Центральноукраїнський національний технічний університет
Центр заочної та дистанційної освіти
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
“____” _____ 2026 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи адаптивного регулювання складності мультимедійного аркадного комплексу із застосуванням алгоритмів машинного навчання”

Виконав здобувач вищої освіти
ІІ курсу, групи КН-25МЗ
ОПП «Комп’ютерні науки»
спеціальності F3 «Комп’ютерні науки»
_____ Патенко Н.М.
«____» _____ 2026 р.

Керівник проекту
кандидат технічних наук, доцент
_____ Смірнова Т.В.
«____» _____ 2026 р.

Нормоконтроль проекту
_____ Коваленко А.С.
«____» _____ 2026 р.

Рецензент _____
_____

Підписано КЕП
Центральноукраїнський національний технічний університет
Патенко Нікіта Максимович
Сертифікат: 18DF2C9B93CEA204040000008A9B34006A244306
Дійсний з 07.10.2026 14:38:36 по 07.10.2027 14:38:35

Центральноукраїнський національний технічний університет
Центр *Заочної та дистанційної освіти*
Кафедра *Кібербезпеки та програмного забезпечення*
Рівень вищої освіти *магістр*
Галузь знань _F_____*"*Інформаційні технології*"*_____
Спеціальність F3 *"*Комп'ютерні науки*"*
Освітньо-професійна (освітньо-наукова) програма _*"*Комп'ютерні науки*"*_

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.
_____ Олексій СМІРНОВ
« 1 » вересня 2026 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ДРУГИМ (МАГІСТЕРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Патенку Нікіті Максимовичу
(прізвище, ім'я, по батькові)

1. Тема роботи *Дослідження та програмна реалізація системи адаптивного регулювання складності мультимедійного аркадного комплексу із застосуванням алгоритмів машинного навчання*

2. Керівник роботи *Смірнова Тетяна Віталіївна, кандидат технічних наук, доцент*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 44-13 від 17.08.2026 року

3. Строк подання студентом роботи до захисту *2.12.2026 р.*

4. Мета та завдання випускної кваліфікаційної роботи*: Метою розробки є дослідження та програмна реалізація системи адаптивного регулювання складності мультимедійного аркадного комплексу із застосуванням алгоритмів машинного навчання*

5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Призначення та область використання.	*5. Впровадження системи в промислову експлуатацію*
2. Перегляд аналогічних існуючих систем.	*6. Наукова новизна.*
3. Опис і обґрунтування проектних рішень.	*7. Заходи з охорони праці та техніки безпеки.*
4. Етапи програмування системи.	*8. Висновки.*

6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Наукова новизна	*1 аркуш*
Структурна схема системи	*1 аркуш*
Функціональна схема системи	*1 аркуш*
Діаграма процесів	*1 аркуш*
Блок-схема алгоритму роботи додатку	*2 аркуша*

7. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Марченко К.М., к.т.н., доцент	06.10.2026	16.11.2026

8. Дата видачі завдання « 1 » вересня 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.10.2026 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2026 р.	
3.	Розробка моделі компонента	20.10.2026 р.	
4.	Розробка структур даних	25.10.2026 р.	
5.	Розробка алгоритмів системи	30.10.2026 р.	
6.	Програмування алгоритмів	10.11.2026 р.	
7.	Розрахунки з охорони праці та техніки безпеки	15.11.2026 р.	
8.	Оформлення ПЗ	17.11.2026 р.	
9.	Попередній захист роботи	2.12.2026 р.	

Дата видачі завдання
« 1 » вересня 2026 р.

Підпис керівника

_____ _____
(прізвище та ініціали)

Завдання прийнято до виконання
« 1 » вересня 2026 р.

Підпис здобувача

_____ _____
(прізвище та ініціали)

АНОТАЦІЯ

Патенко Н.М. Дослідження та програмна реалізація системи адаптивного регулювання складності мультимедійного аркадного комплексу із застосуванням алгоритмів машинного навчання. ФЗ Комп'ютерні науки. Центральноукраїнський національний технічний університет. Кропивницький. 2026.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи адаптивного регулювання складності мультимедійного аркадного комплексу із застосуванням алгоритмів машинного навчання.

Метою розробки є дослідження та програмна реалізація системи адаптивного регулювання складності мультимедійного аркадного комплексу із застосуванням алгоритмів машинного навчання.

Об'єктом дослідження є процес адаптивного регулювання складності мультимедійного аркадного комплексу із застосуванням алгоритмів машинного навчання.

Предметом дослідження є методи адаптивного регулювання складності мультимедійного аркадного комплексу із застосуванням алгоритмів машинного навчання.

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

Результат роботи – програмна реалізація системи адаптивного регулювання складності мультимедійного аркадного комплексу із застосуванням алгоритмів машинного навчання.

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, адаптивне регулювання, мультимедійний аркадний комплекс, машинне навчання

ABSTRACT

Patenko N.M. Research and Software Implementation of an Adaptive Difficulty Adjustment System for a Multimedia Arcade Complex Using Machine Learning Algorithms. F3 Computer Science. Central Ukrainian National Technical University. Kropyvnytskyi. 2026.

This final qualification thesis, submitted for the second (master's) level of higher education, presents the development of software designed for an adaptive difficulty adjustment system for a multimedia arcade complex, utilizing machine learning algorithms.

The objective of this development is to research and implement a software-based adaptive difficulty adjustment system for a multimedia arcade complex using machine learning algorithms.

The object of the research is the process of adaptively adjusting the difficulty of a multimedia arcade complex through the application of machine learning algorithms.

The subject of the research consists of methods for the adaptive adjustment of difficulty in a multimedia arcade complex using machine learning algorithms.

The research methodology draws upon principles from the theory of graphical information processing, mathematical statistics, and software engineering.

The outcome of the work is a software implementation of an adaptive difficulty adjustment system for a multimedia arcade complex, employing machine learning algorithms.

During the development of the software model, an analysis of existing hardware and software tools was conducted. All components of the developed software are described in detail.

A user-friendly interface has been developed, and instructions for operating the software are provided.

The software is designed for use on personal computers running the Windows 10/11 operating system. The program was developed using the Phaser framework.

Keywords: computer science, adaptive control, multimedia arcade system, machine learning

ВІДОМІСТЬ ПРОЕКТУ

		Текстові документи			
1	А4	Пояснювальна записка	86	–	
2	А4	Робоча програма	20	–	Додаток А
		Графічні документи			
3	А4	Наукова новизна	1	–	
4	А4	Структурна схема	1	–	
5	А4	Функціональна схема роботи системи	1	–	
6	А4	Діаграма взаємодії процесів	1	–	
7	А4	Блок-схема основної програми	1	–	
8	А4	Блок-схема роботи підпрограми	1	–	

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ 10
ВСТУП..... 11
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ 15
1.1 Призначення системи..... 15
1.2 Область застосування 17
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ 18
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти..... 18
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування 25
2.3 Розгорнута постановка завдання 26
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ 27
3.1 Опис функціонування системи 27
3.2 Розробка структурної схеми..... 33
3.3 Розробка функціональної схеми 36
3.4 Розробка діаграми процесів 38
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ 40
4.1 Розробка блок-схем та опис алгоритмів функціонування системи 40
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ..... 61
6 НАУКОВА НОВИЗНА 66

7 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ 67
7.1 Вступ 67
7.2 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста ... 69
7.3 Розробка заходів з поліпшення стану охорони праці 71
7.4 Техніка безпеки та протипожежна профілактика 73
7.5 Розрахункова частина 74
8 ОСНОВНІ ВИСНОВКИ 77
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 79

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ПЗ – програмне забезпечення

ВСТУП

Актуальність теми. Глобальний ринок аркадних ігор переживає значне зростання, зумовлене зростанням доходів населення, технологічним прогресом та відродженням інтересу до розваг, що базуються на враженні. У 2025 році ринок оцінювався в 4,75 млрд доларів США, а до 2033 року, за прогнозами, він досягне 7,69 млрд доларів США, що демонструє сукупний річний темп зростання (CAGR) на рівні 5,5% протягом прогнозованого періоду (2026-2033). Це розширення підживлюється як ностальгією за ретро-іграми, так і інтеграцією передових технологій, таких як штучний інтелект, Інтернет речей та віртуальна/доповнена реальність (VR/AR), які покращують захопливий досвід для гравців.

Масштаб ринку та зростання:

– Розмір ринку та прогнози. Очікується, що ринок аркадних ігор зросте на 1999,9 мільйона доларів між 2025 і 2029 роками, зі середньорічним темпом зростання 1,9%. Це зростання значною мірою пояснюється зростанням кількості комерційних ігрових залів та встановленням ігрових автоматів у громадських місцях, таких як торгові центри.

– Регіональне домінування. Північна Америка наразі займає найбільшу частку світового ринку аркадних ігор, складаючи 40%, завдяки технологічному прогресу та ефективним маркетинговим стратегіям. Однак очікується, що Азіатсько-Тихоокеанський регіон збільшить свою частку ринку в найближчі роки.

Рушійні сили зростання:

– Збільшення доходу, що залишається. У світовому масштабі зростання доходів населення призводить до збільшення витрат на дозвілля, і аркадні ігри безпосередньо виграють від цієї тенденції.

– Технологічні інновації. Інтеграція штучного інтелекту, машинного навчання та підключення Інтернету речей створює персоналізований та

захопливий ігровий досвід, знижує експлуатаційні витрати та підвищує попит на передові аркадні автомати.

– Експериментальні розваги: Зміна поведінки споживачів у бік мультисенсорних, високотехнологічних розваг, особливо в міських центрах, стимулює розвиток сучасних аркадних установок.

– Кіберспорт та змагальні ігриЗростаюча популярність кіберспорту ще більше стимулює попит на високопродуктивні аркадні автомати, що підтримують змагальну гру.

Конкурентний ландшафт:

– Ключові гравці. Ринок характеризується жорсткою конкуренцією, де лідерами галузі є такі великі виробники, як Namco, Chicago Gaming, Valley-Dynamo, True Nightmare та Innovative Concepts in Entertainment. Серед інших значних гравців – Sega Corporation, Bandai Namco Entertainment, Konami Holdings Corporation та Raw Thrills, Inc.

– Стратегічний фокус. Компанії зосереджуються на інноваціях продуктів, дослідженнях та розробках, стратегічних партнерствах, злиттях та поглинаннях, щоб розширити свій охоплення ринку та покращити портфель продуктів.6Також робиться акцент на розробці продуктів, орієнтованих на потреби клієнтів та регіону, для задоволення різноманітних уподобань споживачів.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи адаптивного регулювання складності мультимедійного аркадного комплексу із застосуванням алгоритмів машинного навчання.

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

– Огляд існуючих систем адаптивного регулювання складності мультимедійного аркадного комплексу із застосуванням алгоритмів машинного навчання.

– Дослідження системи адаптивного регулювання складності мультимедійного аркадного комплексу із застосуванням алгоритмів машинного навчання.

– Програмна реалізація системи адаптивного регулювання складності мультимедійного аркадного комплексу із застосуванням алгоритмів машинного навчання.

Об'єктом дослідження є процес адаптивного регулювання складності мультимедійного аркадного комплексу із застосуванням алгоритмів машинного навчання.

Предметом дослідження є методи адаптивного регулювання складності мультимедійного аркадного комплексу із застосуванням алгоритмів машинного навчання.

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 адаптивного регулювання складності мультимедійного аркадного комплексу із застосуванням алгоритмів машинного навчання.

– Розроблено вітчизняний продукт адаптивного регулювання складності мультимедійного аркадного комплексу із застосуванням алгоритмів машинного навчання, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі адаптивного регулювання складності мультимедійного аркадного комплексу із застосуванням алгоритмів машинного навчання.

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 LVIII науково-технічній конференції здобувачів вищої освіти «Наука-виробництву», основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у статті збірника праць молодих науковців ЦНТУ, випуск №17.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи адаптивного регулювання складності мультимедійного аркадного комплексу із застосуванням алгоритмів машинного навчання, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Штучний інтелект (ШІ) та цифрові ігри розвиваються разом протягом багатьох років. Ігри забезпечують контрольоване, вимірюване та різноманітне середовище для тестування ідей ШІ, тоді як ШІ, у свою чергу, забезпечує багатший, більш адаптивний та креативний ігровий досвід. Протягом останнього десятиліття цей зв'язок прискорився з розвитком глибокого навчання та, останнім часом, генеративних моделей, змінюючи як фундаментальні дослідження, так і промислові практики. Комплексний огляд підкреслює, як ігри служать випробувальними майданчиками для методів ШІ, таких як планування, навчання з підкріпленням, генерація процедурного контенту та бенчмаркінг, водночас унікальні вимоги ігрового дизайну стимулюють методологічний прогрес у ШІ [1]. Паралельно, моделі великих мов вже використовувалися для розробки наративного контенту та квестів, наприклад, рольових ігор, демонструючи як їхню технічну доцільність, так і проблеми узгодженості, контролю та безпеки [2]. На більш теоретичному рівні Капраро та ін. запропонували теорію ігор на основі мови, яка ілюструє, як лінгвістичний контент впливає на стратегічні рішення, тим самим виходячи за межі традиційних теоретичних рамок ігор, зосереджених лише на економічних вигодах [3].

Застосування штучного інтелекту до аналітики гравців та балансування ігор також значно зросло. Масштабна телеметрія дозволяє виявляти нові тактики та підтримує дизайнерські рішення в реальних комерційних умовах [4]. Додатково, були розроблені підходи до поведінкового профілювання для оцінки рівнів майстерності та стилів гри, що поєднують сигнали мультимодальних датчиків з алгоритмами машинного навчання [5]. Прогнозна аналітика також поширюється на виявлення відтоку, особливо в цифровому навчанні на основі

ігор (DGBL), де такі моделі, як логістична регресія, дерева рішень та випадкові ліси, були застосовані для прогнозування рівня відсіву та утримання, пропонуючи практичні висновки для розробників освітніх ігор [6].

Окрім ігрової механіки, штучний інтелект дедалі більше формує взаємодію гравців та їхнє навчання за допомогою генеративних та адаптивних підходів. Наприклад, було показано, що інтеграція ChatGPT у неігрових персонажів у середовищах, керованих сюжетними лініями, впливає на письмову успішність учнів та їхню залученість до ігрового навчання [7]. Водночас вивчається роль поведінки, пов'язаної з пошуком допомоги, коли учні взаємодіють з чат-ботами, керованими штучним інтелектом, у сценаріях DGBL, оскільки ці взаємодії можуть забезпечити інструментальну підтримку, не витісняючи важливі когнітивні процеси [8].

Імерсивні технології також взаємодіють з іграми на основі штучного інтелекту. Ігри з доповненою реальністю, такі як системи вивчення алфавіту на основі доповненої реальності для дітей молодшого віку, були оцінені на предмет зручності використання та прийнятності, що свідчить як про педагогічний потенціал, так і про практичні проблеми [9]. Крім того, серйозні ігри на основі штучного інтелекту все частіше використовуються для вирішення соціальних проблем. Наприклад, відеоігри, які вбудовують алгоритми для виявлення гендерних упереджень у взаємодії в класі, підкреслюють, як серйозні ігри можуть сприяти рівності та боротися зі стереотипами в середній освіті [10]. Аналогічно, були розроблені фреймворки штучного інтелекту для підтримки розвитку соціально-емоційних навичок та розпізнавання емоцій у дітей з розладом аутистичного спектру, тоді як методи скринінгу на основі серйозних ігор, що використовують відстеження погляду та глибокі нейронні мережі, пропонують інноваційну діагностичну підтримку [11, 12].

1.2 Область застосування

У рамках професійної підготовки та вищої освіти можна спостерігати дві збіжні тенденції. По-перше, соціально інтелектуальний ШІ все більше інтегрується в парадигми кооперативних ігор, де досліджуються компоненти співпраці людини та ШІ, що базуються на даних, на їх вплив на залученість та спільну діяльність [13]. По-друге, віртуальні гравці на базі ШІ були вбудовані в симуляційні платформи для навчання управлінню виробництвом, що дозволяє адаптувати рівні складності та покращувати прийняття рішень за реалістичних обмежень [14]. Паралельно, вища освіта спостерігає нові способи використання генеративного ШІ для проектування цифрових квест-кімнат, де студенти спільно створюють навчальний досвід на основі ігор, тим самим розвиваючи креативність, навички вирішення проблем та цифрової грамотності [15].

