

Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
“ ____ ” _____ 2025 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи
інтелектуального аркадного мультимедіа проекту на основі
фрейворку Phaser”

Виконав здобувач вищої освіти
II курсу, групи КН-24М
ОПП «Комп’ютерні науки»
спеціальності 122 «Комп’ютерні науки»
_____ Нагірняк Д.В.
« ____ » _____ 2025 р.

Керівник проекту
кандидат технічних наук
_____ Лисенко І.А.
« ____ » _____ 2025 р.
Рецензент _____

АНОТАЦІЯ

Нагірняк Д.В. Дослідження та програмна реалізація системи інтелектуального аркадного мультимедіа проекту на основі фрейворку Phaser. 122 Комп'ютерні науки. Центральноукраїнський національний технічний університет. Кропивницький. 2025.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи інтелектуального аркадного мультимедіа проекту на основі фрейворку Phaser.

Метою розробки є дослідження та програмна реалізація системи інтелектуального аркадного мультимедіа проекту на основі фрейворку Phaser.

Об'єктом дослідження є процес інтелектуального аркадного мультимедіа проекту на основі фрейворку Phaser.

Предметом дослідження є методи інтелектуального аркадного мультимедіа проекту на основі фрейворку Phaser.

Методи дослідження базуються на методах теорії обробки графічної інформації, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи інтелектуального аркадного мультимедіа проекту на основі фрейворку Phaser.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ з ОС Windows 10/11.

Програму розроблено в середовищі фреймворку Phaser.

Ключові слова: Комп'ютерні науки, аркадний мультимедіа проект

ABSTRACT

Nahirniak D.V. Research and software implementation of the intelligent arcade multimedia project system based on the Phaser framework. 122 Computer Science. Central Ukrainian National Technical University. Kropyvnytskyi. 2025.

In this final qualification work for the second (master's) level of higher education, software has been developed, which is intended for the intelligent arcade multimedia project system based on the Phaser framework.

The purpose of the development is the research and software implementation of the intelligent arcade multimedia project system based on the Phaser framework.

The object of the research is the process of the intelligent arcade multimedia project based on the Phaser framework.

The subject of the research is the methods of the intelligent arcade multimedia project based on the Phaser framework.

The research methods are based on the methods of the theory of graphic information processing, methods of mathematical statistics, methods of software development.

The result of the work is a software implementation of the intelligent arcade multimedia project system based on the Phaser framework.

In the process of working on the software model, an analysis of existing hardware and software was performed. All components of the developed software are fully described.

A convenient user interface has been developed. Instructions for working with the software are provided.

The program can be used on a PC with OS Windows 10/11.

The program was developed in the Phaser framework environment.

Keywords: computer science, arcade multimedia project

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	6
1.1 Призначення системи.....	6
1.2 Область застосування.....	7
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	9
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	9
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	13
2.3 Розгорнута постановка завдання	14
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	15
3.1 Опис функціонування системи	15
3.2 Розробка структурної схеми.....	19
3.3 Розробка функціональної схеми	27
3.4 Розробка діаграми процесів.....	31
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	33
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	33
4.2 Захист розробленого програмного забезпечення.....	46
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	47
6 НАУКОВА НОВИЗНА	54

						ВКРМ-122.25.0049.00.00.ПЗ		
Вим.	Арк.	№ докум.	Підп.	Дата				
Розроб.	Нагірняк Д.В.				Дослідження та програмна реалізація системи інтелектуального аркадного мультимедіа проекту на основі фрейворку Phaser	Літ.	Аркуш	Аркушів
Перев.	Писенко І.А.					М	1	78
Н.контр.	Коваленко А.С.				ЦНТУ КН-24М			
Затв.	Смірнов О.А.							

7	МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ІТ-ПРОЄКТУ	55
7.1	Визначення цільової аудиторії кінцевого готового продукту	55
7.2	Оцінка привабливості шляхом застосування методів експертних оцінок ...	56
7.3	Вибір методу оцінки вартості ПЗ	56
7.4	Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості.....	57
7.5	Пропозиція алгоритму просування проєкту розробки ПЗ	59
7.6	Оптимізація каналів збуту та шляхів реалізації ПЗ	59
7.7	Визначення ключових факторів успіху конкретного проєкту.....	60
8	ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	61
8.1	Вступ.....	61
8.2	Пожежна безпека.....	62
8.3	Аналіз санітарно-гігієнічних умов праці на робочому місці програміста ...	64
8.4	Розробка заходів з умов поліпшення охорони праці.....	67
8.5	Розрахункова частина	68
9	ОСНОВНІ ВИСНОВКИ.....	70
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	72

КБПЗ-2025

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ПЗ – програмне забезпечення

КБПЗ – 2025

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

ВСТУП

Актуальність теми. Створення захопливих та неповторних ігрових вражень вимагає поєднання технічних навичок, креативності та глибокого розуміння психології гравця. Зі зростанням очікувань гравців та розвитком технологій розробники постійно розширюють межі можливостей, щоб запропонувати захопливий ігровий досвід. Цей динамічний ландшафт підкреслює зростання інновацій та розширення ігрової індустрії. В основі цієї трансформації лежать фреймворки та движки для розробки ігор, які надають необхідні інструменти та функції для втілення віртуальних світів у життя. Фреймворки для розробки ігор – це бібліотеки та платформи, що надають структурований набір ресурсів та інструментів для створення ігор на замовлення. Ці фреймворки спрощують процес розробки ігор, надаючи попередньо вбудовані функції, такі як фізичне моделювання, рендеринг графіки, управління звуком та керування вводом. Ці фреймворки дозволяють розробникам зосередитися на творчих аспектах гри для швидшої розробки та ефективнішого використання ресурсів.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи інтелектуального аркадного мультимедіа проекту на основі фрейворку Phaser.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

- Огляд існуючих систем інтелектуального аркадного мультимедіа проекту на основі фрейворку Phaser.
- Дослідження системи інтелектуального аркадного мультимедіа проекту на основі фрейворку Phaser.
- Програмна реалізація системи інтелектуального аркадного мультимедіа проекту на основі фрейворку Phaser.

Об'єктом дослідження є процес інтелектуального аркадного мультимедіа проекту на основі фрейворку Phaser.

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

Предметом дослідження є методи інтелектуального аркадного мультимедіа проекту на основі фрейворку Phaser.

Методи дослідження базуються на методах теорії обробки графічної інформації, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

– Удосконалено метод інтелектуального аркадного мультимедіа проекту на основі фрейворку Phaser.

– Розроблено вітчизняний продукт інтелектуального аркадного мультимедіа проекту на основі фрейворку Phaser, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі інтелектуального аркадного мультимедіа проекту на основі фрейворку Phaser.

Достовірність наукових результатів підтверджена теоретичними викладеннями, даними комп'ютерного моделювання, коректними дослідженнями параметрів на функціонуючій обчислювальній мережі, а також відповідністю отриманих результатів окремим результатам, наведеним у науковій літературі.

Робота апробована на LVII Науково-технічній конференції здобувачів вищої освіти LV науково-технічної конференції «Наука в ЦНТУ: основні досягнення та перспективи розвитку» (2025 р.), основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у статті збірника праць молодих науковців ЦНТУ, випуск №15.

Таким чином, виходячи з вищеперахованого, дослідження та програмна реалізація системи інтелектуального аркадного мультимедіа проекту на основі фрейворку Phaser, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Розробка ігор стосується процесу створення відеоігор, включаючи все: від початкової концепції та дизайну до кодування, тестування, огляду та випуску у виробничому середовищі.

Це багатогранна справа, яка включає мистецтво, розповідь історій, звуковий дизайн та кодування. Розробка ігор вимагає співпраці між програмістами, геймдизайнерами, художниками, звукорежисерами та аналітиками якості, щоб створити щось видатне.

Що таке ігрові двигуни та фреймворки?

Ігровий рушій – це програмна платформа, що пропонує бібліотеки, інструменти та API, необхідні для розробки графіки, візуалізації та відеоігор.

Ці ігрові двигуни забезпечують основні функції, необхідні для створення гри, такі як симуляція фізики, управління звуком, рендеринг графіки, системи анімації, середовище написання сценаріїв та обробка вхідних даних.

Деякі популярні ігрові рушії, які використовують розробники, включають Unity, CryEngine, Unreal Engine та Godot.

З іншого боку, фреймворки для розробки ігор – це компактні та гнучкі набори коду та бібліотек, які можна повторно використовувати. Фреймворки надають бібліотеки коду, які можна повторно використовувати, що виконують основні завдання розробки ігор, допомагаючи командам розробників проектувати архітектуру гри з нуля, не пишучи чи не впроваджуючи власні функції.

Phaser, Three.js, MonoGame, Pygame та LibGDX – це кілька прикладів фреймворків для розробки ігор.

Хоча ігрові фреймворки та рушії пов'язані між собою та використовуються разом для розробки ігор, вони не є однаковими.

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

1.2 Область застосування

Областю застосування є аркадні ігри. Деякі відеоігри можуть мати більший потенціал для залежності, ніж інші, завдяки своїм структурним характеристикам. Структурні характеристики стосуються особливостей відеоігор, які ініціюють, розвивають та підтримують залученість гравця з часом. Характеристики відеоігор різняться залежно від різних систем або платформ відеоігор. Наприклад, внутрішньоігрові функції аркадних ігрових автоматів суттєво відрізняються від спеціалізованих ігрових консолей. Ранні форми відеоігор були концептуалізовані як нефінансова форма азартних ігор через структурну схожість між аркадними відеоіграми та електронними гральними автоматами. Обидва типи автоматів мають барвисту графіку та звукові ефекти, ігрові винагороди та бонуси за виграшні ходи, цифрове відображення балів правильної поведінки та швидкий проміжок часу гри, який певною мірою залежить від майстерності гравця. Аркадні автомати також мають таблицю рекордів або соціально ранжований бал, який заохочує гравців змагатися та перевершувати своїх суперників. Дослідженнями в 1980-х роках було виявлено меншість підлітків, які регулярно грали на платних аркадних автоматах, іноді втікаючи зі школи, щоб пограти. Ці гравці витрачали на автомати вільний дохід, включаючи гроші на обід та проїзд, і часто крали гроші у батьків та/або інших осіб, щоб фінансувати свою звичку грати у відеоігри. Використовуючи стандартизований тест на патологічну азартну гру (термін «азартні ігри» замінено на «грати у відеоігри»), дослідники виявили, що деякі з цих підлітків-гравців відповідали критеріям залежності.

Сучасні відеоігри структурно складніші, ніж ранні аркадні автомати. Деякі дослідники запропонували п'ять основних категорій ознак для пояснення психологічної привабливості відеоігор. До них належать:

– соціальні ознаки (тобто соціалізаційні аспекти відеоігор, такі як комунікаційні функції, що сприяють створенню соціальних мереж);

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

– ознаки маніпулювання та контролю (тобто способи, за допомогою яких гравець може взаємодіяти та контролювати ігрові властивості за допомогою схеми фізичного контролю та створювати відчуття майстерності та контролю над грою);

– наративні та ідентичні ознаки (тобто способи, за допомогою яких гравець може прийняти іншу ідентичність у грі та стати частиною інтерактивного досвіду розповіді історій);

– ознаки винагороди та покарання (тобто способи, за допомогою яких гравці отримують підкріплення за вмілу гру (тобто виграш) та карають за програш);

– ознаки презентації (тобто естетичні якості відеогри, такі як те, як гра виглядає та звучить для гравця).

Ознаки винагороди та покарання мають найсильніший зв'язок з надмірною поведінкою у відеоіграх. Подібні спостереження були зроблені в дослідженнях проблемних азартних ігор, зокрема щодо гри на електронних ігрових автоматах. Теорія оперантного обумовлення припускає, що часті та змінні графіки підкріплення у відеоіграх роблять їх дуже психологічно захопливими та стійкими до поведінкового згасання. Очікування великих винагород та періодичних менших винагород призводить до швидкої та наполегливої ігрової поведінки. Багато відеоігрових залежних неодноразово виконують невеликі дії у відеогрі, щоб з часом отримати велику винагороду. Ця практика відома серед гравців як гриндінг, і вона може займати значну частину їхнього ігрового часу.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи інтелектуального аркадного мультимедіа проекту на основі фрейворку Phaser, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти

Найкращі ігрові двигуни та фреймворки, які формують ігрову індустрію та дають розробникам змогу втілювати свої ідеї в життя:

1. Unity

Unity, кросплатформний ігровий рушій, розроблений Unity Technologies і зарекомендував себе як одна з домінуючих сил в індустрії розробки ігор.

Це власне ліцензійне програмне забезпечення, яке дозволяє розробникам створювати 2D, 3D та захопливі ігри в реальному часі на понад 20 платформах за допомогою своїх комплексних інструментів та послуг.

Можливості рендерингу в реальному часі в Unity дозволяють створювати захопливі візуальні ефекти, а багате сховище ресурсів забезпечує велику колекцію ресурсів для пришвидшення розробки ігор. Залежно від вимог ігрового проекту, Unity пропонує безкоштовну, Plus та Pro версію на вибір.

2. Unreal Engine

Ще одним популярним ігровим движком серед розробників є Unreal Engine. Розроблений Epic Games, движок Unreal Engine є відомим вибором для розробників, які шукають неймовірні можливості рендерингу та приголомшливу графіку.

Unreal містить настроювані плагіни та попередньо вбудовану модульну систему, що робить його рушійною силою кількох AAA-ігор. Він також забезпечує безпрецедентну налаштування та гнучкість для розробників завдяки своїй надійній системі Blueprint (потужній візуальній системі сценаріїв) та підтримці програмування на C++.

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

Двигун Unreal має модульну архітектуру, що дозволяє розробникам підтримувати чистоту коду для легкої модифікації та оновлення гри.

Оснащений великою колекцією інструментів рендерингу та можливостей візуальних ефектів, Unreal пропонує широкий набір функцій, таких як аудіосистеми, симуляція фізики, інструменти анімації тощо.

3. CryEngine

CryEngine – потужна платформа для розробки ігор, розроблена, щоб допомогти розробникам створювати високоякісні 3D-ігри. Подібно до Unity та Unreal, CryEngine також має приголомшливу візуальну якість та можливості рендерингу в реальному часі, що робить її гарним вибором для розробки AAA-ігор.

Він постачається з редактором «пісочниці» для створення безшовних світів, рендерингом для ультрареалістичних середовищ та фізичними симуляціями для створення конфігурацій сцен реального світу.

