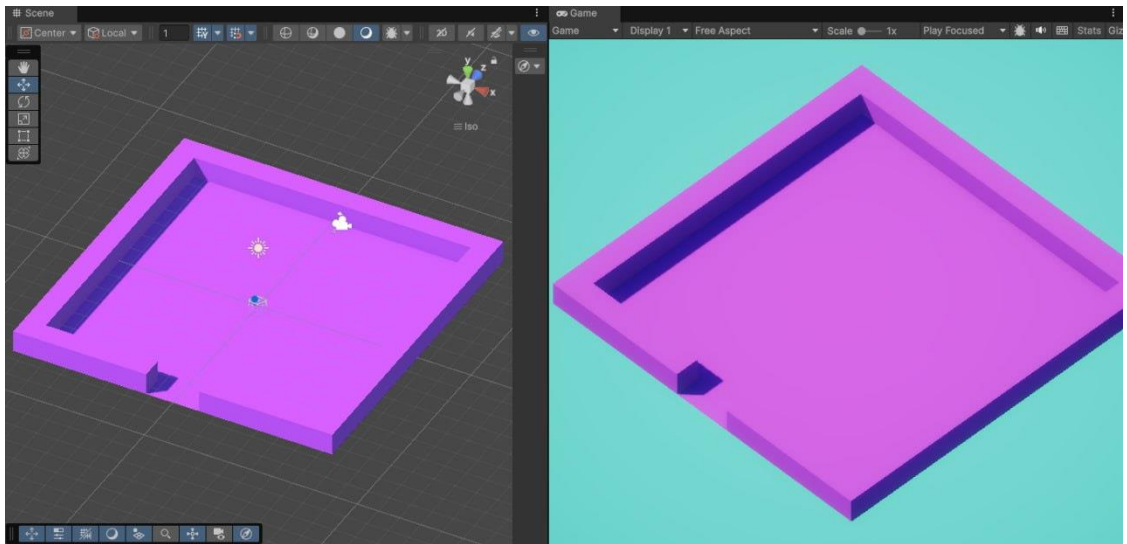


Рисунок 3.4 – Ігрове поле

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		34

Центральним елементом сцени є тривимірна модель, що являє собою ігрове поле. Цей об'єкт, виконаний у вигляді квадратної платформи з невисокими бортами по периметру, слугуватиме основним простором для взаємодії гравця з елементами головоломки. Геометрія моделі є простою, але функціональною: борти створюють замкнену область, у межах якої відбудуватимуться ігрові події. В одному з кутів платформи передбачено спеціальний паз, який відіграватиме ключову роль у механіці рівня, як фінішна точка.

Наступним логічним кроком у розробці стало додавання до сцени об'єкта, керованого гравцем, та створення базової логіки для його функціонування. У якості протагоніста було введено тривимірну модель куба, що гармонійно вписується в раніше обрану мінімалістичну стилістику проєкту. Цей об'єкт, для кращої візуальної ідентифікації пофарбований у контрастний рожевий колір, розміщено всередині ігрової платформи, і він слугуватиме аватаром гравця (рисунок 3.5).

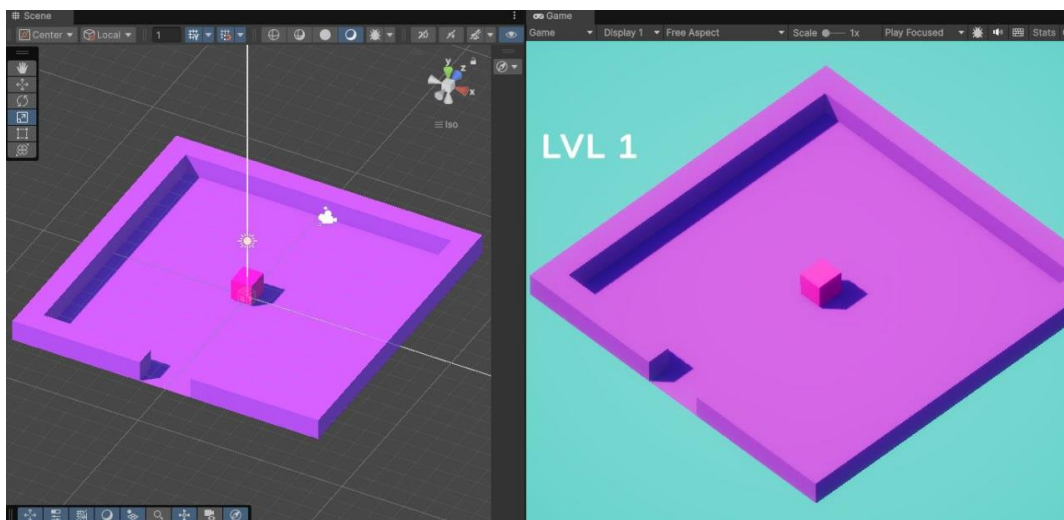


Рисунок 3.5 – об'єкт керований гравцем

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		35

Ключовим елементом цього етапу є розробка програмної складової. Для забезпечення інтерактивності до об'єкта-гравця було прикріплено спеціальний скрипт, написаний мовою програмування C#. Цей скрипт є фундаментальним компонентом, що відповідає за обробку даних введення від користувача (наприклад, з клавіатури або геймпада) і транслює їх у переміщення куба в тривимірному просторі. На даному етапі закладається основа для реалізації фізичної моделі поведінки об'єкта, яка в подальшому включатиме обробку зіткнень (колізій) з бортами рівня та іншими ігровими елементами.

Для забезпечення інтерактивності ігрового процесу було створено клас Player, що успадковує MonoBehaviour — базовий клас для всіх скриптів Unity. Цей компонент прикріплюється до ігрового об'єкта, що представляє гравця, та містить у собі всю ключову логіку. Від зчитування дій користувача до керування станами рівнів.

Метод Start() викликається один раз при запуску сцени та використовується для початкового налаштування.

Фрагмент коду метод Start():

```
void Start ()
{
    rb = GetComponent<Rigidbody>();
    currentSceneIndex =
SceneManager.GetActiveScene().buildIndex;
    nextSceneIndex = currentSceneIndex + 1;
}
```

Фрагмент коду метод LoadNextScene():

```
public void LoadNextScene ()
{
    if (nextSceneIndex <
SceneManager.sceneCountInBuildSettings)
    {
        SceneManager.LoadScene (nextSceneIndex);
    }
    else
    {
        Debug.Log ("LAST LEVELss!");
    }
}
```

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		36

Цей публічний метод завантажує наступну сцену. Він перевіряє, чи існує наступна сцена в налаштуваннях збірки (Build Settings), щоб уникнути помилки при спробі завантажити неіснуючий рівень.

Обробка фізики та введення у FixedUpdate:

Основна логіка руху гравця та обробки дій користувача реалізована у методі FixedUpdate(), який викликається з фіксованою періодичністю, що є найкращою практикою для роботи з фізичним рушієм.

Фрагмент коду метод FixedUpdate():

```
private void FixedUpdate()
{
    if (Input.GetKey(keyOne) || isMovingPositive)
    {
        rb.linearVelocity += moveDirection;
    }
    if (Input.GetKey(keyTwo) || isMovingNegative)
    {
        rb.linearVelocity -= moveDirection;
    }
}
```

Цей скрипт постійно перевіряє, чи натиснуті клавіші, визначені в keyOne та keyTwo або прапорець isMovingPositive встановлений в true (тобто затиснута екранна кнопка). Якщо одна з клавіш натиснута, скрипт отримує доступ до компонента Rigidbody (який відповідає за фізику об'єкта) і змінює його лінійну швидкість (linearVelocity), додаючи або віднімаючи вектор moveDirection. Це створює плавний та фізично коректний рух.

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		37

Обробка тригерних подій:

Логіка завершення рівня та взаємодії з іншими об'єктами реалізована через систему тригерів Unity.

Фрагмент коду Метод OnTriggerEnter(Collider other)

```
private void OnTriggerEnter(Collider other)
{
    if (this.CompareTag("Player") &&
other.CompareTag("Finish"))
    {
        UnlockNewLevel();
        LoadNextScene();
    }
}
```

Метод OnTriggerEnter автоматично викликається, коли колайдер гравця входить у зону іншого колайдера, позначеного як “тригер”. У коді виконується перевірка тегів об'єктів, що зіткнулися. Якщо об'єкт з тегом “Player” входить у тригер з тегом “Finish”, це розцінюється як успішне завершення рівня. У такому випадку послідовно викликаються методи UnlockNewLevel() для збереження прогресу та LoadNextScene() для переходу на наступний рівень.