Крім того, штучний інтелект проникає в контекст підприємницької та бізнес-навчальної діяльності. Нещодавня робота демонструє, як сценарії, згенеровані штучним інтелектом, у серйозних іграх можуть бути використані для моделювання процесів прийняття рішень у технологічних стартапах, піддаючи учнів невизначеності, ітераціям та інноваційним практикам, що відображають реальні виклики [16]. Цей перетин штучного інтелекту, ігор та підприємництва підкреслює ширший потенціал навчання на основі ігор, що виходить за рамки традиційної освіти та розваг.

Цей спектр застосувань ілюструє, як ШІ в іграх охоплює континуум від фундаментальних підходів до високоприкладних контекстів

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи адаптивного регулювання складності мультимедійного аркадного комплексу із застосуванням алгоритмів машинного навчання, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

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

Програмне забезпечення для створення ігор для початківців

Construct 3

Наразі, у своїй третій ітерації, Construct 3 – це інструмент для розробки блокових ігор, який не потребує встановлення, працює безпосередньо з вашого браузера і навіть може бути підключений до вашого улюбленого хмарного сервісу. Якщо ви хочете створювати ігри в найпростішій формі, це інструмент із простим у використанні інтерфейсом перетягування.

Construct 3 чудово підходить для створення 2D-ігор, і він також відомий своєю простотою використання. Крива навчання цьому програмному забезпеченню дуже поверхнева, насправді, вам навіть не потрібні жодні навички мови програмування чи знання кодування, щоб користуватися цим програмним забезпеченням.

Construct 3 має скіни, пропонує анімацію на часовій шкалі, вбудований редактор зображень і пропонує розробнику набір візуальних ефектів, що включає ефекти частинок, художні поєднання, спотворення та відкидання тіней, серед іншого. Він також дуже швидкий і працює на всіх популярних платформах, що може допомогти вам у розробці ваших ігрових продуктів.

Це, безумовно, один із варіантів, який варто розглянути, якщо ви тільки починаєте свою кар'єру розробника відеоігор або якщо написання коду не є сильною стороною вашого ігрового бізнесу.

GameMaker Studio 2

GameMaker Studio 2 – ще один ігровий рушій, створений з урахуванням потреб початківців-розробників ігор та ентузіастів, і є одним із найбільш інтуїтивно зрозумілих та простих у використанні варіантів.

GameMaker Studio також є ігровим рушієм, який пропонує простий інтерфейс з функцією перетягування та варіанти робочого процесу для різних платформ, що дозволяє експортувати гру на настільні комп'ютери, веб-сайти, UWP, мобільні пристрої та консольні платформи. GameMaker Studio 2 є чудовим варіантом з точки зору програмного забезпечення для створення ігор для початківців і містить список інструментів, що включає скінінг, стикування, редактор об'єктів і редактор скриптів.

Він також постачається з системою робочих процесів, готовою до перетягування, попередньо встановленою бібліотекою подій та дій, а також функцією перевірки коду для програмістів, які розробляють ігри без великого досвіду.

Мова GameMaker на основі програмування на C також дозволяє вам використовувати інші мови залежно від ваших потреб.

Також є спеціальні опції, такі як редактор кімнат, набори плиток, редактор зображень, а також підтримка фізичних двигунів, таких як Box2D та двигун фізики частинок LiquidFun від Google, а також аніматор Spine 2D від Esoteric Software.

AppGameKit

AppGameKit, інструмент, розроблений з метою допомогти у створенні ігор для мобільних пристроїв одночасно, корисний для 2D та 3D ігор і кросплатформних додатків сьогодні.

Якщо ви хочете використовувати ігрові двигуни для створення гри спеціально для App Store або Google Play Marketplace, це програмне забезпечення варто обрати.

Цей рушій найкраще підходить для розробників із середнім рівнем навичок, він пропонує такі функції, як система перетягування, повністю

інтегрований редактор, рендеринг Vulcan та OpenGL, редактор сцен, редактор звуку та музики тощо.

Він також безкоштовний та пропонує підтримку 3D- та 2D-ігор на різних платформах, включаючи Windows, Mac, Linux, HTML5 та інші.

Чудове рішення для ігрових компаній з обмеженими ресурсами, яким потрібен інструмент розробки широкого спектру.

Stencyl

Якщо ви хочете створювати ігри та стратегії без програмування та пов'язаних з ним проблем, Stencyl – один із варіантів. Stencyl має простий інтерфейс перетягування елементів для гейм-дизайнерів.

Він дозволяє створювати прості світи за допомогою дизайнера сцен Stencyl та пропонує можливості створення персонажів за допомогою редактора акторів Stency. Це ідеальний інструмент для 2D-ігор, орієнтованих на ринки iOS та Android.

Фактично, ігрові експерти та розробники, які використовували Stencyl, бачили, як їхні 2D-ігри займають верхні позиції в App Store та Android Marketplace.

Тож, якщо ви новачок і хочете створити, скажімо, пригодницьку гру без жодного досвіду написання коду, Stencyl пропонує вам шлях для досягнення цієї мети.

Програмне забезпечення для відеоігор для розробників середнього рівня

Godot

Повністю безкоштовний та з відкритим вихідним кодом, Godot – це фантастичний ресурс для створення ігор як у 2D, так і в 3D.

Godot використовує архітектуру на основі вузлів для спрощення свого інтерфейсу, пропонує систему перетягування, яку дизайнер відеоігор може використовувати для елементів інтерфейсу користувача, предметів, спрайтів та інших ігрових ресурсів, має спеціальний рушій розробки 2D-ігор та пропонує

чудовий інструмент під назвою 3D-рендерер, щоб легко втілювати тривимірні творіння та 3D-моделі в життя.

Godot використовує власну мову сценаріїв під назвою GDScript, а також підтримує різноманітність у знаннях кодування, включаючи C#, C++ та візуальні сценарії за допомогою цього безкоштовного програмного забезпечення.

Godot має зростаючу спільноту підтримки програмістів, які взаємодіють один з одним та діляться знаннями з членами цієї спільноти, що також користуються платформою.

RPG Maker

Ще один варіант для тих, хто має обмежений досвід програмування. RPG Maker, як випливає з назви, – це конструктор рольових ігор, який насправді не вимагає навичок програмування для створення ігор з нуля. Його можна використовувати для створення ігор на таких платформах, як Windows, Mac, IOS, Android – і все це без досвіду програмування.

RPG Maker ідеально підходить для інді-розробників та досвідчених ігор, які шукають ігрові рушії, орієнтовані на пригодницькі ігри в стилі RPG.

Що стосується категорій ігрового дизайну, рольові ігри є великим підрозділом, і цей пакет розробки ігор – це ваш універсальний магазин для цього жанру.

Він постачається в різних ітераціях і, як і всі хороші програми для створення ігор, пропонує своїм користувачам інтерфейс перетягування.

Програмне забезпечення для створення 3D-ігор для досвідчених користувачів

CryEngine

Цю ігрову платформу спочатку створювали не для масового створення ігор, а для розробки однієї конкретної гри. Якщо її назва вас не підказує, то цією популярною грою був легендарний шутер від першої особи Far Cry. CryEngine – це кросплатформна програма для розробки ігор, яка досі вважається спеціалізованим інструментом для шутерів від першої особи, але це не все, що вона може робити.

Він має чудову функцію «пісочниці», завдяки якій гра дотримується філософії та принципу «що бачиш, те й отримуєш».

Пісочниця включає підтримку FBX, інтуїтивно зрозумілий дизайн рівнів, кінематографічний редактор Trackview, систему скриптів Flowgraph та інтеграцію Substance з редактором матеріалів. Вона також пропонує деякі вражаючі фізичні симуляції як частину свого набору навичок.

Інші ресурси включають можливість рендерингу в режимі реального часу та простий у використанні шар перетворення аудіо – один з унікальних ігрових інструментів програмного забезпечення, який дозволяє спростити робочий процес обробки звуку в іграх.

Інші рішення також включають спеціалізований інструмент штучного інтелекту та анімації з параметричною скелетною анімацією та технологією персонажів, багатошарову навігаційну сітчасту структуру та кеш геометрії.

Unity

Якщо ви залежні від інді-ігор або маєте мобільну гру, від якої просто не можете відірватися (і так, ми маємо на увазі ігри для IOS, Android, Windows, Mac, 2D-ігри та 3D-ігри), є велика ймовірність, що вона була розроблена на Unity. З охопленням мільярдів пристроїв, Unity, безумовно, є одним з найкращих ігрових рушіїв AAA.

Причини популярності платформи Unity пов'язані з її потужним набором інструментів, який включає потужну продуктивність, зокрема системи віртуальної реальності та можливості створення AR-ігор, а також конвеєр рендерингу високої чіткості, що підтримує можливості прямого та відкладеного рендерингу.

Крім того, Unity також має префаби для згладжування робочих процесів, графік візуальних ефектів, який дозволяє перетягувати готові вузли ефектів, інтерфейс системи інтелектуальних камер Cinemachine та зразкові можливості фотограмметрії для зручного перетворення фотографій та відео на ігрові ресурси як частину свого набору інструментів. І на додачу до всього цього, це безкоштовне програмне забезпечення для розробки ігор!

Unreal Engine

Розроблений Epic Games, Unreal Engine вважається найкращим програмним забезпеченням для геймдизайну, особливо для ігор AAA-класу, багатьма людьми в індустрії та провідними студіями.

Одна з багатьох причин, чому Epic Games та їхня флагманська платформа для розробки ігор Unreal має таку шановану репутацію серед геймдизайнерів, незважаючи на крутий процес навчання, полягає в їхньому фізичному двигуні, який дозволяє розробникам створювати неймовірно реалістичні візуальні ефекти.

Функції, що пропонуються Unreal Engine, включають попередньо створені модульні системи, налаштовувані плагіни, інтерфейс інтеграції системи керування версіями та багато іншого.

Unreal також має цілий список платформ, які підтримує, включаючи консолі сучасного покоління, такі як XBOX, Playstation, Nintendo Switch, Valve, Oculus Rift, VR, AR, iOS, Android, Windows, HTML5 та багато інших. Спільнота Unreal також є чудовим ресурсом для розробників, на який можна покластися, якщо вам потрібні запитання, навчальні посібники та всебічна підтримка.

Amazon Lumberyard

Amazon Lumberyard – це безкоштовний пакет розробки ігор у всіх сенсах цього слова, безкоштовний. Вам не потрібно купувати бізнес-ліцензію чи платити роялті за публікацію ваших консольних чи мобільних ігор, турбуватися про авторські права від Amazon чи щось подібне – як для любителів, так і для професіоналів.

Єдина умова, яку Amazon ставить перед творцями, полягає в тому, що якщо вони використовуватимуть хмару для розміщення багатокористувацьких онлайн-ігор, хмарний сервіс має бути власним AWS бренду, який вам доведеться придбати.

Amazon Lumberyard базується на CryEngine, хоча Amazon внесла кілька ключових оновлень у своє програмне забезпечення. Тож як геймдизайнер ви

можете покластися на здатність CryEngine розробляти чудові ігрові механіки та створювати ігри для IOS, Android, Xbox One, Playstation 4 та ПК.

Amazon Lumberyard також пропонує підтримку Twitch, функцію спільноти, а також пропонує графічний компонент, який дозволяє створювати приголомшливі візуальні ефекти за допомогою шейдерів, динамічного освітлення, інструментів рослинності, системи ефектів частинок, об'ємного туману тощо.

Звичайно, це не єдині доступні варіанти програмного забезпечення! Інші, такі як Buildbox, Cocos2D та Clickteam Fusion, також визнані популярним програмним забезпеченням для створення ігор як для людей з різним рівнем навчання, як для початківців, так і для тих, хто має великий досвід програмування.

Однак, якщо ви прагнете стати одним із розробників ігор елітного рівня, який досягне вищих ешелонів ігрового світу, буде важко не звертати уваги на програмне забезпечення для розробки ігор, перелічене вище.

Вони є найкращими в галузі та використовуються командами розробників по всьому світу для створення ігор та додатків різної якості та можливостей, а також у різних категоріях ігрового дизайну. Багато з них пропонують унікальні та спеціалізовані набори інструментів, які роблять їх більш придатними для певного типу гри, тому вибирайте з розумом, виходячи зі стилю гри, який уявляє ваша команда, та інвестуйте відповідно.

Важливо пам’ятати, що багато з цих програмних рішень надаються безкоштовно, якщо ви дотримуєтеся певних параметрів, а платежі нараховуються, якщо ваша гра досягає певного рівня успіху. Тому обов’язково прочитайте всі умови та, можливо, уважно перегляньте відгуки.

Також доступні різні версії цього програмного забезпечення. Деякі з них можуть пропонувати лише обмежені функції у безкоштовних версіях, і не всі, що ми перерахували вище, тому слідкуйте за цим також.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Для розробки клієнтської частини мультимедійного аркадного комплексу та реалізації інтерактивного ігрового середовища було обрано сучасний фреймворк Phaser.js. Цей інструментарій є безкоштовним рішенням із відкритим кодом, оптимізованим для створення кросплатформних двовимірних (2D) ігор за допомогою мов програмування JavaScript та TypeScript.

Особливості фреймворку та його застосування в аркадних комплексах

Phaser.js забезпечує повний цикл розробки мультимедійних додатків, включаючи управління ігровими ресурсами, відтворення аудіоефектів, обробку анімації та інтеграцію вбудованих фізичних рушіїв.

У контексті розробки системи адаптивного регулювання складності ігрового процесу ключовими перевагами Phaser.js є:

– Кросплатформна сумісність: можливість стабільного запуску аркадного комплексу на будь-яких сучасних настільних та мобільних платформах безпосередньо в середовищі веббраузерів.

– Адаптивний рендеринг (WebGL та Canvas): автоматичний вибір оптимального методу візуалізації залежно від апаратних можливостей пристрою користувача, що гарантує високу продуктивність, відсутність затримок і плавність зміни кадрів (FPS), критично важливі для динамічних аркад.

– Гнучкість інтеграції з алгоритмами машинного навчання: відкрита архітектура фреймворку дозволяє легко пов'язувати ігрову логіку з зовнішніми або вбудованими аналітичними модулями для реального часу збору телеметрії, оцінки дій гравця та динамічної зміни параметрів складності.

Завдяки своїй швидкодії, потужним засобам роботи з мультимедіа та кросплатформності, Phaser.js забезпечує створення ефективного, візуально насиченого і технологічно гнучкого програмного середовища для дослідження методів адаптивного управління в аркадних комплексах.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускну кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи адаптивного регулювання складності мультимедійного аркадного комплексу із застосуванням алгоритмів машинного навчання.

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) сформувати висновки про виконаний обсяг робіт та одержані результати.

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Термін «серйозні ігри» позначає «ігри, основною метою яких не є розвага, задоволення чи веселощі» [1]. Хоча деякі автори пов’язують їх виключно з освітою [2], насправді вони можуть служити й іншим цілям [3], таким як донесення змістовного повідомлення до гравця (і, якщо можливо, сприяння змінам за допомогою цього повідомлення; див., наприклад, [4]) або створення прямих результатів у реальному світі (наприклад, вирішення відкритих дослідницьких задач [5]).

Хоча серйозні ігри можуть бути і були реалізовані як нецифрові, наприклад, настільні ігри, наразі вони здебільшого реалізуються як відеоігри [6]. Хоча ми не дотримуємося ідеї обмеження серйозних ігор виключно цифровою сферою, як це роблять деякі автори (див., наприклад, [7]), тим не менш, у даній магістерській роботі ми розглядатимемо лише цифрові серйозні ігри. Хоча такі ігри мають спільні ключові елементи та методи процесу розробки з несерйозними, тобто розважальними, відеоіграми, їхнє призначення, відмінне від розважального, спричиняє низку відмінностей. Основна відмінність полягає в необхідності поєднання захопливого ігрового досвіду для гравців (вимога до кожної гри) з утилітарною функцією, для якої була розроблена серйозна гра (вимога, типова для корпоративних інформаційних систем, а не відеоігор) [8]. Не менш важливою є потреба в оцінюванні або еквівалентності як у серйозних іграх, так і в самих іграх: перше – внутрішнє вимірювання продуктивності гравця з метою надання йому відповідного зворотного зв'язку, а також забезпечення адаптивності та персоналізації для задоволення індивідуальних потреб, наприклад, щодо стилів навчання або швидкості надання інформації, а друге – перевірка ефективності серйозної гри в досягненні поставлених цілей [9].

Генеративний штучний інтелект, або GenAI, являє собою «обчислювальні методи, здатні генерувати, здавалося б, новий, змістовний контент, такий як текст, зображення або аудіо, з навчальних даних» [10]. Яскравими прикладами рішень на основі GenAI є OpenAI GPT (для генерації тексту) [11], OpenAI DALL-E 2 (для генерації зображень) [12], SUNO (для генерації музики) [13] та GitHub Copilot (для генерації програмного коду) [14].