Конвертація та оптимізація ресурсів у реальному часі в CryEngine пришвидшують процес розробки. Це найкращий вибір для створення ігор від першої особи, таких як Hunt: Showdown.

4. GoDot

Godot – це потужний ігровий рушій з відкритим кодом, який пропонує безкоштовну та універсальну платформу для розробки ігор на різних операційних системах. Він надає широкий набір інструментів, призначених для полегшення створення як 2D, так і 3D ігор, що робить його чудовим варіантом для незалежних розробників та невеликих студій.

Двигун відомий своєю мінімалістичністю, що забезпечує плавну роботу та ефективно використання ресурсів, що особливо корисно для проектів з обмеженим бюджетом або невеликими командами.

Однією з видатних особливостей Godot є гнучкі можливості написання скриптів, що дозволяють розробникам писати код різними мовами, а GDScript адаптований спеціально для цього движка.

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

Також, аспект Godot, орієнтований на спільноту, є ще однією значною перевагою, оскільки він сприяє постійним оновленням та вдосконаленням.

5. Phaser

Phaser – це 2D-фреймворк для розробки ігор на HTML5. Розроблений Photon Storm, Phaser – це безкоштовний, швидкий фреймворк з відкритим кодом, який внутрішньо використовує WebGL та рендерер Canvas і дозволяє автоматичне перемикання між ними залежно від підтримки браузера.

Активна спільнота Phaser також пропонує багату колекцію розширень та плагінів, які надають розробникам потрібні інструменти та ресурси для оптимізації розробки ігор.

Крім того, його простота використання та прості характеристики роблять його улюбленим серед розробників, які хочуть створювати кросплатформні або браузерні додатки.

6. MonoGame

MonoGame – це безкоштовний фреймворк з відкритим вихідним кодом, побудований на .Net, розроблений MonoGame Foundation за ліцензією Microsoft Public License.

Цей фреймворк є водночас простим та надійним, що дозволяє розробникам створювати ігри для різних платформ, включаючи настільні комп'ютери, мобільні пристрої та ігрові консолі, використовуючи C# та інші мови програмування .Net.

MonoGame підтримує низку платформ, таких як Windows, Linux, iOS, Xbox, PlayStation, Android та macOS. Він використовувався для комерційних ігрових проектів і вирізняється тим, що не має моделі підписки, роялті, плати за виконання або ліцензійних витрат, пов'язаних з його використанням.

7. libGDX

libGDX є одним з найкращих фреймворків, доступних в ігровій індустрії. Це кросплатформний фреймворк для розробки, заснований на OpenGL API. libGDX написаний на Java з деякими компонентами C та C++ для коду, залежного від продуктивності.

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

Ліцензована під Apache 2, вона дозволяє розробникам створювати 2D та 3D ігри.

libGDX також має надійне та добре протестоване середовище, яке дозволяє швидке прототипування та швидкі ітерації. Крім того, він має активну підтримку спільноти та надійну документацію, яка допомагає новачкам зрозуміти, як він працює, та використовувати його для своїх проєктів.

8. PyGame

PyGame – це набір модулів, розроблених на Python для кросплатформної розробки ігор. Він використовує бібліотеки Simple DirectMedia Layer Library, viper, Mirra та PyUI2 для виконання низькорівневих завдань, таких як рендеринг графіки, керування аудіо та інші поширені функції розробки ігор.

Використовуючи PyGame, розробники можуть створювати повнофункціональні мультимедійні програми мовою Python.

Його популярність залежить від простоти, великої спільноти та гнучкості, які пропонує PyGame. Він використовується для розробки відеоігор, освітнього програмного забезпечення, симуляцій та інтерактивних художніх інсталяцій.

9. Panda3D

Panda3D – це кросплатформний, безкоштовний рушій з відкритим кодом для візуалізації, експериментів, симуляцій та, звичайно ж, ігор. Він має аудіо, графіку, введення/виведення, виявлення зіткнень тощо для створення реалістичних 3D-ігор. Panda3D поєднує простоту Python та швидкість C++, що дає розробникам можливість швидше створювати проєкти без шкоди для їхньої продуктивності. Створений для комерційної розробки ігор, Panda3D досі використовується з тією ж метою. Panda3D робить акцент на чотирьох аспектах: швидкості, потужності, стійкості до помилок та повноті, що робить його одним з провідних ігрових двигунів на ринку.

10. Cocos2d-X

Cocos2d-X – це модульний та гнучкий фреймворк з відкритим вихідним кодом для розробки 2D-ігор. Він базується на OpenGL ES 2.0 та Metal для

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

рендерингу графіки. Він має високу сумісність з платформами Android та легко налаштовується.

Cocos2d можна використовувати не лише для ігор, але й для розробки інших графічних програмних застосунків для різних платформ, таких як iOS, macOS, Windows, Android, Linux, HarmonyOS, Web та OpenHarmony. Фреймворк Cocos2d, створений за допомогою бібліотеки Pyglet, написаний на Python та C++.

Це легко та ефективно, з великою підтримкою спільноти. Більше того? Cocos2d є безкоштовним у використанні, тобто не вимагає підписки чи роялті для створення інтерактивних програм на основі графічного інтерфейсу.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Phaser.js – це швидкий, безкоштовний JavaScript-фреймворк з відкритим кодом, який використовується для створення ігор для настільних комп'ютерів та мобільних платформ. Він надає велику бібліотеку функцій, що дозволяють розробникам створювати ігри з багатою графікою, фізикою, звуком та анімацією.

Phaser.js – це популярний ігровий фреймворк HTML5, який дозволяє розробникам створювати захопливі 2D-ігри за допомогою JavaScript або TypeScript. Він забезпечує вбудовану підтримку фізичних двигунів, анімації, аудіо та управління ресурсами, що робить його комплексним рішенням для розробки ігор.

– Міжплатформна сумісність: ігри можна запускати на будь-якому пристрої.

– Рендеринг WebGL та Canvas: Для високої продуктивності Phaser автоматично вибирає оптимальний метод рендерингу для пристрою.

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи інтелектуального аркадного мультимедіа проекту на основі фрейворку Phaser.

В процесі розробки випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти необхідно виконати наступний обсяг роботи:

а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;

б) вибрати та обґрунтувати методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;

г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації в роботі технологічного обладнання;

д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;

з) сформулювати висновки про виконаний обсяг робіт та одержані результати.

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Аркадні ігри не зникли. Вони перетворилися на дизайнерську основу, яка продовжує формувати те, як будується цифрова взаємодія. Ранні машини змушували розробників зосереджуватися на швидкості, зрозумілості та реакції користувача. Обмежене обладнання та механіка оплати за гру призвели до ігор, які навчали гравців без інструкцій. Замість складних меню гравці навчалися на практиці.

Сьогоднішні студії часто застосовують принципи аркадних ігор, навіть не називаючи їх такими. Короткий ігровий цикл, зростання складності та зворотний зв'язок щодо винагороди досі є частиною найуспішніших продуктів. Дизайнери, які вивчають логіку аркадних ігор, розуміють, як утримувати увагу та залучати гравців. Погляд у минуле – це не ностальгія. Йдеться про використання перевірених методів, які працюють на сучасних платформах.

Незмінна привабливість фізичних ігрових просторів

Аркадні автомати досі привертають увагу, оскільки вони пропонують прямий та соціальний досвід. На відміну від домашніх консолей чи смартфонів, аркадні автомати вимагають від гравців діяти публічно. Таке середовище сприяє дружньому змаганняю та спільній грі. Діти, підлітки та дорослі часто збираються навколо автомата, щоб грати по черзі, порівнювати рахунки та реагувати разом. Навіть невеликі аркади в торгових центрах або розважальних центрах можуть генерувати довгі черги, коли автомати пропонують швидкий та корисний ігровий процес.

Фізичне керування також сприяє привабливості. Джойстик та набір кнопок із адаптивною реакцією дозволяють вводити дані, що відрізняються від сенсорного екрана. Мобільні аркадні ігри намагаються привнести подібну

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

динаміку в смартфони, імітуючи темп і чіткість класичних ігор. Розробники часто створюють їх з простими правилами, короткими раундами та швидким зворотним зв'язком. Ці механіки відображають сильні сторони аркадних ігор і допомагають мобільним іграм досягати масштабного успіху. Незалежно від того, чи розміщені вони у фізичному кабінеті, чи завантажені з магазину додатків, ігри, що дотримуються цього формату, продовжують добре працювати.

Сучасна монетизація ретро-досвіду

Ностальгія повертає гравців, але ефективна монетизація забезпечує сталий розвиток продуктів. Багато студій зараз відтворюють або перевидають ігри в ретро-стилі, щоб залучити аудиторію, яка пам'ятає оригінали. Використовуючи сучасні движки, такі як Unity, розробники можуть створювати мобільні ігри, які виглядають класично, але працюють з високою стабільністю. Розробка мобільних ігор на Unity спрощує публікацію на кількох платформах, одночасно інтегруючи рекламні моделі, покупки в додатках та підписки.

Методи монетизації різняться, але одна поширена структура включає пропонування основного ігрового процесу безкоштовно з платними покращеннями візуальних елементів, бонусів або нового контенту. Графіка в ретро-стилі в поєднанні із сучасною монетизацією часто призводить до сильної залученості. Розробники, які розуміють, як збалансувати емоційну привабливість зі структурованою монетизацією, як правило, досягають кращих результатів на переповненому ринку.

Наприклад, деякі мобільні ігри, створені на Unity, використовують функції таблиці лідерів та щоденні винагороди для підвищення утримання користувачів. Інші відтворюють відомі механіки, такі як піксельні платформери або фіксовані шутери, пропонуючи їх у коротких сесіях, що відповідають сучасним звичкам користувачів. Однієї лише ностальгії не гарантує успіху. Дохід надходить від точного вибору дизайну в поєднанні з доступними бізнес-моделями.

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

Популярність аркадних відеоігор у 2025 році

Ігрова аудиторія у 2025 році все ще виявляє значний інтерес до аркадних відеоігор. Короткі сесії, високі бали та цикли повторів продовжують приваблювати гравців усіх вікових груп. Розробники, які розуміють структуру та простоту, часто швидше охоплюють ширшу аудиторію. Замість того, щоб прагнути складних наративів, ігри в стилі аркад надають гравцям пряму мету та негайний зворотний зв'язок.

Нові релізи в цьому жанрі з'являються як у вигляді інді-проектів, так і в каталогах великих видавців. Файтинги, шутери з кулями, ритмічні випробування та класичні перегони є одними з найпопулярніших ігор у магазинах та на онлайн-платформах. Багато творців зараз поєднують аркадну механіку з оновленою візуальною складовою або гібридними жанрами, пропонуючи ігри, які здаються одночасно знайомими та актуальними.

У світі жанрів відеоігор аркадні формати залишаються одними з найстабільніших. Гравці часто перемикаються між масштабними іграми з відкритим світом та короткими аркадними іграми залежно від вільного часу чи настрою. Ця гнучкість робить аркадні відеоігри актуальними, особливо коли вони представлені в сучасному стилі та мають орієнтовану на користувача взаємодію.

Визначення принципів дизайну аркадних ігор, які працюють

Потужний дизайн аркадних ігор зосереджений на швидкій взаємодії, видимих результатах та утриманні гравців. Розробники використовують короткі цикли, негайну реакцію та чіткість цілей, щоб зосередити увагу. На відміну від жанрів з насиченим сюжетом, аркадні ігри часто пропускають експозицію та одразу переходять до викликів та винагород. Елементи керування зазвичай мінімальні, тому немає потреби в інструкціях з експлуатації. Робоча структура дизайну в аркадних іграх часто добре застосовується до сучасних мобільних, консольних та браузерних ігор. При правильному застосуванні ці принципи створюють ігри, які легко освоїти та в які важко відірватися.

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

Швидкий зворотний зв'язок та щільні цикли

Розробники аркадних ігор працюють із короткими циклами, які спонукають гравців діяти швидко та бачити результати негайно. Цей цикл підвищує залученість, змушуючи кожен секунду відчувати себе корисною. Найкращі ігри в цій категорії усувають затримку між дією та відповіддю. Коли ви натискаєте кнопку, щось відбувається одразу. Цей зворотний зв'язок навчає механіці без письмових пояснень.

Сучасні розробники застосовують цю модель на мобільних, VR та ПК платформах. Ключ у циклі: діяти, спостерігати, що відбувається, покращувати, повторювати. Вузькі цикли також полегшують балансування. Дизайнери можуть точно налаштувати швидкість, оцінювання та складність з меншим ризиком розриву довгих сюжетних ліній.

Ключові технології та платформи, що рухають сучасний дизайн аркадних ігор

Дизайн сучасних аркадних ігор залежить від поєднання передових двигунів, апаратного забезпечення та підключених сервісів. Розробники ігор використовують Unity, Unreal Engine та low-code інструменти для розробки як цифрових, так і кабінетних ігор. Мережеві функції, хмарне сховище та датчики розширюють можливості аркадної гри. На відміну від ранніх кабінетних ігор, обмежених апаратним забезпеченням, сучасні системи дозволяють грати в багатокористувацькому режимі, відстежувати рахунок у реальному часі та масштабовану доставку контенту. Вибір правильної платформи або налаштування обладнання залежить від цілей гри та фізичного середовища розгортання.

Рішення Unity, Unreal та Low-Code для шаф

Розробники ігор сьогодні часто обирають рушії, виходячи з необхідної гнучкості, вартості та простоти розгортання. Unity широко використовується для аркадних проєктів завдяки своїй портативності та ефективній підтримці 2D та 3D. Багато комерційних ігрових рушіїв зараз запускають ігри, створені на Unity, на

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

рушіїв для розробників усіх рівнів кваліфікації.

Кому варто використовувати ігровий движок Phaser?

Фазер ідеально підходить для:

– Розробники ігор: Phaser пропонує зручний API та розширену документацію.

– Незалежні розробники: безкоштовність, відкритий код та масштабованість Phaser приваблюють невеликі команди та незалежних творців.

– Професіонали: Студії можуть використовувати Phaser для швидкого прототипування та створення ігор на різних платформах.

Залежно від потреб та вимог, Phaser може бути підходящим фреймворком для багатьох команд розробників ігор.

Переваги ігрового двигуна Phaser

– Відкритий код та безкоштовний: Phaser знаходиться у вільному доступі, а активна спільнота робить внесок у його розробку та підтримку.

– Підтримка кросплатформності: ігри, створені за допомогою Phaser, можуть працювати у веббраузерах, настільних комп'ютерах та мобільних пристроях.

