

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій**

До захисту допущено:

Завідувач кафедри

_____ Олександр РОЛІК

«__» _____ 20__ р.

**Дипломний проєкт
на здобуття ступеня бакалавра
за освітньо-професійною програмою «Інформаційні управляючі системи
та технології»
спеціальності 126 «Інформаційні системи та технології»
на тему: «Система для пошуку найкращого шахового дебюту за
статистикою гравців різних рівнів»**

Виконав:

студент IV курсу, групи ІС-91
Мінін Павло Андрійович _____

Керівник:

Доцент кафедри, к.ф.-м.н.
Жереб Костянтин Анатолійович _____

Рецензент:

Доц., к.т.н, доц. каф. ОТ
Волокита Артем Миколайович _____

Засвідчую, що у цьому дипломному
проєкті немає запозичень з праць інших
авторів без відповідних посилань.

Студент _____

Київ – 2023 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційні управляючі системи та технології»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олександр РОЛІК

«__» _____ 20__ р.

ЗАВДАННЯ

на дипломний проєкт студенту

Мініну Павлу Андрійовичу

1. Тема проєкту «Система для пошуку найкращого шахового дебюту за статистикою гравців різних рівнів», керівник проєкту Жереб Костянтин Анатолійович, доцент кафедри, к.ф.-м.н., затверджені наказом по університету від «31» травня 2023 р. № 2101-с
2. Термін подання студентом проєкту: 12 червня 2023 року
3. Вихідні дані до проєкту: мова програмування Python.
4. Зміст пояснювальної записки: розробити систему для пошуку найкращого шахового дебюту за статистикою гравців різних рівнів, описати предметне середовище, пояснити мету розробки та проблему, що вирішується, обґрунтувати вибір використаного програмного та технічного забезпечення, надати опис та керівництво з рекомендаціями щодо використання функцій програми, протестувати основний функціонал.
5. Перелік графічного матеріалу: схема алгоритму пошуку найкращого шахового дебюту, схема алгоритму рекурсивного пошуку, діаграма компонентів, діаграма послідовності.
6. Дата видачі завдання 17 квітня 2023 року

Календарний план

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту	Примітка
1.	Ознайомлення з завданням	23.04.2023	
2.	Аналіз предметної області та огляд існуючих аналогів	30.04.2023	
3.	Розробка функціональної моделі	07.05.2023	
4.	Постановка задачі та формування цілей та задач розробки	14.05.2023	
5.	Розробка ПЗ	21.05.2023	
6.	Налагодження програми	24.05.2023	
7.	Виконання графічних документів	27.05.2023	
8.	Оформлення пояснювальної записки	01.06.2023	
9.	Подання ДП на попередній захист	05.06.2023	
10.	Подання ДП на основний захист	12.06.2023	
11.	Подання ДП рецензенту	19.06.2023	

Студент

Павло МІНІН

Керівник

Костянтин ЖЕРЕБ

АНОТАЦІЯ

Мінін П.А. Система для пошуку найкращого шахового дебюту за статистикою гравців різних рівнів. КПІ ім. Ігоря Сікорського, Київ, 2023.

Проект містить 60 с. тексту, 11 рисунків, 15 таблиць, посилання на 28 літературних джерел та 4 графічні матеріали.

Ключові слова: HTTP-сервер, пошук з параметрами, програмне забезпечення, система пошуку, шаховий дебют.

Об'єктом дослідження проекту є процес пошуку найкращих шахових дебютів.

Мета розробки – надання практичного рішення проблеми побудови шахового дебютного репертуару шляхом автоматизації пошуку у відкритих базах даних партій.

У дипломному проекті розроблено фрагменти системи для пошуку найкращого шахового дебюту за статистикою гравців різних рівнів, а саме: алгоритм пошуку найкращого шахового дебюту за обраними параметрами, що використовує рекурсивний пошук вглиб, а також графічний користувацький інтерфейс. Проведено ретельний аналіз та вибір технологічного забезпечення, котре було б найбільш оптимальним для вирішення поставлених задач. Значну увагу було приділено налаштуванню комунікації між системою та сервером, що обробляє статистику, і обробці даних різних видів для побудови сучасного програмного забезпечення у сфері шахів.

Отримані результати можуть бути корисними шахістам усіх рівнів та будь-яким іншим шаховим ентузіастам при побудові дебютних репертуарів чи проведенні власних досліджень.

SUMMARY

Minin P.A. System for searching the best chess opening based on players' statistics at different levels. Igor Sikorsky Kyiv Polytechnic Institute, Kyiv, 2023.

The project consists of 60 pages of text, 11 figures, 15 tables, references to 28 literary sources, and 4 graphical materials.

Keywords: HTTP server, parameterized search, software, search system, chess opening.

The object of the project research is the process of searching for the best chess openings.

The development goal is to provide a practical solution to the problem of building a chess opening repertoire by automating the search in open databases of games.

The graduation project includes the development of fragments of a system for searching the best chess opening based on players' statistics at different levels, namely: an algorithm for searching the best chess opening based on selected parameters using recursive depth-first search and a graphical user interface. A thorough analysis and selection of technological infrastructure were carried out to find the most optimal solution for the defined tasks. Significant attention was given to configuring communication between the system and the statistics-processing server, as well as data processing of various types to build modern software in the field of chess.

The obtained results can be useful for chess players of all levels and any other chess enthusiasts in building opening repertoires or conducting their own research.

Номер рядка	Форм	Позначення	Найменування	Кільк. аркушів	Номер екзем. Номер	Примітка
1			<u>Документація загальна</u>			
2						
3			Знову розроблена			
4						
5	A4	IC91.180БАК.005 ПЗ	Пояснювальна записка	60		
6	A3	IC91.180БАК.005 Д1	Система для пошуку	1		
7			найкращого шахового дебюту			
8			за статистикою гравців різних			
9			рівнів. Схема алгоритму			
10			пошуку найкращого шахового			
11			дебюту			
12	A3	IC91.180БАК.005 Д2	Система для пошуку	2		
13			найкращого шахового дебюту			
14			за статистикою гравців різних			
15			рівнів. Схема алгоритму			
16			рекурсивного пошуку			
17						
18	A3	IC91.180БАК.005 ДЗ	Система для пошуку	1		
19			найкращого шахового дебюту			
20			за статистикою гравців різних			
21			рівнів. Діаграма компонентів			
22			програмного забезпечення			
23						
24	A3	IC91.180БАК.005 Д4	Система для пошуку	1		
25			найкращого шахового дебюту			
26			за статистикою гравців різних			
27			рівнів. Діаграма послідовності			
28						
IC91.180БАК.005 ТП						
3	A	№ докум.	Підп	Д		
Розроб.	Мінін П.А..				Літ.	Арку
Керівн.	Жереб К.А.				1	1
Затв.					КПІ ім. Ігоря Сікорського Група IC-91	
				Система для пошуку найкращого шахового дебюту за статистикою гравців різних рівнів. Відомість проєкту		

Пояснювальна записка
до дипломного проєкту
на тему: «Система для пошуку найкращого
шахового дебюту за статистикою гравців різних
рівнів»

Київ – 2023 року

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ	4
ВСТУП.....	5
1 ЗАГАЛЬНІ ПОЛОЖЕННЯ	7
1.1 Опис предметного середовища.....	7
1.1.1 Опис процесу діяльності	8
1.1.2 Опис функціональної моделі	8
1.2 Огляд наявних аналогів	9
1.3 Постановка задачі	13
1.3.1 Призначення розробки	13
1.3.2 Цілі та задачі розробки	14
Висновки до розділу	14
2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ	16
2.1 Вхідні дані.....	16
2.2 Вихідні дані	18
2.3 Структура масивів інформації	19
Висновки до розділу	21
3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ.....	23
3.1 Змістовна постановка задачі	23
3.2 Математична постановка задачі	23
3.3 Опис методу розв'язання	24
3.4 Обґрунтування методу розв'язання	25
Висновки до розділу	27
4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ	29
4.1 Засоби розробки	29
4.2 Вимоги до технічного забезпечення	30
4.2.1 Загальні вимоги	31
4.3 Архітектура програмного забезпечення	31

					ІС91.180БАК.005 ПЗ				
		№ докум.	Підпис						
Розробив	Мінін П.А.				Система для пошуку найкращого шахового дебюту за статистикою гравців різних рівнів. Пояснювальна записка	Літ.	Арк.	Аркушів	
Перевірив	Жереб К.А.					Т	2	60	
						КПІ ім. Ігоря Сікорського			
Затв.						Група ІС-91			

4.3.1 Схеми алгоритмів.....	33
4.3.2 Діаграма послідовності	35
4.3.3 Діаграма компонентів.....	35
4.3.4 Специфікація функцій	36
Висновки до розділу	39
5 ТЕХНОЛОГІЧНИЙ РОЗДІЛ	41
5.1 Керівництво користувача	41
5.2 Випробування програмного продукту	45
5.2.1 Мета випробувань.....	46
5.2.2 Загальні положення	46
5.2.3 Результати випробувань	46
Висновки до розділу	55
ВИСНОВКИ	56
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	58

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

Варіація – у контексті шахів, можливе продовження шахового дебюту. Наприклад, Сицилійський захист, варіант Дракона, класична варіація позначає такий дебют: 1.e4 c5 2.Nf3 d6 3.d4 cxd4 4.Nxd4 Nf6 5.Nc3 g6 6. Be2.

PGN – це "Portable Game Notation", стандарт, призначений для представлення даних шахової гри за допомогою текстових файлів ASCII.

FEN (Нотація Форсайта – Едвардса) – це стандартний метод опису шахових позицій за допомогою простого тексту.