Впровадження GenAI у розробку ігор є справді революційним, оскільки воно не лише може заощадити час і гроші, але й здатне надавати високоякісні ресурси [15]. Особливо важливими перевагами є те, що творчий потенціал стає доступним для людей, які не мають художніх навичок, що інструменти GenAI можуть створювати велику кількість варіацій ітеративним способом (без розчарування, як це зробив би художник-людина після отримання запиту на зміни кілька разів), і що після навчання моделі результати доступні майже миттєво [15].

Цей потенціал був помічений індустрією відеоігор, і до середини 2025 року використання GenAI під час виробництва або ігрового процесу було розкрито для 7818 ігор, опублікованих у Steam, що відповідає 7% від усього каталогу Steam [16]. Найчастіше розкриваним варіантом використання для розробки ігор (зазначеним приблизно в 60% розкриттів) було створення візуальних ресурсів (персонажі, фони, 3D-моделі, текстури тощо), далі йдуть аудіо (наприклад, фонова музика, голоси персонажів, закадровий голос), текст і розповідь (наприклад, описи предметів або сюжетна лінія), маркетингові та рекламні матеріали, а також ігровий код/логіка [16]. Все більша кількість ігор (наприклад, AI Roguelite, inZOI, Never Ending Dungeon, DREAMIO, AIdventure) використовують GenAI під час виконання, наприклад, для створення аудіовізуальних ресурсів, рівнів гри та діалогових відповідей та поведінки неігрових персонажів [16].

Хоча впровадження GenAI ігровою індустрією гальмується кількома бар'єрами, включаючи проблеми інтелектуальної власності та авторських прав (на що вказали 59% респондентів, опитаних Boston Consulting Group), низьку

надійність результатів GenAI (50%), вважаючи поточну технологію GenAI неготовою або непридатною для впровадження (43%), проблеми кібербезпеки та конфіденційності даних (33%), а також брак можливостей та практичних знань у співробітників (28%) [17], очікувані переваги роблять подальший прогрес неминучим. З цієї причини варто дослідити безліч способів, за допомогою яких GenAI може підтримувати розробку серйозних ігор, враховуючи не лише фактичну фазу розробки, але й весь життєвий цикл гри. А оскільки, як пояснювалося раніше, розробка серйозних ігор відрізняється від розробки відеоігор, розроблених виключно для розваги, обсяг можливої підтримки також відрізняється, що вимагає досліджень, які безпосередньо зосереджені на серйозних іграх, а не на відеоіграх загалом. Дана магістерська робота прагне усунути цю прогалину, досягнувши наступних цілей:

– Схемувати модель життєвого циклу розробки серйозних ігор, яка вказує на його ключові фази з відмінним характером.

– Визначити літературні джерела, які не обов'язково присвячені використанню GenAI у сфері серйозних ігор, але які, тим не менш, описують застосування GenAI, що можуть бути використані в життєвому циклі розробки серйозних ігор.

– Призначити визначені програми GenAI відповідним фазам запропонованої моделі життєвого циклу розробки серйозних ігор.

Перша з цих цілей була досягнута шляхом критичного порівняння класичного життєвого циклу розробки системи (SDLC) [18], життєвого циклу розробки відеоігор (GDLC) [19] та попередньої пропозиції моделі життєвого циклу, розробленої спеціально для серйозних ігор [20], а потім специфікації нової моделі, що усуває їхні недоліки щодо адекватного охоплення фаз життєвого циклу серйозних ігор.

Дві інші цілі були досягнуті шляхом проведення наративного огляду відповідної літератури в контексті запропонованої моделі життєвого циклу. Ми обрали наративний, а не систематичний огляд, оскільки його метою було не зіставлення існуючої літератури, а зіставлення можливих на даний момент форм

підтримки (підкріплених існуючою літературою) для серйозних етапів життєвого циклу розробки ігор. Таким чином, ми уникнули трудомісткого аналізу всієї відповідної літератури (натомість ми навели лише кілька достатніх прикладів для відповідних типів застосувань GenAI) та не були обмежені стратегією пошуку, визначеною в протоколі систематичного огляду (натомість ми розробили додаткові вузькофокусовані запити для пошуку статей, що описують застосування GenAI, про існування яких було відомо, але які не були ідентифіковані попередніми запитами ширшого масштабу).

Життєвий цикл розробки серйозних ігор

Оскільки серйозні ігри, реалізовані як цифрові ігри, є типом програмного забезпечення, їхній життєвий цикл можна змоделювати за класичним життєвим циклом розробки системи (SDLC), що складається з семи фаз. Перша з них – планування, коли визначаються обсяг проекту, цілі та вимоги, а також створюється початковий план проекту. Друга – фаза аналізу, яка включає збір та перегляд даних про вимоги до проекту для створення повністю детальної документації вимог. Далі йде фаза проектування, яка включає визначення архітектури проекту та створення документа проектування програмного забезпечення (SDD). На наступній фазі кодування розробники пишуть початковий код та створюють функціональний прототип програмного забезпечення. Потім починається тестування: код перевіряється та усуваються помилки для досягнення вдосконаленої, оптимізованої версії програмного забезпечення. Після завершення тестування, на фазі розгортання, програмне забезпечення переміщується у виробниче середовище та стає доступним для кінцевих користувачів. Заключна фаза обслуговування оновлює код постійними виправленнями та вдосконаленнями, щоб забезпечити його функціональність з часом [18].

Як згадувалося раніше, розробка ігор відрізняється від розробки корпоративних інформаційних систем (для яких спочатку був призначений SDLC). Відповідно, різні автори запропонували кілька визначень життєвого циклу розробки ігор (GDLC), які пізніше синтезували в модель, що складається з

шести фаз [19]:

– Ініціація, на якій створюється приблизне уявлення про гру, що розробляється.

– Передпродакшн, який включає створення та подальше доопрацювання ігрового дизайну й ітеративну розробку ігрових прототипів. Цей етап завершується затвердженням переробленого ігрового дизайну.

– Виробництво, що складається з розробки вихідного коду, ігрових ресурсів та інтеграції обох цих елементів.

– Тестування, метою якого є перевірка зручності використання та іграбельності гри. Це внутрішній процес, у якому беруть участь члени команди розробників ігор.

– Бета-тестування, яке передбачає залучення зовнішніх тестувальників (запрошених або самостійно залучених) для спроби виявити помилки, які не були виявлені внутрішньо на попередньому етапі.

– Реліз, що позначає час, коли гра досягла своєї остаточної форми та була випущена для публіки. Він включає планування технічної підтримки та майбутніх продовжень гри.

Хоча розробку серйозних ігор можна розглядати як з точки зору SDLC (див., наприклад, [22]), так і GDLC (див., наприклад, [23]), більш доречною була б модель, яка не лише вибирає та поєднує найважливіші елементи цих двох, але й додає елементи, які вони обидва пропускають (наприклад, оцінку того, чи досягає серйозна гра своїх запланованих утилітарних цілей). І хоча існує модель життєвого циклу, розроблена спеціально для серйозних ігор, вона дуже спрощена, оскільки визначає лише три загальні фази: проектування (включаючи навчальний дизайн та дизайн гри), розробка (гри та аналітичної моделі) та оцінка (включаючи метрики та взаємодії) [20]. З огляду на це, ми вирішили запропонувати власну модель життєвого циклу розробки серйозної гри, що складається з таких фаз:

– Розробка ідей, під час якої висувається початкова концепція та визначаються основні принципи проектування. Зазвичай розглядається кілька

концепцій-кандидатів, з яких вибирається одна, допрацьовується та зрештою затверджується для реалізації.

– Дизайн, що складається з дизайну контенту та дизайну програмного забезпечення, в рамках якого розробляється документація та матеріали, але не створюється життєздатний прототип. Зауважте, що ми формально не розрізняємо ігровий контент та навчальний контент, оскільки навіть у випадку, коли вони не є однаковими, їх можна розробляти за допомогою однакових методів. Аналогічно, ми не розрізняємо ігрову механіку та ігрові ресурси, оскільки перші є або різновидом контенту, подібним до другого (коли використовується багаторазовий ігровий движок, а правила гри вказані як його вхідні дані), або вбудованими в програмне забезпечення. Фаза завершується, коли приймається рішення про розробку прототипу.

– Прототипування, що складається зі створення прототипів контенту та програмного забезпечення, в якому два компоненти об'єднуються в ігрову форму, більш-менш схожу, але не ідентичну, майбутній релізній версії гри. Після розробки прототип представляється тестувальникам.

– Зворотній зв'язок з користувачем, під час якого тестувальники-люди відтворюють прототип і повідомляють про свої думки. Ці думки об'єднуються та призводять або до запиту на зміни та повернення до етапу проектування, або до прийняття користувачем, що призводить до наступного етапу.

– Розробка, що складається з розробки контенту та програмного забезпечення, в рамках якої розробляється гра у версії, яка вважається фінальною.

– Тестування, що включає закриту групу тестувальників-людей, яким представлена альфа-версія гри. На відміну від попередніх прототипів, альфа-версія не повинна містити очевидних (тобто відомих розробникам) помилок, коду scaffolding та відсутнього вмісту. На відміну від релізної версії, альфа-тестери можуть очікувати виявлення серйозних програмних помилок та збоїв ігрового процесу. Виявлені помилки повідомляються, і гра повертається до попередньої фази. Якщо важливих помилок не залишається, гра розгортається

для кінцевих користувачів.

– Використання, під час якого користувачі грають у гру. Навіть на цьому етапі можуть бути виявлені помилки, про які можна повідомити команді розробників для розробки патчів. Незважаючи на це, збираються дані про використання та оцінки користувачів.

– Оцінювання, яке проводиться після збору достатньої кількості даних. Його результати не лише підтверджують ефективність гри у досягненні її серйозної мети, але й можуть вказати на її сильні та слабкі сторони, і, як таке, можуть спонукати до ідеї наступної версії гри.

3.2 Розробка структурної схеми

Структурну схему розробленої системи наведено на рисунку 3.1. Вона відображає архітектуру комплексу адаптивного регулювання складності мультимедійного аркадного середовища на основі алгоритмів машинного навчання. Схема складається з таких основних компонентів:

– ігровий рушій Phaser.js;

– блок функцій Phaser.js;

– блок прикладних застосувань;

– блок введення даних.

Блок функцій Phaser.js

Фреймворк Phaser пропонує широкий набір інструментів, що значно спрощують і прискорюють процес розробки ігор. Серед його ключових можливостей слід виділити:

– Підтримка різних фізичних движків: інтеграція Arcade Physics, Matter.js та ImpactJS забезпечує інструменти для імітації гравітації, прискорення та точного виявлення зіткнень.

– Управління сценами: зручна організація переходів, ігрових рівнів та інтерфейсів меню.

– Завантаження та обробка ресурсів: вбудовані модулі для роботи із

зображеннями, аудіо, відео, спрайтами та тайлмапами.

– Анімація: підтримка спрайтових аркушів та покадрової анімації «з коробки».

– Мультиплатформний ввід: обробка сигналів із клавіатури, миші, сенсорних екранів та ігрових контролерів.

– Аудіосистема: вбудовані інструменти для відтворення фонової музики, звукових ефектів та аудіопетель.

– Керування камерою: масштабування, обертання та прокручування видимої ігрової області.

– Тайлмапи: швидке створення масштабних структурованих світів і рівнів.

– Готові ігрові об'єкти: наявність вбудованих компонентів (спрайтів, кнопок, текстових полів і фігур), що мінімізує потребу писати базовий код з нуля.

Блок застосувань

До складу блоку застосувань входять такі напрями використання фреймворку:

– Казуальні веб-ігри: завдяки своїй легкості та широкій сумісності з сучасними браузерами, Phaser є оптимальним вибором для веб-розробки.

– Освітні ігри: простота у використанні та високий рівень інтерактивності роблять фреймворк популярним інструментом для створення дитячих навчальних ігор.

– Мобільна розробка: вбудована підтримка сенсорного управління та акселерометра дозволяє створювати кросплатформні мобільні застосунки.

– Прототипування та MVP: фреймворк часто застосовують для швидкого проектування та створення мінімально життєздатних продуктів.

– Тематичні заходи та хакатони: завдяки швидкому циклу розробки Phaser активно використовується для подій зі швидкого прототипування ігор.

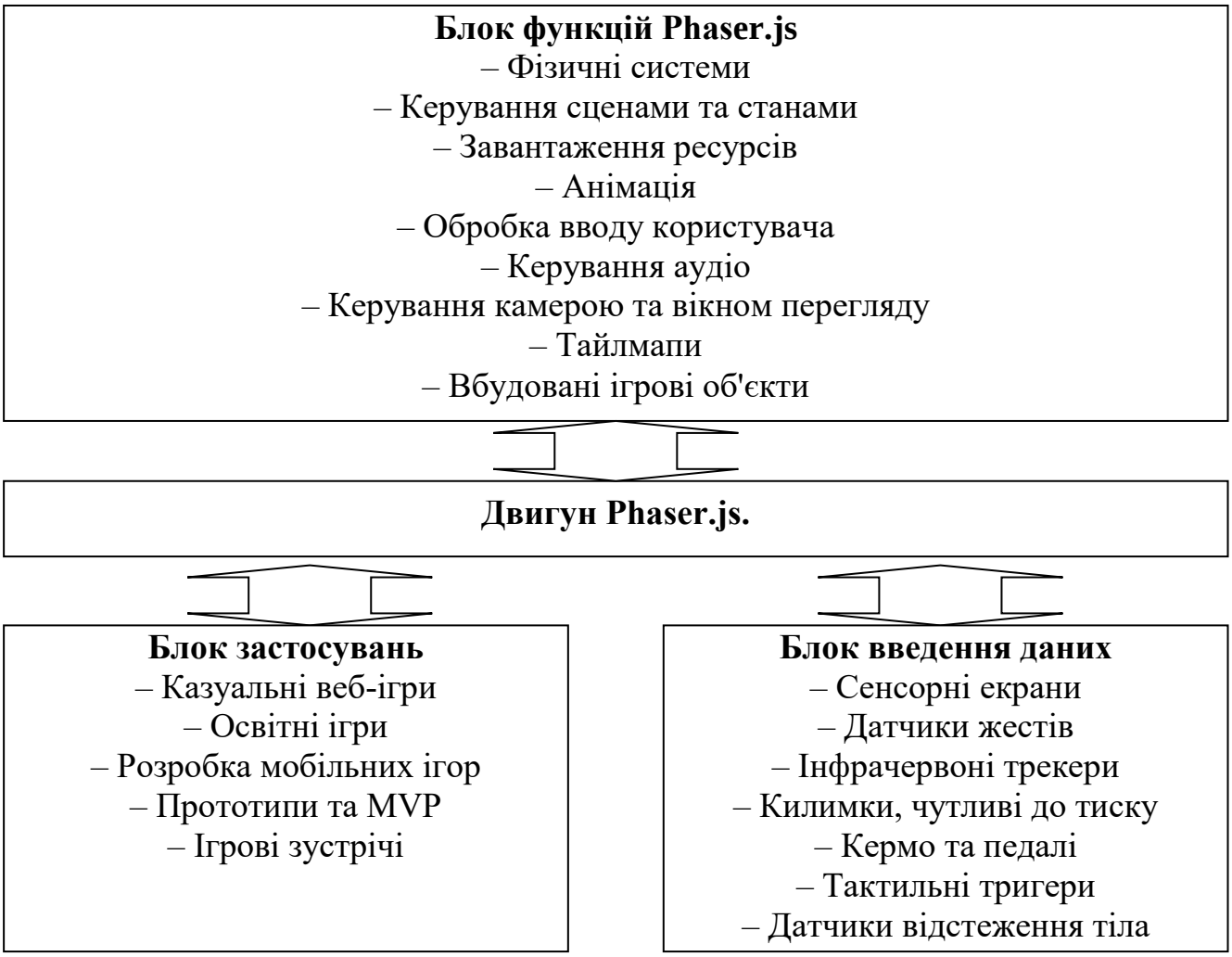

Рисунок 3.1 – Структурна схема системи

Блок введення даних

Сенсорні екрани, датчики та альтернативні інтерфейси введення**.** Сучасні аркадні комплекси виходять за межі стандартних механічних елементів керування (кнопок та джойстиків). Розробники впроваджують сенсорні дисплеї, сенсори руху та інтерактивні панелі для реалізації нових сценаріїв взаємодії. Такий підхід створює можливості для розробки ритм-ігор, фітнес-симуляторів та інтерактивних тренувальних комплексів. Сенсорні екрани широко застосовуються в ігрових автоматах квиткового типу та мобільних адаптаціях для аркадних кабінетів. Завдяки інтуїтивно зрозумілому інтерфейсу вони ідеально підходять для коротких розважальних сесій у громадських місцях. Водночас датчики жестів (наприклад, Leap Motion або інфрачервоні трекери)

збагачують функціонал інтерактивних музейних інсталяцій та спортивних рішень.