Система збереження прогресу та переходу між рівнями:

Для збереження прогресу гравця та керування завантаженням сцен було створено окремі методи.

Фрагмент коду метод UnlockNewLevel():

```
void UnlockNewLevel()
{
    if (SceneManager.GetActiveScene().buildIndex >=
PlayerPrefs.GetInt("ReachedIndex"))
    {
        PlayerPrefs.SetInt("ReachedIndex",
SceneManager.GetActiveScene().buildIndex + 1);
        PlayerPrefs.SetInt("UnlockedLevel",
PlayerPrefs.GetInt("UnlockedLevel", 1) + 1);
        PlayerPrefs.Save();
    }
}
```

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		38

Фрагмент коду метод LoadNewLevel():

```
public void LoadNextScene ()
{
    if (nextSceneIndex <
SceneManager.sceneCountInBuildSettings)
    {
        SceneManager.LoadScene (nextSceneIndex);
    }
    else
    {
        Debug.Log ("LAST LEVEL!");
    }
}
```

Метод UnlockNewLevel використовує вбудований клас PlayerPrefs для збереження даних на пристрої користувача. Він перевіряє, чи є поточний рівень найвищим з пройдених. Якщо так, він оновлює збережений індекс досягнутого рівня (ReachedIndex) та кількість відкритих рівнів, після чого зберігає зміни викликом PlayerPrefs.Save().

Метод LoadNextScene відповідає за фактичне завантаження наступного рівня. Він перевіряє, чи існує наступна сцена в налаштуваннях проєкту. Якщо так завантажує її. Якщо ж гравець пройшов останній рівень, у консоль виводиться відповідне повідомлення. Цей підхід створює надійну систему прогресії гравця.

До ігрової сцени було додано два нові типи взаємодіючих об'єктів: активатор (представлений у вигляді плаского жовтого циліндра на підлозі) та динамічна перешкода (жовтий прямокутний блок, що перегороджує шлях). Суть нової механіки полягає у встановленні прямого причинно-наслідкового зв'язку між цими елементами, який ініціюється діями гравця.

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		39

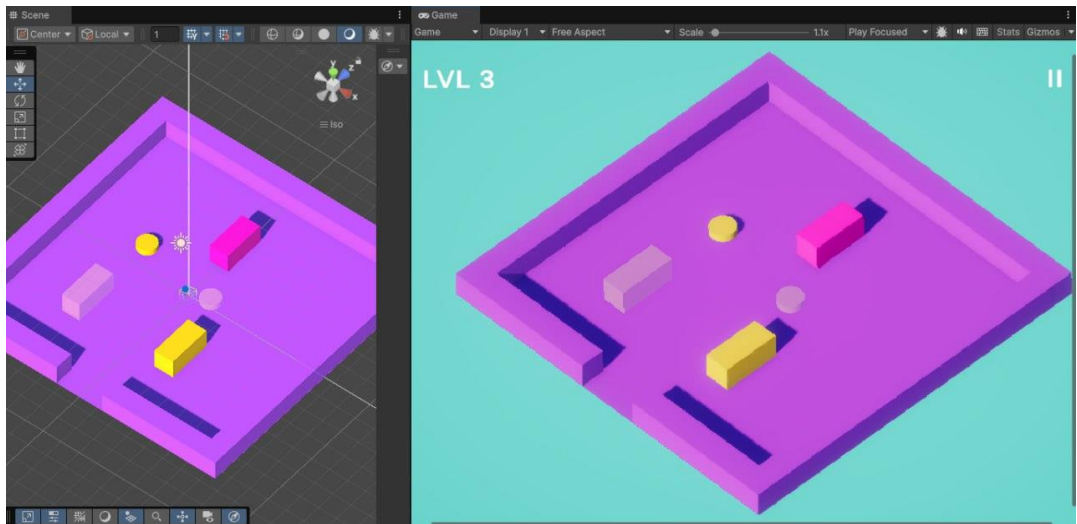


Рисунок 3.6 – Нова механіка

Коли керований гравцем об'єкт (куб) наїжджає на платформу активатора, спрацьовує відповідний тригер. Це, у свою чергу, викликає зміну стану пов'язаної з ним динамічної перешкоди. Згідно з програмною логікою, при активації кнопки матеріал жовтого блоку-перешкоди змінює свої властивості, стаючи візуально прозорим. Важливо зазначити, що ця візуальна трансформація супроводжується і зміною фізичних властивостей: колайдер об'єкта вимикається, що дозволяє гравцю вільно проходити крізь нього. Як тільки гравець залишає активатор, перешкода повертається у свій початковий, непрозорий та твердий стан.

Впровадження такої системи “кнопка-двері” є фундаментальним кроком у проектуванні головоломок. Ця механіка вимагає від гравця не просто навігації по рівню, а й аналізу оточення та виконання послідовності логічних дій для подолання перешкод. Це значно підвищує глибину ігрового процесу та збільшує когнітивне навантаження, що є ключовою метою розробки гри в даному жанрі.

Центральним елементом ігрових головоломок є скрипт `Activator.cs`, що успадковує клас `MonoBehaviour`. Цей компонент призначається об'єктам-кнопкам і містить всю логіку, що керує динамічними перешкодами на рівні.

Основна логіка механізму реалізована в методі OnTriggerEnter, який автоматично викликається рушієм Unity, коли будь-який об'єкт з компонентом Collider входить у тригерну зону даного активатора.

Фрагмент коду Метод OnTriggerEnter(Collider other)

```
private void OnTriggerEnter(Collider other)
{
    if (canPush)
    {
        if (other.CompareTag("Cube") ||
other.CompareTag("Player"))
        {
            foreach (GameObject first in firstGroup)
            {
                first.GetComponent<Renderer>().material =
normal;
                first.GetComponent<Collider>().isTrigger =
false;
            }

            foreach (GameObject second in secondGroup)
            {
                second.GetComponent<Renderer>().material =
transparent;
                second.GetComponent<Collider>().isTrigger =
true;
            }
        }
    }
}
```

Перевірка можливості активації: Перш за все, код перевіряє if (canPush). Якщо значення false, кнопка є неактивною, і подальша логіка не виконується. Ідентифікація об'єкта: Наступний крок — перевірка if (other.CompareTag("Cube") || other.CompareTag("Player")). Це гарантує, що активатор спрацює лише від контакту з об'єктом гравця або іншим спеціальним кубом, а не від випадкових елементів. Обробка груп об'єктів: Після успішних перевірок скрипт послідовно обробляє масиви об'єктів:

Для кожного об'єкта з firstGroup встановлюється матеріал normal, а його колайдер переводиться в режим твердого тіла (isTrigger = false). Це робить об'єкт непрохідною перешкодою.

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		41

Для кожного об'єкта з secondGroup встановлюється матеріал transparent, а колайдер переводиться в режим тригера (isTrigger = true). Це робить об'єкт прохідним.

Зміна стану кнопок: У фінальній частині методу (не показана повністю для стислості) відбувається зміна стану самої кнопки та пов'язаної з нею, забезпечуючи візуальний зворотний зв'язок та передачу "ходу" іншій кнопці, що дозволяє створювати складні логічні ланцюги.

Подальший розвиток проєкту був зосереджений на розширенні функціональності інтерфейсу користувача (UI) та імплементації ключових елементів керування ігровою сесією. На цьому етапі було реалізовано систему паузи, що є невід'ємним компонентом для забезпечення комфортного ігрового досвіду. Ця система дозволяє гравцю тимчасово призупинити ігровий процес, що не тільки надає можливість для перерви, але й слугує центральним вузлом для доступу до важливих функцій.

На рисунку 3.7, у правому верхньому куті екрана було додано іконку паузи. При її активації ігровий процес зупиняється, а на екран накладається інтерфейс меню паузи. Візуально це реалізовано за допомогою напівпрозорої панелі, яка затемнює ігровий світ на фоні, концентруючи увагу користувача на елементах меню, але водночас зберігаючи контекст гри. У центрі панелі розміщено заголовок "ПАУЗА", що чітко інформує гравця про поточний стан.