					ІС91.180БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		4

ВСТУП

Шахи – це гра, яка здобула величезну популярність серед людей з різних країн та культур. Вони привертають увагу своєю складністю та глибиною, а також можливістю розвивати розумові здібності та навички. Історія шахів налічує більше тисячі років і за цей час гра пройшла великий шлях розвитку. Однією з найважливіших складових шахів є дебютна теорія, яка вивчає початкові ходи гри. Вона розвивалася впродовж століть та була вдосконалена багатьма великими гравцями, такими як Ріхард Реті, Боббі Фішер та Гаррі Каспаров. Завдяки розвитку дебютної теорії було створено безліч варіантів початкових ходів, що дозволяє гравцям знаходити нові та цікаві стратегії для гри.

Актуальність проблеми. Зараз актуальною проблемою є знаходження дебюту, який би приніс найбільший успіх шахісту. Так як рівні гравців та їх уподобання можуть сильно відрізнятись, не існує одного дебюту, який би був найкращим для кожного. Саме тому, я обрав систему для пошуку найкращого шахового дебюту за статистикою гравців різних рівнів темою свого дипломного проекту.

Об'єкт роботи. Об'єктом дослідження проекту є процес пошуку найкращих шахових дебютів. Пошук включає у себе розгляд можливих варіацій та порівняння за певною оцінкою позицій, які можуть виникнути у дебюті.

Предмет роботи. Предметом роботи є система для автоматизації пошуку найкращих шахових дебютів на основі статистики партій. Система, яка була розроблена, враховує різні параметри пошуку, які можуть цікавити користувача, та знаходить дебюти, які найбільше наближують перемогу. Для цього використовуються проценти перемог у кінцевих позиціях дебюту та історія популярностей ходів. Таким чином отримані результати будуть статистично обґрунтовані та не будуть залежати від повної кооперації опонента.

Мета. У цьому контексті, метою мого дипломного проекту є надання практичного рішення проблеми побудови шахового дебютного репертуару шляхом автоматизації пошуку у відкритих базах даних партій.

					ІС91.180БАК.005 ПЗ	Арк.
						5
Зм.	Лист	№ докум.	Підпис	Дата		

Для досягнення цієї мети довелося вирішити такі завдання:

- надати користувачу інтерфейс для вводу параметрів пошуку, що його цікавлять;
- реалізувати алгоритм, що надаватиме пріоритет ходам, які стабільно приводять до виграшних позицій;
- реалізувати представлення отриманих результатів, що буде сумісним з найсучаснішими технологіями у шаховому просторі.

Практичне значення. Я вважаю, що така система має практичне значення для гравців будь-якого рівня, від початківців до професіоналів, тому що вона допоможе знайти й дослідити дебюти, які дають найбільші шанси на перемогу спираючись на статистику підлаштовану під бажання користувача.

Структура основних розділів:

- у 1 розділі описано сучасний стан предметного середовища, розроблено функціональну модель, проведено огляд наявних аналогів з їх недоліками та поставлено задачу розробки;
- у 2 розділі надається опис вхідних та вихідних даних системи, розглянуто масиви інформації, що використовуються;
- у 3 розділі змістовно та математично поставлено задачу дипломного проектування, надано та обґрунтовано метод розв’язання;
- у 4 розділі визначено засоби розробки та вимоги до програмного забезпечення, у схемах представлено архітектуру проєкту;
- 5 розділ має керівництво користувача та тести програмного забезпечення.

Крім основних розділів, робота включає в себе вступ, висновки, список використаних джерел та 4 кресленики формату А3. Загальний обсяг пояснювальної записки 60 сторінок.

					ІС91.180БАК.005 ПЗ	Арк.
						6
Зм.	Лист	№ докум.	Підпис	Дата		

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Опис предметного середовища

Дебютна теорія вивчається шахістами з метою мати можливість вийти з попередньої підготовки з вигранною позицією і бути готовим до перших атак зі сторони супротивника. Це дозволяє шахістам орієнтуватися у різних варіантах, які можуть виникнути на дошці, і забезпечує їм конкурентну перевагу. Вивчення дебютної теорії є важливим елементом шахових тренувань, оскільки стартова позиція завжди однакова і може бути детально вивчена, дозволяючи гравцеві матеріалізувати свої знання та стратегію з самого початку гри.

Останні роки свідчать про стрімкий ріст популярності шахових вебсайтів (дивитись рисунок 1.1), де значна кількість партій стає доступною для широкої аудиторії. Цей доступ до великої кількості шахових партій надає ідеальну основу для статистичного аналізу та виявлення найкращих дебютів. За допомогою таких даних можна вивчати успішність різних варіантів дебютів, оцінювати ефективність певних стратегій та рекомендувати гравцям оптимальні початкові ходи для досягнення успіху в грі.

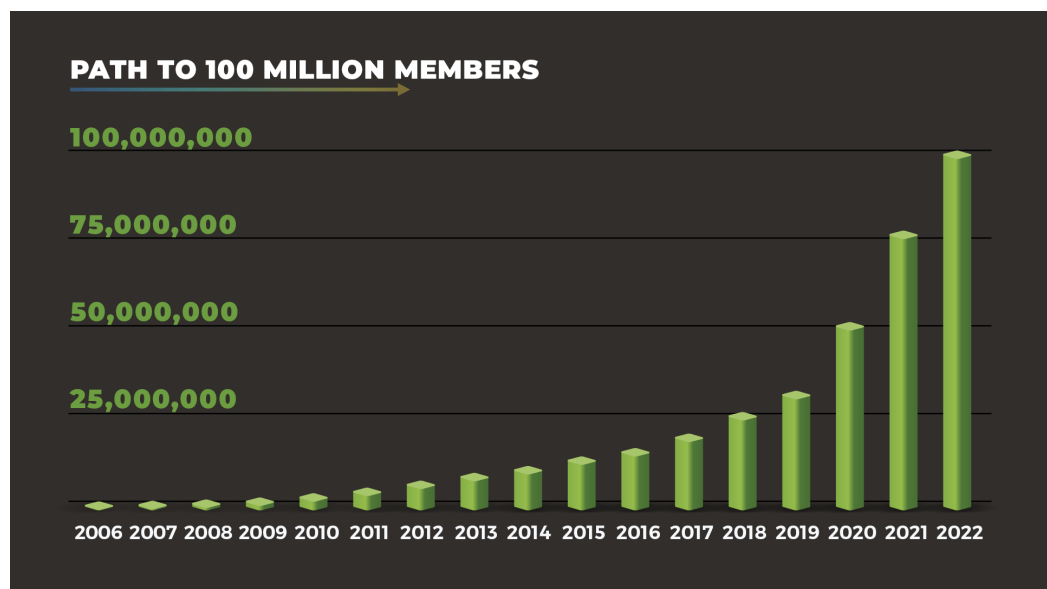


Рисунок 1.1 – Ріст кількості користувачів найпопулярнішого шахового сайту chess.com з роками. [1]

У відкритому доступі можна знайти сервери, що працюють з дампами партій у форматі .pgn, які можуть повернути оброблену інформацію по HTTP-запиту. Саме такий сервер, а саме lila-openingexplorer від некомерційного сайту Lichess [2], буде використаний при розробці системи пошуку найкращого дебюту. За допомогою нього можна буде отримувати для кожної позиції найбільш популярні наступні ходи та відсотки перемог кожної зі сторін при цьому.

1.1.1 Опис процесу діяльності

Для системи пошуку найкращого шахового дебюту за статистикою гравців різних рівнів основними процесами діяльності є:

- вибір користувачем параметрів пошуку: позиції початку пошуку, кольору, який робитиме перший хід, кольору своїх фігур, глибини та ширини пошуку, середнього рейтингу гравців з вибірки та часового контролю партій серед запропонованих, мінімальну популярність для відповідей суперника та мінімальну кількість зустрічей позиції у базі даних;
- здійснення пошуку за обраними параметрами;
- збереження результатів у форматі .pgn.

1.1.2 Опис функціональної моделі

Для роботи з системою передбачено одну дієву особу – користувача. До основних функціональних можливостей належать:

- вибір користувачем параметрів пошуку;
- здійснення пошуку за обраними параметрами;
- збереження результатів у форматі .pgn.

Таблиця 1.1 – Варіанти використання системи та їх опис

Актор	Варіант використання	Опис
Користувач	Вибір параметрів пошуку	Користувач може обрати

Актор	Варіант використання	Опис
		позицію початку пошуку, колір, який робитиме перший хід, колір своїх фігур, глибину та ширину пошуку, середній рейтинг гравців з вибірки та часовий контроль партій серед запропонованих, мінімальну популярність для відповідей суперника та мінімальну кількість зустрічей позиції у базі даних.
Користувач	Здійснення пошуку	Користувач може виконати пошук найкращих дебютів за обраними ним параметрами та побачити результат у створеному файлі формату .pgn чи у вікні результатів

1.2 Огляд наявних аналогів

На ринку програмного забезпечення подібних рішень не існує, тому проєкт можна вважати частково інноваційним. У цьому плані основними конкурентами є більш традиційні методи знаходження дебютів.

Шахові двигуни, такі як Stockfish, пропонують ходи, які будуть найкращими для комп'ютера (роботу зображено на рисунку 1.2). Зазвичай ці ходи приводять до ретельно досліджених дебютів, внаслідок чого позиція частіше за все залишиться рівною і дебютна підготовка шахіста не надасть йому значимої переваги над суперником. Незважаючи на це, шахові двигуни є важливим компонентом сучасних дебютних досліджень, особливо коли досліджуються

									ІС91.180БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата						9

непопулярні варіації.