Окрім сенсорних екранів, ігрові комплекси можуть використовувати інші альтернативні пристрої введення даних:

1. Платформи (килимки), чутливі до тиску: знаходять застосування в танцювальних симуляторах та реабілітаційних комплексах.

2. Ігрові рулі та педалі: забезпечують високий ступінь реалізму в симуляторах перегонів.

3. Тактильні тригери (haptic triggers): реалізують зворотний зв’язок, підвищуючи занурення в екшн-іграх та шутерах.

4. Системи відстеження положення тіла (Body Tracking): уможливлюють управління за допомогою рухів всього тіла без використання додаткових гарнітур чи датчиків.

Застосування різноманітних інтерфейсів введення дозволяє розробникам охоплювати ширшу аудиторію, адаптувати системи під принципи доступності (accessibility) та ефективно інтегрувати ігри в різноманітні фізичні простори.

3.3 Розробка функціональної схеми

Функціональну схему розробленої системи наведено на рисунку 3.2. Аналіз архітектури програмного забезпечення вказує на те, що комплекс побудований на базі сучасного фреймворку Phaser і включає такі ключові підсистеми:

– 2D-рендеринг: підтримка спрайтів, тайлмапів та текстур для формування високоякісної та деталізованої графіки.

– Фізичні системи: інтеграція вбудованих фізичних рушіїв (Arcade Physics та Matter.js) для реалістичного моделювання взаємодії ігрових об'єктів.

– Інструменти анімації: засоби створення динамічних переходів на основі спрайт-листів і кадрові послідовності.

– Підтримка аудіо та обробка введення: робота з мультимедійними API

для відтворення звукових ефектів та обробки сигналів із різних пристроїв керування (клавіатура, миша, сенсорні екрани, геймпади).

– Керування активами: оптимізоване завантаження ігрових ресурсів, зображень, звуків та структурованих файлів конфігурації (JSON).

– Інструменти зворотного зв'язку та стратегія ранньої взаємодії.



Рисунок 3.2 – Функціональна схема системи

Для забезпечення ефективної взаємодії користувача з системою реалізовано комплекс механізмів зворотного зв'язку:

– Візуальні та звукові підказки: миттєва поява числових значень рахунку, кольорові ефекти та супровідні звукові реакції на дії гравця.

– Миттєва реакція на вхідні дані: мінімізація затримок (input lag) для досягнення максимальної синхронізації між діями користувача та реакцією ігрового середовища.

– Концепція «однохвилинного гачка» (one-minute hook): стратегія ранньої взаємодії, що замінює складні навчальні етапи практичним зануренням в ігровий цикл протягом першої хвилини. Завдяки простому керуванню, швидким циклам перемог/поразок, видимим цілям (таймери, лічильники) та можливості миттєвого перезапуску підвищується показник утримання користувачів (retention).

Системи підрахунку очок та хмарна інтеграція

Механізми оцінювання та рейтингові системи перетворюють короткі ігрові сесії на повторюваний процес. Структура мотиваційних систем включає миттєве відображення результатів, послідовні правила нарахування балів, багаторівневі пороги винагород та динамічне масштабування складності на пізніх етапах.

Упровадження хмарних таблиць лідерів та інтеграція із зовнішніми серверами (Photon, PlayFab, Firebase) забезпечують оновлення рейтингів у реальному часі, синхронізацію профілів між пристроями, збереження статистики та підтримку багатокористувацьких режимів. Це дозволяє органічно поєднати класичну аркадну механіку з можливостями сучасних хмарних і розподілених платформ.

3.4 Розробка діаграми процесів

Процесну діаграму розроблюваної системи наведено на рисунку 3.3. Її детальний аналіз дозволяє простежити характер взаємодії внутрішніх компонентів та маршрутизацію інформаційних потоків у межах

мультимедійного аркадного комплексу. У процесі проєктування застосовується модель графічного представлення потоків даних (Data Flow Diagram, DFD), яка наочно відображає рух, трансформацію та збереження інформації (зокрема, телеметричних даних гравця, ігрових станів, метрик складності та результатів роботи моделі машинного навчання).

Діаграми взаємодії процесів є фундаментальним інструментом структурного проєктування для візуалізації обробки даних. Традиційно розробка розпочинається з побудови контекстної діаграми потоків даних, яка ілюструє загальну взаємодію розроблюваної системи адаптивного регулювання зі зовнішнім середовищем (користувачем, клієнтським браузером, хмарними серверами та модулями прийняття рішень). У подальшому ця модель деталізується шляхом поетапної декомпозиції процесів і потоків даних з метою повної та всебічної демонстрації архітектури комплексу, що створюється.

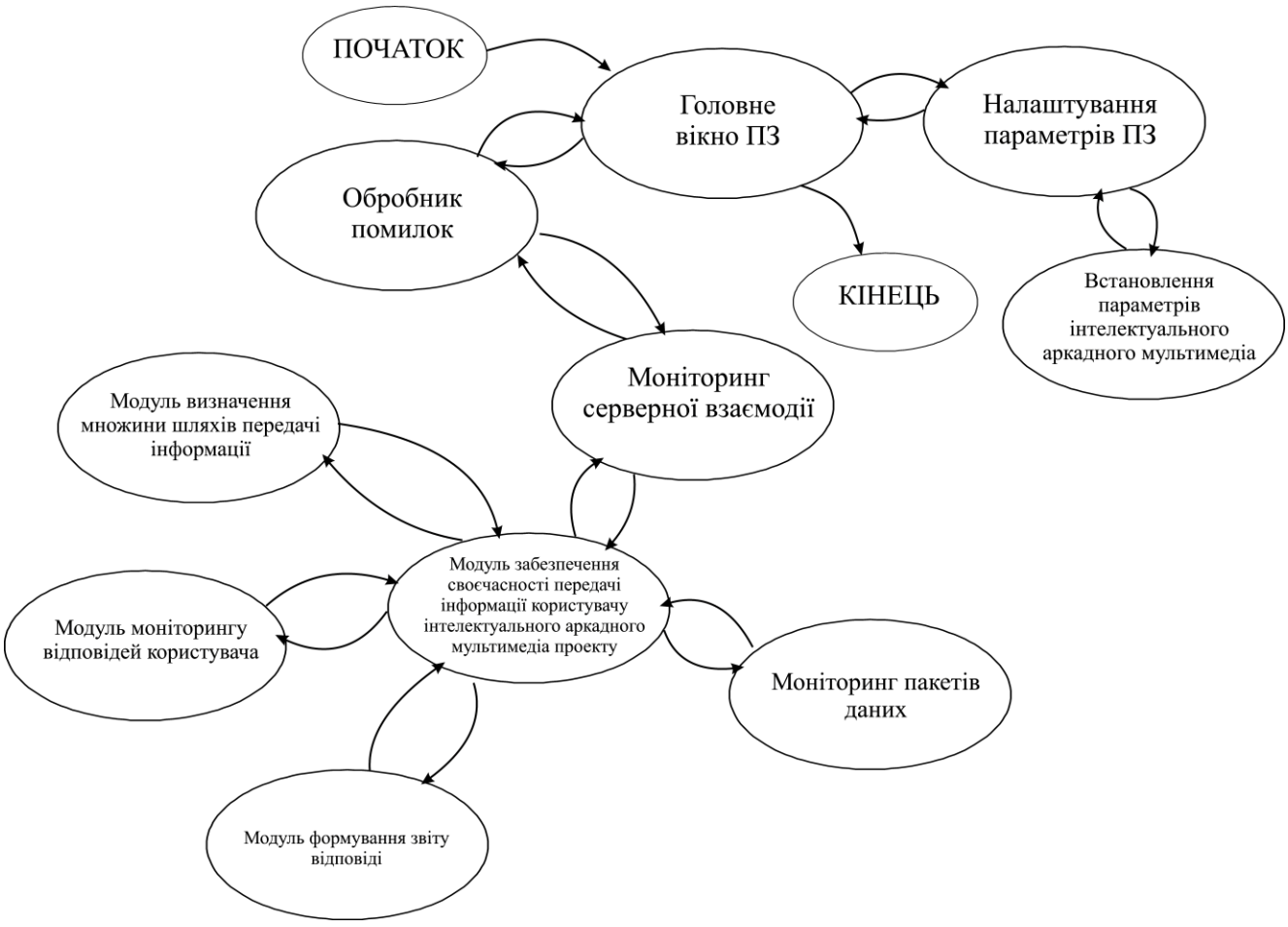

Рисунок 3.3 – Діаграма взаємодії процесів

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

4.1 Блок-схеми та опис алгоритмів функціонування системи

Алгоритмічне забезпечення системи адаптивного регулювання складності мультимедійного аркадного комплексу призначене для автоматизованого визначення рівня підготовленості гравця, аналізу його поточної результативності та динамічної зміни параметрів ігрового процесу. Основним завданням є підтримання такого рівня складності, який відповідає можливостям користувача та забезпечує поступове ускладнення ігрових завдань без надмірного збільшення кількості невдалих спроб. Для цього передбачається використання алгоритмів машинного навчання, які встановлюють залежності між характеристиками поведінки гравця, параметрами ігрового середовища та результатами проходження окремих етапів. На основі отриманих прогнозів система визначає необхідність зміни інтенсивності перешкод, швидкості руху об'єктів, часових обмежень та інших характеристик складності.

Загальну послідовність функціонування програмного комплексу відображено на рисунку 4.1. Основний алгоритм починається з ініціалізації ігрового середовища та створення користувацького сеансу. Під час проходження рівня система реєструє результати дій гравця, визначає показники його результативності та формує поведінковий профіль. Після накопичення необхідної кількості спостережень виконується аналіз поточного стану за допомогою моделі машинного навчання. На основі прогнозованої успішності проходження наступного етапу визначається доцільний рівень складності. Обрана конфігурація передається ігровому модулю та застосовується в дозволений момент. Подальші результати використовуються для перевірки правильності прийнятого рішення.

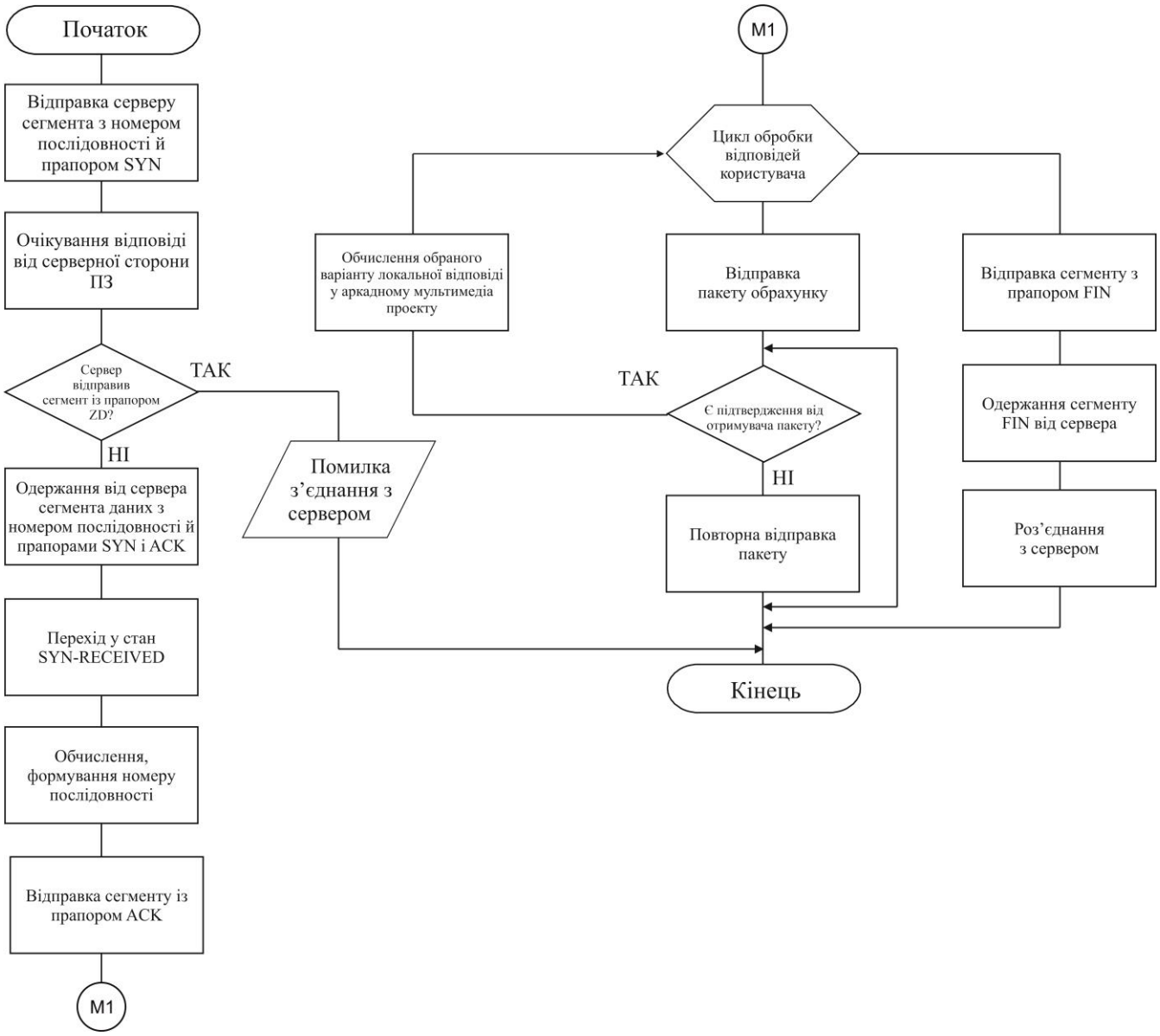

Рисунок 4.1 – Блок-схема основної програми

Для організації роботи системи доцільно передбачити архітектуру з окремими компонентами керування ігровим процесом, реєстрації подій, формування поведінкових характеристик і машинного навчання. Ігровий модуль відповідає за відображення графічних об'єктів, оброблення керувальних дій та виконання правил аркадної гри. Компонент телеметрії накопичує інформацію про результати проходження. Аналітична підсистема формує вектори характеристик і визначає поточний стан гравця. Модуль адаптації використовує прогноз моделі для вибору наступної конфігурації складності. Окремий компонент зберігання

забезпечує доступ до параметрів моделей, профілів користувачів та історії проходження ігрових етапів.

Особливістю мультимедійного аркадного комплексу є взаємодія декількох складових ігрового середовища. Складність може залежати від кількості об'єктів на екрані, швидкості їхнього переміщення, тривалості доступного часу, частоти появи перешкод і правил нарахування результатів. Додатково використовуються графічні ефекти, звукові повідомлення та інші мультимедійні засоби зворотного зв'язку. При цьому параметри візуального й звукового оформлення необхідно відокремлювати від безпосередніх характеристик складності. Наприклад, зміна швидкості появи цілей може впливати на вимоги до реакції гравця, тоді як зміна музичного супроводу не обов'язково повинна змінювати умови виконання завдання.

Алгоритм роботи підпрограми інтелектуального вибору рівня складності представлено на рисунку 4.2. Підпрограма отримує поведінковий вектор гравця, поточну конфігурацію ігрового середовища та перелік допустимих варіантів наступного етапу. Для кожного кандидата модель машинного навчання визначає прогнозовану ймовірність успішного проходження. Після цього результати порівнюються з установленим цільовим діапазоном. Якщо прогноз свідчить про надмірну простоту завдання, може бути обрано складнішу конфігурацію. Якщо очікувана результативність недостатня, перевага надається доступнішому варіанту. Остаточний вибір обмежується правилами поступової адаптації, які запобігають різким змінам ігрового процесу.

Під час реалізації адаптивного регулювання необхідно розрізняти фактичну результативність і рівень ігрових навичок. Висока кількість набраних балів не завжди свідчить про достатню підготовленість. Вона може бути наслідком проходження простого рівня або використання допоміжних механізмів. Аналогічно невдала спроба не обов'язково означає низький рівень навичок. На результат можуть впливати випадкові події, тимчасове відволікання та особливості керування. Тому оцінювання необхідно виконувати за сукупністю

показників, які враховують умови проходження та історію попередньої активності.

Алгоритм ініціалізації ігрового середовища

Початковим етапом функціонування системи є підготовка мультимедійного комплексу до нового сеансу. Вхідними даними алгоритму є вибраний ігровий режим, доступні параметри керування та відомості про користувацький профіль, якщо він існує.