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		42

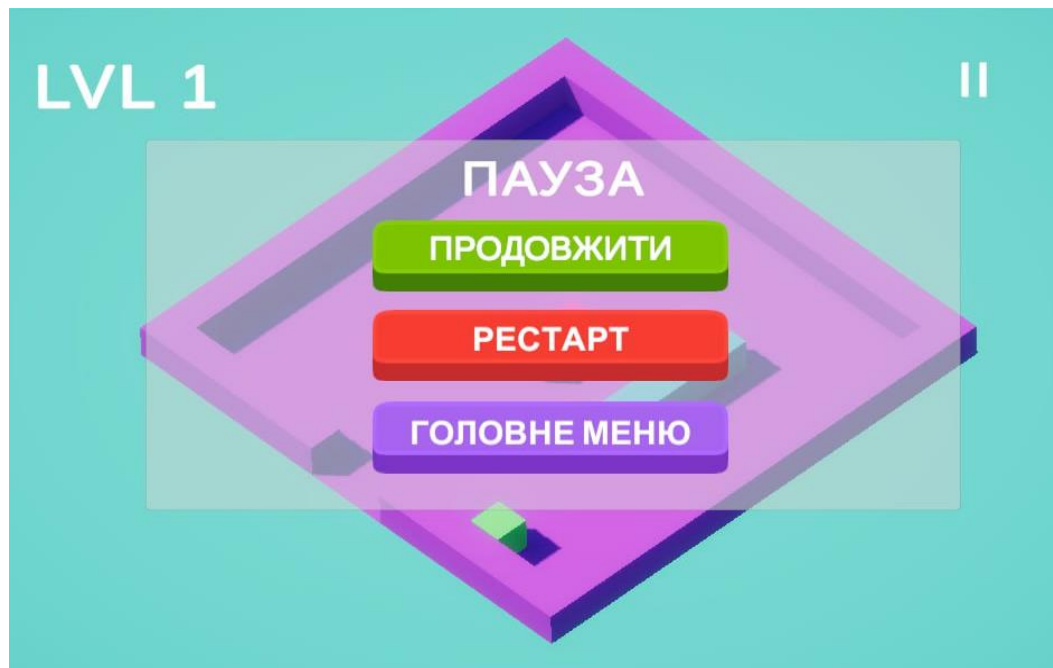


Рисунок 3.7 – Меню паузи

Нижче заголовка розташовано три основні функціональні кнопки, розроблені з урахуванням принципів інтуїтивності та зручності для користувача. Зелена кнопка “ПРОДОВЖИТИ” дозволяє гравцю закрити меню та миттєво повернутися до перерваного моменту гри. Червона кнопка “РЕСТАРТ” надає можливість негайно перезапустити поточний рівень, що є критично важливим для жанру головоломок, де часто виникає потреба почати спробу заново. Фіолетова кнопка “ГОЛОВНЕ МЕНЮ” забезпечує вихід з поточної ігрової сесії.

Технічно реалізація виконана за допомогою стандартної системи Unity UI, що включає об'єкти Canvas, Panel та Button. Вибір кольорової палітри та форми кнопок (яскраві, контрастні кольори та заокруглені прямокутники) свідомо наслідує загальну мінімалістичну та стилізовану графіку гри, створюючи цілісний та гармонійний візуальний досвід. Таким чином, на даному етапі проєкт отримав повноцінну та функціональну систему паузи, що значно покращує юзабіліті та надає гравцеві необхідний контроль над ігровим процесом.

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		43

Для керування функціональністю меню паузи, візуальний інтерфейс якого було розглянуто раніше, створено спеціалізований скрипт `PauseMenu.cs`. Цей компонент є центральним вузлом, що обробляє дії користувача в меню та керує станом ігрової сесії, зокрема зупинкою та відновленням ігрового часу.

Механізм зупинки гри та керування часом:

Ключова функціональність паузи реалізована через маніпуляцію системним часом гри за допомогою `Time.timeScale`.

Фрагмент коду метод `TogglePauseMenu()`:

```
public void TogglePauseMenu()
{
    pauseMenuPanel.SetActive(!pauseMenuPanel.activeSelf);

    if (pauseMenuPanel.activeSelf)
    {
        Time.timeScale = 0f;
    }
    else
    {
        Time.timeScale = 1f;
    }
}
```

Фрагмент коду метод `ContinueGame()`

```
public void ContinueGame()
{
    pauseMenuPanel.SetActive(false);
    Time.timeScale = 1f;
}
```

Метод `TogglePauseMenu` призначений для виклику кнопкою паузи в основному ігровому інтерфейсі. Він інвертує активний стан панелі меню (`pauseMenuPanel.SetActive(!pauseMenuPanel.activeSelf)`), роблячи її видимою або невидимою. Головне тут — керування властивістю `Time.timeScale`. Встановлення її значення в `0f` ефективно "заморожує" весь ігровий час, зупиняючи фізичні симуляції, анімації та виклик методу `FixedUpdate`. Це і є механізм паузи. Встановлення значення `1f` повертає плин часу до нормального. Метод `ContinueGame`, прив'язаний до кнопки "Продовжити", виконує ту саму дію — ховає меню і відновлює час.

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		44

Для реалізації функцій перезапуску рівня та виходу в головне меню використовуються методи класу SceneManager.

Фрагмент коду метод RestartGame()

```
public void RestartGame()  
{  
SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex)  
;  
    Time.timeScale = 1f;  
}  
public void LoadMainMenu()  
{  
    SceneManager.LoadScene("mainmenu");  
    Time.timeScale = 1f;  
}
```

Метод RestartGame (для кнопки “Рестарт”) використовує SceneManager.LoadScene() для перезавантаження поточної активної сцени, отримуючи її індекс через SceneManager.GetActiveScene().buildIndex. Метод LoadMainMenu (для кнопки "Головне меню") завантажує сцену за її назвою, в даному випадку "mainmenu". В обох методах критично важливим є рядок Time.timeScale = 1f. Він гарантує, що при завантаженні нової сцени ігровий час буде відновлено до нормального значення, адже стан Time.timeScale не скидається автоматично при переході між сценами. Без цього сцена завантажилася б, але залишалася б “на паузі”.

Після реалізації ключових програмних компонентів проєкту, розробку базового функціонала можна вважати завершеною. На даний момент у грі імплементовано систему керування персонажем, основні ігрові механіки на основі взаємодії активаторів та динамічних перешкод, а також створено повноцінний інтерфейс користувача, що включає меню паузи та елементи навігації між сценами.

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		45

З цього моменту проєкт переходить до наступної логічної та критично важливої фази — безпосереднього створення ігрових рівнів. На даному етапі фокус розробки зміщується на розробку ігрових рівнів. Головною задачею стає не написання нового коду, а вмiле конструювання унікальних та цікавих завдань для гравця, використовуючи вже готовий набiр iнструментiв та механiк.

Першим було створено “Рiвень 1” (Рисунок 3.8), що виконує функцiю вступного та навчального етапу для гравця. Дизайн цього рiвня є навмисно простим та мiнiмалiстичним, що вiдповiдає загальнiй стилiстичнi гри та його дидактичним цiлям. Iгровий простiр являє собою короткий, звивистий коридор, утворений за допомогою кiлькох статичних перешкод рiзної форми та кольору, якi розмiщенi на основнiй платформi.