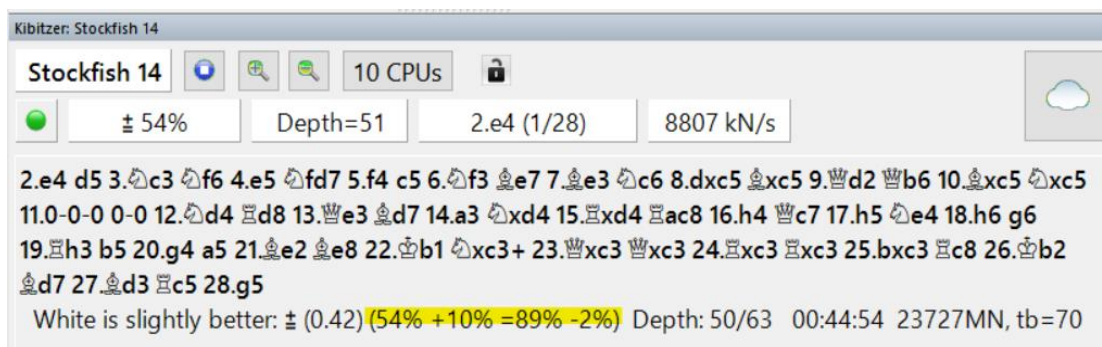


Рисунок 1.2 – Приклад роботи шахового двигуна Stockfish

Таблиця 1.2 – Переваги та недоліки використання шахових двигунів для пошуку дебютів

Переваги	Недоліки
Шахові двигуни пропонують ходи, які переможуть найсильніших шахістів	Людині неможливо вивчити усі можливі продовження партій, які може проаналізувати двигун
Рекомендовані ходи не залежать від популярності дебюту та його дослідженості	Рекомендуються теоретичні дебюти, які постійно зустрічаються у грі
У деяких дебютах може знайтись недосліджена варіація, силу якої гарантуватиме двигун	Шахові двигуни оцінюють позицію вважаючи, що обидві сторони гратимуть ідеально

Література присвячена дебютній теорії зазвичай фокусує увагу на одному конкретному дебюті та не має статистичного обґрунтування вибору перших ходів. [4] Побудова репертуару шляхом порівняння інформації отриманої з ретельного розбору книг присвяченим різним дебютам є непрактично складною задачею та має якийсь сенс тільки для титулованих гравців найвищих ешелонів. Такі книжки ретельно досліджують та розкривають красу дебюту, що запов на око автору. Вивчення цього матеріалу є чудовим наступним кроком після отримання цього

дебюту як результату виконання системи для пошуку найкращого шахового дебюту за статистикою гравців різних рівнів.

Таблиця 1.3 – Переваги та недоліки використання шахової літератури для пошуку дебютів

Переваги	Недоліки
Глибоко досліджуються різні варіації у дебюті	Зазвичай розглядається тільки один дебют без статистичного обґрунтування вибору
Надаються практичні рекомендації по вивченню та використанню	Неможливо врахувати особисті побажання читача
Присутні розгляди партій відомих шахістів у дебюті	Має багато суб'єктивних оцінок автора

Одним з резонних варіантів є порада шахового експерту. І навіть враховуючи, що не кожний має доступ до консультації з такою людиною один на один, її досвід все одно поступатиметься базі даних з мільйонами партій і програмному забезпеченню, що зможе ці дані проаналізувати.

Таблиця 1.4 – Переваги та недоліки використання порад шахових експертів для побудови дебютного репертуару

Переваги	Недоліки
Поради базовані на багатому досвіді експерта	Суб'єктивні оцінки експерта грають велику роль
При особистому знайомстві можуть бути рекомендовані дебюти, що будуть збігатися з характером людини	Важкий процес отримання особистої консультації
Дебюти можуть супроводжуватися рекомендаціями по грі	Цілі дослідження дебютної теорії можуть відрізнятись

Нарешті, залишається самостійний пошук серед відкритих баз партій (дивитись рисунок 1.3). Цей процес є дуже трудомістким. Щоб отримати зважений результат, доводиться власноруч продивлятися сотні чи тисячі позицій. Тому такий аналіз зазвичай здійснюється поверхово та не розглядає глибокі варіації. Саме автоматизація пошуку серед баз партій і є метою цього проєкту. Звільнення людини від перегляду великої кількості різних можливостей продовження партії та надання оцінки дебютам для порівняння, що враховуватиме проценти перемог на кінцевих позиціях та шанси потрапити до цих позицій, дозволять пришвидшити і поглибити пошук та досягти неочікуваних недосліджених результатів.



Рисунок 1.3 – Дослідник дебютів від Lichess. Відображені дані для початкової позиції

Таблиця 1.5 – Переваги та недоліки самостійного пошуку шахових дебютів серед відкритих баз партій

Переваги	Недоліки
Багато налаштувань параметрів	Трудомісткий процес
Наявність статистики по кожній з позицій	Статистика по позиціям не враховує, які ходи шахіст планує грати у відповідь на ходи суперника
Можливість заглибитися у будь-яку варіацію	Може зникати доступ до бази партій

1.3 Постановка задачі

Шахісти часто стикаються з проблемою незліченого обсягу інформації про різні шахові дебюти, що може бути важким для освоєння. Внаслідок цього, статистична експертиза потребує автоматизації процесу, що і є задачею дипломного проєкту.

1.3.1 Призначення розробки

Шахісти завжди зацікавлені у вивченні нових ідеї, що можуть дати їм конкурентну перевагу в грі. Призначення розробки системи пошуку найкращого шахового дебюту полягає в тому, щоб задовольнити цікавість шахових ентузіастів і надати їм інструмент для дослідження нових перспективних варіацій дебютів.

Система пошуку найкращого шахового дебюту може підійти для широкого спектру користувачів. Вона буде корисна для шахістів різних рівнів, починаючи від початківців, які хочуть покращити свої початкові навички, до досвідчених гравців, які бажають розширити свій дебютний арсенал і вдосконалити стратегію гри. Також ця система може бути корисною для тренерів, які надають інструкції та рекомендації своїм вихованцям, а також для шахових дослідників та ентузіастів, які цікавляться аналізом шахових варіацій та статистичними даними.

Загалом, будь-яка особа, яка хоче покращити свої шахові навички та бути кращим у дебютах, може знайти користь у використанні такої системи.

1.3.2 Цілі та задачі розробки

Основною ціллю є надання шаховим ентузіастам можливості провести пошук найкращих дебютів за параметрами вказаними користувачем. Аналіз баз партій дозволить віднайти «приховані скарби» – такі варіації перших ходів, з якими користувач матиме найбільші шанси опинитися у виграшній позиції.

Задачі стоять використати статистику успішності сторін у позиціях, що отримуються у дебюті, надану сервером, врахувати імовірність потрапити у таку позицію, оцінити можливі варіанти початкового розвитку партії задля визначення найкращого дебюту.

Система для пошуку, що розробляється, здійснює комунікацію з сервером, що систематизує дані з дампу партій. Статистика з цих даних використовується для знаходження найкращих перших ходів у алгоритмі.

Висновки до розділу

Розділ "ЗАГАЛЬНІ ПОЛОЖЕННЯ" розкриває загальний контекст і обґрунтування розробки системи пошуку найкращого шахового дебюту за статистикою гравців різних рівнів.

Дебютна теорія є важливим елементом шахових тренувань, оскільки надає можливість шахістам мати виграшну позицію з самого початку гри і забезпечує конкурентну перевагу.

Завдяки доступності великої кількості шахових партій на вебсайтах, статистичний аналіз може бути використаний для виявлення найкращих дебютів, оцінювання стратегій та рекомендацій гравцям.

Система має одного актора – користувача. Користувач може обрати потрібні йому параметри, запустити пошук та отримати результат у файлі та у

					ІС91.180БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		14

новому вікні.

На ринку програмного забезпечення немає подібних рішень, що робить цей проект частково інноваційним. Основні конкуренти включають шахові двигуни, літературу та самостійний пошук серед відкритих баз партій на онлайн-сайтах.

Ця система має широкий спектр користувачів, включаючи шахістів різних рівнів, тренерів і дослідників. Вона надає можливість поліпшити шахові навички, розширити дебютний арсенал і аналізувати статистичні дані.

Основною ціллю розробки є надання шаховим ентузіастам зручного інструменту для пошуку найкращих дебютів з врахуванням параметрів, вибраних користувачем.

					ІС91.180БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		15

2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Вхідні дані

Вхідні дані у системі представлені у вигляді параметрів пошуку, що задає користувач та інформації з сервера після обробки бази партій.

Для кожного з параметрів вибір представлено таким чином:

– вибрати шахову позицію для початку пошуку:

1) ввести FEN обраної позиції. FEN (Нотація Форсайта – Едвардса) – це стандартний метод опису шахових позицій за допомогою простого тексту; [6]

2) залишити стартову за замовчанням;

– вибрати чий хід у цій позиції:

1) хід білих;

2) хід чорних;

– обрати сторону для якої здійснити пошук найкращого дебюту:

1) для білих;

2) для чорних;

– обрати глибину пошуку:

1) ввести натуральне число;

2) залишити глибину за замовчанням;

– обрати ширину пошуку (кількість варіантів ходів, що проглядаються для кожної позиції):

1) ввести натуральне число;

2) залишити ширину за замовчанням;

– вказати середні рейтинги гравців у партіях, за якими буде проведено пошук (можливо обрати декілька варіантів):

1) 400;

2) 1000;

3) 1200;

4) 1400;

					ІС91.180БАК.005 ПЗ	Арк.
						16
Зм.	Лист	№ докум.	Підпис	Дата		

- 5) 1600;
- 6) 1800;
- 7) 2000;
- 8) 2200;
- 9) 2500;

– вказати контролі часу, за якими були зіграні партії, за якими буде проведено пошук (можливо обрати декілька варіантів):

- 1) UltraBullet;
- 2) Bullet;
- 3) Blitz;
- 4) Rapid;
- 5) Classical;
- 6) Correspondence;

– вибрати мінімальну популярність для відповідей суперника для розглядання у процентах:

- 1) ввести натуральне число;
- 2) залишити значення за замовчанням;

– вибрати мінімальну кількість зустрічей позиції у базі даних:

- 1) ввести натуральне число;
- 2) залишити значення за замовчанням.

Дамп партій може використовуватися будь-який, якщо він поданий у форматі .pgn чи .pgn.zst. Під час розробки була обрана збірка партій за вересень 2019 року зіграних на сайті lichess.org, що налічує 36,996,010 партій. На щастя, Lichess надає можливість безкоштовного користування їх колекціями партій. Сайт, де можна скачати збірку партій, зображено на рисунку 2.1.