– Розширена документація: Вичерпні посібники, навчальні посібники та приклади роблять навчання та використання Phaser простим.

– Гнучкість рендерингу: Розробники можуть вибирати між WebGL та Canvas, оптимізуючи ігри для різних пристроїв та потреб у продуктивності.

– Багата екосистема плагінів: Різноманітні плагіни розширюють можливості Phaser, від фізичних двигунів до розширеної анімації.

Ігровий движун Phaser та Incredibuild

Incredibuild покращує ігровий движок Phaser, пришвидшуючи процес збірки та розгортання. Розподіляючи обчислювальні завдання між кількома машинами, Incredibuild скорочує час збірки ресурсів та коду.

Однією з головних переваг фреймворку є його простота використання, і це, ймовірно, одна з причин, чому він набрав такої популярності за такий

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

короткий проміжок часу (фреймворку трохи більше року).

Фактично, Phaser абстрагує всю складну математику, яка зазвичай потрібна для створення гри, надаючи вам більше, ніж просто ігрові компоненти. Це дозволяє вам пропустити частину, яку ви витрачаєте на роздуми про те, як можна реалізувати певну спеціальну функцію та який рівень обчислення вона вимагає.

Як працює Phaser.js?

Ігри Phaser розробляються з використанням об'єкта конфігурації гри, який визначає такі властивості, як розміри, тип рендерингу, налаштування фізики та сцени. Кожна сцена має три основні методи життєвого циклу.

- `preload()`: Завантажує такі ресурси, як зображення та звуки.
- `create()`: Ініціалізує об'єкти та логіку гри.
- `update()`: Виконується безперервно для обробки логіки гри та взаємодії з користувачем.

Порівняння з іншими ігровими двигунами

– Phaser проти Cocos2d-x: Cocos2d-x підтримує розробку як 2D, так і 3D ігор і є кросплатформним. Однак, зосередженість Phaser на HTML5 робить його дуже придатним для швидкої ітерації.

– Phaser проти Unity 2D: Unity пропонує потужні інструменти для розробки як 2D, так і 3D ігор, забезпечуючи розширену підтримку редакторів та платформ. Phaser, з іншого боку, легший та простий у вивченні, що робить його ідеальним для розробників, які зосереджені на веб-іграх.

– Phaser проти Three.js: Хоча Three.js використовується для створення анімованої 3D-графіки у веб-браузерах, Phaser зосереджений на розробці 2D-ігор, що робить його кращим для браузерних ігор з насиченим 2D-контентом.

Структурна схема розробленої системи зображена на рисунку 3.1. На ній показано структуру розробленої системи інтелектуального аркадного мультимедіа проекту на основі фрейворку Phaser. Вона включає в себе:

- Двигун Phaser.js.
- Блок функцій Phaser.js.
- Блок застосувань.
- Блок введення даних.

Блок функцій Phaser.js

Phaser пропонує широкий спектр функцій, які роблять розробку ігор дуже простою та плавною.

- Багаті фізичні системи: Phaser підтримує кілька фізичних движків, таких як Arcade Physics, Matter.js та ImpactJS, які надають розробникам інструменти для виявлення зіткнень, гравітації, прискорення тощо.

- Керування сценами та станами: Phaser пропонує керування сценами та станами, що спрощує створення меню, ігрових рівнів та переходів.

- Завантаження ресурсів: Phaser спрощує керування ресурсами завдяки вбудованим завантажувачам ресурсів для зображень, аудіо, відео, спрайтів, тайлмап тощо.

- Анімація: Створення анімації в Phaser стає дуже простим завдяки вбудованій підтримці спрайтових аркушів та анімації на основі кадрів.

- Обробка вводу користувача: Phaser підтримує різноманітні методи введення, включаючи клавіатуру, мишу, сенсорне керування та навіть ігрові контролери.

- Керування аудіо: Phaser забезпечує вбудовану підтримку відтворення аудіо, включаючи звукові ефекти, музику та аудіопетлі.

- Керування камерою та вікном перегляду: Система камер Phaser дозволяє контролювати видиму область гри, забезпечуючи функції прокручування, масштабування та обертання.

- Тайлмапи: Завдяки підтримці рівнів на основі тайлмапів ви можете

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

швидко створювати величезні ігрові світи та структуровані рівні.

– Вбудовані ігрові об'єкти: Phaser надає вбудовані об'єкти, такі як спрайти, текстові об'єкти, кнопки, графіка та фігури. Це зменшує необхідність створювати ігрові компоненти з нуля.

Блок застосувань

Блок застосувань включає в себе:

– Казуальні веб-ігри: Phaser ідеально підходить для створення браузерних ігор завдяки легкій конструкції та сумісності з браузерами.

– Освітні ігри: Цей фреймворк часто використовується для розробки освітніх ігор для дітей завдяки своїй простоті та інтерактивним можливостям.

– Розробка мобільних ігор: Завдяки функціям, адаптованим для мобільних пристроїв, легко розробляти кросплатформні мобільні ігри з підтримкою сенсорного керування та акселерометра.

– Прототипи та MVP: Розробники часто використовують Phaser для швидкого створення прототипів та мінімально життєздатних продуктів (MVP).

– Ігрові зустрічі: Завдяки швидкому циклу розробки, Phaser є популярним вибором для ігрових зустрічей та заходів зі швидкого прототипування.

Блок введення даних

Сенсорні екрани, сенсори та альтернативні входи. Сучасні ігрові автомати використовують більше, ніж просто кнопки та джойстики. Розробники тепер створюють ігри, які використовують сенсорні екрани, датчики руху та натискні панелі для підтримки нових типів взаємодії. Цей зсув відкриває простір для дизайну ритм-ігор, ігор, орієнтованих на фітнес, або захопливих тренувань.

Сенсорні екрани поширені в іграх на основі квитків та мобільних портах, адаптованих для кабінетів. Їхня низька крива навчання робить їх ідеальними для коротких публічних ігор. Датчики жестів, такі як Leap Motion або інфрачервоні трекери, додають цінності інтерактивним музейним експозиціям або фітнес-додаткам.

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

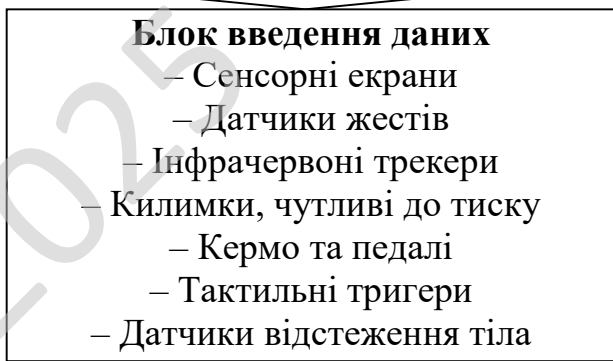
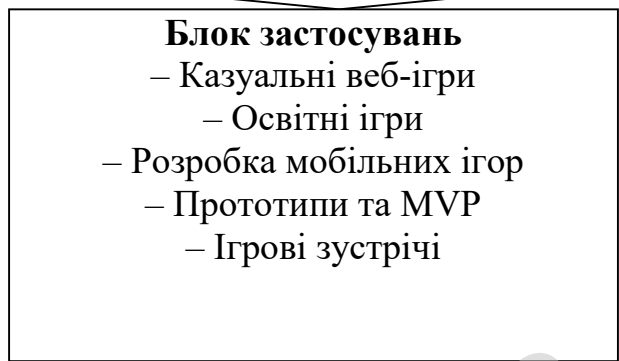
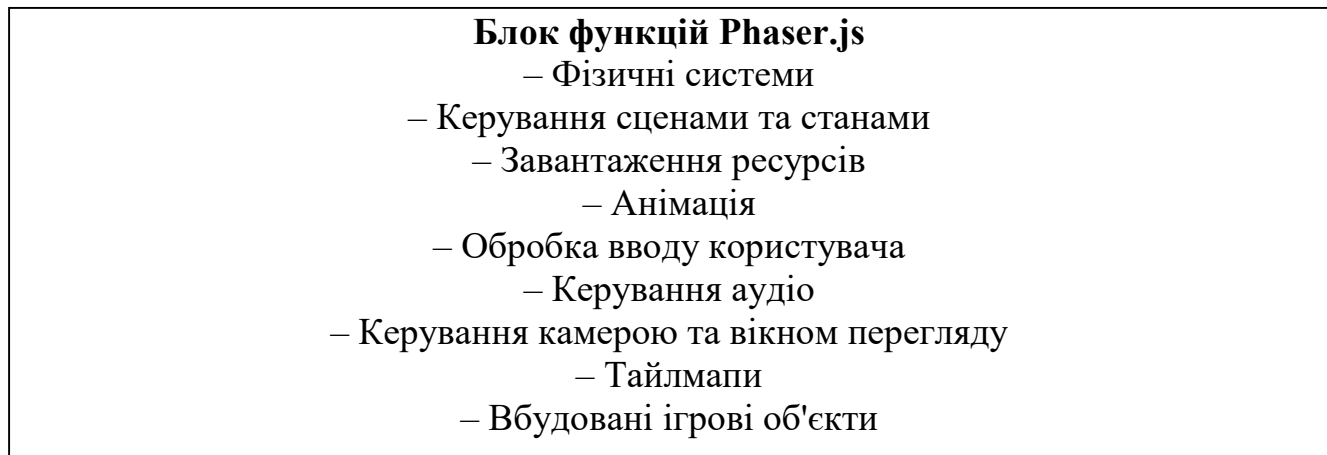


Рисунок 3.1 – Структурна схема системи

Альтернативні засоби введення виходять за рамки дотику:

1. Килимки, чутливі до тиску. Використовуються в танцювальних іграх або ігрових залах для фізичної реабілітації.
2. Кермо та педалі. Застосовується в гоночних іграх для додаткового реалізму.
3. Тактильні тригери. Забезпечують зворотний зв'язок у стрілялках або екшн-іграх.
4. Датчики відстеження тіла. Дозволяють грати всім тілом без носимих пристроїв.

Використання різноманітних вхідних даних дозволяє розробникам

охоплювати нових гравців, адаптуватися до доступності та підтримувати ширший спектр фізичних просторів.

Тенденції розробки ігор на 2025 рік і далі

Якщо ви зайшли так далеко, вам може бути цікаво дізнатися про тенденції розробки ігор, які змінять та трансформуватимуть ігрову індустрію в найближчі роки.

Нижче ми перерахували тенденції, які допоможуть вам охопити більше людей і збільшити ваш дохід.

1. Захоплюючі ігри

Імерсивні ігри покращують розширену реальність (XR), поєднуючи віртуальну реальність, доповнену реальність та змішану реальність для забезпечення розширеного ігрового досвіду.

Завдяки захопливим іграм ми рухаємося до більш реалістичних ігор. Поєднуючи реалістичну графіку та високоякісний звуковий дизайн, розробники тепер готові втілити в життя ігрових персонажів та сюжети, дозволяючи гравцям глибше зануритися в гру.

З розвитком технологій та їхньою доступністю, захопливий ігровий досвід приваблюватиме ширшу аудиторію, стимулюючи подальші інновації в ігровій індустрії.

Кросплатформний геймплей: Ігровий ландшафт зазнає значних змін у 2025 році та надалі зі зростанням поширеності кросплатформної розробки ігор.

Зростаюча кількість розробників забезпечують безперебійну гру для гравців на різних платформах, таких як Xbox, PlayStation, ПК та мобільних пристроях. Цей зсув знаменує собою занепад ексклюзивних для платформ багатокористувацьких режимів, що дозволяє гравцям грати з родиною та друзями на різних пристроях.

Не тільки це, це також дозволяє іграм перемикатися між платформами, зберігаючи свій прогрес, зберігаючи свої серії незмінними, щоб їх не довелося починати спочатку.

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

2. Хмарні ігри

Зростання популярності хмарних ігор набирає обертів як серед гравців, так і серед розробників, завдяки віддаленим серверам. Хоча хмарні ігри існують вже деякий час, акцент змістився на те, щоб зробити їх більш доступними та захопливими за допомогою передових технологій.

Завдяки таким сервісам, як NVIDIA GeForce NOW та Xbox Cloud Gaming, гравці тепер мають можливість насолоджуватися іграми найвищого рівня практично на будь-якому пристрої. Ця інновація руйнує бар'єри та сприяє розвитку активних, пов'язаних ігрових спільнот.

3. Штучний інтелект та машинне навчання

Штучний інтелект (ШІ) та машинне навчання (МН) трансформують відеоігри, покращуючи занурення та взаємодію. Очікується, що в найближчі кілька років неіграбельні персонажі (NPC) вийдуть за рамки сценарної поведінки, адаптуючись до стратегій гравців та навчаючись на їхніх діях.

Ми також можемо передбачити процедурно згенерований контент, який розвиватиметься в режимі реального часу, дозволяючи ігровим середовищам та сюжетним лініям змінюватися залежно від вибору гравця.

Це створить персоналізовані наративи, поглиблюючи зв'язок гравців зі світом гри та його персонажами. Загалом, інтеграція штучного інтелекту та машинного навчання призведе до більш динамічного ігрового досвіду з високою реграбельністю, оскільки кожна сесія пропонує унікальні завдання та історії, адаптовані до індивідуальних стилів гри. Майбутнє ігор виглядає інтерактивним та високо персоналізованим.

4. Реалістична графіка

Удосконалення ігрової графіки стрімко розвивається, і інтеграція технології трасування променів відіграє значну роль у цьому прогресі. У майбутньому ми можемо очікувати, що ігри демонструватимуть ще більш захоплюючі та реалістичні візуальні ефекти.

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

Технологія трасування променів імітує поведінку світла в реальному світі, створюючи неймовірно реалістичні відблиски, тіні та світлові ефекти. Це вдосконалення обіцяє перетворити ігровий досвід на кінематографічні пригоди.

Для тих, хто цінує ігри з сюжетом та приголомшливою візуальною складовою, потенційний вплив трасування променів на естетичну якість ігор є особливо захопливим, оскільки це, ймовірно, підніме загальну графічну точність до безпрецедентного рівня.

Вартість розробки ігор

Вартість розробки гри може суттєво варіюватися, починаючи приблизно від 35 000 доларів США до понад 600 000 доларів США. На цю велику різницю у вартості розробки впливає багато факторів, які впливають на загальну складність та обсяг проекту.

Прості 2D-ігри з обмеженим функціоналом часто мають менші витрати, тоді як 3D-ігри з насиченим сюжетом та складними механізмами вимагають вищого бюджету.