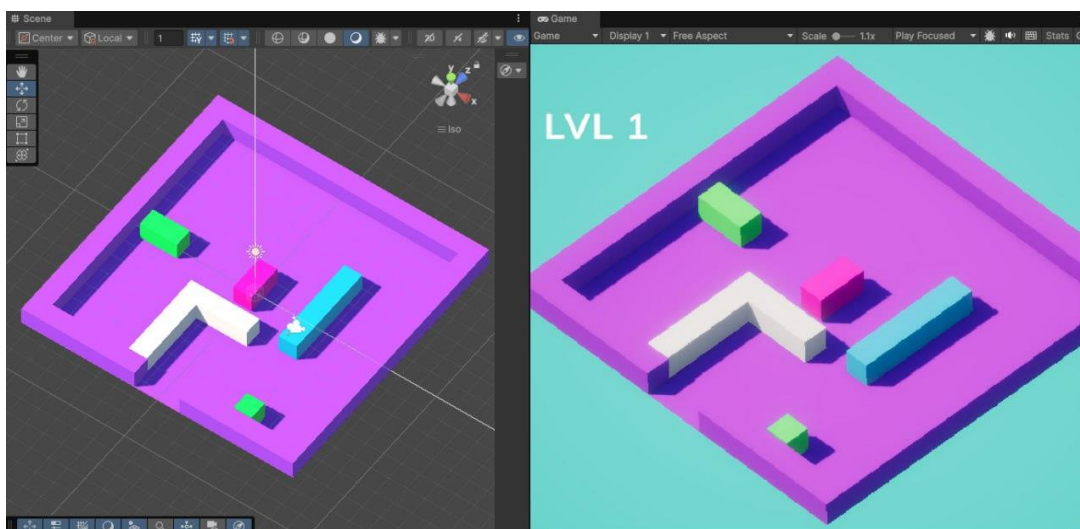


Рисунок 3.8 – Рiвень 1

Основна мета цього рiвня полягає не у створеннi складноi логiчноi задачi, а в плавному ознайомленнi гравця з фундаментальною механiкою гри — керуванням рухом iгрового персонажа. Проходячи цей простий маршрут вiд стартової точки до фiнiшу, гравець має можливiсть на практицi освоїти

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Пiдпис	Дата		46

елементи керування, відчуті фізику руху куба, його інерцію та особливості взаємодії з твердими перешкодами.

Наступним кроком у проєктуванні ігрового досвіду є “Рівень 2” (Рисунок 3.9), який логічно продовжує та розвиває концепції, закладені на першому етапі, підвищуючи при цьому вимоги до навичок гравця. Якщо перший рівень був вступним, то другий слугує для перевірки та закріплення отриманих навичок керування.

На відміну від простого та лінійного коридору першого рівня, структура другого є значно складнішою та щільнішою. Ігровий простір заповнено більшою кількістю статичних перешкод, які розташовані таким чином, що формують вузький та заплутаний лабіринт. Це змушує гравця не просто рухатися вперед, а ретельно планувати свій маршрут, уникати тупиків та виконувати більш точні маневри.

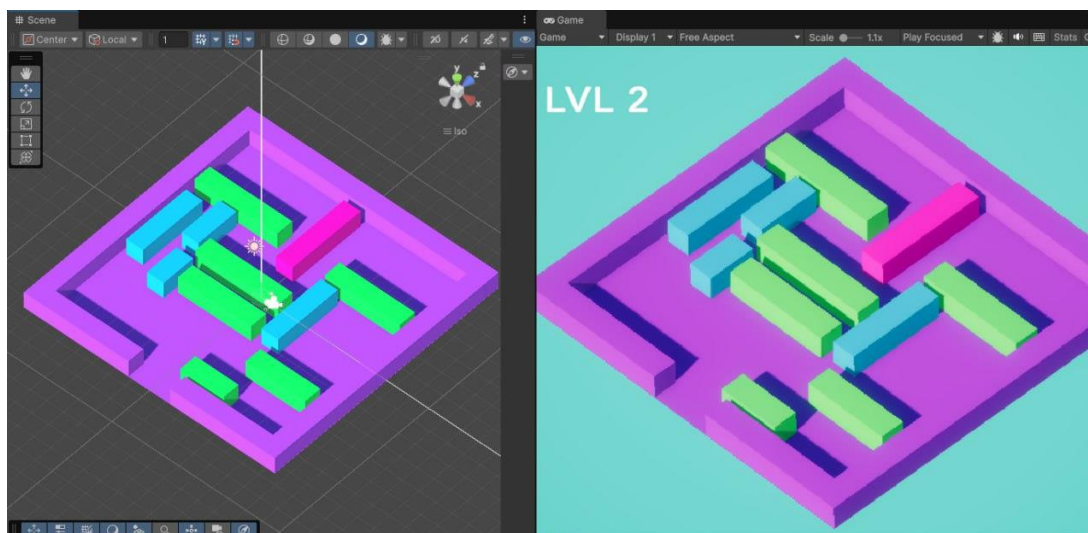


Рисунок 3.9 – Рівень 2

Основний виклик для гравця тут зміщується від простого освоєння керування до його впевненого та майстерного застосування. Рівень вимагає кращої просторової орієнтації та більш тонкого контролю над рухом куба. Як і на попередньому етапі, тут навмисно не використовуються складні

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		47

інтерактивні механіки, такі як кнопки чи динамічні блоки. Це свідоме дизайнерське рішення, спрямоване на те, щоб гравець повністю освоїв та відточив основний навик — навігацію — перед тим, як гра почне вводити нові логічні елементи. Таким чином, “Рівень 2” є важливим елементом у кривій складності гри, що слугує мостом між базовим навчанням та майбутніми повноцінними головоломками.

Третій рівень є поворотним моментом у ігровому процесі та кривій навчання, оскільки саме на ньому відбувається введення ключової головоломної механіки, що кардинально змінює характер викликів для гравця. Після того, як гравець засвоїв та відточив основи навігації на попередніх двох рівнях, “Рівень 3” представляє йому нові, раніше розроблені інтерактивні елементи: систему активаторів (жовті циліндричні кнопки на підлозі) та динамічних перешкод (жовті блоки, що перекривають шлях).

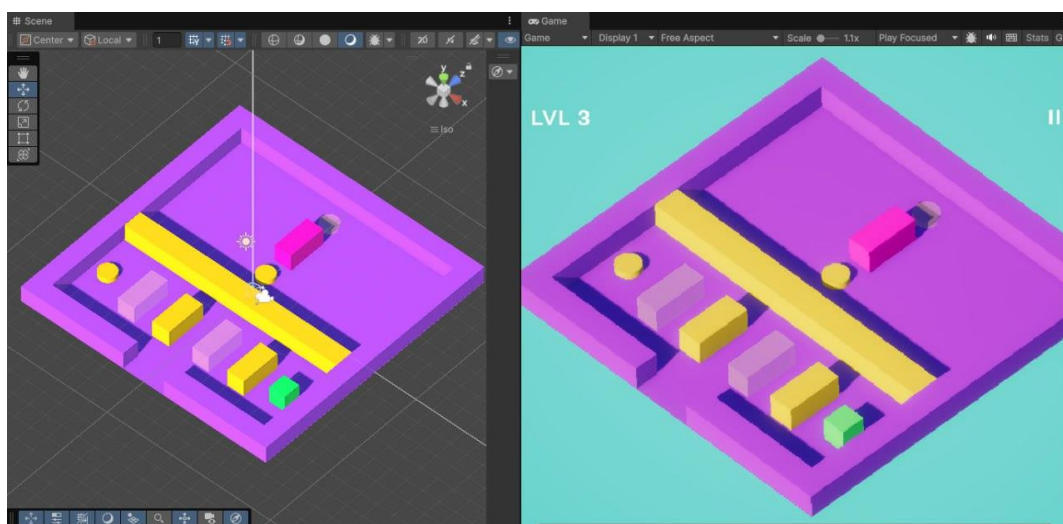


Рисунок 3.10 – Рівень 3

Дизайн рівня побудований таким чином, щоб підштовхнути гравця до експериментального дослідження цієї нової взаємодії. На локації розміщено два активатори та дві динамічні перешкоди.

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		48

Логіка головоломки, що базується на раніше розробленому скрипті Activator.cs, полягає у необхідності послідовного перемикання стану цих перешкод. Натискання на одну кнопку робить одну перешкоду прохідною (прозорою та без колізії), але водночас може робити іншу перешкоду твердою. Щоб дістатися до фінішу, гравець повинен не просто знайти шлях, а й розробити правильну послідовність дій для маніпуляції ігровим оточенням.

Цей рівень ефективно поєднує раніше засвоєний навик точної навігації з новим шаром логічного мислення. Тепер успіх залежить не тільки від моторних навичок гравця, але і від його здатності аналізувати ситуацію, розуміти причинно-наслідкові зв'язки та прогнозувати результат своїх дій. Таким чином, "Рівень 3" виконує свою дидактичну функцію, успішно навчаючи гравця ключовій механіці гри та переводячи її з жанру простої навігаційної аркади до повноцінної логічної головоломки.