Після обробки завантаженої бази партій сервером система для пошуку найкращого шахового дебюту може користуватися отриманою статистикою, яка повертається у форматі JSON як відповідь на HTTP-запит. Програмне забезпечення комунікує з сервером для отримання даних для кожної позиції, після чого знаходить найкращі дебюти та будує дерево ходів.

					ІС91.180БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		17



lichess.org

open database

Lichess games and puzzles are released under the **Creative Commons CC0 license**. Use them for research, commercial purpose, publication, anything you like. You can download, modify and redistribute them, without asking for permission.

STANDARD CHESS

VARIANTS

PUZZLES

4,519,926,692 standard rated games, played on lichess.org, in PGN format. Each file contains the games for one month only; they are not cumulative.

Month	Size	Games	Download
2023 - May	33.7 GB	104,193,153	.pgn.zst / .torrent
2023 - April	32.9 GB	101,706,224	.pgn.zst / .torrent
2023 - March	34.9 GB	108,201,825	.pgn.zst / .torrent
2023 - February	31.8 GB	98,471,537	.pgn.zst / .torrent
2023 - January	33.5 GB	103,178,407	.pgn.zst / .torrent
2022 - December	30.2 GB	93,501,009	.pgn.zst / .torrent
2022 - November	28.9 GB	89,319,297	.pgn.zst / .torrent
2022 - October	30 GB	92,629,656	.pgn.zst / .torrent
2022 - September	28.9 GB	89,174,810	.pgn.zst / .torrent
2022 - August	30 GB	92,670,440	.pgn.zst / .torrent
2022 - July	29.8 GB	92,055,571	.pgn.zst / .torrent
2022 - June	28.3 GB	87,649,571	.pgn.zst / .torrent

Рисунок 2.1 – Відкрита база даних партій Lichess [19]

2.2 Вихідні дані

Для системи вихідними даними є дерево ходів, що формують найкращі варіанти розвитку в залежності від можливих ходів суперника. Воно записується у файл формату .pgn та представляється у текстовому вікні звідки його можна скопіювати. Дерево має розгалуження у випадках, коли існує декілька популярних відповідей для суперника. Варіант ходу для сторони пошуку завжди один –

					ІС91.180БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		18

найкращий. Останній ступінь дерева – параметр глибини пошуку заданий користувачем.

Файл цього формату можна відкрити і проаналізувати у будь-якому шаховому графічному користувацькому інтерфейсі. Існують безкоштовні рішення у вигляді програмного забезпечення (Scid PC, Arena, Chessbase Reader) та онлайн-ресурсів на шахових сайтах (lichess.org/analysis).

2.3 Структура масивів інформації

Перед початком роботи з системою сервер потребує імпортований дамп партій. Сервер приймає партії у вигляді одного великого текстового файлу розширення .pgn або .pgn.zst. Для кожної партії у файлі є інформація про неї, така як псевдоніми гравців, їх рейтинги, дата проведення, часовий контроль. Після інформації йде нотація ходів. Для більшості партій записано також час на годиннику на момент ходу. Побачити як виглядає запис однієї партії можна на рисунку 2.2.

```
[Event "Rated Rapid tournament https://lichess.org/tournament/bc4LtQZc"]
[Site "https://lichess.org/9Hr1XRrx"]
[Date "2019.09.01"]
[Round "-1"]
[White "Groz"]
[Black "fischer2016"]
[Result "1-0"]
[UTCDate "2019.09.01"]
[UTCTime "00:00:04"]
[WhiteElo "1782"]
[BlackElo "1695"]
[WhiteRatingDiff "+17"]
[BlackRatingDiff "-7"]
[ECO "B00"]
[Opening "Owen Defense"]
[TimeControl "60+0"]
[Termination "Normal"]

1. e4 { [Nc1k 0:10:00] } b6 { [Nc1k 0:10:00] } 2. Nf3 { [Nc1k 0:09:54] } Bb7 { [Nc1k 0:09:59] } 3. Nc3 { [Nc1k 0:09:51] } g6 { [Nc1k 0:09:58] } 4. d4 { [Nc1k 0:09:47] } Bg7 { [Nc1k 0:09:56] } 5. Be3 { [Nc1k 0:09:35] } Nf6 { [Nc1k 0:09:50] } 6. Bd3 { [Nc1k 0:09:31] } 0-0 { [Nc1k 0:09:48] } 7. Qd2 { [Nc1k 0:09:25] } Ng4 { [Nc1k 0:09:39] } 8. 0-0-0 { [Nc1k 0:09:13] } d6 { [Nc1k 0:09:34] } 9. Bf4 { [Nc1k 0:09:03] } e5 { [Nc1k 0:09:27] } 10. dxe5 { [Nc1k 0:08:41] } dxe5 { [Nc1k 0:09:07] } 11. Bg5 { [Nc1k 0:08:32] } f6 { [Nc1k 0:08:51] } 12. h3 { [Nc1k 0:08:22] } Nhg4 { [Nc1k 0:08:43] } 13. Kb1 { [Nc1k 0:07:55] } Nae6 { [Nc1k 0:08:41] } 14. Bc4+ { [Nc1k 0:07:41] } Kh8 { [Nc1k 0:08:36] } 15. Nd5 { [Nc1k 0:06:44] } Nc5 { [Nc1k 0:08:18] } 16. h4 { [Nc1k 0:04:55] } Nxe4 { [Nc1k 0:08:13] } 17. Qe3 { [Nc1k 0:04:26] } Nxf3 { [Nc1k 0:07:52] } 18. fxg3 { [Nc1k 0:04:22] } Nf5 { [Nc1k 0:07:41] } 19. Qf2 { [Nc1k 0:04:11] } Nd6 { [Nc1k 0:07:30] } 20. Bb3 { [Nc1k 0:04:07] } a5 { [Nc1k 0:07:24] } 21. a3 { [Nc1k 0:03:54] } a4 { [Nc1k 0:07:22] } 22. Ba2 { [Nc1k 0:03:41] } Ne4 { [Nc1k 0:06:52] } 23. Qe1 { [Nc1k 0:03:29] } Nd6 { [Nc1k 0:06:38] } 24. g4 { [Nc1k 0:03:25] } e4 { [Nc1k 0:06:25] } 25. Nd4 { [Nc1k 0:03:15] } Re8 { [Nc1k 0:05:45] } 26. h5 { [Nc1k 0:02:38] } g5 { [Nc1k 0:05:38] } 27. h6 { [Nc1k 0:02:28] } Bf8 { [Nc1k 0:05:29] } 28. Rf1 { [Nc1k 0:02:21] } Bxd5 { [Nc1k 0:05:01] } 29. Bxd5 { [Nc1k 0:02:15] } Rb5 { [Nc1k 0:04:49] } 30. Bc6 { [Nc1k 0:02:07] } Re5 { [Nc1k 0:04:30] } 31. Nf5 { [Nc1k 0:01:57] } Rac5 { [Nc1k 0:03:52] } 32. Bxa4 { [Nc1k 0:01:51] } Nxf5 { [Nc1k 0:03:41] } 33. Rxd8 { [Nc1k 0:01:48] } Kg8 { [Nc1k 0:03:19] } 34. Rxf5 { [Nc1k 0:01:44] } Rxf5 { [Nc1k 0:03:11] } 35. gxf5 { [Nc1k 0:01:44] } Rxf5 { [Nc1k 0:03:07] } 36. Bb3+ { [Nc1k 0:01:39] } 1-0
```

Рисунок 2.2 – Формат подання інформації у файлі з базою партій

Під час виконання програми алгоритм пошуку здійснює HTTP-запити до сервера для кожної позиції, що розглядається. Оброблені дані сервер повертає у форматі JSON. Формат відповідей сервера представлений на рисунку 2.3. Ці відповіді містять такі змінні як white, draws, black та масив moves. Ці змінні

						Арк.
						19
Зм.	Лист	№ докум.	Підпис	Дата	ІС91.180БАК.005 ПЗ	

показують скільки разів обрана позиція зустрічалася у партіях з вибірки та який був результат. Ця інформація використовується для визначення проценту перемог обраної користувачем сторони.

```
{
  - "opening": {
    "eco": "D10",
    "name": "Slav Defense: Exchange Variation"
  },
  "white": 1443,
  "draws": 3787,
  "black": 1156,
  - "moves": [
    - {
      "uci": "c6d5",
      "san": "cxd5",
      "averageRating": 2423,
      "white": 1443,
      "draws": 3787,
      "black": 1155,
      "game": null
    },
    + { ... }
  ],
  + "topGames": [ ... ],
  "recentGames": [ ],
  + "history": [ ... ]
}
```

Рисунок 2.3 – Формат подання інформації у відповідях сервера

Масив moves містить найпопулярніші ходи у відповідь на цю позицію та змінні white, draws та black для цих ходів. Ходи записані у двох форматах: UCI (Universal Chess Interface) та SAN (Standard Algebraic Notation).

UCI є протоколом комунікації між шаховими програмами та їхніми користувачами. Запис ходу у форматі UCI складається з двох полів: початкове поле та кінцеве поле. Наприклад, хід "e2e4" позначає переміщення пішака з клітинки "e2" на клітинку "e4". Цей запис використовується при виклику функції push_uci() на симульованій шахівниці для здійснення ходу.

В SAN, кожен хід позначається символами, які відображають тип фігури та переміщення. Наприклад, пішаки позначаються лише буквою поля, куди вони

перейшли, наприклад, "e4". Інші фігури позначаються першою літерою свого типу, за якою слідує поле, куди вони перейшли. Наприклад, "Nd2" позначає переміщення коня на поле "d2". Цей запис допомагає легко дізнатися чи є хід матом, чи ні. У ходів, що ставлять мат, у кінці запису SAN стоїть #.