Деякими іншими факторами, що впливають на цей спектр витрат, є розмір та кваліфікація команди розробників, їхнє географічне розташування, жанр гри, обрана графіка та стиль малювання, час розробки, використані інструменти та технології, платформа, для якої розробляється гра, інтеграції (наприклад, покупки в додатку або підтримка багатокористувацької гри), залежності від сервісів або програмного забезпечення, звуковий дизайн, дизайн інтерфейсу користувача та вибір платформ та движків розробки.

3.3 Розробка функціональної схеми

Функціональна схема розробленої системи зображена на рисунку 3.2.

З рисунку видно, що розроблена система складається з наступних частин:

– 2D-рендеринг: Phaser підтримує спрайти, тайлмапи та текстури, що забезпечує детальну та яскраву 2D-графіку.

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

- Фізичні системи: Інтегровані фізичні двигуни, такі як Arcade Physics та Matter.js, дозволяють створювати реалістичну взаємодію об'єктів.
- Інструменти анімації: Створюйте динамічні анімації за допомогою вбудованих інструментів для спрайт-листів та послідовностей на основі кадрів.
- Підтримка аудіо: Легке додавання музики та звукових ефектів за допомогою аудіо API Phaser.
- Обробка введення: Підтримка керування за допомогою клавіатури, миші, сенсорного керування та геймпада.
- Керування активами: Ефективно завантажуйте та керуйте ігровими активами, включаючи зображення, аудіо та JSON-файли.

Серед поширених інструментів зворотного зв'язку є:

- Спливаючі вікна з рахунками. Числа з'являються миттєво після успіху або невдачі.
- Звукові підказки. Кожна дія супроводжується звуком, який підтверджує введення або результат.
- Спалах або кольорові ефекти. Візуальні реакції підсилюють ритм і допомагають гравцям навчатися.
- Реакція на вхідні дані. Затримки уникаються, тому елементи керування відчуються пов'язаними з рухом.

Навіть у нових форматах ці системи зворотного зв'язку продовжують забезпечувати надійну взаємодію.

Однохвилинний гачок: стратегія ранньої взаємодії

Перші враження вирішують, чи гравці продовжать гру, чи припинять її. Дизайн аркадних ігор вирішує цю проблему за допомогою чіткої стратегії: показати основний цикл протягом першої хвилини. Ця структура так само добре застосовується в мобільних іграх, веб-іграх та коротких навчальних модулях.

Замість складного адаптаційного процесу, одномінутна приманка замінює пояснення діями. До моменту завершення першого раунду гравець має розуміти, що робити, чому це важливо та як виглядають винагороди.

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

потрібно відстежувати.

– Миттєвий перезапуск. Програш не змушує користувачів повертатися до меню. Гра швидко відновлюється.

Зосереджуючись спочатку на швидкості та зрозумілості, розробники підвищують утримання та заохочують формування звички у ігрових циклах.

Системи підрахунку очок та мотивація гравців

Очки допомагають гравцям вимірювати прогрес і порівнювати результати. Це єдине число стає причиною повертатися, вдосконалюватися та змагатися. В аркадному дизайні рахунок – це не лише результат, а й рушійна сила поведінки.

Сучасні ігри часто замінюють або покращують результати таблицями лідерів, значками чи індикаторами прогресу, але основна ідея залишається. Вимірний результат дає людям щось, з чим можна змагатися з самими собою чи іншими.

Мотивуючі системи оцінювання зазвичай мають чіткі структури:

1. Миттєве відображення. Результати з'являються одразу після дії.
2. Послідовні правила. Гравці знають, як збільшити або втратити очки.
3. Багаторівневі пороги винагород. Певні бали розблоковують новий контент, скіни або визнання.
4. Соціальне порівняння. Результати можна ділитися або порівнювати між користувачами.
5. Динамічне масштабування. Пізніші рівні пропонують більше очок, що підвищує азарт.

Логіка підрахунку очок заслуговує на пильну увагу під час створення або оцінювання проєкту в аркадному стилі. Вона перетворює короткі ігри на повторювані враження та дає гравцям привід повертатися.

Хмарні таблиці лідерів та інтеграції з багатокористувацьким режимом

Раніше табло результатів було фізичним. Сьогодні розробники створюють ігри з хмарними таблицями лідерів, які оновлюються в режимі реального часу. Ця

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

зміна впливає на мотивацію гравців та дизайн системи. Замість того, щоб писати ініціали на машині, користувачі тепер бачать, як вони порівнюються з гравцями з усього світу.

Додавання підтримки хмарних технологій дає аркадним іграм друге життя поза межами ігрової шафи. Користувачі можуть відстежувати рейтинги зі своїх телефонів або повернутися, щоб перевершити суперника пізніше. Така механіка сприяє формуванню звички до гри та створює справжню конкуренцію.

Багатокористувацькі можливості також додають нову цінність. Розробники можуть використовувати такі сервіси, як Photon, PlayFab або Firebase, щоб підключити кілька аркадних кабінетів або дозволити мобільним користувачам приєднуватися ззовні.

Ось кілька особливостей:

1. Оновлення рейтингу в режимі реального часу. Результати синхронізуються з центральним сервером.
2. Синхронізація між пристроями. Оцінки та профілі переміщуються між кабінетом та мобільним пристроєм.
3. Підбір гравців у сеансі. Деякі ігри автоматично підбирають гравців у змагальних режимах.
4. Зберігання статистики гравця. Історія ігор та прогрес у рейтингу відстежуються у хмарі.

Онлайн-інтеграція наближає аркадні ігри до мейнстрімних цифрових ігор, не втрачаючи при цьому їхнього основного дизайну.

3.4 Розробка діаграми процесів

Діаграма процесів розробленої системи зображена на рисунку 3.3. При детальному її розгляді можна побачити як саме проходить взаємодія у розробленій системі. Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі.

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

Діаграма взаємодії процесів використовується для візуалізації процесів обробки даних (структурне проектування). Для розробника вважається звичним спочатку креслити діаграму взаємодії процесів даних рівня контексту, завдяки чому буде показано взаємодію системи. Ця діаграма в подальшому підлягає уточненню шляхом деталізації процесів та потоків даних з метою показати систему що розробляється.

Діаграми потоків даних містять чотири типи елементів:

- Процеси які являють собою трансформацію даних в рамках описуваної системи.
- Сховища даних (репозиторії).
- Зовнішні по відношенню до системи сутності.
- Потоки даних між елементами трьох попередніх типів.

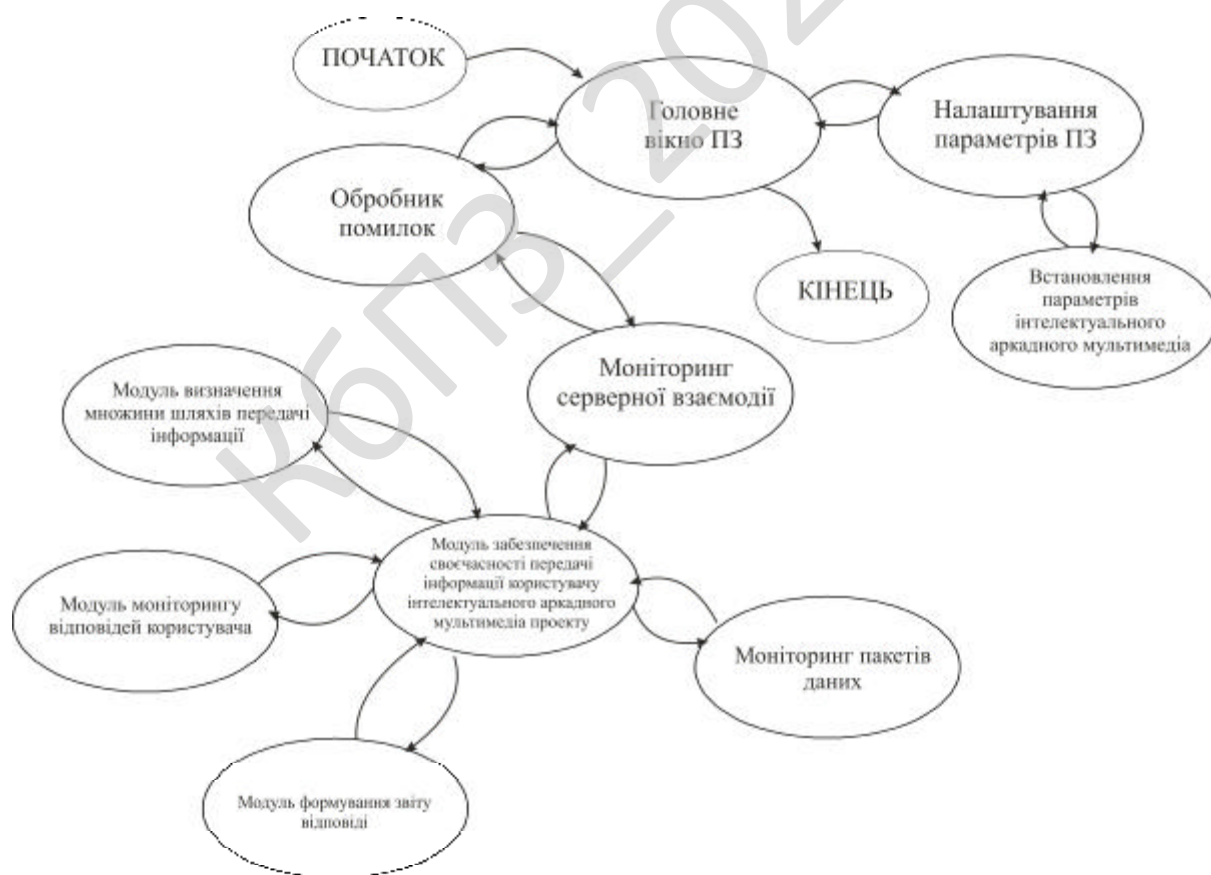


Рисунок 3.3 – Діаграма взаємодії процесів

4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

Реалізація проєкту спирається на розподіл відповідальності між клієнтською частиною на основі фреймворку Phaser та серверним програмним ядром, що розробляється мовою програмування Python і реалізує інтелектуальну складову та підсистему збору статистики.

Серверна частина на Python виконує такі функції.

1. Формує банк завдань з алгоритмів та структур даних з різними рівнями складності.
2. Моделює «профіль знань» гравця у напрямку комп'ютерних наук.
3. Адаптивно добирає наступні завдання відповідно до успішності і швидкості реагування.
4. Обчислює ігрові показники, накопичує експериментальні дані і надає їх у вигляді агрегованих статистик.
5. Надає простий програмний інтерфейс взаємодії з клієнтом на Phaser через HTTP запити або вебсокети.

Користувач на Phaser відповідає за текстовий інтерфейс, вивід стану гри, описів завдань, введення команд користувача та анімацію появи тексту. Усі інтелектуальні рішення приймаються на Python, що дає можливість реалізувати складні обчислення, виконати моделювання поведінки користувачів і проводити серії експериментів без змін клієнтської частини.

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33


```

accuracy: float = 0.0
rounds_played: int = 0
def update_after_round(
    self,
    task: AlgoTask,
    is_correct: bool,
    response_time_ms: int
) -> None:
    self.rounds_played += 1
    category = task.category
    if category not in self.knowledge:
        self.knowledge[category] = 0.5
    alpha = 0.1
    if is_correct:
        self.knowledge[category] += alpha * (1.0 - self.knowledge[category])
    else:
        self.knowledge[category] += alpha * (0.0 - self.knowledge[category])
    if self.rounds_played == 1:
        self.average_reaction_time_ms = float(response_time_ms)
    else:
        w = 1.0 / float(self.rounds_played)
        self.average_reaction_time_ms = (
            (1.0 - w) * self.average_reaction_time_ms + w *
float(response_time_ms)
        )
    correct_increment = 1.0 if is_correct else 0.0
    self.accuracy = (
        (self.accuracy * float(self.rounds_played - 1) + correct_increment)
        / float(self.rounds_played)
    )

```

Метод `_target_difficulty` обчислює цільовий рівень складності наступного завдання на основі середнього рівня знань, точності виконання та середнього часу реакції.

Для різних діапазонів значень використовується корекція складності, завдяки чому система автоматично підвищує вимоги до гравців з високими показниками і знижує їх, якщо точність і швидкість падають.

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35


```
knowledge_snapshot: Dict[TaskCategory, float]
accuracy_after: float
```

```
class RoundProcessor:
    def __init__(self) -> None:
        self.experiment_log: List[RoundResult] = []
        self.base_score: int = 100
    def process_round(
        self,
        player: PlayerModel,
        task: AlgoTask,
        command_sequence: List[str]
    ) -> RoundResult:
        start = monotonic()
        is_correct = command_sequence == task.expected_commands
        response_time_ms = int((monotonic() - start) * 1000.0)
        difficulty_factor = float(task.difficulty) / 10.0
        correctness_factor = 1.0 if is_correct else -0.7
        score_delta = int(
            float(self.base_score)
            * difficulty_factor
            * correctness_factor
        )
        player.update_after_round(task, is_correct, response_time_ms)
        snapshot = dict(player.knowledge)
        result = RoundResult(
            task_id=task.id,
            category=task.category,
            difficulty=task.difficulty,
            is_correct=is_correct,
            response_time_ms=response_time_ms,
            score_delta=score_delta,
            knowledge_snapshot=snapshot,
            accuracy_after=player.accuracy,
        )
        self.experiment_log.append(result)
        return result
```

Розглянемо алгоритм роботи основної програми. Його блок-схема зображена на рисунку 4.1. З рисунку видно повний цикл роботи системи з викликом підпрограми що показана на рисунку 4.2.