3.3 Тестування гри

Тестування є невіддільним та критично важливим етапом у процесі розробки будь-якого ігрового проєкту, включно з грою-головоломою. Його головна мета — не лише виявлення та виправлення програмних помилок (багів), але й забезпечення коректної роботи ігрових механік, збалансованості та позитивного досвіду для гравця. Для цього проєкту процес тестування допоміг покращити загальну надійність та продуктивність гри, а також переконатися, що кінцевий продукт відповідає початковому задуму.

Процес перевірки якості розпочався з тестування, яке було зосереджене на окремих компонентах гри. На цьому етапі перевірялася функціональність кожного елемента, наприклад, коректність руху об'єкта гравця при натисканні клавіш, правильність збереження прогресу в PlayerPrefs при завершенні рівня, а також логіка спрацювання активаторів при взаємодії кубів.

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		49

Тестування функціоналу додатку:

На пристрої Android головне меню успішно завантажується та дозволяє гравцеві безперешкодно перейти до екрана вибору рівня. Екран вибору рівня, як показано на зображенні, також функціонує бездоганно. Кнопки вибору рівнів (1, 2, 3) є достатньо великими для зручного натискання на сенсорному екрані та успішно завантажують відповідну ігрову сцену.

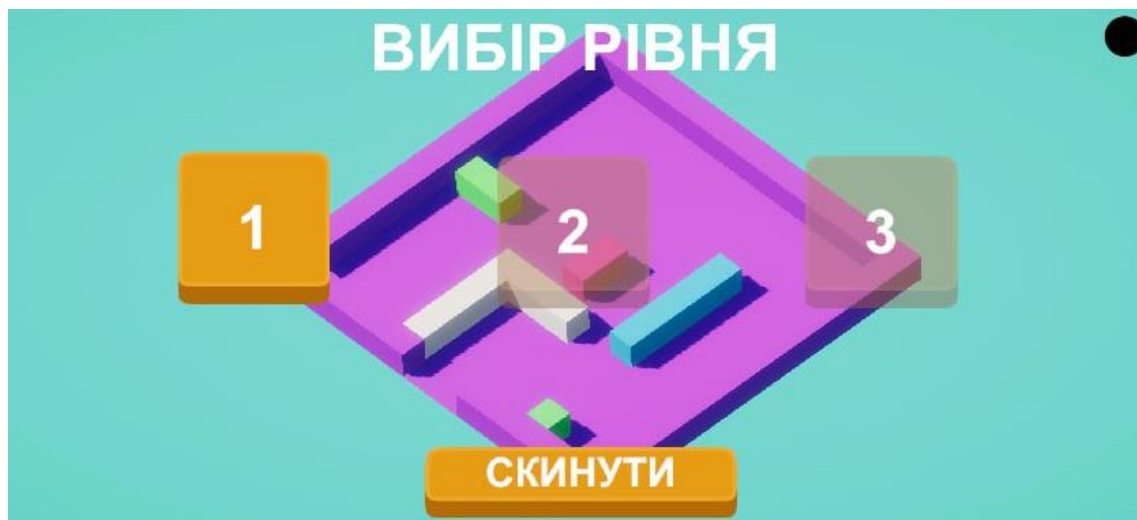


Рисунок 3.11 – Меню вибору рівнів

При першому запуску, як показано на Рисунку 3.11, гравцеві доступний лише перший рівень. Інші рівні візуально позначені як заблоковані за допомогою напівпрозорості, і стануть доступними для гри лише після того, як буде пройдено попередній.

Після вибору першого рівня з меню, ігрова сцена успішно завантажується, що продемонстровано на Рисунку 3.12. Тестування підтвердило, що ігровий процес запускається справно, а всі елементи на екрані відображаються коректно.

Інтерфейс користувача (UI) розташований правильно та не заважає огляду ігрового поля. У верхньому лівому куті екрана відображається індикатор поточного рівня (LVL 1), а у правому верхньому куті - функціональна кнопка паузи .

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		50

Особливу увагу було приділено адаптації інтерфейсу під сучасні екрани мобільних пристроїв: елементи UI враховують “безпечні зони” і не перекриваються вирізом для камери. Це підтверджує, що ігровий екран повністю функціональний, а його компоненти готові до використання на широкому спектрі пристроїв.

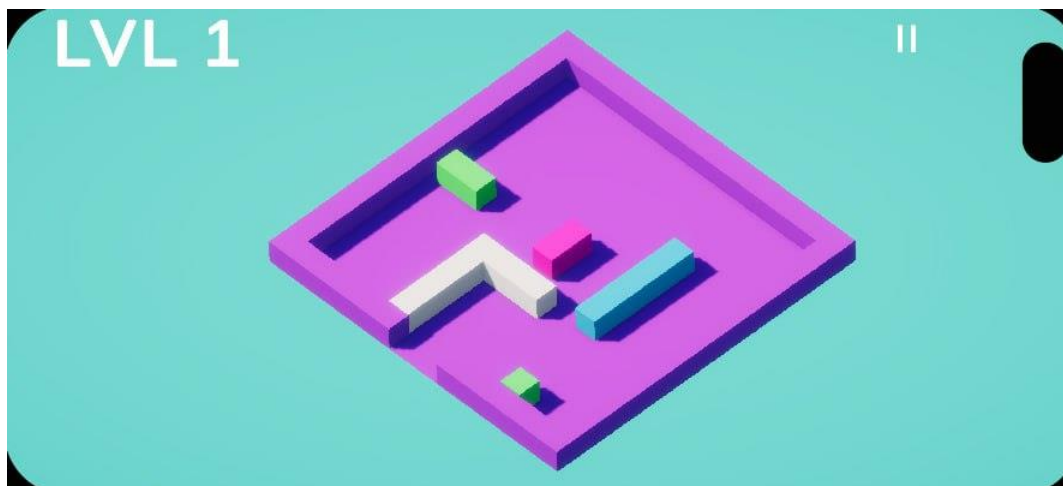


Рисунок 3.12 – Рівень 1

Під час тестування ігрового процесу було перевірено функціональність меню паузи, що продемонстровано на Рисунку 3.13. При натисканні на кнопку паузи у правому верхньому куті екрана, гра коректно зупиняється, а поверх ігрової сцени відображається відповідне меню. Його реалізовано у вигляді напівпрозорої панелі, що дозволяє гравцеві не втрачати візуальний контекст рівня.

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		51

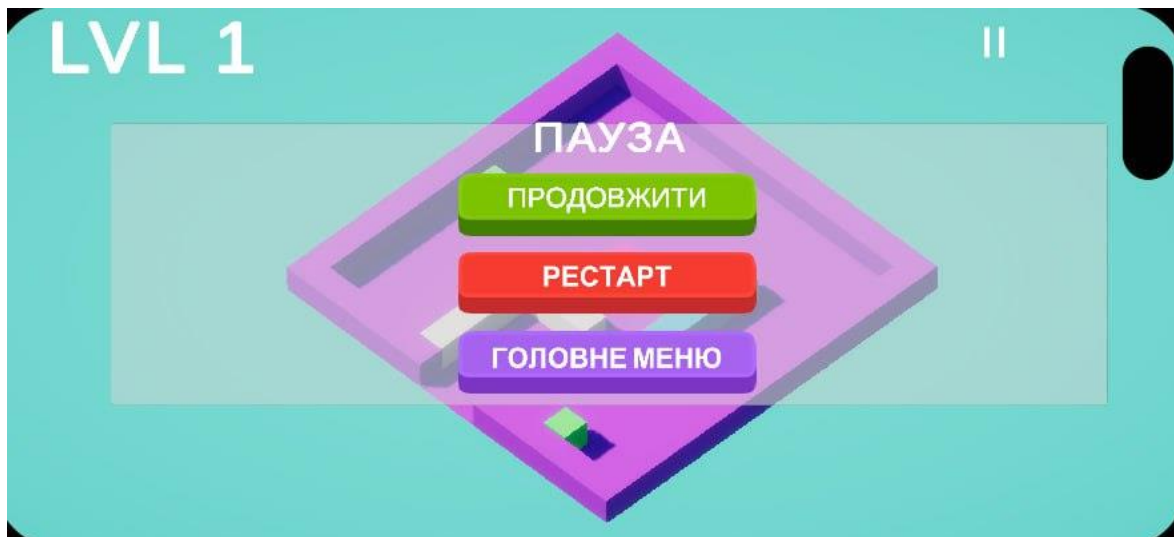


Рисунок 3.13 – Меню паузи

Тестування підтвердило, що всі кнопки меню паузи працюють справно. “Продовжити” — успішно закриває меню та відновлює ігровий процес з тієї ж точки.