На першому етапі перевіряється доступність необхідних програмних ресурсів. До них можуть належати графічні об'єкти, звукові файли, конфігурації рівнів і параметри ігрових механік. Якщо необхідний ресурс недоступний, система повинна визначити можливість використання резервної конфігурації або завершити ініціалізацію з відповідним повідомленням.

Далі створюється ідентифікатор ігрового сеансу. Для нього визначаються початковий рівень складності, обраний режим та поточний стан проходження. Якщо користувач має збережений профіль, його характеристики можуть використовуватися для вибору стартової конфігурації.

Для нового користувача доцільно встановлювати початковий рівень, який дозволяє виконати базові дії та отримати перші спостереження. Завдання не повинно вимагати попереднього знання складних ігрових механік, якщо вони ще не були представлені користувачу.

Після підготовки конфігурації активується ігровий цикл. Він забезпечує оновлення стану об'єктів, оброблення дій користувача, визначення зіткнень і відображення мультимедійних елементів.

Результатом алгоритму є підготовлений ігровий сеанс із визначеними початковими параметрами.

Алгоритм керування станами ігрового процесу

Для узгодження ігрової логіки з механізмами машинного навчання доцільно використовувати модель станів. Вона дозволяє визначати моменти, у які дозволяється оновлювати параметри складності.

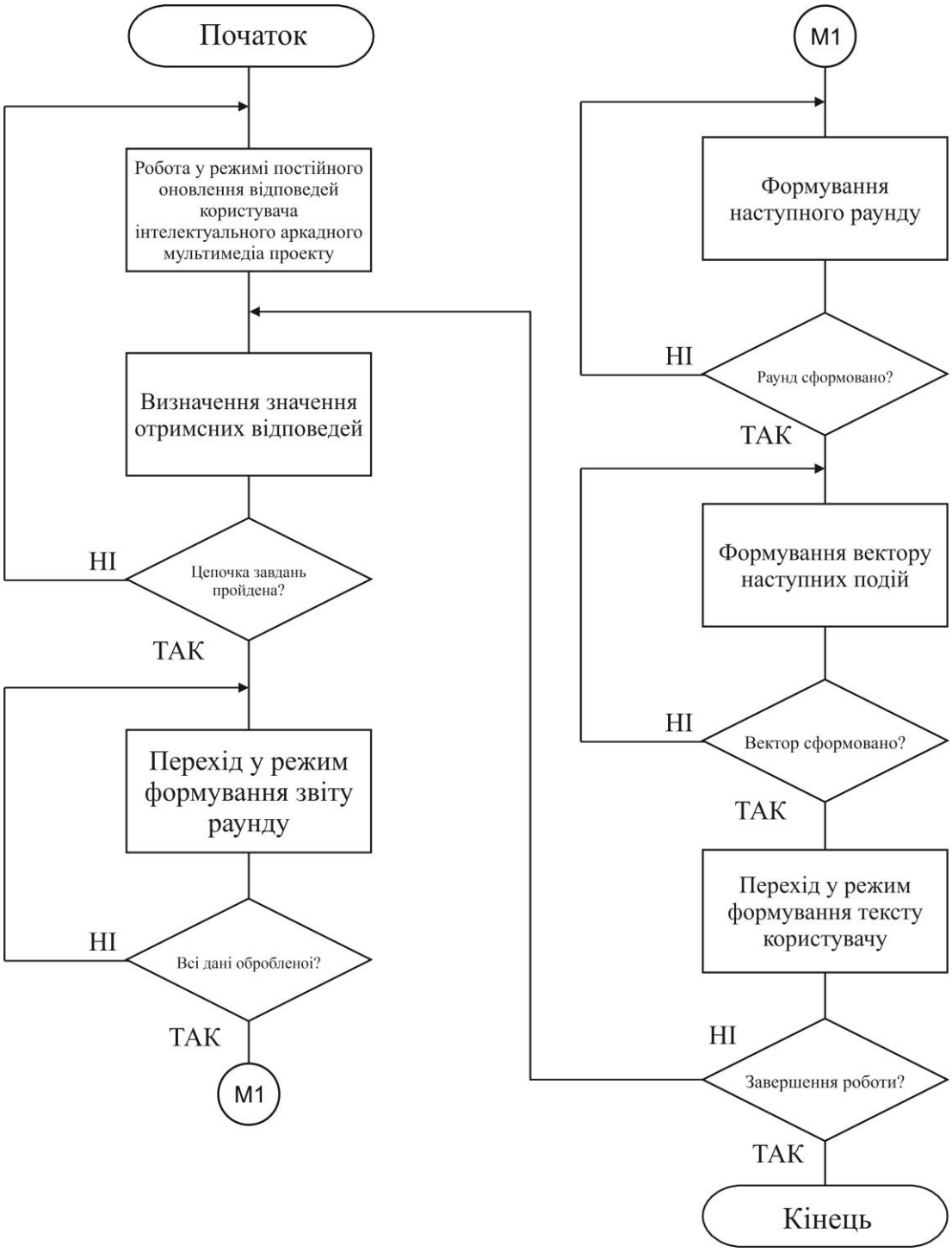

Рисунок 4.2 – Блок-схема роботи підпрограми

Початковим станом є підготовка сеансу. Після завершення ініціалізації виконується перехід до активного проходження. У цьому стані ігровий модуль обробляє керувальні дії та змінює положення об'єктів відповідно до поточної конфігурації.

Після завершення етапу система переходить до стану оцінювання. У ньому обчислюються підсумкові характеристики результативності та формується запит до модуля адаптації. Після отримання рішення виконується підготовка наступного етапу.

Окремо доцільно передбачити стани паузи, завершення сеансу та відновлення після технічної помилки. Перебування в стані паузи не повинно збільшувати показник часу реакції або враховуватися як активне проходження.

Зміна складності під час незавершеної ігрової дії може призвести до некоректної оцінки результату. Наприклад, раптове прискорення об'єктів безпосередньо перед завершенням завдання порушує однаковість умов проходження. Тому основним моментом застосування адаптації доцільно обрати перехід між рівнями, хвилями або іншими завершеними етапами.

Результатом алгоритму є керований життєвий цикл сеансу, у якому ігрова логіка та аналітичні операції виконуються без взаємних порушень.

Алгоритм реєстрації користувацької активності

Для побудови поведінкової моделі необхідно отримувати інформацію про дії гравця в процесі виконання завдань. Такими діями можуть бути натискання клавіш, переміщення керувального об'єкта, вибір цілі, виконання успішної дії або допущення помилки.

Вхідними даними алгоритму є внутрішні повідомлення ігрового модуля. На першому етапі визначається тип події. Далі до неї додаються ідентифікатор сеансу, номер етапу, час виникнення та параметри поточної складності.

Для реєстрації реакції може використовуватися інтервал між появою відповідного стимулу та правильною дією користувача. Однак такий показник необхідно застосовувати лише для завдань, у яких визначено момент появи стимулу та очікувану реакцію.

Окремо реєструються результати виконання. До них можуть належати кількість успішних дій, кількість помилок, тривалість проходження, кількість використаних спроб та підсумковий ігровий результат.

Важливо не зберігати необмежений потік усіх подій без практичної необхідності. Для подальшого аналізу часто достатньо агрегованих характеристик завершеного етапу. Детальні записи доцільно використовувати переважно під час тестування алгоритмів або діагностики.

Результатом є структурований набір повідомлень про активність гравця з відомостями про умови її виникнення.

Алгоритм попереднього оброблення ігрової телеметрії

Отримані події можуть містити повтори, пропущені значення та нерівномірні часові інтервали. Перед застосуванням машинного навчання необхідно виконати їхню перевірку.

На початковому етапі визначається належність кожного повідомлення до відповідного ігрового сеансу. Події, які не мають коректного ідентифікатора або виникли після завершення сеансу, обробляються окремо.

Далі виконується перевірка часових характеристик. Якщо час реакції має від'ємне значення або перевищує допустимі межі через технічну помилку, він не використовується як звичайне спостереження.

Наступним етапом є групування подій за завершеними етапами. Для кожного етапу формуються агреговані показники. Це дозволяє зменшити обсяг інформації та забезпечити однакове представлення результатів різних проходжень.

Окремо контролюються події, що виникли під час паузи або втрати фокуса ігрового вікна. Такі інтервали можуть виключатися з обчислення активного часу. Інакше модель може помилково інтерпретувати тимчасову відсутність дій як погіршення навичок.

Результатом алгоритму є очищена ігрова телеметрія, придатна для формування характеристик поведінки.

Алгоритм формування вектора результативності гравця

Для використання алгоритмів машинного навчання результати проходження необхідно перетворити на числовий вектор. Він повинен відображати не лише підсумкові бали, а й характер виконання завдань.

Вхідними даними є агреговані показники завершених етапів та інформація про їхню складність. На першому етапі визначається частка успішних дій. Для цього враховується кількість правильно виконаних операцій відносно загальної кількості спроб.

Далі аналізується частота помилок. Для аркадних режимів, які потребують швидкого реагування, може використовуватися кількість невдалих дій за одиницю активного часу. Для інших режимів доцільніше враховувати помилки відносно кількості завдань.

Окремо визначається швидкість виконання. Наприклад, час реакції може характеризувати здатність користувача своєчасно взаємодіяти з рухомими об'єктами. Для зменшення впливу випадкових затримок доцільно використовувати медіану або інші стійкі статистичні показники.

Важливим є врахування поточної складності. Однакова частка успішних дій на різних рівнях не обов'язково означає однакову підготовленість. Тому до вектора додаються параметри середовища, за яких отримано відповідний результат.

Після формування числових характеристик виконується їхнє масштабування. Для цього можуть застосовуватися стандартизація або нормалізація до визначених діапазонів. Параметри перетворення необхідно визначати на навчальній вибірці та зберігати разом із моделлю.

Результатом алгоритму є узгоджений поведінковий вектор, який використовується для оцінювання навичок і прогнозування проходження.

Алгоритм формування індивідуального профілю гравця

Для персоналізації складності необхідно накопичувати інформацію про характерні результати конкретного користувача. Такий профіль дозволяє враховувати не лише останній етап, а й попередню динаміку проходження.

Вхідними даними є поведінкові вектори завершених етапів. На першому етапі перевіряється достатність спостережень. Якщо користувач лише розпочав гру, профіль може бути недостатньо інформативним.

Для накопичених даних обчислюються узагальнені характеристики. До них можуть належати середня успішність, типова частота помилок, характерний час реакції та показники стабільності результатів.

Для врахування змін навичок може використовуватися експоненційне згладжування. Нові результати впливають на поточний профіль із визначеним коефіцієнтом. Це дозволяє поступово враховувати навчання користувача.

Водночас надмірна залежність від останнього результату є небажаною. Одна невдала спроба не повинна різко змінювати оцінку підготовленості. Тому доцільно використовувати декілька попередніх спостережень і контролювати розкид показників.

Результатом є динамічний профіль результативності, який зберігається в межах сеансу або між сеансами за наявності відповідного користувацького механізму.

Алгоритм початкового оцінювання навичок

Для нового гравця модель не має достатньої історії поведінки. У такому випадку необхідно використовувати процедуру початкового оцінювання, яка не потребує тривалого проходження.

Вхідними даними є результати перших ігрових етапів. На початку встановлюється базова складність із передбачуваними умовами виконання завдань.

Після завершення першого етапу система визначає основні характеристики користувача. Якщо він демонструє стабільну результативність, наступний етап може отримати дещо вищу складність. Якщо кількість помилок значна, поточні параметри зберігаються або коригуються в бік спрощення.

Для початкового оцінювання доцільно обмежувати максимальний розмір зміни. Це запобігає ситуації, коли випадково успішний результат призводить до різкого ускладнення гри.

Після накопичення достатньої кількості даних система може перейти від початкової евристичної процедури до прогнозування на основі машинного навчання.

Результатом алгоритму є початковий рівень підготовленості та набір спостережень для подальшої персоналізації.

Алгоритм групування гравців за поведінковими характеристиками

Для підвищення якості початкового налаштування може застосовуватися кластеризація історичних поведінкових профілів. Вона дозволяє визначати групи користувачів із подібними характеристиками проходження.

Вхідними даними є нормалізовані вектори результативності, сформовані за порівнянних ігрових умов. На першому етапі визначається набір ознак, які характеризують навички та стиль проходження.

Для групування може використовуватися K-means. Алгоритм формує визначену кількість кластерів у просторі поведінкових характеристик. Кожен кластер описується відповідним центром.

У результаті можуть бути сформовані групи, які відрізняються швидкістю реакції, точністю виконання дій або стабільністю результатів. Однак такі групи не слід без додаткової перевірки ототожнювати з фіксованими категоріями майстерності.

Для нового користувача визначається найближча група за доступними початковими спостереженнями. Її характеристики можуть використовуватися для формування попереднього прогнозу.

Зі збільшенням кількості індивідуальних результатів вплив групового профілю може поступово зменшуватися. Таким чином система переходить від узагальненої моделі до персоналізованого оцінювання.

Результатом алгоритму є груповий контекст, який допомагає формувати стартові параметри складності за недостатньої користувацької історії.

Алгоритм навчання моделі прогнозування успішності

Центральним елементом інтелектуального регулювання є модель, яка прогнозує результат проходження наступного ігрового етапу залежно від характеристик гравця та складності завдання.

Вхідними даними алгоритму є історія завершених етапів. Кожен навчальний запис повинен містити характеристики гравця, доступні до початку етапу, параметри його складності та фактичний результат.

Як цільова ознака може використовуватися успішне завершення етапу. У такому випадку завдання машинного навчання формулюється як бінарна класифікація з оцінюванням імовірності успіху.

Для початкової реалізації можуть використовуватися Random Forest, Gradient Boosting або інші алгоритми класифікації. Вони дозволяють аналізувати нелінійні залежності між поведінковими характеристиками та параметрами складності.

Важливо, щоб навчальний набір містив приклади проходження за різних рівнів складності. Якщо користувачі з високою підготовленістю завжди отримували лише складні завдання, а нові гравці лише прості, модель може засвоїти залежність між рівнем складності та характеристиками користувача без достатніх даних для оцінювання альтернативних умов.

Для зменшення такого зміщення доцільно використовувати контрольовані експериментальні проходження з обмеженим варіюванням складності. Отримані результати формують навчальні спостереження для оцінювання впливу різних конфігурацій.

Під час формування вибірок необхідно уникати витоку інформації. До вхідного вектора не можна включати підсумкові характеристики того етапу, успішність якого прогнозується. Для оцінювання якості на нових користувачах доцільно розділяти навчальні та тестові дані за гравцями.

Після навчання модель перевіряється на окремій вибірці. Крім точності класифікації, необхідно оцінювати відповідність прогнозованих імовірностей фактичній частоті успіху. Для цього може застосовуватися калібрування ймовірностей.

Результатом алгоритму є модель прогнозування, яка отримує поведінкові характеристики гравця та складність кандидата і повертає оцінку ймовірності успішного проходження.

Алгоритм інтелектуального вибору рівня складності

Після отримання результатів машинного навчання необхідно визначити конкретну конфігурацію наступного етапу. Для цього формується перелік допустимих кандидатів.

Вхідними даними алгоритму є індивідуальний профіль, прогнозна модель, поточний рівень та встановлені обмеження адаптації.

На першому етапі визначається допустимий діапазон складності. Він залежить від ігрового режиму та попередньої конфігурації. Для запобігання різким змінам максимальний перехід між сусідніми етапами доцільно обмежувати.

Після формування кандидатів модель прогнозує ймовірність успішного проходження кожного варіанта. Для цього використовується один і той самий поведінковий профіль із різними значеннями параметрів складності.

Наступним етапом є порівняння результатів із цільовою зоною успішності. Її значення визначається завданнями ігрового дизайну та уточнюється експериментально.

Якщо прогнозована ймовірність успіху істотно перевищує бажану, система може обрати складніший варіант. Якщо вона нижча за допустиму межу, доцільно спростити завдання.

За наявності декількох прийнятних кандидатів перевага може надаватися конфігурації, яка найменше відрізняється від поточної. Це забезпечує більш передбачувану адаптацію.

Якщо прогноз моделі недостатньо достовірний, система повинна зберігати попередню конфігурацію або використовувати заздалегідь визначене евристичне правило.

Результатом алгоритму є рішення про рівень складності наступного етапу з поясненням підстав його вибору.

Алгоритм обмеження швидкості зміни складності

Безпосереднє застосування прогнозованого рівня не завжди забезпечує комфортне проходження. Якщо після одного успішного етапу складність різко зростає, користувач може отримати непропорційно складне завдання.