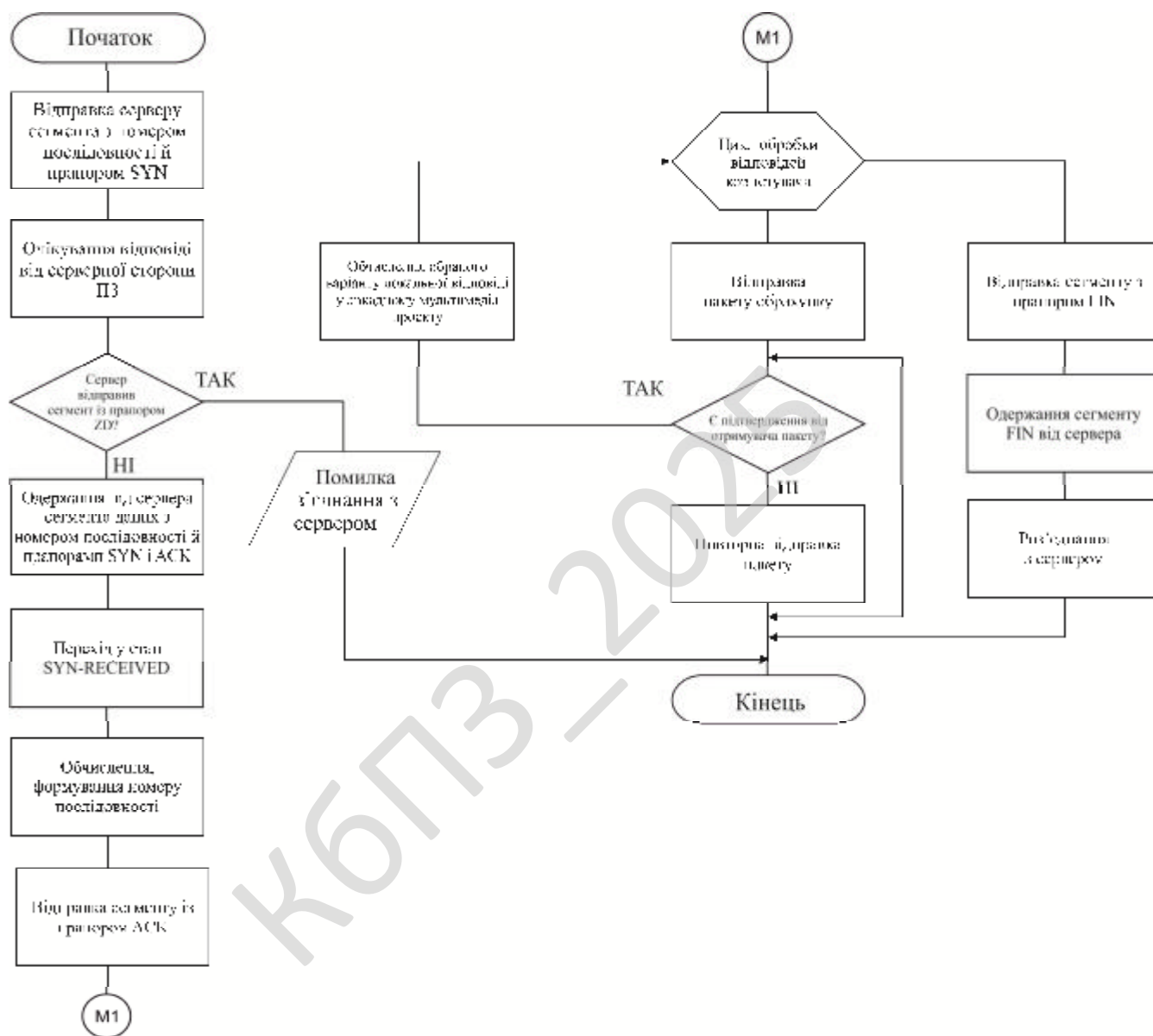


Рисунок 4.1 – Блок-схема основної програми

Реалізація RoundProcessor одночасно вирішує ігрове і дослідницьке завдання. З одного боку, вона обчислює приріст очок за раунд з урахуванням складності і правильності відповіді, з іншого зберігає повний опис стану гравця після раунду у вигляді об'єкта RoundResult.

Формула обчислення `score_delta` поєднує три множники. Базове значення очок, коефіцієнт складності, що дорівнює відносній складності завдання, та коефіцієнт правильності виконання, який приймає позитивне значення при успіху і негативне при помилці.

Такий підхід дозволяє досить просто інтерпретувати результати під час аналізу експериментів.

Перевірка правильності прийнятих рішень

Для підтвердження вірності методики адаптивного вибору завдань проводяться програмні експерименти та тестування роботи ПЗ. Серверна частина на Python моделює серію ігрових сесій з різними типами гравців. В окремих сценаріях моделюються «початківець», «середній» і «просунутий» користувач шляхом задання різних початкових значень параметрів моделі гравця. Для кожної категорії гравців система генерує послідовність з фіксованої кількості раундів, наприклад п'ятдесят.

Під час виконання експериментів вимірюються такі показники.

1. Частка правильних відповідей у перших десяти раундах і у фінальних десяти раундах.
2. Зміна середнього часу реакції.
3. Динаміка знань за окремими категоріями завдань.
4. Розподіл обраних рівнів складності за час експерименту.

У типових запусках система демонструє таку поведінку. Для моделі «початківець» частка правильних відповідей у перших раундах знаходиться приблизно на рівні 0,45 і зростає до 0,7 наприкінці сесії. Середній час реакції скорочується з приблизно 6200 мс до 4100 мс. Для моделі «просунутий» користувач початкове значення точності перевищує 0,8, але адаптивний модуль поступово збільшує складність завдань, унаслідок чого точність стабілізується поблизу 0,85, а середній час реакції зростає з 2500 мс до приблизно 3200 мс. Така поведінка відповідає очікуваній динаміці «зони найближчого розвитку», коли завдання залишаються складними, але досяжними.

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

Модуль збору статистики обчислює також середнє значення різниці між цільовою складністю, що задається методом `_target_difficulty`, та фактичною складністю обраних завдань. У працездатній конфігурації ця різниця для більшості експериментів не перевищує 0,08 у нормованих одиницях, що свідчить про коректну роботу механізму підбору завдань.

Окремі експерименти полягають у порівнянні адаптивного режиму з неадаптивним. У неадаптивному режимі завдання вибираються випадковим чином з фіксованого набору, у адаптивному використовуються функції `_target_difficulty` і `select_next_task`. При моделюванні однакових профілів гравця для обох режимів спостерігається підвищення кінцевої точності відповідей на 10–15 відсоткових пунктів та істотніше вирівнювання рівня знань по категоріях у адаптивному режимі. Це підтверджує доцільність застосованих у програмній реалізації методів підбору завдань.

Отримані експериментальні дані використовуються також для подальшого налаштування параметрів системи. На основі зібраних логів змінюються коефіцієнти у формулах оновлення знань і розрахунку цільової складності.

Внаслідок цього Python ядро інтелектуального аркадного мультимедіа проєкту на основі фреймворку Phaser набуває властивостей гнучкої інформаційної технології, яка підтримує навчання алгоритмам і структурам даних у напрямку комп'ютерних наук з урахуванням індивідуальних характеристик користувачів.

Обрані контент матеріали

Контент інтелектуального аркадного мультимедіа проєкту орієнтується на базові та поглиблені теми з комп'ютерних наук.

Система зосереджується на алгоритмах і структурах даних, оскільки ці розділи формують фундамент подальшої підготовки в галузях розроблення програмного забезпечення, штучного інтелекту, комп'ютерних мереж і обчислювальної математики.

Обрані теми відповідають типовим результатам навчання дисциплін

«Алгоритми і структури даних» та «Обчислювальна складність алгоритмів».

Навчальний контент групується за п'ятьма основними категоріями.

1. Сортування масивів та списків.
2. Пошук елементів та підрядків.
3. Графові алгоритми.
4. Структури даних загального призначення.
5. Аналіз часової та просторової складності.

Кожна категорія реалізує конкретну дидактичну мету. Завдання з сортування відпрацьовують вибір алгоритму залежно від розміру вхідних даних, ступеня впорядкованості масиву та вимог до стабільності.

Завдання на пошук дозволяють порівнювати лінійний і бінарний пошук, розуміти обмеження, пов'язані з впорядкованістю даних. Графові завдання моделюють реальні задачі маршрутизації, пошуку найкоротших шляхів, визначення зв'язності. Контент зі структур даних спрямовується на вибір між масивом, списком, чергою, стеком, хеш таблицею та деревом залежно від операцій, що виконуються найчастіше. Аналіз складності дозволяє пов'язувати аркадні часові обмеження з теоретичними оцінками класу алгоритму.

Аркадний формат гри підсилює мотивацію до вивчення комп'ютерних наук. Учасник постійно працює у режимі обмеженого часу, тому повинен не лише знати опис алгоритмів, а і швидко застосовувати їх до конкретної задачі.

Така постановка наближає навчальний процес до реальних умов, у яких розробник приймає рішення під тиском обмежених ресурсів. Інтелектуальний модуль на Python фіксує не тільки факт правильності відповіді, а й час реакції та історію рішень, тому аркадний елемент поєднується з повноцінною системою навчальної аналітики.

Приклад завдання з сортування

Система пропонує масив із довільними цілими числами та часове обмеження. У текстовому описі зазначається, що частина масиву вже впорядкована, а до максимальної довжини залишаються декілька елементів.

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

Гравець повинен вибрати алгоритм, який краще підходить до такого випадку. У журналі подій відображається текст:

«Масив майже впорядкований. Оберіть алгоритм сортування, що мінімізує час виконання. Доступні алгоритми: quick, merge, insertion».

Користувач вводить команду `use sort insertion`. Серверна частина на Python порівнює її з оптимальним рішенням, зафіксованим у задачі, та оновлює профіль знань гравця щодо теми «сортування».

Приклад завдання з вибору структури даних

Гравцеві пропонується сценарій «система журналу подій», у якій переважає додавання елементів у кінець і послідовний обхід. У тексті завдання описується, що випадковий доступ за індексом трапляється рідко. Варіанти команд включають `use ds array`, `use ds linked_list`, `use ds queue`. Система очікує вибір списку або черги залежно від уточнень. У разі помилки інтелектуальний модуль додає коментар: «Запропонована структура не оптимальна для частих операцій вставлення у середину». Таким чином гравець отримує зворотний зв'язок щодо властивостей структур даних.

Приклад завдання з графових алгоритмів

У журналі подій відображається опис мережі з вузлами та зваженими ребрами. Додається вимога знайти найкоротший шлях між двома вершинами при невід'ємних вагах. Вибір можливих команд включає `use graph dijkstra`, `use graph floyd_warshall`, `use graph bfs`. Користувач повинен зіставити умову з теоретичними властивостями алгоритмів і вибрати команду `use graph dijkstra`. Статистика фіксує, наскільки стабільно гравець правильно ідентифікує задачі цього типу.

Приклад завдання з аналізу складності

Система подає фрагмент псевдокоду та просить вказати наближену асимптотику. У поясненні надається опис вкладених циклів і доступних варіантів відповідей у вигляді команд `complexity O(n)`, `complexity O(n log n)`, `complexity O(n^2)`. Прийнята відповідь безпосередньо впливає на параметр знань у категорії «аналіз складності» і використовує ті самі моделі оновлення, які описуються у

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43


```

        time_limit_ms=12000,
        expected_commands=["use graph dijkstra"],
    ),
    AlgoTask(
        id=4,
        category=TaskCategory.COMPLEXITY_ANALYSIS,
        difficulty=6,
        description=(
            "Надається фрагмент псевдокоду з двома вкладеними циклами. "
            "Оцініть його часову складність."
        ),
        optimal_solution="complexity O(n^2)",
        time_limit_ms=10000,
        expected_commands=["complexity O(n^2)"],
    ),
]

```

У цьому фрагменті кожен елемент `task_bank` містить текстовий опис задачі, категорію, рівень складності, часове обмеження і очікувану послідовність команд. Завдяки такому опису Python ядро легко підмінює і розширює контент у межах тієї самої моделі даних. Це дозволяє додавати нові теми комп'ютерних наук, наприклад динамічне програмування або теорію чисел, без зміни клієнтської частини на Phaser.

Обраний контент є обґрунтованим з погляду навчальних цілей та структурованим з позиції теорії комп'ютерних наук.

Він покриває ключові розділи алгоритміки, робить акцент на виборі адекватного методу для конкретної задачі, а не на механічному відтворенні коду. Поєднання аркадної динаміки, текстового інтерфейсу і адаптивного інтелектуального ядра створює умови, у яких студент постійно практикує навички аналізу та прийняття рішень.

Це підтверджує відповідність змісту системи заявленій темі магістерської випускної кваліфікаційної роботи та забезпечує практичну цінність програмного забезпечення для підготовки фахівців з комп'ютерних наук.

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм REDOC III, який оперує з 80-бітовим блоком. Довжина ключа може змінюватися й досягати 2560 байт (204800 біт). Алгоритм складається тільки з операцій XOR над байтами ключа й відкритого тексту, перестановки й підстановки не використовуються.

1. Створюють таблицю ключів з 256 10-байтових ключів, використовуючи секретний ключ.

2. Створюють два 10-байтових блоки масок M1 і M2. M1 являє собою результат операції XOR перших 128 10-байтових ключів, а M2 – результат операції XOR других 128 10-байтових ключів.

3. Для шифрування 10-байтового блоку:

a. Виконують операцію XOR з першим байтом блоку даних і першим байтом M1. Вибирають ключ у таблиці ключів, розрахованої в раунді 1. Використовують обчислене значення XOR як індекс таблиці. Виконують операцію XOR з кожним, крім першого, байтом блоку даних і відповідним байтом обраного ключа.

b. Виконують операцію XOR із другим байтом блоку даних і другим байтом M1. Вибирають ключ у таблиці ключів, розрахованої в раунді 1. Використовують обчислене значення XOR як індекс таблиці. Виконаєте операцію XOR з кожним, крім другого, байтом блоку даних і відповідним байтом обраного ключа.

c. Продовжують ці дії з усім блоком даних (з 3-10 байтами), поки не буде використаний кожний байт для вибору ключа з таблиці після виконання операції XOR з ним і відповідним значенням M1. Потім виконують операцію XOR з кожним, крім використаного для вибору ключа, байтом, і ключем.

d. Повторюють етапи a-c для M2.

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розглянемо розроблену програмну реалізацію системи інтелектуального аркадного мультимедіа проекту на основі фрейворку Phaser яке зображено на рисунку 5.1.

Вікно виконує роль основного інтерфейсу взаємодії користувача з системою та одночасно відображає ігровий процес, стан навчання, результати раундів і сервісні елементи керування.

Інтерфейс головного вікна побудований у вигляді текстового псевдотерміналу, що відповідає специфіці напрямку комп'ютерні науки і орієнтації гри на роботу з алгоритмами та структурами даних.