Під час виконання алгоритму створюється симуляція шахівниці за допомогою бібліотеки python-chess. Позиції, які підлягають аналізу, отримуються шляхом виконання ходів, що повертаються з HTTP-запитів, на симульованій дошці. Стан дошки передається як параметр рекурсивної функції, що запобігає повторному створенню елементів. На виході ця рекурсивна функція повертає список з найкращими ходами з розглянутих, оцінкою варіації, що повертається, та найкращі результати виконання з попередніх етапів.

Результуючий список передається до функції побудови .pgn файлу, що спеціально налаштована на обробку інформації у такому вигляді. Вона використовує модуль pgn бібліотеки python-chess для правильного формату запису ходів та різних варіацій.

Висновки до розділу

Вхідні дані системи для пошуку найкращого шахового дебюту представлені у вигляді параметрів пошуку, які користувач вказує, а також інформації з сервера після обробки бази партій. Для більшості з параметрів є рекомендовані значення, що обрані за замовчанням. Для деяких – можливий вибір декількох варіантів.

Параметри пошуку включають вибір шахової позиції для початку пошуку, чий хід у цій позиції, сторону для якої здійснити пошук найкращого дебюту, глибину та ширину пошуку, рейтинги гравців та контроль часу партій, мінімальну популярність відповідей суперника та мінімальну кількість зустрічей позиції у базі даних.

База партій надається серверу у форматі .pgn або .pgn.zst. Такий файл може бути завантажений з відкритої бази даних партій, наприклад, з сайту database.lichess.org.

					ІС91.180БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		21

Після обробки бази партій сервером, система отримує статистичні дані у форматі JSON, які використовуються для пошуку найкращих дебютів та побудови дерева ходів. Цю інформацію вона отримує в ході виконання запитів до сервера для кожної з позиції, що розглядаються.

Вихідними даними системи є дерево ходів, яке записується у файл формату .pgn. Це дерево містить найкращі варіанти розвитку в залежності від можливих ходів суперника.

Файл формату .pgn можна відкрити та проаналізувати у шахових графічних інтерфейсах або онлайн-ресурсах на шахових сайтах.

Для аналізу та обробки партій використовуються бібліотеки, такі як python-chess, які допомагають моделювати шахівницю та виконувати ходи.

Результати пошуку найкращого шахового дебюту представлені у вигляді списку найкращих ходів з розглянутих, оцінки варіацій та найкращих результатів виконання з попередніх етапів.

					ІС91.180БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		22

3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Змістовна постановка задачі

Задача системи пошуку – максимізувати середній шанс перемоги на глибині пошуку опираючись на результати гравців з вибірки у тих самих позиціях та ймовірність кооперації суперника, що дозволить туди дістатися. У результаті пошуку користувач отримає дерево ходів, де на усіх кінцевих вузлах будуть позиції з високим процентом перемог обраної користувачем сторони, у яких при цьому є велика імовірність опинитися, так як вони вимагатимуть одних з найпопулярніших відповідей від суперника.

Таким чином, якщо розбити завдання на підзадачі, задача виглядає наступним чином:

- 1) Надати користувачу вибір параметрів пошуку.
- 2) Здійснити пробіг по різним дебютам, уникаючи непопулярні відповіді суперника та позиції з замалою кількістю партій.
- 3) Оцінити отримані позиції за відсотком перемог обраної сторони у них.
- 4) Оцінити кожний попередній хід як середню зважену оцінку можливих наступних ходів в залежності від відповіді суперника на цей. Вагою у цьому випадку слугуватиме популярність потрібної для цієї варіації відповіді суперника.
- 5) Повторювати пункт 4 до того, як алгоритм опиниться у стартовій позиції.
- 6) Побудувати дерево з ходів з найвищими оцінками.
- 7) Зберегти дерево у файл .pgn.

3.2 Математична постановка задачі

Можна представити задачу як цільову функцію максимізації:

$$Z = \max \sum_0^n x_i p_i, \quad (1)$$

де x_i – процент перемог у позиції i , p_i – загальний шанс, що суперник

					ІС91.180БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		23

зробить усі потрібні ходи для можливості опинитися у позиції i , n – кількість кінцевих вузлів дерева.

Не важко розрахувати, що загальний шанс p_i визначається добутком шансів на усі попередні ходи:

$$p_i = \prod_1^k r_j, \quad (2)$$

де r_j – шанс на потрібний від суперника j -ий хід, k – глибина ходу p_i . Варто зазначити, що k не завжди дорівнює глибині пошуку, так як партія може передчасно закінчитися матом.

3.3 Опис методу розв'язання

Задача розв'язується шляхом побудови системи пошуку найкращого шахового дебюту, використовуючи підзадачі з підрозділу 3.1 як алгоритм.

Пробіг по дебютам можна представити як пробіг по дереву, де вершинами є різні позиції, що можуть виникнути. Так як не існує точної межі, коли партія виходить з дебюту, тому дерево потрібно обмежити штучно. Для цього можна застосовувати під час пошуку заздалегідь задану певну межу глибини L . Це означає, що вузли на глибині L розглядаються таким чином, як якщо б вони не мали наступників. Такий підхід називається пошуком з обмеженням глибини. [20]

Для системи для пошуку найкращого шахового дебюту за статистикою гравців різних рівнів було обрано пробіг вглиб з використанням умовної оптимізації. Для успішного виконання система потребуватиме граничних значень для замалої вибірки партій, популярної/непопулярної відповіді суперника та меж параметрів пошуку.

Окремо має бути реалізована система запису до файлу формату .pgn.

					ІС91.180БАК.005 ПЗ	Арк.
						24
Зм.	Лист	№ докум.	Підпис	Дата		

3.4 Обґрунтування методу розв'язання

Пробіг можна здійснювати як вшир, так і вглиб, що зображено на рисунку 3.1.

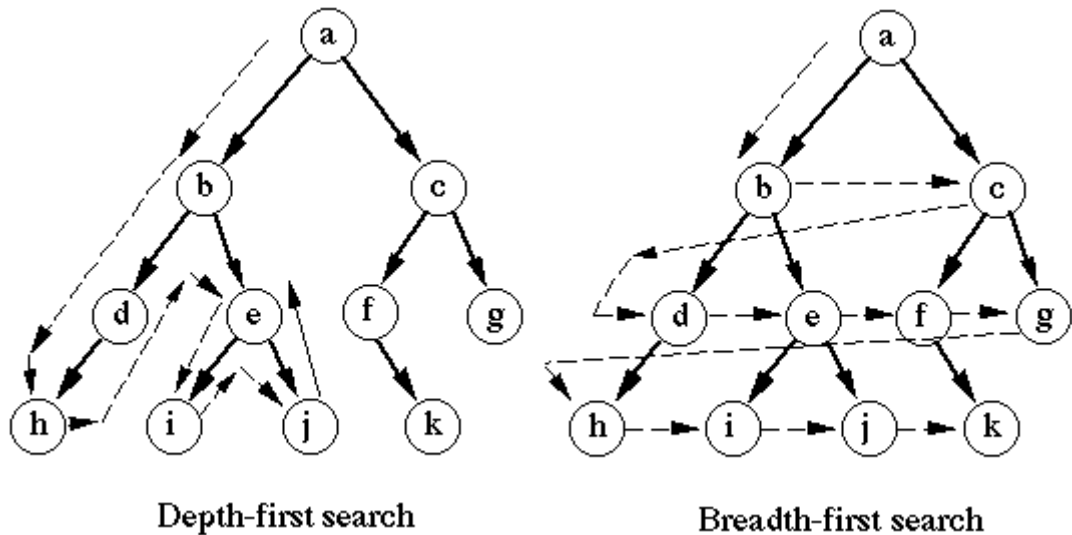


Рисунок 3.1 – Пробіг вглиб (зліва) та пробіг вшир (справа)

Пошук у глибину починається з якнайшвидшого занурення вниз до листових вузлів дерева. Щоб здійснити пробіг спочатку відвідується початковий вузол, потім його дочірні елементи (обхід попереднього порядку): a b d h e i j c f k g.

Пошук у ширину перетинає дерево рівень за рівнем, відвідуючи вузли дерева вище в порядку a b c d e f g h i j k. [21]

Пробіг у глибину було обрано, так як він має просту реалізацію та є більш адаптивним. Для системи для пошуку найкращого дебюту, де статистика будується з даних з кінцевих позицій, логічніше використовувати пробіг, де кінцеві вершини зустрічаються раніше. Це може стати в нагоді при імплементації нового функціоналу у майбутніх оновленнях програмного забезпечення.

Під час вибору методу оптимізації було два логічних кандидати: умовна оптимізація та евристична.

У багатьох випадках є можливість використовувати деяку додаткову

інформацію, що стосується задачі, щоб сприяти скороченню часу і обсягу пам'яті, необхідних для виконання пошуку. Інформацію такого роду називають евристичною. Стратегія пошуку, яка використовує евристичну інформацію, називається евристичним пошуком. Процедура вибору альтернативних вершин, що враховує евристичну інформацію про завдання, називається евристикою. Часто евристичну інформацію знаходять, проводячи поглиблене дослідження умов завдання, будуючи теорію рішення відповідних завдань. [22]

У нашому випадку нескладно було б оптимізувати пошук за евристичними успішності та популярності дебютів, проте впровадження такого рішення мало би більше негативного впливу, ніж позитивного. На перший погляд здається логічним виконувати пошук у варіаціях, де процент перемог збільшується з кожним ходом. Але такий підхід суперечить з ідеєю системи у знаходженні найкращого дебюту. Так як евристичний пошук не має сенсу при повному проході графу і проценти перемог у кінцевих позиціях дебюту не мають прямої залежності від процентів перемог у попередніх, то знайдений таким чином дебют не можна вважати дійсно найкращим.