“Рестарт” — коректно перезавантажує поточний рівень, повертаючи всі об’єкти на їхні початкові позиції.

“Головне меню” — здійснює вихід з ігрової сцени та повертає гравця на головний екран застосунку”

Особливу увагу під час тестування було приділено ключовій ігровій механіці — кнопкам-перемикачам, яка вводить новий рівень складності в головоломки. Тестування проводилося на прикладі третього рівня, що продемонстровано на рисунках 3.14 та 3.15.

На першому рисунку 3.14 показано початковий стан рівня, де певні шляхи заблоковані набором перешкод (прозорі та жовті блоки). На ігровому полі також присутні кнопки-активатори (жовті кола). Другий рисунок 3.15 демонструє результат активації однієї з кнопок, відповідна група перешкод (жовті блоки) змінила свій стан, ставши прозорою, що дозволяє гравцеві пройти крізь неї.



Рисунок 3.14 – Активована перша кнопка

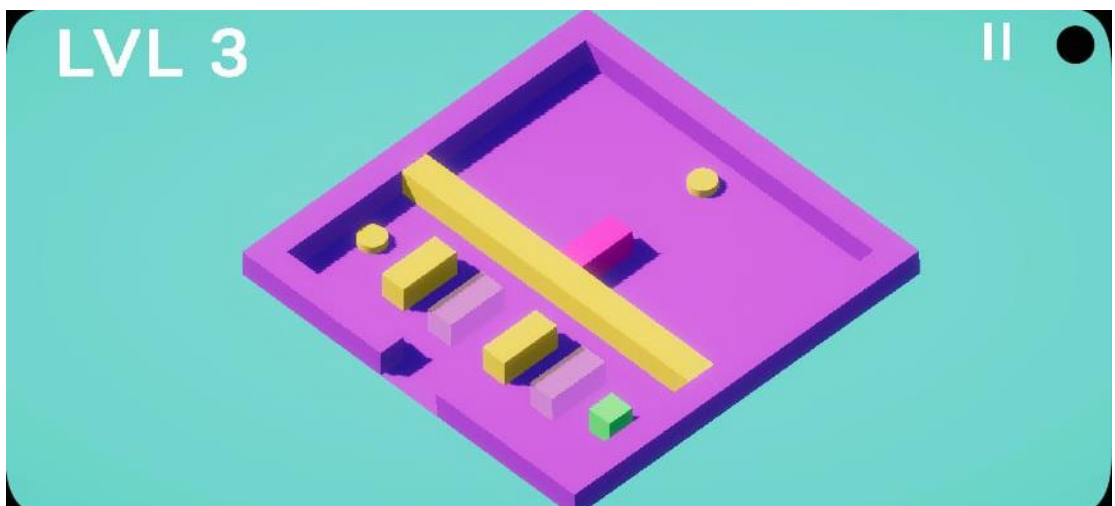


Рисунок 3.15 – Активована друга кнопка

Це підтверджує, що логіка зв'язку між кнопкою та її об'єктами працює коректно. Тестування показало, що механіка кнопок-перемикачів є повністю функціональною: кожна кнопка впливає лише на свою групу об'єктів, а візуальна зміна стану перешкод є чіткою та зрозумілою для гравця.

Змн.	Арк.	№ докум.	Підпис	Дата

Одним із ключових аспектів тестування була перевірка системи збереження прогресії, яка відповідає за розблокування нових рівнів. Тестування підтвердило її повну та надійну працездатність, що продемонстровано на Рисунках 3.14 та 3.15.

На першому зображенні (Рисунок 3.16) показано стан меню, коли гравець вже пройшов перший рівень, тому другий рівень є доступним для гри, а третій - ще заблокований.

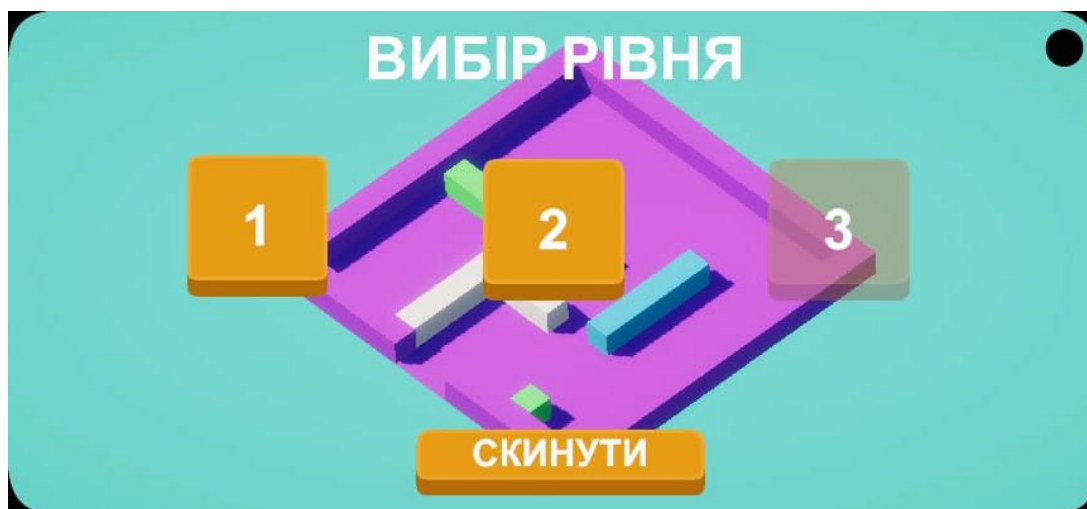


Рисунок 3.16 – Меню після проходження першого рівня

Після успішного проходження другого рівня та повернення до меню вибору, система коректно оновлює стан прогресу. Як видно на другому зображенні (Рисунок 3.17), третій рівень успішно розблокувався і став доступним для вибору.

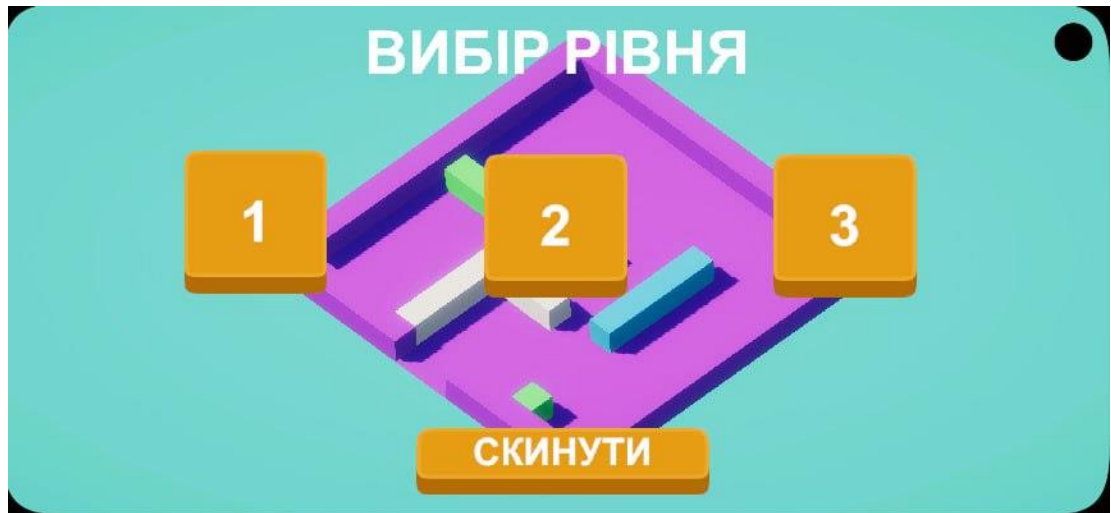


Рисунок 3.17 – Меню після проходження другого рівня

Цей процес підтверджує, що дані про проходження рівнів успішно зберігаються за допомогою механізму PlayerPrefs. Система надійно зберігає прогрес навіть після повного закриття та повторного відкриття застосунку, забезпечуючи безперервність ігрового досвіду для користувача.