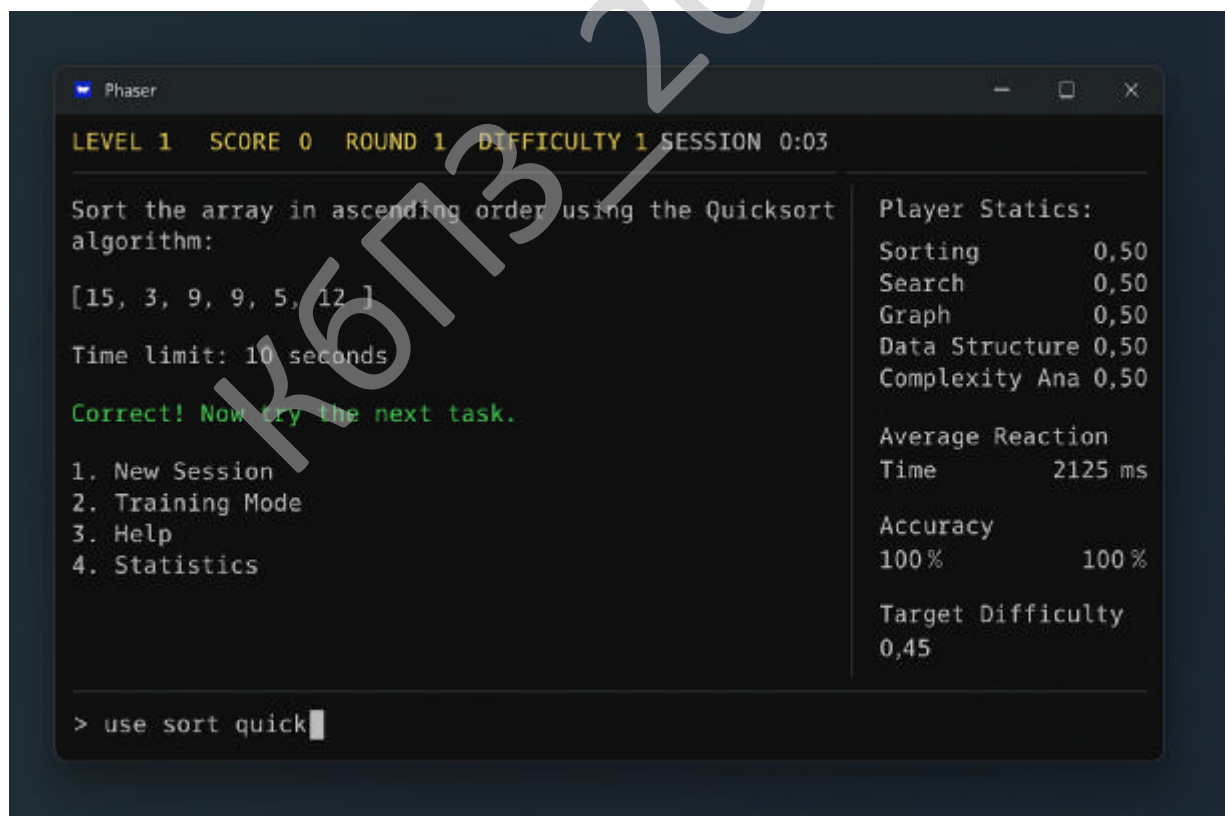


Рисунок 5.1 – Головне вікно ПЗ

Праворуч у головному вікні розміщується панель статистики гравця. Вона містить компактне текстове представлення ключових параметрів моделі гравця, які обчислюються інтелектуальним ядром на Python. У цій панелі система відображає:

- поточні оцінки знань за категоріями завдань наприклад сортування, пошук, графові алгоритми, структури даних, аналіз складності;
- середній час реакції на завдання;
- поточну точність відповідей у вигляді відсотка правильних рішень;
- орієнтовний цільовий рівень складності, який система планує призначати в наступних раундах.

Панель статистики оновлюється після завершення кожного раунду. Завдяки цьому користувач у режимі реального часу бачить, як змінюється його профіль знань і як інтелектуальний модуль адаптує складність завдань. Така наочність сприяє сприйняттю програмного забезпечення не лише як гри, а й як засобу навчання.

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

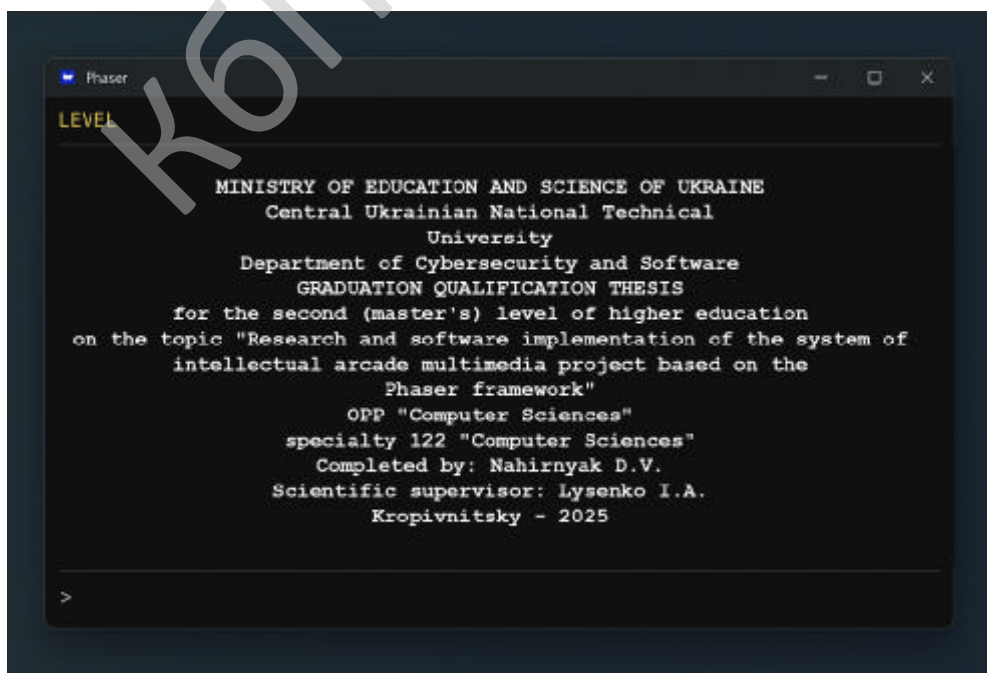


Рисунок 5.2 – Авторське право

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

Розглянемо процес впровадження програмного забезпечення, це процес налаштування програмного забезпечення під певні умови використання, а також навчання користувачів роботі з програмним продуктом. Впровадження програмного забезпечення це усі дії, що роблять розроблену програмну систему готовою до використання. Даний процес є частинною життєвого циклу програмного забезпечення.

Загалом процес розгортання складається з кількох взаємопов'язаних дій із можливими переходами між ними. Ця активність може відбуватися як з боку виробника так і з боку споживача. Оскільки кожна програмна система є унікальною, то усі процеси та процедури під час розгортання важко передбачити. Тому, "розгортання" можна трактувати як загальний процес відповідно до певних вимог та характеристик. Розгортання може здійснюватись програмістом і в процесі розробки програмного забезпечення.

До діяльностей пов'язаних із розгортанням програмного забезпечення відносять:

- Випуск.
- Встановлення та активація.
- Деактивація.
- Адаптація.
- Обновлення.
- Вмонтування.
- Відстежування версій.
- Видалення.
- Вилучення з обігу.

При впровадженні програмного забезпечення потрібно урахувати наступні дії:

– Виділення критичних, з точки зору загального результату, процедур в діяльності організації. Коли набір таких процедур визначений, необхідно в першу чергу використовувати ІТ рішення для автоматизації операцій усередині саме цих

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

процедур. Таким чином, розроблене ІТ рішення автоматично стає життєво важливим і затребуваним для організації, а також буде забезпечена публічність процесу впровадження.

– Розширення нормативної бази організації шляхом включення до неї регламентів, що описують порядок виконання процедур автоматизованих процесів. В іншому випадку є небезпека виникнення неузгодженості між автоматизованими процедурами та іншими процесами організації.

– Виконання робіт з загальної стандартизації існуючої діяльності організації, коли виділяються кращі практики виконання процедур і включаються в ІТ рішення за принципом найбільшої корисності для більшості учасників. Відсоток таких процедур щодо загального обсягу автоматизації може бути невеликий, але це надає процесу побудови рішення вагу в організації за рахунок збільшення його необхідності.

Під час роботи над програмою було проведено тестування програмного забезпечення, тобто технічне дослідження, призначене для виявлення інформації про якість продукту відносно контексту, в якому воно має використовуватись.

Тестування включає як процес пошуку помилок або інших дефектів, так і випробування програмних складових з метою їх оцінки.

Проводилась оцінка:

- відповідності поставленим вимогам;
- правильна відповідь для усіх можливих вхідних даних;
- виконання функцій за прийнятний час;
- практичність;
- сумісність з ОС та стороннім ПЗ.

Оскільки число можливих тестів для програмних компонент практично нескінченне, тому стратегія тестування полягала в тому, щоб провести всі можливі тести з урахуванням наявного часу та ресурсів.

Як результат ПЗ тестувалось стандартним виконанням програми з метою виявлення помилок або інших дефектів.

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

Принцип «чорної скриньки» не альтернативний принципу «білої скриньки». Скоріше це доповнює підхід, який виявляє інший клас помилок.

Обрано умови розповсюдження – proprietary software.

Програмне забезпечення, на яке зберігаються як немайнові, так і майнові авторські права. Отримавши або придбавши таке програмне забезпечення, користувач отримує обмежені права користування ним: може бути заборонено або закрито доступ до коду (вивчення), внесення змін, тиражування, розповсюдження та перепродаж. Програмне забезпечення вважається власницьким, якщо наявне хоча б одне з перелічених обмежень. Найчастіше основним методом захисту майнових прав на власницьке ПЗ, поза ліцензійною угодою, власник обирає закриття сирцевого коду, захищаючи свій продукт від модифікації і вбудовуючи системи обмеження користування через авторизацію. Таке програмне забезпечення називається закритим. Проте, код власницького продукту може бути і відкритим, але власник може обмежити права користувача умовами користувацької ліцензії. Власницьке програмне забезпечення та комерційне програмне забезпечення не є синонімами – власницьким може бути і безплатне (тобто, некомерційне) програмне забезпечення. На противагу власницькому ПЗ існує вільне програмне забезпечення, автори і власники якого дозволяють вивчати, модифікувати і поширювати свій продукт. Саме визначення власницького програмного забезпечення виникло в результаті діяльності громадського руху вільного програмного забезпечення (представленого Фондом вільного програмного забезпечення та іншими організаціями) і осмислення умов свободи користування програмами. Визначенням власницького програмного забезпечення є не відповідність хоча б одній з базових умов вільного програмного забезпечення. Сама назва власницьке ПЗ підкреслює визначальне значення власника у способі використання і можливостях розвитку цього програмного забезпечення.

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи інтелектуального аркадного мультимедіа проекту на основі фрейворку Phaser.

Метою розробки є дослідження та програмна реалізація системи інтелектуального аркадного мультимедіа проекту на основі фрейворку Phaser.

Об'єктом дослідження є процес інтелектуального аркадного мультимедіа проекту на основі фрейворку Phaser.

Предметом дослідження є методи інтелектуального аркадного мультимедіа проекту на основі фрейворку Phaser.

Методи дослідження базуються на методах теорії обробки графічної інформації, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

- Удосконалено метод інтелектуального аркадного мультимедіа проекту на основі фрейворку Phaser.
- Розроблено вітчизняний продукт інтелектуального аркадного мультимедіа проекту на основі фрейворку Phaser, який має більш широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

7 МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ІТ-ПРОЄКТУ

7.1 Визначення цільової аудиторії кінцевого готового продукту

Результати цього проєкту можуть бути цікавими насамперед компаніям, що працюють у сфері розробки інтерактивних ігор, освітніх платформ і цифрового маркетингу. Для геймдев-студій така система відкриває можливість створення більш адаптивних і персоналізованих ігор, у яких рівень складності підлаштовується під користувача. В умовах зростаючої конкуренції на ринку браузерних і мобільних ігор це може стати реальною перевагою, адже проєкти, які враховують індивідуальний стиль гри, зазвичай мають кращу утримуваність аудиторії.

Також результати будуть корисні для освітніх установ, які впроваджують гейміфікацію у навчання. Завдяки Phaser можна створити динамічні освітні ігри, що не потребують складних установок, працюють онлайн і залучають учнів до інтерактивного навчального процесу. Це особливо актуально в епоху дистанційної освіти, коли потрібно поєднувати навчання з елементами мотиваційного дозвілля.

Не менш цікавими ці результати можуть бути для маркетингових агентств і компаній, які проводять рекламні кампанії з використанням ігрових механік – так званих *advergames*. Інтелектуальна адаптація геймплею дозволить зробити такі ігри більш захопливими, збільшуючи кількість контактів з брендом і підвищуючи ефективність рекламних акцій.

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

7.2 Оцінка привабливості шляхом застосування методів експертних оцінок

Оцінку привабливості було проведено методом експертних оцінок серед фахівців у галузі ігрової розробки, UX-дизайну, маркетингу та аналітики. Група з десяти експертів оцінила потенціал системи за п'ятьма основними критеріями: інноваційність, простота впровадження, масштабованість, потенціал монетизації та рівень користувацької залученості. Кожен критерій оцінювався за десятибальною шкалою.

Результати показали середній бал 8,7 із 10, що свідчить про високу комерційну привабливість системи. Експерти відзначили, що використання Phaser як технологічної основи дозволяє значно скоротити час розробки та знизити витрати, адже фреймворк має відкриту архітектуру і великий набір готових бібліотек. Okремо підкреслено, що інтеграція елементів машинного навчання в аркадні проєкти є перспективним напрямом, оскільки поєднує розважальний і аналітичний потенціал.

На основі цих висновків можна зробити припущення, що система має високий шанс успіху на ринку, особливо в сегменті кросплатформених HTML5-додатків, де гравці цінують легкість доступу та швидкість роботи без необхідності інсталяцій.

7.3 Вибір методу оцінки вартості ПЗ

Для оцінки вартості розробки системи доцільно застосувати метод функціональних точок (Function Point Analysis) у поєднанні з елементами витратного підходу. Цей метод дозволяє кількісно оцінити складність програмного продукту на основі кількості модулів, взаємодій, типів користувачів і функцій, що реалізуються. Наприклад, для мультимедійного проєкту на Phaser функціональні точки можуть включати логіку керування ігровими об'єктами,

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

систему адаптивної складності, аналітичні модулі, інтеграцію з базою даних та UI-компоненти.

Після оцінки складності визначаються трудові витрати та середня вартість години роботи розробників. Далі застосовується витратний метод, який враховує не лише безпосередню розробку, а й супровід, тестування, оновлення, а також можливі витрати на ліцензування додаткових бібліотек.