У умовній оптимізації додаткові обмеження вводяться у задачу з метою забезпечення дотримання певних умов чи обмежень на вихідні результати. Оптимізація з обмеженнями вимагає знаходження оптимального рішення, яке максимізує або мінімізує цільову функцію, враховуючи одночасно набір обмежень. [23]

У контексті знаходження найкращого дебюту це має повний сенс. Є багато параметрів, таких як популярність відповідей суперника, кількість зустрічей позицій у базі даних і т.д., що користувач міг би зазначити для пришвидшення та підлаштування системи під свої інтереси.

Один із ключових аспектів цього методу полягає у використанні даних з бази партій для оцінки проценту перемог у кожній позиції. Це дозволяє отримати практичну оцінку ефективності дебютів на основі досвіду реальних гравців. Використання реальних даних забезпечує більш об'єктивну оцінку, оскільки враховуються реальні результати партій.

					ІС91.180БАК.005 ПЗ	Арк.
						26
Зм.	Лист	№ докум.	Підпис	Дата		

Крім того, історія популярності відповідей суперника є важливою складовою методу. Ця інформація дозволяє врахувати типову стратегію суперника та його найбільш ймовірні відповіді на кожний хід. Це дозволяє побудувати дебютний репертуар, який ефективно реагує на типові ходи суперника і забезпечує високі шанси перемоги.

Також важливо врахувати, що метод не вимагає глибоких теоретичних знань у дебюті. Замість цього він базується на емпіричних даних та статистиці. Це дозволяє гравцю зрозуміти ефективні ходи та стратегії на основі практичного досвіду та результатів реальних партій.

В цілому, метод розв'язання, описаний у підрозділі 3.3, є надійним та практичним підходом до побудови шахового дебютного репертуару. Він поєднує в собі історію успішності позицій та популярності відповідей суперника для досягнення максимальної ефективності у виборі найкращого дебюту.

Висновки до розділу

Задача системи пошуку полягає в максимізації середнього шансу перемоги на певній глибині пошуку, використовуючи результати гравців у вибірці та імовірність кооперації суперника.

Цільова функція максимізації враховує процент перемог у кожній позиції та загальний шанс, що суперник зробить потрібні ходи для досягнення цієї позиції.

Для вирішення задачі використовується система пошуку найкращого шахового дебюту, яка включає в себе вибір параметрів пошуку та оцінку позицій на основі шансів перемоги та популярності відповідей суперника.

Побудований метод розв'язання задачі задовольняє вимоги поставлених підзадач і надає практичний підхід до пошуку найкращого шахового дебюту, заснований на реальних даних та стандартній поведінці суперника.

Для успішного виконання системи необхідно враховувати граничні значення для замалої вибірки партій, популярної/непопулярної відповіді суперника та межі параметрів пошуку.

					ІС91.180БАК.005 ПЗ	Арк.
						27
Зм.	Лист	№ докум.	Підпис	Дата		

Результати пошуку найкращого дебюту представляються у вигляді дерева ходів, яке може бути збережене у файл формату .pgn для подальшого аналізу.

					ІС91.180БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		28

4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

4.1 Засоби розробки

Для написання алгоритму була використана мова програмування Python. Python є популярною мовою програмування, яка має кілька переваг для написання систем пошуку:

1) Простота та зрозумілість: Python має простий та читабельний синтаксис, що полегшує розробку та розуміння коду. Це робить його ідеальним вибором для швидкого написання алгоритмів.

2) Багатофункціональність: Python має широкий вибір стандартних бібліотек та сторонніх модулів, які дозволяють ефективно виконувати різноманітні завдання. Це включає роботу з мережевими запитами, обробку даних, роботу з файлами та багато іншого.

3) Велика спільнота та підтримка: Python має активну спільноту розробників, що означає, що завжди є доступ до підтримки, документації та різноманітних ресурсів. Це сприяє швидкому вирішенню проблем та розвитку ефективних рішень.

4) Швидкість розробки: Python відомий своєю високою продуктивністю та швидкістю розробки. Він дозволяє розробникам швидко створювати та тестувати прототипи, що сприяє ефективному впровадженню алгоритму пошуку.

5) Портативність: Алгоритм, написаний на Python, може працювати на різних операційних системах без необхідності у великих змінах.

Враховуючи ці переваги, Python є потужним інструментом для реалізації алгоритму пошуку і дозволяє швидко та ефективно створювати високоякісний код.

Для роботи з запитами та даними у форматі JSON використана бібліотека requests. Вона має зручний і простий інтерфейс для виконання HTTP-запитів, що спрощує взаємодію з веб-серверами, підтримує усі методи HTTP та дозволяє зручно отримувати та обробляти дані у форматі JSON, які надходять з серверу.

Для симуляції шахівниці та запису партій у формат PGN використано

					ІС91.180БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		29

бібліотеку python-chess. Вона забезпечує генерацією ходів, перевірку легальності ходів та підтримує усі поширені формати. [14]

Графічний користувацький інтерфейс був побудований з використанням бібліотеки tkinter з двох причин.

По-перше, tkinter є простою у використанні бібліотекою. Вона надає легкий спосіб створення основних елементів інтерфейсу, таких як кнопки, полі для введення тексту і прапорці. Також вона має добре задокументований API, який допомагає розробникам швидко орієнтуватися і розуміти, як використовувати різні функції і можливості бібліотеки.

По-друге, tkinter є переносною бібліотекою, що означає, що програми, створені з її допомогою, можуть працювати на різних платформах, таких як Windows, macOS і Linux, без змін у вихідному коді. Це робить tkinter відмінним вибором для створення крос-платформеного графічного інтерфейсу.

Хоча є інші бібліотеки для створення графічного інтерфейсу в Python, такі як PyQt і wxPython, tkinter є популярним вибором для простих інтерфейсів або проєктів з обмеженими вимогами, завдяки своїм простоті, доступності і переносимості.

4.2 Вимоги до технічного забезпечення

Успішне розгортання та функціонування системи залежить від технічних вимог, які визначають необхідне апаратне та програмне забезпечення. Врахування цих вимог є важливим для забезпечення швидкості, стабільності та зручності використання системи користувачами.

Вимоги до технічного забезпечення клієнтської машини:

- процесор: рекомендовано використовувати процесор з частотою не менше 3 ГГц;
- оперативна пам'ять (RAM): рекомендовано мати мінімум 8 ГБ RAM.

					ІС91.180БАК.005 ПЗ	Арк.
						30
Зм.	Лист	№ докум.	Підпис	Дата		

4.2.1 Загальні вимоги

Програмне забезпечення потребує налаштований метод комунікації з сервером-аналізатором з HTTP-запитами та JSON відповідями правильного формату (див. API lila-openingexplorer). Сервер повинен мати імпортовану базу партій.

4.3 Архітектура програмного забезпечення

Архітектура програмного забезпечення для системи пошуку найкращого дебюту за статистикою гравців різних рівнів поділена на такі компоненти:

– Графічний інтерфейс: Графічний інтерфейс складається з двох вікон. Головне вікно програми надає користувачу можливість обрати усі параметри для пошуку, такі як позицію початку пошуку, глибину, ширину, рейтинги гравців тощо. Вікно результатів показує результати пошуку у вигляді дерева ходів. Також містить кнопку "Копіювати результат".

– Алгоритм пошуку: Цей компонент приймає вибрані параметри з графічного інтерфейсу для подальшого використання. Він також відповідає за взаємодію з сервером. Він здійснює HTTP-запити до сервера для отримання статистики по позиціям та популярних відповідей суперників з бази партій. Алгоритм пошуку аналізує отриману статистику та популярність відповідей для побудови дерева ходів з найкращими відповідями на можливі популярні ходи суперника. Після закінчення виконання алгоритм передає отримані результати до графічного інтерфейсу для відображення, а також зберігає результати пошуку у файл формату .pgn у призначену для цього папку results.

– Сервер: Сервер містить імпортовану базу партій для аналізу та статистики. Може використовувати будь-яку базу даних. Сервер lila-openingexplorer, який використовується у нашому випадку, працює з MongoDB і запускається локально. HTTP-сервер надає API для взаємодії з алгоритмом пошуку, обробки запитів та передачі необхідної інформації.

					ІС91.180БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		31

Ця архітектура передбачає, що графічний інтерфейс взаємодіє з алгоритмом пошуку, який в свою чергу взаємодіє з сервером, отримуючи необхідну статистику та інформацію. Результати пошуку відображаються у вікні результатів, з якого їх можна скопіювати та зберігати до файлу.

Переваги такої архітектури програмного забезпечення включають:

– Зручний графічний інтерфейс: Користувач може взаємодіяти з програмою шляхом вибору параметрів за допомогою головного вікна програми. Це робить користування програмою більш зручним і інтуїтивно зрозумілим.

– Налаштування параметрів: Користувач може вибрати різні параметри для здійснення пошуку, такі як кольори дебютного репертуару, позиції, глибина та ширина пошуку, фільтри бази партій тощо. Це дозволяє користувачеві налаштувати пошук під свої потреби та вимоги.

– Використання сервера для отримання інформації: Алгоритм пошуку здійснює HTTP-запити до сервера, що містить імпортовану базу партій. Це дозволяє отримувати актуальні статистичні дані та результати на основі популярності відповідей суперника.

– Ефективність алгоритму пошуку: Алгоритм пошуку порівнює популярність відповідей суперника та виконує обчислення для побудови дерева ходів. Це дозволяє отримувати найкращі відповіді на можливі популярні ходи суперника. Такий підхід сприяє ефективному пошуку та представленню результатів користувачеві.

– Зручне збереження результатів: Результати пошуку відображаються у новому вікні та можуть бути скопійовані користувачем. Крім того, результати зберігаються до файлу з розширенням .pgn у папку results, що дозволяє зберігати та повторно використовувати знайдені дебютні ходи.

Все це сприяє зручному використанню програми, ефективному пошуку та збереженню результатів для подальшого використання.

Програма була розроблена як виконуваний файл з графічним інтерфейсом, а не, наприклад, веб- або мобільний додаток, з декількох причин:

– Виконуваний файл надає більшу незалежність і доступність. Користувач

					ІС91.180БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		32

може встановити програму на свій комп'ютер і використовувати її навіть без підключення до інтернету. Це особливо корисно, коли користувач хоче мати постійний доступ до програми незалежно від наявності інтернет-з'єднання.