Для запобігання цьому передбачається алгоритм згладжування керувальних рішень.

Вхідними даними є рекомендована складність, поточна конфігурація та історія попередніх змін. На першому етапі визначається різниця між поточним і рекомендованим значеннями.

Якщо різниця перевищує встановлене обмеження, зміна скорочується до допустимого кроку. Таким чином навіть значне відхилення прогнозованої успішності не призводить до неконтрольованого переходу.

Окремо використовується механізм гістерезису. Він дозволяє уникати багаторазового переключення між двома сусідніми рівнями за незначних коливань результативності.

Для цього можуть застосовуватися різні пороги підвищення та зниження складності. Додатково визначається мінімальна кількість етапів між послідовними змінами.

За повторюваних невдач може передбачатися швидше зменшення складності, ніж її підвищення. Однак така асиметрія повинна перевірятися з урахуванням особливостей ігрового режиму.

Результатом алгоритму є згладжена конфігурація, яка відповідає обмеженням поступової адаптації.

Алгоритм перетворення рівня складності на ігрові параметри

Рекомендований рівень складності є узагальненим показником. Для його практичного використання необхідно визначити конкретні параметри ігрового середовища.

Вхідними даними є вибраний рівень складності та характеристики активного аркадного режиму.

Для гри, орієнтованої на швидкість реакції, можуть налаштовуватися швидкість руху цілей, тривалість їхньої доступності та часовий інтервал між появою об'єктів.

Для режиму ухилення від перешкод доцільно змінювати щільність перешкод, швидкість переміщення або характеристики безпечних зон. При цьому необхідно зберігати можливість виконання завдання за передбаченими правилами.

Для аркадного режиму з нарахуванням очок можуть використовуватися параметри частоти появи цілей, кількості одночасних об'єктів і допустимої кількості помилок.

На початковому етапі визначається відповідна конфігурація. Після цього виконується перевірка її допустимості. Наприклад, швидкість руху об'єктів не повинна перевищувати значення, за якого завдання стає технічно невиконуваним у межах установлених правил.

Окремо необхідно контролювати взаємний вплив параметрів. Одночасне збільшення кількості об'єктів і швидкості їхнього переміщення може створювати значно більший ефект, ніж окреме налаштування кожного показника.

Результатом алгоритму є конкретний набір ігрових параметрів, який передається модулю виконання наступного етапу.

Алгоритм керування мультимедійним зворотним зв'язком

Мультимедійні засоби використовуються для відображення результатів дій і повідомлення про зміни ігрового середовища. До них можуть належати графічні індикатори, анімації, звукові ефекти та текстові підказки.

Вхідними даними алгоритму є стан ігрового сеансу, результати проходження та обрана конфігурація складності.

На першому етапі визначається тип повідомлення. Успішне виконання завдання може супроводжуватися відповідним графічним або звуковим сигналом. Невдала спроба може відображатися через окрему анімацію.

Якщо система змінює складність між етапами, доцільно забезпечити зрозуміле повідомлення про нові умови. Це дозволяє уникати ситуації, коли користувач сприймає зміну швидкості об'єктів як технічну помилку.

Окремо необхідно враховувати налаштування доступності. Звукова гучність, контрастність і можливість відключення окремих ефектів не повинні довільно використовуватися як засоби підвищення складності.

За потреби може застосовуватися система підказок. Її активація повинна враховувати правила конкретного режиму та фіксуватися в телеметрії. Це необхідно для правильного інтерпретування результативності проходження з допомогою.

Результатом алгоритму є узгоджене мультимедійне представлення ігрового процесу без порушення логіки адаптації.

Алгоритм застосування рішення до ігрового модуля

Після вибору конфігурації необхідно передати її компоненту, який безпосередньо керує виконанням гри.

Вхідними даними є рекомендований рівень, його конкретні параметри та поточний стан ігрового сеансу.

На першому етапі перевіряється можливість застосування зміни. Якщо гравець ще виконує поточне завдання, рішення зберігається до завершення етапу.

Після переходу до стану підготовки наступного рівня ігровий модуль отримує нову конфігурацію. Далі перевіряється допустимість усіх параметрів.

Для забезпечення узгодженості доцільно застосовувати набір параметрів як єдину конфігурацію. Якщо частина значень некоректна, система повинна відхилити зміну повністю або використати заздалегідь перевірений резервний варіант.

Після успішного застосування в історії сеансу фіксується фактична конфігурація. Це необхідно для подальшого навчання, оскільки результат проходження повинен зіставлятися саме з реально використаними параметрами, а не з попередньою рекомендацією. Результатом алгоритму є підготовлений наступний етап з узгодженими характеристиками складності.

Алгоритм оновлення поведінкової моделі

Під час експлуатації користувач може поступово вдосконалювати навички. Крім того, до комплексу можуть додаватися нові ігрові режими. Тому попередньо навчена модель не завжди зберігатиме однакову якість прогнозування.

Для врахування таких змін передбачається процедура періодичного оновлення моделі.

Вхідними даними є нові записи завершених етапів, попередня навчальна вибірка та результати перевірки поточної моделі. На першому етапі виконується контроль повноти нових даних.

Далі формуються ознаки з урахуванням фактичної складності проходження. Необхідно виключити записи, у яких результат був суттєво спотворений технічною помилкою, втратою фокуса або іншими неврахованими чинниками.

Після накопичення достатнього обсягу даних може виконуватися повторне навчання. Для моделей, які підтримують інкрементальне оновлення, допускається використання відповідного механізму. Для інших алгоритмів необхідне повторне навчання на підготовленому наборі.

Нова модель перевіряється на контрольних спостереженнях. Якщо якість прогнозування не відповідає встановленим вимогам, попередня версія залишається активною.

Результатом алгоритму є оновлена модель, яка може використовуватися для подальшого регулювання після підтвердження її придатності.

Алгоритм виявлення нестабільної результативності

Окремим завданням є визначення ситуацій, у яких короткочасні результати не відповідають звичайній поведінці користувача. Причиною може бути випадкова невдача, зміна умов керування або тимчасове зниження концентрації.

Вхідними даними алгоритму є послідовність результатів декількох завершених етапів.

На першому етапі аналізується розкид показників результативності. Далі визначається наявність різких змін точності, часу реакції та кількості помилок.

Якщо спостерігається значна нестабільність, доцільно зменшити швидкість адаптації. Система може зберігати поточну конфігурацію протягом додаткових етапів для накопичення більш достовірних даних.

Для розрізнення короткочасного відхилення та стійкого погіршення може використовуватися ковзне вікно спостережень. Після накопичення даних перевіряється, чи зберігається відповідна тенденція.

Результатом алгоритму є оцінка стабільності користувацької результативності, яка впливає на допустиму інтенсивність зміни складності.

Алгоритм оброблення помилок і відмов компонентів

Під час виконання ігрового комплексу можливі помилки отримання телеметрії, недоступність моделі машинного навчання, некоректні параметри адаптації та порушення збереження результатів.

Для запобігання впливу таких ситуацій на основний ігровий процес необхідно відокремити роботу модуля адаптації від механізмів відображення графіки та оброблення керувальних дій.

На першому етапі кожного запиту перевіряється коректність сформованого поведінкового вектора. Якщо частина необхідних характеристик відсутня, модель не повинна отримувати довільно підставлені значення без передбаченої процедури їхньої обробки.

Якщо прогнозування завершилося помилкою, система може зберегти попередню складність або використати базове евристичне правило. Відмова аналітичного компонента не повинна призводити до аварійного завершення ігрового сеансу.

Окремо контролюється час виконання прогнозу. Якщо відповідь не надходить у межах допустимого інтервалу, ігровий модуль повинен продовжити підготовку наступного етапу з безпечною конфігурацією.

Результатом алгоритму є кероване функціонування комплексу за умов часткової недоступності аналітичних компонентів.

Особливості програмної реалізації системи

Програмну реалізацію доцільно організувати таким чином, щоб ігрова логіка не залежала від конкретного алгоритму машинного навчання. Для цього між ігровим модулем та аналітичним компонентом використовується визначений програмний інтерфейс.

Ігровий модуль передає характеристики завершеного етапу та отримує результат вибору наступної складності. Для нього не повинні мати значення внутрішня структура класифікатора або спосіб навчання моделі.

Як один із можливих технологічних варіантів для аналітичного компонента може використовуватися Python. Для підготовки даних доцільно застосовувати NumPy та pandas, а для навчання моделей класифікації scikit-learn. За потреби можуть використовуватися нейромережеві засоби PyTorch.

Для безпосередньої реалізації мультимедійного середовища може використовуватися ігровий рушій із підтримкою двовимірної графіки, звукових ефектів та оброблення подій. Конкретний рушій і мова програмування визначаються фактичною реалізацією комплексу.

Важливо забезпечити незалежність частоти оновлення графічного середовища від швидкості виконання аналітичної моделі. Прогнозування доцільно виконувати асинхронно або в окремому компоненті. Це дозволяє уникати блокування ігрового циклу під час оброблення даних.

Для зберігання результатів проходження може застосовуватися локальна база даних або відповідне серверне сховище. Необхідно забезпечити зв'язок між користувацьким сеансом, параметрами складності та отриманим результатом.

Забезпечення конфіденційності та коректності користувацьких даних

Система адаптивного регулювання накопичує відомості про дії користувачів і характер їхнього проходження. Для виконання основних алгоритмів зазвичай достатньо технічних характеристик ігрової активності без збирання додаткової особистої інформації. На початковому етапі визначається перелік необхідних показників. До нього можуть належати частота помилок, тривалість проходження та результати виконання завдань.

Для збереження історії може використовуватися псевдонімізований ідентифікатор користувача. Якщо персоналізація між сеансами не потрібна, профіль може існувати лише протягом одного запуску гри.

Окремо необхідно контролювати доступ до збережених даних і параметрів моделей. Некоректна зміна навчальної вибірки або підміна моделі може призводити до неправильного вибору складності.

Для забезпечення відтворюваності результатів доцільно зберігати версію моделі та конфігурацію адаптації, які використовувалися під час конкретного ігрового сеансу.

Результатом відповідних механізмів є контрольована обробка користувацької телеметрії з обмеженням зайвого накопичення інформації.

Перевірка коректності алгоритмів та оцінювання ефективності

Для дослідження роботи запропонованого алгоритмічного забезпечення необхідно передбачити функціональні та експериментальні випробування. Основним завданням є визначення правильності оброблення користувацької активності, якості прогнозування результативності та впливу адаптивного регулювання на проходження гри.

Початковим етапом є перевірка алгоритмів реєстрації подій. Для цього використовуються контрольовані сценарії, у яких відомі час появи стимулу, момент дії користувача та очікуваний результат. Необхідно перевірити правильність визначення часових характеристик і формування підсумкових показників.

Далі перевіряється побудова поведінкових векторів. Для контрольованих наборів подій обчислюються очікувані значення точності, частоти помилок і часу реакції. Отримані результати порівнюються з даними програмного компонента.

Наступним етапом є перевірка моделі машинного навчання. Для цього формується набір історичних проходжень, розділений на навчальну та тестову частини. Доцільно забезпечити, щоб тестові користувачі не були представлені в навчальному наборі під час оцінювання здатності моделі працювати з новими гравцями.

Для оцінювання прогнозованої ймовірності успіху можуть використовуватися Brier Score та Log Loss. Додатково перевіряється калібрування прогнозів, тобто відповідність передбачених імовірностей фактичній частоті успішного проходження.

Окремо необхідно оцінити алгоритм вибору складності. Для цього формуються контрольовані поведінкові профілі з різною результативністю. Система повинна коректно визначати допустимі конфігурації та дотримуватися обмежень на величину зміни.

Для перевірки стабільності можуть використовуватися послідовності чергування успішних і невдалих проходжень. Під час аналізу необхідно визначити, чи не призводять невеликі коливання результатів до постійного переключення між рівнями.

Експериментальне порівняння може передбачати три режими роботи комплексу. Перший використовує фіксовану складність, другий реалізує евристичне регулювання, а третій застосовує запропонований алгоритм машинного навчання. Порівняння необхідно виконувати за узгодженими правилами проходження.

Основними показниками можуть бути частка успішно завершених етапів, кількість повторних спроб, тривалість проходження та частота зміни складності. Для оцінювання стабільності адаптації доцільно враховувати величину й напрямок відповідних змін.

Додатково може досліджуватися суб'єктивне сприйняття складності користувачами. Для цього застосовуються опитування після проходження або інші дозволені способи отримання зворотного зв'язку. Необхідно враховувати, що висока успішність сама по собі не підтверджує зацікавленості чи задоволеності грою.

Для визначення впливу машинного навчання доцільно виконати окреме порівняння прогнозної моделі з простими правилами зміни складності. Такий підхід дозволяє встановити, чи забезпечує використання складнішого алгоритму практично значущу відмінність.

Під час навантажувальних випробувань оцінюються затримка обчислення прогнозу, використання оперативної пам'яті та вплив аналітичного компонента на частоту оновлення графічного середовища. Важливо переконатися, що операції машинного навчання не створюють помітних затримок під час керування грою.

Для перевірки стійкості до відмов можуть моделюватися недоступність моделі, некоректні поведінкові дані та порушення роботи сховища профілів. Очікуваним результатом є продовження ігрового сеансу з базовими параметрами та коректне відображення обмежень адаптації.

Кількісні висновки щодо точності прогнозування, стабільності регулювання та впливу адаптивної складності на результати проходження можуть бути сформовані лише після виконання відповідних експериментів.

Запропоноване алгоритмічне забезпечення формує послідовний процес адаптивного регулювання складності мультимедійного аркадного комплексу від реєстрації користувацької активності до прогнозування результативності та зміни параметрів ігрового середовища. Основне значення має використання моделі машинного навчання для оцінювання ймовірності успішного проходження за різних конфігурацій. Поєднання прогнозування з механізмами поступової адаптації, контролю стабільності та індивідуального профілювання створює основу для персоналізованого керування складністю. Подальша програмна реалізація та експериментальна перевірка повинні визначити придатність запропонованих алгоритмів для різних режимів мультимедійного аркадного комплексу.

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ЕКСПЛУАТАЦІЮ

Програмне забезпечення мультимедійного аркадного комплексу призначене для автоматизованого регулювання складності ігрових завдань відповідно до індивідуальних характеристик гравця. Система реєструє результати проходження, аналізує швидкість реакції, точність виконання дій та кількість помилок. На основі отриманих показників алгоритми машинного навчання прогнозують успішність наступного етапу й визначають доцільні параметри складності. Головне вікно програми показане на рисунку 5.1.

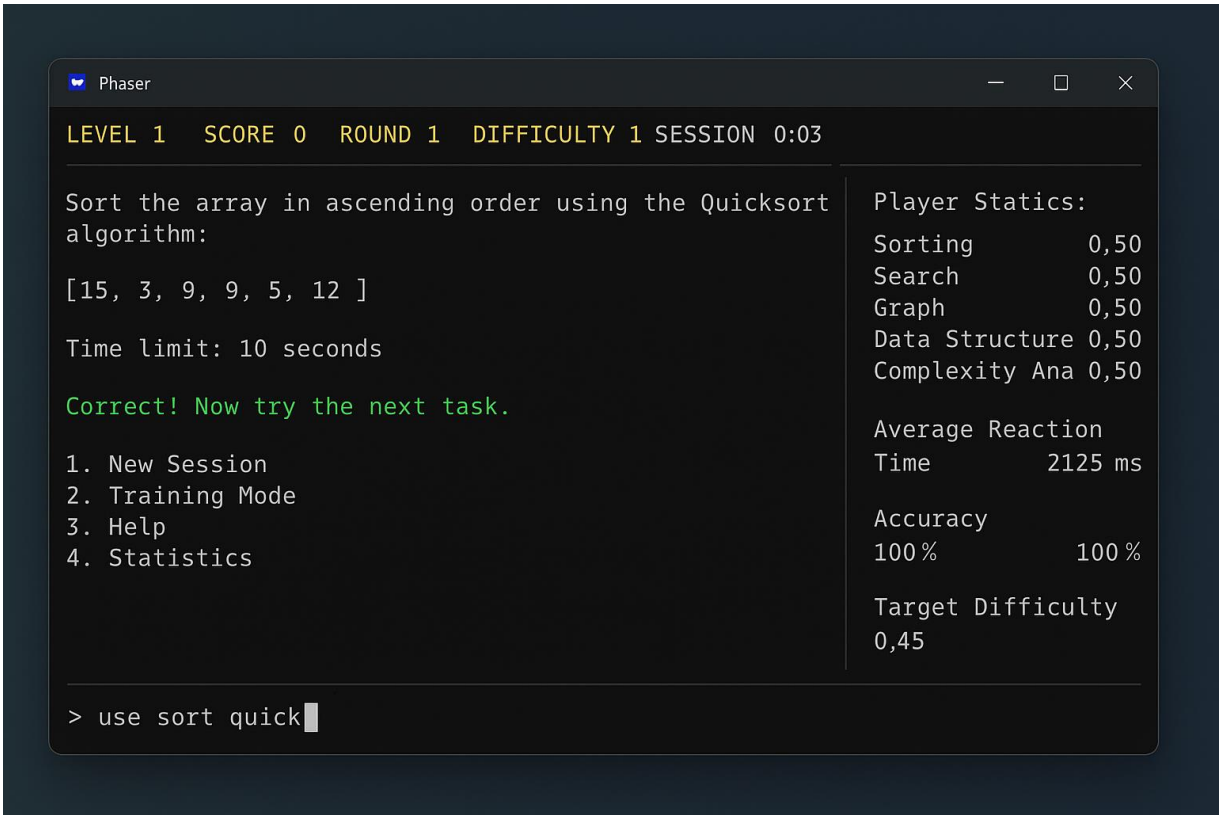

Рисунок 5.1 – Головне вікно ПЗ

Впровадження програмного комплексу передбачає підготовку середовища виконання, встановлення програмних залежностей, налаштування ігрових параметрів та підключення модуля машинного навчання. Основними функціональними складовими є мультимедійний ігровий модуль, компонент реєстрації користувацької активності, підсистема формування поведінкових характеристик і модуль адаптивного керування складністю. Окремо передбачається зберігання профілів гравців, параметрів навченої моделі та результатів ігрових сеансів.