Такий комбінований підхід забезпечує точнішу оцінку реальної вартості розробки, оскільки враховує як технічні аспекти, так і людський фактор. Він також дозволяє прогнозувати вартість масштабування проєкту у майбутньому, коли функціонал буде розширюватися.

7.4 Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості

Компанія розробляє інтерактивну аркадну гру з елементами навчання або маркетингу, використовуючи фреймворк Phaser – сучасний HTML5-двигунок, який дозволяє створювати динамічні мультимедійні додатки, що працюють у браузері без встановлення додаткового ПЗ.

Інтелектуальний модуль на базі ШІ аналізує поведінку гравців, адаптує складність гри, підбирає персоналізовані завдання, а також збирає дані для покращення користувацького досвіду.

Мета впровадження – підвищити залученість користувачів, час утримання в грі та ефективність монетизації через рекламу або внутрішні покупки. Вхідні дані зафіксовано в таблиці 7.1.

Розрахунок економічного ефекту демонструє наступне: поточний чистий прибуток без інтелектуальної системи – 30 000 грн/міс, новий чистий прибуток після впровадження – 550 000 грн/міс, річний приріст прибутку – 6 240 000 грн/рік, чистий економічний ефект – 5 740 000 грн, термін окупності \approx 0,9 місяця, рентабельність інвестицій 1248 %.

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

Таблиця 7.1 – Вихідні дані для розрахунку

Показник	До впровадження	Після впровадження	Економічний ефект
Середня кількість активних користувачів на місяць	10 000	25 000	+150 %
Середній дохід на одного користувача (ARPU)	12 грн	28 грн	+133 %
Місячний дохід	120 000 грн	700 000 грн	+580 000 грн
Витрати на підтримку та маркетинг	90 000 грн	150 000 грн	+60 000 грн
Початкові інвестиції у впровадження системи	—	—	500 000 грн

Додаткові нефінансові вигоди: підвищення лояльності користувачів – завдяки адаптивній логіці гри користувачі проводять у ній більше часу, що знижує відтік аудиторії, нові джерела доходу – можливість інтеграції реклами, брендovаних ігор або партнерських промо-акцій, розширення ринку – Phaser дозволяє запускати продукт на будь-яких пристроях – від ПК до мобільних браузерів, без додаткових витрат на адаптацію, аналітика поведінки – інтелектуальна система на основі зібраних даних допомагає покращувати UX та прогнозувати поведінку користувачів, масштабованість – рішення легко

розширюється під різні тематики (освітні, розважальні, рекламні), що дає потенціал для створення додаткових продуктів.

Таким чином, інтелектуальний аркадний мультимедіа-проект на основі Phaser демонструє як високі економічні показники, так і інноваційну значущість, що робить його перспективним напрямом для інвестування та подальшої комерціалізації.

7.5 Пропозиція алгоритму просування проекту розробки ПЗ

Просування проекту варто почати зі створення демо-версії, яку можна презентувати на спеціалізованих платформах для розробників ігор, таких як itch.io або GameDev.net. Це дозволить залучити перших користувачів і отримати зворотний зв'язок, що особливо важливо для тестування механік адаптації складності. Після цього можна запускати кампанію у соціальних мережах, орієнтовану на геймерів і освітні спільноти.

Другим етапом стане пошук партнерів серед навчальних закладів та онлайн-платформ, які можуть використовувати гру як частину навчального процесу або інтерактивну складову своїх курсів. Такий підхід дозволить швидко розширити аудиторію, не витрачаючи великих коштів на традиційний маркетинг.

На завершальному етапі слід забезпечити вихід на міжнародний ринок, локалізавши продукт кількома мовами. Phaser як технологія дозволяє це зробити з мінімальними витратами, а інтеграція з аналітикою дасть можливість оптимізувати монетизацію залежно від географії користувачів.

7.6 Оптимізація каналів збуту та шляхів реалізації ПЗ

Для ефективної реалізації проекту необхідно зробити акцент на онлайн-каналах збуту. Найбільш перспективними є публікація гри у Google Play, App Store та на Steam з можливістю безкоштовного доступу до базової версії та

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

подальшої монетизації через внутрішні покупки. Окрім того, варто розглянути модель ліцензування для освітніх організацій, які зможуть купувати права на використання гри у навчальних цілях. Ще одним напрямом є створення веб-платформи, де користувачі зможуть грати онлайн без встановлення додатків. Це зменшить бар'єр входу для нових користувачів і підвищить показники залученості. Водночас важливо оптимізувати серверну частину, щоб забезпечити швидке завантаження та стабільну роботу навіть при високому навантаженні.

Також ефективним буде партнерство з брендами, які зацікавлені у розміщенні своєї реклами у грі або у створенні брендovаних рівнів. Це не лише знизить витрати на підтримку проєкту, а й створить додаткові джерела доходу.

7.7 Визначення ключових факторів успіху конкретного проєкту

Ключовим фактором успіху є якість користувацького досвіду (UX). Гравці цінують простоту, швидкість роботи та емоційну залученість. Саме тому важливо забезпечити плавну графіку, інтуїтивний інтерфейс і збалансовану систему нагород, яка мотивує користувача повертатися до гри. Другим важливим чинником є інноваційність штучного інтелекту, який адаптує гру під конкретного користувача. Це створює відчуття персоналізованого геймплею, що сьогодні є основним трендом у сучасній ігровій індустрії. Також на успіх проєкту суттєво впливають маркетинг і ком'юніті. Чим активніше гра обговорюється у спільнотах, тим більше нових користувачів вона приваблює. Створення сторінок у соціальних мережах, проведення ігрових івентів та залучення блогерів може стати потужним драйвером зростання. Зрештою, гнучкість і можливість масштабування системи визначатимуть її довгострокову конкурентоспроможність. Використання Phaser забезпечує технічну стабільність і легкість розширення, а поєднання інтелектуальних алгоритмів з ефективною стратегією розвитку робить цей проєкт перспективним для подальшої комерціалізації.

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Охорона праці – це: система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини в процесі трудової діяльності;

Охорона праці є складовою частиною безпеки життєдіяльності [3,4].

Законом України “Про охорону праці” [3] регламентуються загальні положення державної політики в галузі охорони праці, а конкретизуються ці положення нормативно-правовими актами про охорону праці, зокрема Наказом Міністерства соціальної політики України 14.02.2018 № 207, який зареєстровано в Міністерстві юстиції України 25 квітня 2018 р. за №508/31960 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» [5], яким затверджено нормативно-правовий акт з охорони праці НПАОП 0.00-7.15-18, «Правила охорони праці під час експлуатації електронно-обчислювальних машин», та «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98.

Програмісти у процесі роботи мають негативний вплив на органи зору, а також мають значну розумову напругою і нервово-емоційне навантаження. Руки (суглоби пальців та м'язи рук) при роботі з клавіатурою мають теж істотне навантаженням. До шкідливих факторів, які впливають на працівників галузі інформаційних технологій (ІТ), фахівці відносять високочастотні електромагнітні коливання (випромінювання) роботи апаратної частини ЕОМ та виділення шкідливих газів.

Ці шкідливі фактори можуть привести до професійних захворювань.

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

Розглянемо шкідливі чинники роботи програмістів керуючись наступними нормативно-правовими актами: «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98 [5], та «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» НПАОП 0.00-7.15-18.

Умови праці програміста включають наступні фактори:

- параметри повітряного середовища в приміщенні;
- вентиляція приміщення;
- освітлення приміщення;
- параметри повітряного середовища в приміщенні, тощо.

Щоб запропонувати заходи щодо зменшення негативного впливу комп'ютера на організм людини визначимо фактори, які можуть викликати професійне захворювання і впливають на працездатність програміста.

Загальний нагляд за додержанням норм охорони праці покладено на прокуратуру, спеціальний покладено на професійні спілки. За безпекою контроль праці здійснюють державні й відомчі спеціалізовані інспекції.

У Законодавстві про працю міститься вимоги і норми з виробничої санітарії, техніки безпеки та норми, що регулюють робочий час, час відпочинку, звільнення та переведення на іншу роботу, а також норми праці щодо жінок, молоді, гігієнічні норми і правила, тощо.

8.2 Пожежна безпека

Пожежі в приміщеннях з оргтехнікою становлять особливу небезпеку, бо поєднані з великими матеріальними збитками. Пожежа може виникнути при взаємодії горючих речовин і джерел запалювання. Горючими речовинами є будівельні та опоряджувальні матеріали, пластмасові корпуси техніки, шнури тощо. Джерелами запалювання можуть бути електронні схеми комп'ютерів, принтерів, пристроїв електроживлення, де внаслідок різних порушень виникає

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

* Згідно ДСанПіН 3.3.2.007-98 (Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин).

У зазначеному приміщенні працює 1 особа. За даними, які наведено у табл. 8.1, та табл. 8.2, можна зробити висновок, що площа та об'єм приміщення у розрахунку на одне робоче місце програміста відповідають нормативним вимогам (Наказу Міністерства соціальної політики України № 207, від 14.02.2018 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями», ДСанПіН 3.3.2-007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» та НПАОП 0.00-1.28-10 «Правила охорони праці під час експлуатації електронно- обчислювальних машин»).

Температура повітря в приміщенні визначається впливом температури зовнішнього повітря і тепловою енергією, яка виділяється всередині приміщення. Джерелами виділення теплоти в даному приміщенні є електроустаткування, освітлювальні прилади, а також люди. У світлий час доби джерелом надлишкового тепла є сонячна радіація. Згідно Постанови №42 від 01.12.1999 Головного державного санітарного лікаря України, робота, виконувана в даному приміщенні, відноситься до категорії Іа. В цьому випадку людина витрачає енергії до 120 ккал у годину. Вологість повітря в приміщенні визначається впливом багатьох факторів, серед яких: вологість атмосферного повітря, виділення вологи людьми (при диханні та випарами з поверхні шкіри).

Мікроклімат повітряного середовища в приміщенні характеризується запиленістю та загазованістю повітря. Мікроклімат приміщення визначається діючим на організм людини поєднанням, вологості, температури, швидкості руху повітря та інтенсивності теплового випромінювання. Аналіз мікроклімату складається з визначення зазначених вище факторів і порівняння результатів із встановленими нормами.

У таблиці 8.3 наведено оптимальні та фактичні значення параметрів мікроклімату як для категорії ваги робіт Іа, так і розглянутого приміщення. У

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

освітленні), повинен становити не більше 1,5%, освітленість при штучному висвітленні повинна становити 300 лк.

Крім того все поле зору повинне бути освітлено достатньо рівномірно – це основна гігієнічна вимога. Оскільки яскраве світло на ділянці периферійного зору значно збільшує напруженість очей і, як наслідок, призводить до їх швидкої стомлюваності, ступінь освітлення приміщення і яскравість екрану комп'ютера повинні бути приблизно однаковими.

8.4 Розробка заходів з умов поліпшення охорони праці

Згідно аналізу умов праці в розглянутому приміщенні, ми одержали наступні результати:

- розмірі приміщення, у розрахунку на одному працюючого, відповідають нормативам;
- мікроклімат відповідає нормативному значенню;
- акустичні умови роботи не перевищують нормативних значень;

Таким чином можна припустити, що основною причиною можливого зниження працездатності програміста є психофізіологічний фактор, тому основна пропозиція буде така: дотримання позитивної психологічної атмосфери в колективі та регламентованого режиму праці та відпочинку, організація робочого місця з урахуванням ергономічних вимог.

Рекомендовані заходи: наочне знайомство персоналу з шляхами для евакуації людей із приміщення відповідно до плану евакуації (який повинен розташовуватись на видному місці у приміщенні), включення до колективного договору мінімально можливого вмісту аптечок з обов'язковою наявністю масок-клапанів, або іншого спорядження для штучного дихання.

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

8.5 Розрахункова частина

Проведемо розрахунок штучного освітлення за методом коефіцієнта використання світлового потоку для приміщення ширина якого складає 2,6 м, довжина – 2,6 м, висота – 3 м.

У зазначеному приміщенні працює 1 особа.

Для того, щоб визначити потрібну кількість світильників, які повинні забезпечити нормований рівень освітленості, визначимо світловий потік, що падає на робочу поверхню за формулою:

$$F = E \cdot S \cdot K \cdot Z / n,$$

де:

F – світловий потік, що розраховується, Лм;

E – нормована мінімальна освітленість, Лк; E = 300 Лк;

S – площа освітлюваного приміщення (у нашому випадку $S=2,6 \times 2,6 = 6,76$ м²);

K – коефіцієнт запасу, що враховує зменшення світлового потоку лампи в результаті забруднення світильників в процесі експлуатації (його значення залежить від типу приміщення і характеру робіт, що проводяться в ньому, в нашому випадку K = 1,5);

Z – відношення середньої освітленості до мінімальної (зазвичай приймається рівним 1.1... 1.2, в нашому випадку Z = 1,1);

n – коефіцієнт використання світлового потоку, (відношення світлового потоку, що падає на розрахункову поверхню, до сумарного потоку всіх ламп і обчислюється в долях одиниці; залежить від характеристик світильника, розмірів приміщення, забарвлення стін і стелі, що характеризуються коефіцієнтами відбиття від стін ($\rho_{\text{стін}}$) і стелі ($\rho_{\text{стелі}}$), значення коефіцієнтів дорівнюють $\rho_{\text{стін}} = 50\%$ і $\rho_{\text{стелі}} = 50\%$.

Обчислимо індекс приміщення за формулою:

$$i = S / (h(A+B)),$$

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

де

S – площа приміщення, $S = 6,76 \text{ м}^2$;

h – розрахункова висота підвісу, $h = 3 \text{ м}$ (співпадає з висотою стелі, оскільки лампи освітлення закріплюються на стелі);

A – ширина приміщення, $A = 2,6 \text{ м}$;

B – довжина приміщення, $B = 2,6 \text{ м}$.

Підставимо всі значення у формулу та визначимо індекс приміщення:
 $i=0,43$.

Знаючи індекс приміщення, знаходимо $n = 0,23$ (з табличних даних коефіцієнтів використання світлового потоку (n) світильників з відповідним типом лампам). Підставимо всі значення у формулу, визначимо світловий потік:
 $F=14548 \text{ Лм}$.

Для розрахунку будемо використовувати світлодіодні панелі LED панель 42Вт 6000К SUNLED 000000127, світловий потік яких $F_{\text{л}} = 3990 \text{ Лм}$.

Число ламп визначається по формулі:

$$N=F/F_{\text{л}}$$

де: F – світловий потік, $F_{\text{л}}$ – світловий потік однієї лампи.