– Швидкодія і продуктивність. Виконуваний файл може працювати безпосередньо на локальному комп'ютері користувача, що забезпечує оптимальну продуктивність. Операції пошуку, обробка даних, математичні дії можуть виконуватися швидше і ефективніше.

– Кращий контроль над функціональністю і безпекою. Виконуваний файл дозволяє розробникам забезпечити більший контроль над функціональністю програми і захистити її від несанкціонованого доступу чи модифікацій. Крім того, використання локального сервера HTTP дозволяє уникнути потенційних проблем з безпекою, пов'язаних з публічним доступом до програми через інтернет. Це грає найбільшу роль для шахістів, які готують репертуар до турніру.

Загалом, використання виконуваного файлу з графічним інтерфейсом дозволяє забезпечити більшу доступність, продуктивність і безпеку для користувачів програми.

4.3.1 Схеми алгоритмів

Схеми алгоритмів приведені у графічних матеріалах дипломного проекту.

Після передачі параметрів з графічного інтерфейсу пошук починається зі створення симуляції шахівниці.

Далі одним з попередніх етапів перед викликом рекурсивної функції пошуку є визначення чи здійснюється пошук для кольору, який має робити перший хід у позиції для початку пошуку. Це впливає на кількість ходів, які матимуть бути присутні у кінцевому результаті. Якщо кольори, задані користувачем, співпадають, то рекурсивний пошук викликається на усіх ходах, що отримуються з першого HTTP-запиту. Після того як пройдуть усі рекурсії, з отриманих дерев визначається найкраще. У такий спосіб дерево ходів може розпочинатися і закінчуватися ходами одного кольору. Якщо ж ні, рекурсивний

					ІС91.180БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		33

пошук виконується на стартовій позиції.

Отриманий результат потім передається до функції запису до рgn. Ця функція зчитує інформацію про дерево ходів та переводить її до потрібного формату. Кінцевий текст записується до файлу та повертається до графічного інтерфейсу, щоб бути відкритим у вікні результатів.

Розглядаючи алгоритм рекурсивного пошуку, він починається з перевірки поточної глибини. Причиною слугує те, що можуть відрізнятись, в залежності від глибини, методи обрахунку оцінки позиції. На глибині, заданій в параметрах пошуку, оцінка отримується як середнє зважене відсотків перемог у позиціях, де ваговим коефіцієнтом слугує популярність потрібної для цієї варіації ходу суперника. На попередніх глибинах оцінка отримується шляхом обчислення середнього зваженого оцінок, що отримуються з наступних викликів рекурсивної функції.

Якщо перевірка повертає false, то виконується HTTP-запит до сервера. Для кожного з ходів з відповіді сервера перевіряється, чи ставить він мат.

Якщо так, то це означає, що після цього продовжень вже не існує і колір, для якого здійснюється пошук, програє. Це продовження додається до списку варіацій з оцінкою 0.

Якщо ж ні, то хід виконується і робиться HTTP-запит для отриманої позиції. Перевіряємо умову закінчення циклу та, за непроходження, додаємо варіацію з найвищим відсотком перемог до списку на повернення.

Після закінчення обходу варіацій, якщо результат не пустий, то обчислюємо оцінку як було описано раніше. У випадку коли усі варіації опинилися занадто непопулярні, оцінкою стає процент перемог у позиції з виклику рекурсії. Повертаємо список найкращих відповідей на ходи суперника і оцінку.

Схожим чином працює частина алгоритму і для ітерацій на не останній глибині. Тільки тут порівнюються вже не позиції з їх відсотками перемог, а цілі варіації з їх оцінками. У кінці повертається список з результатами від ітерації на наступній глибині та загальною оцінкою.

4.3.2 Діаграма послідовності

Діаграма послідовності є інструментом, що дозволяє детально описати послідовність виконання операцій та взаємодію об'єктів. Вона фіксує порядок взаємодії та використовує вертикальну вісь для показу часу, коли повідомлення надсилаються та отримуються.

Для уявлення взаємодії між компонентами в застосунку була створена діаграма послідовності, яка дозволяє зрозуміти, як система реагує на дії користувача та які процеси виконуються. Це дозволяє зробити представлення більш простим та зрозумілим, візуально відображаючи послідовність подій та взаємодію між компонентами.

Діаграма послідовності приведена у графічних матеріалах дипломного проєкту.

При роботі з програмою користувач взаємодіє з графічним користувацьким інтерфейсом і вводить значення для параметрів, які його цікавлять.

Коли користувач тисне кнопку пошуку, інтерфейс викликає алгоритм за обраними користувачем параметрами.

Поки виконується алгоритм, він постійно потребує значень статистики по ходам та процентам перемог для різних позицій. За цим він звертається до сервера, який працює з базою даних партій.

Коли пошук закінчується, результат, крім запису до файлу, повертається до графічного інтерфейсу у новому вікні у вигляді текстового віджету.

4.3.3 Діаграма компонентів

Діаграма компонентів служить для розбиття розробленої системи на окремі високорівневі функціональні компоненти. Кожен компонент виконує чітко визначену функцію в системі і взаємодіє лише з іншими необхідними елементами, якщо це необхідно.

Діаграма компонентів, також наведена у графічних матеріалах дипломного

проекту та дозволяє легко візуалізувати структуру та взаємозв'язки компонентів системи.

4.3.4 Специфікація функцій

Розглянемо специфікацію розроблених та використаних функції у даному застосунку.

Таблиця 4.1 – Опис функцій

Назва	Опис
Алгоритм пошуку «search_engine.py»	
search (for_white, white_to_move, starting_pos, time_controls, ratings, depth, width, min_games, min_response_pop)	Функція початку пошуку. Приймає параметри для пошуку, а саме: колір, для якого здійснюється пошук; колір, який ходить перший у позиції; стартова позиція у FEN; часові контролю розділені комами; рейтинги гравців розділені комами; глибину пошуку; ширину пошуку; мінімальну кількість зустрічей позиції; мінімальну популярність відповідей суперника. Виконує попередні дії, регулює виклик recursive_search() в залежності від параметрів, викликає write_to_pgn(). Повертає результат пошуку у вигляді тексту формату PGN.
recursive_search (board_state, cur_depth, depth, time_controls, ratings, width, color, min_games,	Рекурсивна функція, що використовується функцією початку пошуку search(). Проводить більшу

Назва	Опис
min_response_pop)	<p>частину дій алгоритму. Приймає оброблені функцією search() параметри. Знаходить найкращі дебюти з заданої позиції шляхом рекурсивного пошуку вглиб з обмеженнями. Шукає варіації з найвищими оцінками серед розглянутих, де оцінкою слугує середнє зважене наступних оцінок, або процентів перемог у позиціях, якщо наступних оцінок немає. Ваговим коефіцієнтом виступає популярність ходу суперника, потрібного для варіації. Повертає багатовимірний список найкращих ходів з варіаціями, куди вони приводять, та оцінку.</p>
write_to_pgn (tree, starting_position)	<p>Функція запису до файлу. Спеціально налаштована для обробки списків формату повернень від recursive_search(). Приймає дерево ходів записане у багатовимірний список та стартову позицію пошуку. Проходить по дереву, додаючи кожний хід до партії з варіаціями у форматі PGN. Результат записується до файлу з розширенням .pgn у папку results та повертається у вигляді</p>

Назва	Опис
	тексту.
jprint (obj)	Функція зручного виводу JSON відповідей у консоль. Приймає JSON текст і виводить його у консоль з табуляцією.
Графічний користувацький інтерфейс «opening_search_gui.py»	
main()	Функція початку роботи з програмою. Створює головне вікно програми з усіма віджетами для вводу значень параметрів, необхідною інформацією та кнопкою початку пошуку. Забезпечує виклик функції launch_algorithm() за натиском на кнопку.
launch_algorithm ()	Функція, що запускає пошуковий алгоритм та передає йому обрані користувачем параметри. Збирає значення параметрів з віджетів, проводить приведення типів, оброблює некоректно введені значення, запускає алгоритм пошуку search_engine.search() з параметрами користувача, по завершенню створює вікно з результатами.
copy_result ()	Функція, що копіює отриманий текст з поля результатів пошуку до буферу обміну.
add_selected_ratings ()	Функція, що бере значення з віджету

Назва	Опис
	вибору середніх рейтингів гравців та наповнює ними список.
add_selected_speeds ()	Функція, що бере значення з віджету вибору часових контролів та наповнює ними список.

Висновки до розділу

Мова програмування Python була використана для написання алгоритму пошуку найкращого дебюту. Python має простий та зрозумілий синтаксис, багатофункціональність, широку спільноту та велику підтримку, що робить його потужним інструментом для реалізації таких систем.

Для роботи з запитам та даними у форматі JSON була використана бібліотека requests. Вона забезпечує зручний і простий інтерфейс для виконання HTTP-запитів, обробки відповідей у форматі JSON та взаємодії з веб-серверами.

Вимоги до технічного забезпечення клієнтської машини включають процесор з частотою не менше 3 ГГц та мінімум 8 ГБ оперативної пам'яті (RAM). Це рекомендовані вимоги, які забезпечать швидкість та стабільність роботи системи.

Архітектура програмного забезпечення включає графічний інтерфейс, алгоритм пошуку та сервер. Графічний інтерфейс надає зручний спосіб взаємодії з користувачем та налаштування параметрів пошуку. Алгоритм пошуку отримує вибрані параметри, взаємодіє з сервером для отримання статистики та популярних відповідей, та будує дерево ходів з найкращими відповідями. Сервер містить базу партій та надає API для взаємодії з алгоритмом пошуку.

Використання виконуваного файлу з графічним інтерфейсом було обрано з причини його незалежності, швидкодії, кращого контролю над функціональністю та зручності для кінцевого користувача. Користувач може просто запустити програму на своєму комп'ютері без потреби установки додаткових залежностей чи

середовищ виконання.