Програмне середовище повинно забезпечувати відображення двовимірної графіки, відтворення звукових ефектів, оброблення керувальних дій і виконання алгоритмів машинного навчання. Конкретний ігровий рушій, операційна система та версії бібліотек визначаються фактичною програмною реалізацією.

Вимоги до апаратного забезпечення залежать від складності графічного середовища, кількості одночасно активних об'єктів і способу виконання прогнозної моделі. Комп'ютер повинен мати достатню продуктивність процесора, оперативну пам'ять та графічні засоби для стабільного відображення гри. За використання окремого аналітичного сервісу необхідно забезпечити його доступність. Водночас для компактних моделей класифікації можливе локальне виконання без виділеного сервера або графічного прискорювача.

Підготовка програмного середовища починається зі встановлення необхідних компонентів і перевірки їхньої сумісності. Після встановлення перевіряється можливість завантаження бібліотек, виконання числових операцій та отримання прогнозу від тестової моделі. Якщо ігровий модуль реалізований іншою мовою програмування, його взаємодія з аналітичним компонентом може здійснюватися через внутрішній програмний інтерфейс.

Наступним етапом є підготовка мультимедійних ресурсів. Необхідно перевірити доступність графічних зображень, анімацій, звукових файлів і конфігурацій ігрових рівнів. Для кожного режиму визначаються правила проходження, допустимі значення швидкості об'єктів, кількість перешкод і часові обмеження. Важливо забезпечити, щоб усі можливі конфігурації

залишалися технічно виконуваними та відповідали встановленим правилам гри.

Після підготовки ігрового середовища налаштовується реєстрація користувацької активності. Для кожного сеансу необхідно забезпечити формування внутрішнього ідентифікатора та збереження результатів завершених етапів. До відповідних характеристик належать кількість успішних дій, помилок, тривалість проходження й параметри фактичної складності. Інтервали паузи не повинні помилково враховуватися як активний час виконання завдання.

Для збереження результатів може використовуватися локальна база даних або серверне сховище, якщо комплекс передбачає роботу з декількома користувачами. Необхідно забезпечити зв'язок між сеансом, поведінковими характеристиками та застосованою конфігурацією складності. За потреби зберігаються попередні результати користувача для персоналізації наступних запусків. Доступ до цих відомостей повинен відповідати встановленим правилам оброблення користувацьких даних.

Окремої підготовки потребує модель машинного навчання. Для її створення використовуються результати завершених ігрових етапів, що містять поведінкові характеристики, параметри складності та фактичну успішність проходження. Як можливі алгоритми розглядаються Random Forest і Gradient Boosting. Навчання доцільно виконувати до розгортання основного програмного комплексу. Підготовлені параметри моделі зберігаються разом із відомостями про структуру та масштабування вхідних ознак.

Під час підключення навченої моделі необхідно перевірити відповідність її вхідних даних характеристикам, які формує ігровий модуль. Зокрема, однаковими повинні залишатися порядок ознак, одиниці вимірювання та правила їхнього перетворення. Невідповідність цих параметрів може спричиняти некоректне прогнозування навіть за успішного завантаження моделі. Перед використанням необхідно перевірити, що прогнозовані ймовірності мають допустимі значення.

Наступним етапом є налаштування правил адаптації. Для цього

визначаються мінімальний і максимальний рівні складності, допустима величина переходу між етапами та цільовий діапазон успішності. Окремо встановлюється мінімальна кількість спостережень для персоналізованого прогнозування. До накопичення необхідної історії доцільно використовувати початкову конфігурацію або обмежені евристичні правила. Конкретні значення параметрів повинні уточнюватися за результатами експериментальної перевірки.

Застосування нових параметрів складності необхідно узгодити зі станами ігрового процесу. Рекомендовану конфігурацію доцільно передавати після завершення рівня або хвилі. Це запобігає непередбачуваній зміні швидкості об'єктів чи кількості перешкод безпосередньо під час виконання завдання. Після прийняття рішення ігровий модуль перевіряє допустимість параметрів і використовує їх під час підготовки наступного етапу.

Рекомендована послідовність запуску починається з перевірки мультимедійних ресурсів і доступності сховища користувацьких результатів. Далі ініціалізується аналітичний компонент і завантажуються параметри навченої моделі. Після успішної перевірки прогнозування активується ігровий модуль. Під час створення сеансу визначається початкова складність, запускається реєстрація подій та забезпечується взаємодія між компонентами гри й адаптації.

Первинну перевірку працездатності доцільно виконувати на контрольованих ігрових сценаріях. Спочатку перевіряється запуск нового сеансу, відображення об'єктів, оброблення керувальних дій і правильність завершення етапу. Після цього контролюється формування поведінкових показників. Необхідно переконатися, що система правильно визначає кількість успішних дій, помилок і тривалість активного проходження.

Для перевірки адаптивного регулювання можуть використовуватися заздалегідь підготовлені профілі з різною результативністю. Необхідно перевірити правильність отримання прогнозу, вибору наступної складності та дотримання обмежень на її зміну. Окремо контролюються випадки, коли модель не повертає коректного результату. За таких умов комплекс повинен

продовжувати роботу з попередньою або резервною конфігурацією.

Для ідентифікації програмного продукту передбачається представлення відомостей про його розробника. Така інформація використовується під час документування мультимедійного аркадного комплексу та організації подальшого супроводження програмного забезпечення. Авторські дані системи адаптивного регулювання складності із застосуванням алгоритмів машинного навчання представлені на рисунку 5.2.

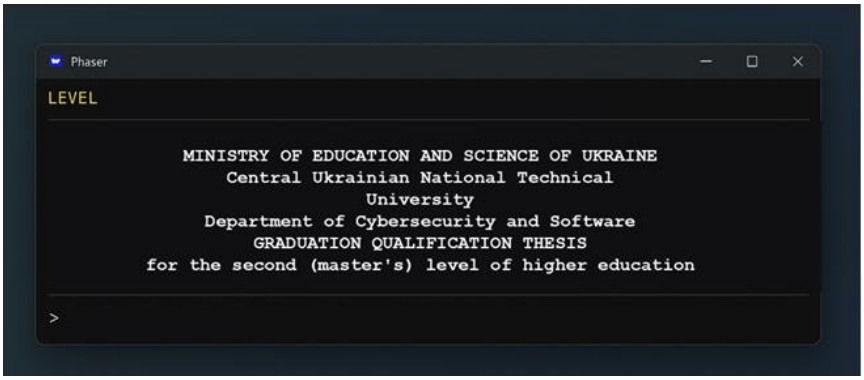

Рисунок 5.2 – Авторське право

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи адаптивного регулювання складності мультимедійного аркадного комплексу із застосуванням алгоритмів машинного навчання.

Метою розробки є дослідження та програмна реалізація системи адаптивного регулювання складності мультимедійного аркадного комплексу із застосуванням алгоритмів машинного навчання.

Об'єктом дослідження є процес адаптивного регулювання складності мультимедійного аркадного комплексу із застосуванням алгоритмів машинного навчання.

Предметом дослідження є методи адаптивного регулювання складності мультимедійного аркадного комплексу із застосуванням алгоритмів машинного навчання.

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 адаптивного регулювання складності мультимедійного аркадного комплексу із застосуванням алгоритмів машинного навчання.

– Розроблено вітчизняний продукт адаптивного регулювання складності мультимедійного аркадного комплексу із застосуванням алгоритмів машинного навчання, який має більш широкі можливості, на відміну від існуючих аналогів.

7 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

7.1 Вступ

Охорона праці є невід'ємною частиною управління будь-яким виробничим процесом, у тому числі й роботою програмістів. Рівень захисту здоров'я працівників безпосередньо впливає на ефективність та безпеку виконання ними своїх завдань. Основні принципи охорони праці передбачають створення таких умов, що дозволяють працівникам уникнути травмування та захворювань, а також забезпечення їхнього комфортного перебування на робочому місці.

У сфері охорони праці в ІТ-галузі діє низка нормативних актів, які регламентують безпечні умови праці. Ось деякі з основних нормативних актів: Закон України «Про охорону праці» (№ 2694-XII від 14.10.1992 р.) встановлює загальні положення щодо охорони праці в усіх галузях, зокрема й у сфері ІТ. Він регулює відносини між роботодавцем і працівниками у сфері безпеки праці, обов’язки щодо створення безпечних умов праці, гарантії прав працівників на охорону здоров’я та безпеку на робочому місці. Відповідно до цього закону, роботодавець зобов’язаний забезпечити належний рівень безпеки на робочому місці та створити умови, що сприятимуть збереженню здоров’я працівників.

Державні санітарні норми і правила (СанПіН): Визначають санітарні вимоги до умов праці, включаючи норми освітлення, рівні шуму, вентиляцію, температуру, які є важливими для офісів і робочих місць, де використовуються комп’ютери.

Державні будівельні норми України (ДБН): Містять норми щодо проектування та облаштування приміщень, де виконуються роботи з використанням комп’ютерної техніки. Наприклад, ДБН В.2.5-28:2006 – норми щодо природного та штучного освітлення, важливі для створення комфортного робочого середовища.

Накази та постанови Державної служби України з питань праці: Наприклад, Наказ № 15 "Про затвердження Правил охорони праці під час роботи з комп’ютерними системами". Цей документ встановлює вимоги до безпеки під час роботи з комп'ютерною технікою, включаючи ергономічні параметри робочого місця та організацію робочого часу.

ДСТУ ISO 45001:2019 – Система управління охороною здоров’я та безпекою праці: Стандарт, що визначає вимоги до систем управління охороною здоров'я і безпекою праці, які підприємства можуть застосовувати для запобігання нещасним випадкам та забезпечення здорових умов роботи.

Також стаття 43 Конституції України яка встановлює право людини на належні, безпечні і здорові умови праці.

До основних законодавчих актів про охорону праці належать і “Основи законодавства України про охорону здоров’я”, які регулюють суспільні відносини в цій сфері. Їхня мета – сприяти гармонійному розвитку фізичних і духовних сил, забезпечувати високу працездатність та активне довголіття громадян, усувати чинники, що негативно впливають на здоров’я, а також запобігати та знижувати рівень захворюваності, інвалідності та смертності, забезпечуючи спадковість покращення.

Кодекс законів про працю (КЗпП, № 322-VIII від 10.12.71), що регулює трудові відносини всіх працівників, встановлює високий рівень умов праці, всебічну охорону трудових прав працівників, (розділ XI “Охорона праці”)

Ці нормативні акти допомагають забезпечити безпечне середовище для працівників ІТ-галузі та знизити ризики, пов’язані з професійною діяльністю.

У цьому розділі розглянуто сучасні підходи до покращення умов праці

програмістів, розрахунки захисного занулення електроустановки та заходи протипожежної профілактики.

7.2 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста

Робота програміста характеризується значним навантаженням на зоровий апарат, опорно-руховий апарат, а також можливістю впливу на організм електромагнітного випромінювання. Дотримання санітарно-гігієнічних умов є критичним для забезпечення продуктивної роботи і запобігання професійним захворюванням.

Згідно з ДБН В.2.5-28:2018, рівень освітленості в офісі має бути не нижче 300-500 lx, що досягається за допомогою комбінації природного та штучного освітлення.

Рекомендується використовувати світлодіодні лампи з тепло-білим спектром світла (колірна температура 4000-5000K).

Мікроклімат робочого приміщення визначається температурою, вологістю та рухом повітря. Для комфортної роботи програмістів рекомендована температура становить 22-24°C, а вологість – 40-60%.

Робоче місце програміста має відповідати нормативним вимогам, що встановлені законодавством України та міжнародними стандартами. Згідно з ДСанПІН 3.3.2-007-98, основними вимогами до робочих місць з використанням комп'ютерної техніки є забезпечення належного рівня освітлення, оптимального мікроклімату та ергономічності робочого простору. Площа приміщення на одного працівника не повинна бути меншою за 6 м², тому для шести програмістів загальна площа приміщення площею 45 м² відповідає нормам (7.5 м² на людину).

Нормативи мікроклімату регулюються санітарними правилами ДСН 3.3.6.042-99. Приміщення площею 45 м², де працюють 6 програмістів, повинно бути обладнане системою кондиціонування повітря з функцією автоматичного підтримання параметрів температури та вологості.

Шум у робочому приміщенні не повинен перевищувати 50 дБ, що забезпечується звукоізоляцією стін та стелі. Ергономіка робочих місць має

особливе значення, оскільки від правильного налаштування висоти столу, положення монітора та крісла залежить збереження здоров'я хребта і запобігання синдромам зап'ястного каналу.

Забезпечення належних умов праці в IT-відділі потребує комплексного підходу, що включає організаційні заходи та використання технічних засобів. Ось основні аспекти такого забезпечення:

Розробка та впровадження політики охорони праці: Включає правила використання обладнання, порядок роботи з комп’ютерними системами, регулярні інструктажі з охорони праці.

Графік роботи та перерви: передбачається затвердження графіка, який включає регулярні перерви для зменшення напруги зору та втоми.

Проведення інструктажів та навчання: Регулярні інструктажі з техніки безпеки та правильного використання обладнання.

Технічні засоби.

Монітори: Перевагу слід надавати моніторам із діагоналлю 24–27 дюймів, роздільною здатністю не менше 1920x1080 пікселів (Full HD), а краще – 2560x1440 пікселів (2K) або вище, щоб зменшити навантаження на очі. Моделі повинні мати функцію захисту від синього світла та регулювання яскравості. Таким вимогам відповідає, наприклад, Dell UltraSharp U2719D

Системні блоки або ноутбуки: Обладнання з процесорами Intel Core i5 або i7, 16+ ГБ оперативної пам’яті, SSD на 512 ГБ або більше для швидкої роботи програм. Наприклад, Lenovo ThinkPad X1 Carbon Gen 9 – ноутбук із процессором Intel Core i7.

Клавіатури та миші: Ергономічні моделі з регульованими параметрами для зниження напруги м’язів рук. Наприклад Logitech MX Keys Advanced Wireless Illuminated Keyboard – ергономічна клавіатура з підсвіткою, оптимальною для довготривалого набору тексту.

Стільці: Офісні крісла з регульованою висотою, підтримкою поперекового відділу хребта, підлокітниками та можливістю нахилу спинки.

Робочі столи: Регульовані за висотою столи, що дозволяють працювати

як сидячи, так і стоячи, з достатньою площею для розміщення обладнання. Наприклад, IKEA BEKANT – стіл із регулюванням висоти, що дозволяє перемикатися між сидячим і стоячим режимом роботи. Виготовлений з міцних матеріалів, має достатньо місця для розміщення комп'ютера та аксесуарів.

Акустичні панелі: Використання панелей, які поглинають звук і знижують рівень шуму до комфортних значень (35–45 дБ). Наприклад Interface Carpet Tiles – килимові плитки, що мають властивість поглинати звук. Легко встановлюються на підлогу та значно знижують рівень шуму.