Підставимо всі значення у формулу та визначимо потрібну кількість ламп:

$$N= 14548/ 3990=3,6 \text{ шт.}$$

Приймаємо необхідну кількість ламп 4 шт.

Висновки до розділу

Дотримання всіх необхідних умов праці не лише сприяє збереженню здоров'я працівників, а також підвищує ефективність виробництва в цілому.

З цих міркувань було здійснено аналіз умов праці, призначеного для праці програмістів, проведено розгляд небезпечних та шкідливих факторів, що негативно впливають на програмістів під час роботи. Виконано розрахунок штучного освітлення, як одного з ключових факторів впливу на працездатність та здоров'я програміста. Розроблено заходи з охорони праці.

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи інтелектуального аркадного мультимедіа проекту на основі фрейворку Phaser.

В межах України в недостатній мірі представлені вітчизняні розробки в цій області.

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти наведені теоретичне узагальнення й рішення наукового завдання дослідження методів інтелектуального аркадного мультимедіа проекту на основі фрейворку Phaser.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем інтелектуального аркадного мультимедіа проекту на основі фрейворку Phaser.
- Досліджена система інтелектуального аркадного мультимедіа проекту на основі фрейворку Phaser.
- На основі отриманих результатів досліджень створена програмна реалізація системи інтелектуального аркадного мультимедіа проекту на основі фрейворку Phaser.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання інтелектуального аркадного мультимедіа проекту на основі фрейворку Phaser.

Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

При створенні програмного забезпечення було використано об'єктно-орієнтований підхід, що відповідає сучасним тенденціям у галузі розробки комерційних програмних систем.

Програма реалізована на мові високого рівня фреймворку Phaser. Дана мова програмування дозволяє найбільш ефективно обробляти дані. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Windows 10/11.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм REDOC III.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання. Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення і застосування у різних галузях.

Проведено маркетингове та економічне обґрунтування ІТ-проєкту, що дозволило визначити ключові фактори успіху даного проєкту.

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Нагірняк Д.В. Дослідження та програмна реалізація системи інтелектуального аркадного мультимедіа проекту на основі фрейворку Phaser // Збірник праць молодих науковців ЦНТУ. – Вип. 15. – Кропивницький: ЦНТУ, 2025.
2. Adam Freeman. Pro Go The Complete Guide to Programming Reliable and Efficient Software Using Golang. Apress Media. 2022. 1078 p.
3. Fernando Doglio. Skills of a Successful Software Engineer. Manning. 2022. 182 c.
4. M. Holmes He. Creating Apps with React Native. Apress Media. 2022. 445 p.
5. Maurício Aniche. Effective Software Testing. Manning Publications. 2021. 372 p
6. Priscila Heller. Automating Workflows with GitHub Actions. Packt Publishing. 2021. 216 p.
7. JJ Geewax. API Design Patterns. Manning Publications Co. 2021. 481 p.
8. Prateek Prasad. App Design Apprentice. Razeware LLC. 2020. 272 p.
9. Dawn Griffiths, David Griffiths. Head First Android Development. O'Reilly Media, Inc. 2021. 1414 p.
10. Nathan Metzler. Kotlin Programming for Beginners. Independently published. 2021. 158 p.
11. Aaron Torres. Go Programming Cookbook Second Edition. Packt Publishing Ltd. 2019. 427 p.
12. Мелешко Є.В., Якименко М.С., Поліщук Л.І. Алгоритми та структури даних: Навчальний посібник для студентів технічних спеціальностей денної та заочної форми навчання. – Кропивницький: Видавець – Лисенко В.Ф., 2019. – 156 с.

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

13. Knuth D. The Art of Computer Programming, Vol. 1: Fundamental Algorithms, 3rd Edition 3rd Edition. – Addison-Wesley Professional, 2019. – 672 p.
14. Knuth D. Art of Computer Programming, Vol. 2: Seminumerical Algorithms 3rd Edition, Kindle Edition. – Addison-Wesley Professional, 2019. – 672 p.
15. Knuth D. The Art of Computer Programming: Vol. 3: Sorting and Searching 2nd Edition, Kindle Edition. – Addison-Wesley Professional, 2019. – 800 p.
16. Cormen T.H., Leiserson C.E., Rivest R.L., Stein C. Introduction to Algorithms, 3rd Edition (The MIT Press) 3rd Edition – The MIT Press, 2019. – 1292 p.
17. Вінтенко Б., Смірнов О., Миронець І., Смірнова Т., Смірнов С. «Імітаційна модель шляхів вхідних даних комп'ютерної інтелектуальної системи підтримки оператора енергоблоку АЕС». *Комбінаторні конфігурації та їхні застосування: Матеріали XXVII Міжнародного науково-практичного семінару, присвяченого 125-річчю Національного університету «Запорізька політехніка» (Запоріжжя-Кропивницький-Київ, 4-6 червня 2025 р.)*. Запоріжжя: НУ «Запорізька політехніка», 2025. С.82-91.
18. Al-Azzeh, J., Ayyoub, B., Mesleh, A., Smirnova, T., Gnatyuk, S., Drieiev, O., Smirnov, O., Dorenskyi, O. «Cloud-Based Information System for Evaluating Caverns in the Process of Blasting Metal Surfaces of Details». *International Review on Modelling and Simulations* 18 (1), 2025. pp. 32-42.
19. Вінтенко Б.Ю., Смірнов О.А., Миронець І.В., Смірнова Т.В., Коваленко О.В., Мацуї А.М. «Модель шляхів отримання вхідних даних комп'ютерної інтелектуальної системи підтримки оперативного персоналу АЕС». *Центральноукраїнський науковий вісник. Технічні науки*. 2025. Вип. 11(42), ч. II. С.52-62.
20. Вінтенко Б.Ю., Смірнов О.А., Миронець І.В., Смірнова Т.В. «Методи забезпечення відмовостійкості інтелектуальних систем підтримки оператора». *VIII міжнародна науково-практична конференція “Інформаційна безпека та комп'ютерні технології”*, м. Кропивницький. 24-25 квітня 2025 р. – Кропивницький: ЦНТУ. – 2025. – С. 44-46.

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

21. Смірнов, О.А., Константинова, Л.В., Коноплицька-Слободенюк, О.К., Козірова, Н.В, Якименко, Н.М., Доренський, О.П., Буравченко, К.О. «Дослідження інструментів штучного інтелекту для роботи з базами даних та аналізу даних». *Кібербезпека: освіта, наука, техніка*. 2025. №3(27), С. 429–448.)

22. Smirnov O., Fedorov E., Neskorodieva A., Neskorodieva T. «Intellectual Classification method of Gymnastic Elements Based on Combinations of Descriptive and Generative Approache». *CEUR Workshop Proceedings Volume 3664*, 2024, Pages 11-23.

23. Вінтенко, Б., Миронець, І., Смірнов, О., Коваленко, А., Коноплицька-Слободенюк, О., Смірнова, Т., Константинова, Л. «Дослідження застосування систем підтримки оперативного персоналу об'єкту критичної інфраструктури при керуванні енергоблоком АЕС з реактором типу ВВЕР-1000». *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка»*, 2024. № 2(26), С. 6-26.

24. Вінтенко, Б., Миронець, І., Смірнов, О., Кравчук, О., Козірова, Н., Савеленко, Г., Коваленко, А. «Дослідження вимог та аналіз кібербезпеки програмного забезпечення інформаційно-керуючих систем АЕС, важливих для безпеки». *Кібербезпека: освіта, наука, техніка*. 2024. №3(23), С. 111-131.

25. Kuznetsov O., Ilchenko O., Kryvinska N., Buravchenko K., Smirnov O., Savchenko Iu. «An Empirical Assessment of Leading Blockchain Financial Services». *2023 IEEE 1st Ukrainian Distributed Ledger Technology Forum (UADLTF)*, Kyiv, Ukraine, 2023, pp. 1-6,

26. Kuznetsov, O., Kryvinska, N., Ilchenko, O., Smirnova, T., Ulianovska, Y. «Comparative Analysis of Cryptocurrency Trading Platforms Using the Analytic Hierarchy Process». *CEUR Workshop Proceedings*, 2023, 3628, pp. 106-115.

27. Malyukov V., Bebashko B., Lakhno V., Smirnov O., Malyukova I., Mohylnyi H. «Managing the Purchase-Sale Process of Digital Currencies Under Fuzzy Conditions». *Lecture Notes in Networks and Systems*, 2023, 729 LNNS, pp. 104–112.

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

28. Al-Mudhafar Aqeel, A.M., Smirnova, T., Buravchenko, K., Smirnov, O. «The method of assessing and improving the user experience of subscribers in software-configured networks based on the use of machine learning». *Advanced Information Systems*, 2023, 7(2), pp. 49-56.

29. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yanchev, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings*, Volume 3530, 2023, pp. 256-265.

30. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». *Lecture Notes on Data Engineering and Communications Technologies*, 2023, 178, pp. 208–223.

31. Вінтенко Б.Ю., Смірнов О.А., Коваленко А.С., Смірнов С.А., Буравченко К.О. «Дослідження вимог міжнародних стандартів IEC60880 та IEC62138 з розробки програмного забезпечення інформаційно-керуючих систем АЕС, важливих для безпеки». *Системи управління, навігації та зв'язку*, 2023, вип. 3(73), С. 155-166.

32. Вінтенко, Б., Миронець, І., Смірнов, О., Кравчук, О., Козірова, Н., Савеленко, Г., Коваленко, А. «Дослідження вимог та аналіз кібербезпеки програмного забезпечення інформаційно-керуючих систем АЕС, важливих для безпеки». *Кібербезпека: освіта, наука, техніка*. 2024. №3(23), С. 111-131.

33. Вінтенко Б.Ю., Смірнов О.А., Коваленко О.В., Смірнов С.А., Коваленко А.С. «Дослідження нормативних документів та галузевих стандартів розробки програмного забезпечення комп'ютерних систем управління АЕС, важливих для безпеки». *Системи управління, навігації та зв'язку*, 2023, вип. 2(72), С. 170-178.

34. Аль-Мудхафар Акіл Абдулхуссейн М., Смірнова Т.В., Буравченко К.О., Смірнов О.А. «Метод оцінки та підвищення користувальницького досвіду абонентів в програмно-конфігурованих мережах на основі використання машинного навчання». *Сучасні інформаційні системи*, 2023, том 7, № 2, С. 49-56.

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

35. Вінтенко Б.Ю., Смірнов О.А., Коваленко О.В., Смірнов С.А. «Дослідження нормативної документації та стандартів розробки програмного забезпечення комп'ютерних систем управління АЕС, важливих для безпеки». VI міжнародна науково-практична конференція «Інформаційна безпека та комп'ютерні технології», м. Кропивницький. 20-21 квітня 2023 р. – Кропивницький: ЦНТУ. – 2023. – С. 35-36.

36. Smirnov, O., Karapetyan, A., Fedorov, E., «Creating Neural Network and Single Solution Human-Based Metaheuristic Methods of Solving the Traveling Salesman Problem». CEUR Workshop Proceedings, Volume 3312, 2022, pp. 47-58.

37. Smirnov O., Kuznetsov A., Kryvinska N., Kiian A., Kuznetsova K. «Full Non-Binary Constant-Weight Codes». SN Computer Science, Vol 2, 337, 2021. <https://doi.org/10.1007/s42979-021-00739-w>.

38. Smirnov O., Kovalenko O., Kovalenko A., Kavun S. «Quantitative Risk Assessment Method Development in the Context of the SDLC-model». 2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), 2021, pp. 203-208, doi: 10.1109/PICST54195.2021.9772143

39. Smirnova T., Gnatyuk S., Berdibayev R., Avkurova Zh., Iavich M. «Cloud-Based Cyber Incidents Response System and Software Tools». Communications in Computer and Information Science, 2021, vol 1486. Springer, Cham. pp 169-184.

40. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». International Journal of Computing; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256.

41. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». CEUR Workshop Proceedings. Volume 2740, 2020, Pages 102-114.

42. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-

quantum application». 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18. 2020. P. 172-177.

43. Smirnov, O., Shekhanin, K., Kuznetsov, A., Krasnobayev, V. «Detecting Hidden Information in FAT». International Journal of Computer Network and Information Security (IJCNIS). Vol. 12, No. 3, 2020. PP.33-43.

44. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», CEUR Workshop Proceedings Volume 2608, 2020, Pages 633-645.

45. Smirnov O. Kuznetsov A., Zaichenko Yu., Pastukhov M., Oleshko O., Kuznetsova K., «Formation of Discrete Signals with Special Correlation Properties». International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2019; Odessa; Ukraine; 9-13 September 2019. P.22-28.

46. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». International Journal of Computing; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407.

47. Smirnov, O., Kuznetsov, A., Reshetniak, O., Ivko, N., Katkova, T., Kuznetsova, T., «Generators of Pseudorandom Sequence with Multilevel Function of Correlation». 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), Kyiv, Ukraine, 8 – 11 October 2019 . P.517-522.

48. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». CEUR Workshop Proceedings, Vol 2588, P. 90-106, 2019.

49. Kuznetsova, T., «Code-Based Schemes for Post-Quantum Digital Signatures», 10th IEEE International Conference on Intelligent Data Acquisition and

Advanced Computing Systems: Technology and Applications, IDAACS 2019; Metz; France; 18-21 September 2019. P. 707-712.

50. Smirnov, O., Kuznetsov, A., Stefanovych, O., Gorbenko, Y., Krasnobaev, V., Kuznetsova K. «Information Hiding Using 3D-Printing Technology», 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019; Metz; France; 18-21 September 2019. P.701-706.

51. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Averchev, A., Pastukhov, M., Kuznetsova, K., «Formation of Pseudorandom Sequences with Special Correlation Properties», 2019 3rd International Conference on Advanced Information and Communications Technologies, AICT -2019/ Lviv, Ukraine, 2-6 July, 2019, P. 395-399.

52. Смірнов, О.А., Усік П.С., Полігенько О.О., Одарченко Р.С., Терещенко Л.Ю. «Інформаційна технологія та програмне забезпечення для підвищення ефективності планування підсистеми базових станцій стільникового зв'язку». Проблеми телекомунікацій. № 1(26). С. 83-96. 2020.

53. Смірнов О.А., Усік П.С., Миронець І.В., Буравченко К.О., Якименко Н.М. «Метод підвищення ефективності розподіленої обробки даних у комп'ютерних системах операторів стільникового зв'язку» Вісник Черкаського державного технологічного університету. Технічні науки. №4. С. 103-110. 2020.

					ВКРМ-122.25.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78