Проведене тестування ігрового застосунку «PuzzleMind» продемонструвало його повну працездатність та стабільність. Перевірка на платформі Android підтвердила коректну роботу всіх елементів інтерфейсу, від головного меню до меню паузи. Було встановлено, що система збереження прогресу надійно працює, правильно розблоковуючи нові рівні після проходження попередніх.

Ключові ігрові механіки, зокрема кнопки-перемикачі, функціонують без помилок та відповідно до ігрового дизайну. Всі виявлені під час тестування незначні недоліки були успішно виправлені. Отже, можна констатувати, що фінальний продукт є якісним, відповідає поставленим вимогам та готовий до використання.

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		55

4 ТЕХНІКО-ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ

4.1 Аналіз ринку збуту продукту

Мобільна гра в жанрі головоломки є програмним продуктом, орієнтованим на користувачів смартфонів з операційною системою Android. Сучасний ринок мобільних ігор має стабільно високі темпи зростання. Зокрема, ігри типу casual, до яких належать головоломки, мають широкую аудиторію серед усіх вікових категорій — від дітей до літніх людей.

Розроблена гра відрізняється унікальним ігровим механізмом, продуманим візуальним стилем та інтуїтивним інтерфейсом. На ринку існує велика кількість конкурентів, таких як Candy Crush Saga, Brain Out, Monument Valley, проте запропонований продукт має свої переваги — наприклад, локалізацію українською мовою, цікаві механіки, що підвищує зацікавленість користувача.

Гра буде безкоштовною але в грі буде присутній платний контент у вигляді косметики для персонажа.

Потенційними покупцями є власники смартфонів, користувачі маркетплейсів Google Play. Очікується, що основним каналом розповсюдження стане Google Play (через модель умовно безкоштовного доступу з рекламою або внутрішніми покупками).

Очікуваний попит — середній. Стимулювання зростання попиту планується за рахунок маркетингових заходів у соціальних мережах та реклами через мобільні рекламні мережі. Сервісне обслуговування (техпідтримка, оновлення, фікси) буде здійснюватися автором або невеликою командою.

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		56

4.2. Розрахункова частина

Розрахунок витрат на розробку гри головоломки:

Ця сума включає оплату праці команди з 3 спеціалістів, залучених до створення гри. До складу команди входитимуть:

- Розробник Unity (1 особа) – 60000 грн за 4 місяці
- UI/UX дизайнер (1 особа) – 45000 грн за 4 місяці
- Тестувальник та контент-менеджер (1 особа) – 48000 грн за 4 місяці

Загальна зарплата розробників: 153000 грн.

Ця сума включає оплату праці команди розробників, які будуть залучені до створення гри. В команду входитимуть фахівці з різних галузей. З урахуванням етапів проектування, кодування, тестування та налагодження, розробка триватиме близько 4 місяців.

Єдиний соціальний внесок: 33600 грн.

Окремим рядком у витратах передбачено єдиний соціальний внесок, який згідно з чинним законодавством України становить 22% від суми нарахованої заробітної плати. ЄСВ — це обов'язковий платіж, що сплачується роботодавцем до державного бюджету для забезпечення соціального страхування працівників, зокрема пенсійного, медичного та страхування на випадок безробіття. У межах даного проекту сума ЄСВ становила 33 660 грн, що відображає соціальну відповідальність роботодавця й реальні витрати, пов'язані з офіційною оплатою праці команди.

Витрати на програмне забезпечення: 4400 грн.

У процесі реалізації проекту використано як безкоштовне, так і платне програмне забезпечення. Основним інструментом для розробки гри стало середовище Unity, яке у версії Personal Edition є безкоштовним для розробників, чий дохід не перевищує встановлений поріг. Це дало змогу зекономити значну частину бюджету без втрати функціональності або якості продукту. ля створення графічного інтерфейсу, ігрових елементів та дизайну

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		57

екранів активно використовувались програми Figma Pro та Adobe Photoshop. Підписка на Figma протягом трьох місяців обійшлася в 1800 грн, а Adobe Photoshop використовувався в рамках стандартного пакету Creative Cloud, ліцензія якого на місяць складає від 650 грн. Використання Photoshop було особливо доцільним для створення якісних текстур, обробки ілюстрацій та експорту графіки у високій роздільності. Комбінація цих інструментів забезпечила швидку й ефективну реалізацію візуальної частини проєкту, що відіграє ключову роль у сприйнятті гри цільовою аудиторією.

Витрати на тестування та розгортання: 22950 грн (15% від зарплати).

До цієї суми входить оплата праці тестувальника, закупівля доступу до інструментів для автоматизованого тестування.

Інші прямі витрати: 38250 грн (25% від зарплати)

Сюди входять витрати на:

- зв'язок (інтернет, мобільний зв'язок);
- канцелярію;
- оренду приміщення для роботи;
- підвищення кваліфікації (курси, ліцензії);
- транспорт;
- реклама в соцмережах;

Усього прямих витрат: 252260грн.

Крім того, до проєкту було включено накладні витрати (76500 грн), які охоплюють утримання приміщення, оплату комунальних послуг, амортизацію техніки, послуги адміністративного характеру.

Також було закладено плановий резерв у розмірі 51000 грн. Це забезпечує гнучкість проєкту у разі змін, нових вимог чи потреби розширення функціоналу гри.

Кошторисна вартість проєкту розробки мобільної гри головоломки становить 382860 грн.

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		58

Згідно з чинним податковим законодавством, додатково необхідно сплатити податок на додану вартість (ПДВ) у розмірі 20% від кошторисної вартості проєкту, що становить 76572 грн.

Отже, загальна вартість розробки гри дорівнює: 456332 грн.

4.3 Обґрунтування необхідності розробки

Гра-головоломка ідеально відповідає сучасним запитам користувачів на якісне, корисне та розвиваюче дозвілля. У світі, де інформаційний потік постійно зростає, а час стає дедалі ціннішим, люди шукають не лише розваги, а й можливості для самовдосконалення. Саме тут на перший план виходить наша гра, яка не просто заповнює вільний час, а й активно сприяє покращенню когнітивних здібностей.

Продуманий дизайн гри безпосередньо впливає на розвиток пам'яті, стимулює логічне мислення та значно покращує концентрацію уваги. Розуміючи, що сучасний користувач цінує доступність і інтерактивність, тому гра розроблена так, щоб її можна було використовувати будь-де і будь-коли. Незалежно від того, чи це перерва на роботі, поїздка в громадському транспорті чи вечірній відпочинок вдома, гра пропонує зручний спосіб тренувати мозок та весело проводити час.

Що отримає користувач:

- Інтелектуальний розвиток користувачів, зокрема школярі, студенти, офісні працівники та люди літнього віку, зможуть ефективно тренувати логіку, увагу та пам'ять.
- Заміна пасивного відпочинку активною грою, яка не просто розважає, а й стимулює мислення та сприяє постійному самовдосконаленню.
- Екологічна та економічна вигода а саме зменшення потреби у друкованих тренажерах і паперових головоломках, що економить кошти користувачів та позитивно впливає на довкілля за рахунок скорочення використання паперу.

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		59

– Гнучкість у використанні - додаток не потребує спеціального обладнання, працює на більшості сучасних смартфонів і не залежить від місця перебування користувача, забезпечуючи можливість грати офлайн.

Загалом, розробка гри головоломки є економічно доцільною завдяки високому попиту на розвивальні ігри, доступності смартфонів у широкій аудиторії, конкурентним перевагам продукту та очікуваному позитивному впливу на користувачів.

Додатково варто зазначити, що впровадження гри дозволить залучити нових користувачів та сформувати лояльну спільноту, адже гра поєднує як розважальну, так і навчальну функції. Зручний інтерфейс, можливість грати офлайн, різні рівні складності та відсутність агресивної монетизації роблять гру привабливою для широкого кола користувачів. Крім того, інтеграція гри із соціальними мережами та функція «Поділитися з другом» сприятимуть вірусному поширенню, що підвищить популярність проєкту і, відповідно, його прибутковість.

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		60

ВИСНОВКИ

Результатом виконання даної кваліфікаційної роботи є розроблена та готова до використання ігрова програма в жанрі головоломки для ОС Android.