Шумопоглинаюче покриття підлоги: Килимове покриття або спеціальні панелі для зменшення відбиття звуку. Наприклад Mitsubishi Electric Lossnay LGH-RVX Series – система вентиляції з рекуперацією тепла, що забезпечує свіжий приплив повітря та відведення вуглекислого газу з робочих приміщень, зберігаючи оптимальний температурний режим. Daikin Ururu Sarara – кондиціонер із функціями зволоження, очищення та регулювання температури повітря. Чудово підходить для підтримки комфортного клімату в офісному приміщенні.

7.3 Розробка заходів з поліпшення стану охорони праці

Освітлення: Для підвищення рівня освітлення необхідно встановити додаткові світильники з регульованою яскравістю, що дозволить співробітникам налаштовувати освітлення під свої потреби. Згідно з ДСанПІН 3.3.2-007-98, використання моніторів з антивідблисковим покриттям сприятиме зменшенню навантаження на очі. Крім того, неякісне або неправильне освітлення може викликати загальну втому і знижувати продуктивність. Світильники мають мати коефіцієнт кольоропередачі (CRI) понад 80 для більш природного освітлення. Гарним варіантом буде BenQ ScreenBar – настільна лампа, яка кріпиться до монітора, має регульовану яскравість і колірну температуру, не створює відблисків. Philips Hue Go – мобільна лампа з можливістю налаштування колірної температури від теплого до холодного

світла. Luxo L-1 LED Task Light – класична настільна лампа з можливістю регулювання напрямку та яскравості, що підходить для тривалої роботи за комп'ютером.

Для загального освітлення при рівні 500 люксів: 45 м^2×500 люкс = 22500 люменів. 45м^2×300люкс = 13500 люменів.

Таким чином, для загального освітлення приміщення площею 45 м² необхідно забезпечити світильники, що сумарно дають від 13 500 до 22 500 люменів.

Вентиляція і кондиціювання: Впровадження системи автоматичного контролю мікроклімату забезпечить підтримку оптимальних умов роботи. Рекомендується встановлення датчиків якості повітря для виявлення підвищеного рівня вуглекислого газу, що буде сигналом для активації системи вентиляції.

Датчики якості повітря: Вони дозволяють автоматично виявляти підвищений рівень CO_2 та інших забруднювачів, сигналізуючи про необхідність провітрювання або активуючи вентиляційну систему. Гарним варіантом буде Netatmo Smart Indoor Air Quality Monitor – вимірює температуру, рівень CO_2, вологість та рівень шуму. Awair Element – вимірює температуру, вологість, концентрацію CO_2, леткі органічні сполуки (VOC) та дрібні частинки PM2.5.

Модернізація робочих місць включає надання програмістам крісел з регулюванням висоти і нахилу, що сприяє підтримці правильної постави. Також важливо використовувати столи з можливістю регулювання висоти для роботи стоячи.

Проведення профілактичних оглядів: періодичні медичні огляди для оцінки стану здоров'я працівників і запобігання професійним захворюванням.

Захист від електромагнітного випромінювання: розміщення обладнання з урахуванням безпечних відстаней і застосування екранування моніторів.

7.4 Техніка безпеки та протипожежна профілактика

Електробезпека: Електроустановки повинні бути захищені заземленням та зануленням відповідно до вимог ДБН В.2.5-28:2017. Встановлення ПЗВ дозволить уникнути ураження електричним струмом у випадку замикань.

Забезпечення протипожежної безпеки включає застосування сучасних засобів раннього виявлення пожежі, таких як системи димової сигналізації та автоматичні вогнегасники. Приміщення, в якому працюють 6 програмістів, повинно бути обладнане щонайменше одним порошковим вогнегасником ємністю не менше 5 кг. Рекомендовано використовувати вогнегасники типу ВП-5 або аналогічні моделі, які є ефективними для гасіння пожеж класу А (тверді речовини) та класу Е (електрообладнання під напругою).

Елементи приміщення з підвищеною пожежонебезпекою**:** Електричні установки та обладнання: Проводка, розетки, розподільчі щити та інші електричні прилади, особливо якщо вони знаходяться в поганому стані або перевантажені, є потенційними джерелами пожежі. Кабелі та проводка: Неправильне прокладання кабелів, їх пошкодження або недостатній захист можуть призвести до короткого замикання та пожежі.

Меблі та оздоблення: Старі або низькоякісні меблі, особливо виготовлені з легкозаймистих матеріалів, також можуть бути джерелом загрози.

Матеріали оздоблення приміщення з низькою здатністю до займання**:** Керамічна плитка: має високу термостійкість і низьку здатність до займання. Ламінат з вогнестійким покриттям: спеціальні моделі ламінату з вогнестійкими властивостями, які відповідають стандартам протипожежної безпеки.

Металеві панелі або декоративні покриття: не підтримують горіння, чудово підходять для офісів з підвищеними вимогами до безпеки.

Фарби з вогнестійкими властивостями: спеціальні фарби та покриття, які зменшують горючість поверхонь і не сприяють поширенню вогню.

Меблі з металу та скла: часто використовуються в офісах для забезпечення не тільки естетичного вигляду, але й безпеки.

Слід регулярно проводити навчання з протипожежної безпеки, включаючи інструктаж працівників щодо дій у разі пожежі та використання первинних засобів пожежогасіння. У робочих приміщеннях також має бути чітко позначений план евакуації.

У приміщенні необхідно встановити світлові вказівники для позначення виходів та шляхів евакуації. Згідно з нормативними вимогами, двері евакуаційних виходів повинні відкриватися у напрямку виходу. Підлога та інші поверхні в приміщенні не повинні бути слизькими, щоб уникнути травмувань під час евакуації.

План евакуації має бути розміщений на видному місці та продубльований в електронному форматі для зручного доступу працівників. Він повинен містити вказівки щодо порядку дій у разі виникнення пожежі, шляхів виходу та місць збору.

7.5 Розрахункова частина

Проведемо розрахунок занулення електроустановки потужністю 4 кВт у приміщенні площею 45 м².

Початкові данні :

1. Довжина магістрального кабелю – L_M , 50м.
2. Коефіцієнт потужності, cosφ = 0.85.
3. Довжина розгалуження – L , 10м.
4. Напруга мережі, U_L =380 В.

Розрахунок номінального струму електроустановки

$$I= P/3 \cdot UL \cdot \cos\varphi$$

де: P=4 кВт – потужність електроустановки, UL=380 В, cosφ=0.85.

Розрахуємо: I=4000/3·380·0.85=4000/557.97≈7.16А

Розрахунок пускового струму електродвигуна.

Пусковий струм визначається як: $I_{пуск}= I * 5$

Розрахуємо: $I_{пуск=} 7.16 \cdot 5 = 35.8$А

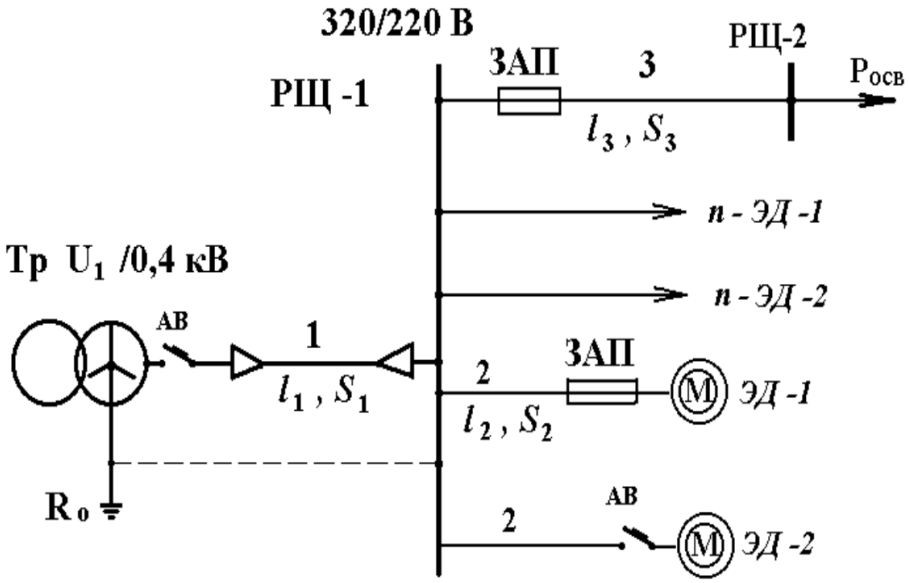

Рисунок 7.1 – Схема електромережі

Номінальна сила струму захисного пристрою: $I_{ном}=I_{пуск}/\beta$ де $\beta = 2.5\beta$ для легких умов пуску.

Розрахуємо: $I_{ном=}35.8/2.5=14.32А$

Для цього значення вибираємо автоматичний вимикач із номінальним струмом захисту 16 А, що є стандартним значенням у доступних таблицях. Мінімальний струм короткого замикання:

$$I_{к\ мін} = I_{ном} * K$$

де K=3 для автоматичного вимикача з інверсно-залежною характеристикою струму $I_{к\ мін=}16 \cdot 3=48А$.

Вибираємо площу перерізу для фазного провідника, щоб він витримував номінальний струм I≈7.16А, і щоб значення тривалого допустимого струму не було меншим за $I_{ном}$=14.32 А.

Вибираємо переріз 2.5 мм² з таблиці для допустимого струму у алюмінієвих проводах, що забезпечує необхідну міцність для даного навантаження.

Максимальний робочий струм: $I_{роб}=I \cdot Ko \cdot Kз$ де Ko=0.8 – коефіцієнт одночасності роботи, Kз=0.85 – коефіцієнт завантаження.

Розрахуємо:

$$I_{роб} = 7.16 \cdot 0.8 \cdot 0.85 \approx 4.87A$$

Переріз нульового захисного провідника обираємо, щоб його площа була не меншою за половину площі фазного провідника: $S_н = 0.5 \cdot S_ф$ де $S_ф = .5$ мм², отже, $S_н = 0.5 \cdot 2.5 = 1.25$мм² Округлюючи значення до стандартного, приймаємо $S_н = 1.5$ мм².

Перевіримо значення струму короткого замикання. Обчислимо дійсне значення струму однофазного короткого замикання за формулою:

$$I_{кр} = U_ф / Z_{петлі}$$

Де $U_ф = 220$В (фазна напруга), а $Z_{петлі}$ розраховується на основі опорів фазного та нульового провідників. підставимо значення опорів для алюмінію (питомий опір $\rho = 0.028$ Ом·мм²/м), довжина фазного та нульового провідників сумарно складає $L = 60$м:

$$R_{петл} = \rho \cdot L / S_ф + S_н = 0.028 \cdot 60 / 2.5 + 1.5 = 0.672\Omega$$

Оскільки Ікр> Імін, обраний переріз проводів є правильним.

Розрахунок проводився відповідно до схеми електромережі.

Висновки з розділу

Розглянуті заходи щодо охорони праці і техніки безпеки вказують на важливість забезпечення комфортних та безпечних умов праці для програмістів. Застосування рекомендацій з освітлення, мікроклімату, ергономіки, а також організація протипожежного захисту та електробезпеки дозволять значно знизити ризики виробничого травматизму та захворювань.

Проведений розрахунок захисного занулення підтверджує необхідність забезпечення електробезпеки у приміщенні за допомогою сучасних засобів захисту.

8 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи адаптивного регулювання складності мультимедійного аркадного комплексу із застосуванням алгоритмів машинного навчання.

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 адаптивного регулювання складності мультимедійного аркадного комплексу із застосуванням алгоритмів машинного навчання.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем адаптивного регулювання складності мультимедійного аркадного комплексу із застосуванням алгоритмів машинного навчання.

– Досліджена система адаптивного регулювання складності мультимедійного аркадного комплексу із застосуванням алгоритмів машинного навчання.

– На основі отриманих результатів досліджень створена програмна реалізація системи адаптивного регулювання складності мультимедійного аркадного комплексу із застосуванням алгоритмів машинного навчання.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання адаптивного регулювання складності мультимедійного аркадного комплексу із застосуванням алгоритмів машинного навчання.

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.

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 створені 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.

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.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Патенко Н.М. Дослідження та програмна реалізація системи адаптивного регулювання складності мультимедійного аркадного комплексу із застосуванням алгоритмів машинного навчання // Збірник праць молодих науковців ЦНТУ. – Вип. 17. – Кропивницький: ЦНТУ, 2027.
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.
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. Богаченко, С., Смірнов, О., Буравченко, К., Смірнова, Т., Конопліцька-Слободенюк, О., Якименко, Н., Смірнов, С. «Обмеження ресурсів мобільних та edge-платформ при розробці програмного забезпечення для реалізації та оптимізації моделей машинного навчання». *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка»*, 2026, т. 2 № 34, С. 677-686.

18. Смірнов, О., Ткачук, Р., Козірова, Н., Константинова, Л., Конопліцька-Слободенюк, О., Якименко, Н., Смірнов, С.. «Дослідження застосування векторних баз даних у генеративному штучному інтелекті». *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка»*, 2026, Том 1 № 33, 667-682.

19. Вінтенко Б.Ю., Смірнова Т.В., Миронець І.В., Смірнов О.А., Буравченко К.О. «Метод оцінки функціональної стійкості комп'ютерно-орієнтованих процедур системи підтримки оперативного персоналу АЕС». *Системи управління, навігації та зв'язку*, 2026, вип. 2(84), С. 62-69.

20. Смірнова Т.В., Усік П.С., Лисенко І.А., Буравченко К.О., Смірнов О.А. «Методологія підтримки технологічних процесів у критичній інфраструктурі з забезпеченням безпеки інформації на основі хмарних технологій». *Безпека інформації.* 2025. Том 31 № 1. С. 49-60.

21. Смірнова Т.В., Буравченко К.О., Добринчук О.А., Смірнов С.А., Якименко Н.М., Смірнов О.А. «Хмарна технологія моніторингу ключових

індикаторів ефективності технологічних процесів у критичній інфраструктурі». *Центральноукраїнський науковий вісник. Технічні науки.* 2025. № 12(43), ч. II. С. 36-54.

22. Лисенко, І., Минайленко, Р., Смірнов, С., Буравченко, К., Якименко, Н., Смірнов, О. «Дослідження інструментів штучного інтелекту для інтелектуального аналізу даних». *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка»,* 2025, № 3(31), С. 227–241.

23. Vintenko, B., Myronets, I., Smirnov, O., Smirnova, T. «Enhancing the Reliability of Input Data for Nuclear Power Plant Operator Support Systems». *International Conference on Sensors, Devices and Systems. SDaS 2025. Lecture Notes in Electrical Engineering,* 2026. vol 1617. pp 3-12 Springer, Cham.

24. Смірнов О., Заріцкий В., Буравченко К., Конопліцька-Слободенюк О., Константинова Л., Якименко Н., Смірнов С. «Оптимізація розпізнавання облич за допомогою CUDA прискореної бібліотеки dlib». *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка»,* 2026, Том 4, № 32, С. 573-582.

25. Kuznetsov O., Frontoni E., Kuznetsova Y., Chevardin V., Smirnov O. «Architectural foundations for adaptive security in edge computing systems». *Cybersecurity Defensive Walls in Edge Computing,* 2025. pp. 21-61.

26. Вінтенко, Б.Ю., Миронець, І.В., Смірнов, О.А., Коваленко, О.В., Усік, П.С., Буравченко, К.О., Лисенко, І.А. «Логіко-структурна модель комп'ютерно-орієнтованої процедури системи підтримки оперативного персоналу АЕС». *Кібербезпека: освіта, наука, техніка*. 2025. Том 2 № 30. С. 413-427, 2025.

27. Смірнова, Т.В. «Дослідження методів, моделей та сучасних ІТ-рішень для підтримки технологічних процесів у критичній інфраструктурі держави». *Кібербезпека: освіта, наука, техніка*. 2025. Том 2 № 30. С.195-208, 2025.

28. 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.

29. Вінтенко Б.Ю., Смірнов О.А., Миронець І.В., Смірнова Т.В., Коваленко О.В., Мацуй 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.

30. Смірнов, О.А., Константинова, Л.В., Конопліцька-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.**)**

31. 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.

32. Вінтенко, Б., Миронець, І., Смірнов, О., Коваленко, А., Конопліцька-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.

33. 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.

34. 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,

35. 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.

36. Malyukov V., Bebeshko 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.

37. 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.

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

39. 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.

40. 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.

41. 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.

42. 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.

43. 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.

44. 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.

45. 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.

46. 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.

47. 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

48. 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.

49. 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.

50. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary

constant weight coding technique». CEUR Workshop Proceedings. Volume 2740, 2020, Pages 102-114.

51. 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.

52. 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.

53. 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.

54. Smirnov O. Kuznetsov A., Zaicenko 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.

55. 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.

56. 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.

57. 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.

58. 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.

59. 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.

60. 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.