У ході роботи було детально розглянуто та реалізовано всі етапи створення тривимірної головоломки, призначеної для розвитку логічного мислення та просторової уяви гравців. Створений ігровий продукт є повноцінним та завершеним, а його функціональність повністю відповідає всім вимогам, що були визначені на етапі проєктування. Тестування підтвердило стабільність роботи застосунку та коректність реалізації ключових ігрових механік.

Під час розробки застосунку були використані сучасні інструменти та технології для створення ігор, зокрема ігровий рушій Unity та мова програмування C#. Цей процес дозволив отримати безцінний практичний досвід у проєктуванні ігрових рівнів, реалізації ігрової логіки та оптимізації продукту. Крім того, виконання роботи сприяло поглибленню та закріпленню знань, отриманих під час вивчення таких дисциплін, як об'єктно-орієнтоване програмування, комп'ютерна графіка та архітектура програмного забезпечення.

Загалом, створення гри стало важливим етапом у професійному становленні, оскільки надало глибоке розуміння повного циклу розробки інтерактивних продуктів — від ідеї до фінальної реалізації. Завдяки виконанню цього проєкту вдалося не лише застосувати теоретичні знання на практиці, але й розвинути навички у вирішенні нетривіальних технічних завдань, що є ключовим для індустрії розробки ігор. Створений застосунок є повноцінним проєктом, що демонструє здатність розробляти якісне програмне забезпечення.

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		61

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Unity Learn. Unity Learn: вебсайт.
URL: <https://learn.unity.com/> (дата звернення: 10.02.2025).
2. Game Programming Patterns. Game Programming Patterns: вебсайт.
URL: <https://gameprogrammingpatterns.com/> (дата звернення: 17.02.2025).
3. C# & Unity [UA]. Проект на C#, YouTube: вебсайт.
URL: <https://www.youtube.com/watch?v=-6BJ3vEApMg> (дата звернення: 26.02.2025).
4. NSDG. Посібник для інтерфейсу Unity C#. Sharp Coder Blog: вебсайт. URL: <https://uk.sharpcoderblog.com/blog/introduction-to-interfaces-in-unity-3d-csharp> (дата звернення: 07.03.2025).
5. Вивчення Unity – Створення інтерфейсу користувача, YouTube: вебсайт. URL: <https://www.youtube.com/watch?v=qWdf79pl4e4> (дата звернення: 21.03.2025).
6. Unity - Manual: Unity 6.1 User Manual. Unity - Manual: Unity 6.1 User Manual: вебсайт. URL: <https://docs.unity3d.com/Manual/index.html> (дата звернення: 05.04.2025).
7. Android game development – AndroidDevelopers: вебсайт.
URL: <https://developer.android.com/games/engines/unity/unity-on-android> (дата звернення: 18.04.2025).

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		62

ДОДАТКИ

Додаток А

Лістинг програмного коду скрипта Activator:

```
using UnityEngine;
using UnityEngine.SceneManagement;
public class Player : MonoBehaviour
{
    [SerializeField] KeyCode keyOne;
    [SerializeField] KeyCode keyTwo;
    [SerializeField] Vector3 moveDirection;
    private bool isMovingPositive = false;
    private bool isMovingNegative = false;
    private Rigidbody rb;
    private int currentSceneIndex;
    private int nextSceneIndex;
    void Start()
    {
        rb = GetComponent<Rigidbody>();
        currentSceneIndex =
SceneManager.GetActiveScene().buildIndex;
        nextSceneIndex = currentSceneIndex + 1;
    }
    public void LoadNextScene()
    {
        if (nextSceneIndex <
SceneManager.sceneCountInBuildSettings)
        {
            SceneManager.LoadScene(nextSceneIndex);
        }
        else
        {
            Debug.Log("LAST LEVELss!");
        }
    }
    void UnlockNewLevel()
    {
        if (SceneManager.GetActiveScene().buildIndex >=
PlayerPrefs.GetInt("ReachedIndex"))
        {
            PlayerPrefs.SetInt("ReachedIndex",
SceneManager.GetActiveScene().buildIndex + 1);
            PlayerPrefs.SetInt("UnlockedLevel",
PlayerPrefs.GetInt("UnlockedLevel", 1) + 1);
            PlayerPrefs.Save();
        }
    }
    private void FixedUpdate()
```

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		63

```

    {
    if (Input.GetKey(keyOne) || isMovingPositive)
        {
            rb.linearVelocity += moveDirection;
        }
    if (Input.GetKey(keyTwo) || isMovingNegative)
        {
            rb.linearVelocity -= moveDirection;
        }
    if (Input.GetKey(KeyCode.R))
        {
            SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex);
        }
    if (Input.GetKey(KeyCode.Q))
        {
            SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
        }
    }
    public void OnPointerDownMovePositive()
    {
        isMovingPositive = true;
    }
    public void OnPointerUpMovePositive()
    {
        isMovingPositive = false;
    }
    public void OnPointerDownMoveNegative()
    {
        isMovingNegative = true;
    }
    public void OnPointerUpMoveNegative()
    {
        isMovingNegative = false;
    }
    private void OnTriggerEnter(Collider other)
    {
        if (this.CompareTag("Player") &&
other.CompareTag("Finish"))
            {
                UnlockNewLevel();
                LoadNextScene();
            }
        if (this.CompareTag("Cube") &&
other.CompareTag("Cube"))
            {
                foreach (Activator button in
FindObjectsOfType<Activator>())
                    {
                        button.canPush = false;
                    }
            }
    }

```

					КР.КН 25.587.06.000 ПЗ	Арк.
ЗМН.	Арк.	№ докум.	Підпис	Дата		64

```

        }
    }
    private void OnTriggerExit(Collider other)
    {
        if (this.CompareTag("Cube") &&
other.CompareTag("Cube"))
        {
            foreach (Activator button in
FindObjectOfType<Activator>())
            {
                button.canPush = true;
            }
        }
    }
}

```

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		65

Додаток Б

Лістинг програмного коду скрипта LoadNextScene:

```
using UnityEngine;
using UnityEngine.SceneManagement;
public class SceneLoader : MonoBehaviour
{
    public void LoadNextScene()
    {
        int currentSceneIndex =
SceneManager.GetActiveScene().buildIndex;
SceneManager.LoadScene(currentSceneIndex + 1);
    }
}
```

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		66

Додаток В

Лістинг програмного коду скрипта PauseMenu:

```
using UnityEngine;
using UnityEngine.SceneManagement;
public class PauseMenu : MonoBehaviour
{
    public GameObject pauseMenuPanel;
    public void TogglePauseMenu()
    {
        pauseMenuPanel.SetActive(!pauseMenuPanel.activeSelf)
;
        if (pauseMenuPanel.activeSelf)
        {
            Time.timeScale = 0f;
        }
        else
        {
            Time.timeScale = 1f;
        }
    }
    public void ContinueGame()
    {
        pauseMenuPanel.SetActive(false);
        Time.timeScale = 1f;
    }
    public void RestartGame()
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene()
.buildIndex);
        Time.timeScale = 1f;
    }
    public void LoadMainMenu()
    {
        SceneManager.LoadScene("mainmenu");
        Time.timeScale = 1f;
    }
}
```

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		67

Додаток Г

Лістинг програмного коду скрипта Activator

```
using UnityEngine;
public class Activator : MonoBehaviour
{
    public GameObject[] firstGroup;
    public GameObject[] secondGroup;
    public Activator button;
    public Material normal;
    public Material transparent;
    public bool canPush;
    private void OnTriggerEnter(Collider other)
    {
        if (canPush)
        {
            if (other.CompareTag("Cube") ||
other.CompareTag("Player"))
            {
                foreach (GameObject first in firstGroup)
                {
                    first.GetComponent<Renderer>().material
= normal;
                    first.GetComponent<Collider>().isTrigger
= false;
                }
                foreach (GameObject second in secondGroup)
                {
                    second.GetComponent<Renderer>().material
= transparent;
                    second.GetComponent<Collider>().isTrigge
r = true;
                }
                GetComponent<Renderer>().material =
transparent;
                button.GetComponent<Renderer>().material =
normal;
                button.canPush = true;
            }
        }
    }
}
```

					КР.КН 25.587.06.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		68