У схемах та специфікації функцій детально розглянуто усі процеси розробленого програмного забезпечення.

					ІС91.180БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		40

5 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

5.1 Керівництво користувача

Перед початком роботи з програмою повинен бути запущений сервер обробки партій з режимом доступу <http://localhost:9002/lichess>. Рекомендовано користуватися сервером `lila-openingexplorer`, що знаходиться у відкритому доступі. Посилання на нього можна знайти у переліку використаних джерел за номером [1]. На GitHub сторінці за посиланням можна знайти інструкцію по встановленню та імпорту баз даних партій.

Користувач, щоб запустити систему, повинен відкрити виконуваний файл. Тоді він потрапляє на головне вікно програми, що зображено на рисунку 5.1. На ньому представлені різні параметри виконання пошуку та поля для заповнення значень цих параметрів. Розглянемо кожний з параметрів.

Параметри позиції для початку пошуку та кольору, який має робити хід першим у заданій позиції, надають додаткові можливості користувачу. З ними можна провести пошук у варіації, що цікавить користувача. Якщо такої цілі не стоїть, за замовченням ці параметри пропонують стартову позицію, що не ускладнить користування програмою.

Параметр кольору для пошуку представлено у вигляді простого прапорця. Мати дебютний репертуар для кожного з кольорів – рівноважливо, тому цей параметр є одним з основних.

Параметр глибини вказує на останній ступінь глибини дерева ходів. Від нього залежатиме тривалість виконання алгоритму та номер ходу, далі якого система не буде більше поглиблюватися. Якщо колір, який має робити хід першим, збігається з кольором, для якого потрібно виконати пошук, то буде виконано ще один темп – тобто хід тільки одного кольору. Зі збільшенням глибини, результуючі дебютні репертуари мають тільки покращуватися, проте водночас збільшиться час виконання алгоритму та розмір дерева.

					ІС91.180БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		41

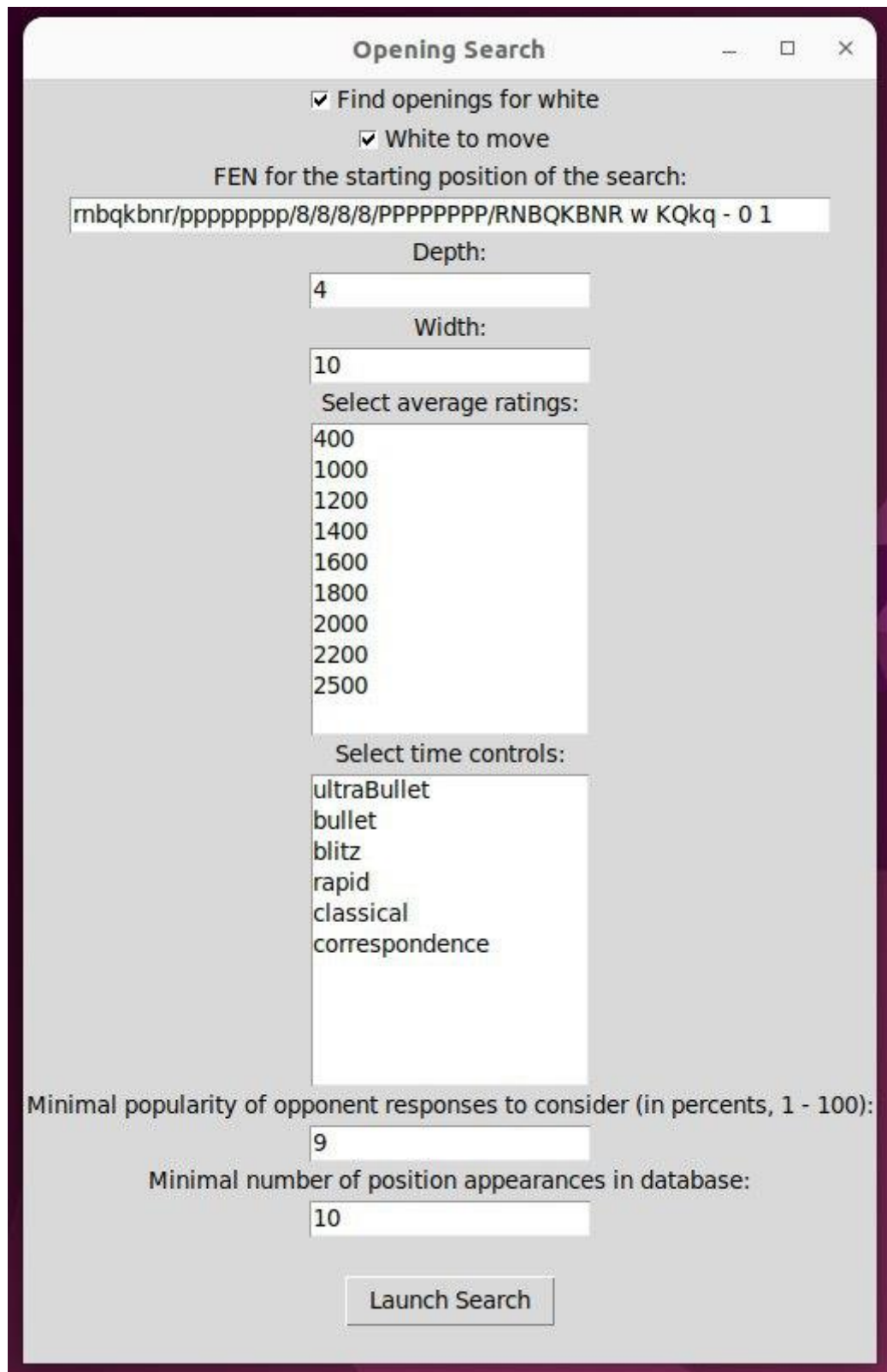


Рисунок 5.1 – Головне вікно системи з параметрами пошуку

Параметр ширини пошуку вказує кількість варіантів ходів, які будуть розглядатися для кожної позиції, де має ходити обраний колір. Це налаштування

дозволить пришвидшити виконання алгоритму використовуючи колективний досвід гравців, чиї партії аналізуються. Не рекомендується вказувати замале значення цього параметру.

Параметр середніх рейтингів гравців дозволить підібрати дебюти, які матимуть найбільший сенс для користувача. Таким чином рекомендовані дебюти повинні бути не заскладними для користувача, що підтверджується результатами інших гравців того ж рівня. Також цей параметр надасть представлення які відповіді суперників зустрічатимуться найчастіше та підготує до них користувача. Для випадків, коли користувачу буде складно дати перевагу тільки одній з представлених категорій рейтингів, було реалізовано можливість вибору декількох варіантів.

Параметр фільтрації партій по часовим контролям дає можливість обрати цікаві користувачу обмеження часу з переліку доступних. Перелік налічує усі найпопулярніші контролю часу.

Параметр мінімальної популярності для відповідей суперника для розглядання є ще одним з інструментів для забезпечення практичності програми. Так як підготуватися до кожного можливого ходу зі сторони суперника неможливо, кожний користувач зможе визначити для самого себе наскільки часто повинна зустрічатися та чи інша варіація, щоб її було резонно досліджувати. Замалі значення цього параметру можуть сильно сповільнити виконання алгоритму та сильно збільшити розміри результуючих дерев ходів, що не є рекомендованим.

Параметр мінімальної кількості зустрічей позиції у базі даних дозволяє користувачу вказати скільки кожна позиція дерева має мінімально налічувати партій. Користувач може вказати вибірки яких розмірів він вважатиме достатньо статистично обґрунтованими.

Після того, як користувач закінчив з виставленням параметрів пошуку, він може запустити алгоритм за натиском кнопки вниз. Поки алгоритм виконуватиметься, користувач буде бачити відповідне повідомлення.

По закінченню, відкривається вікно з результатом. Вигляд вікна зображено

на рисунку 5.2. У вікні є інформація, що результат було записано до файлу з розширенням .pgn та збережено до папки results. Ця папка з записаними до неї результатами пошуку може виглядати так, як на рисунку 5.3. Також надається можливість побачити та скопіювати результат з текстового поля чи за натиском кнопки копіювання.

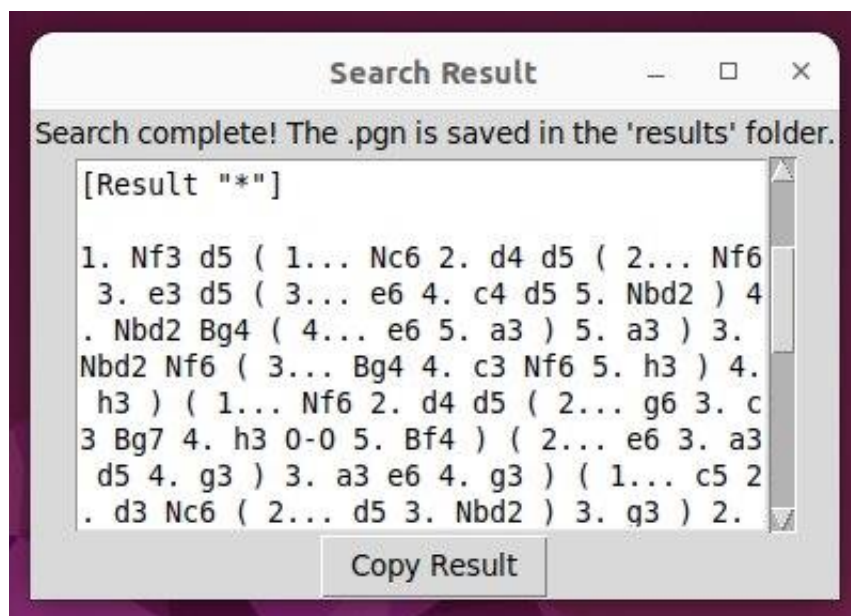


Рисунок 5.2 – Вікно з результатом пошуку

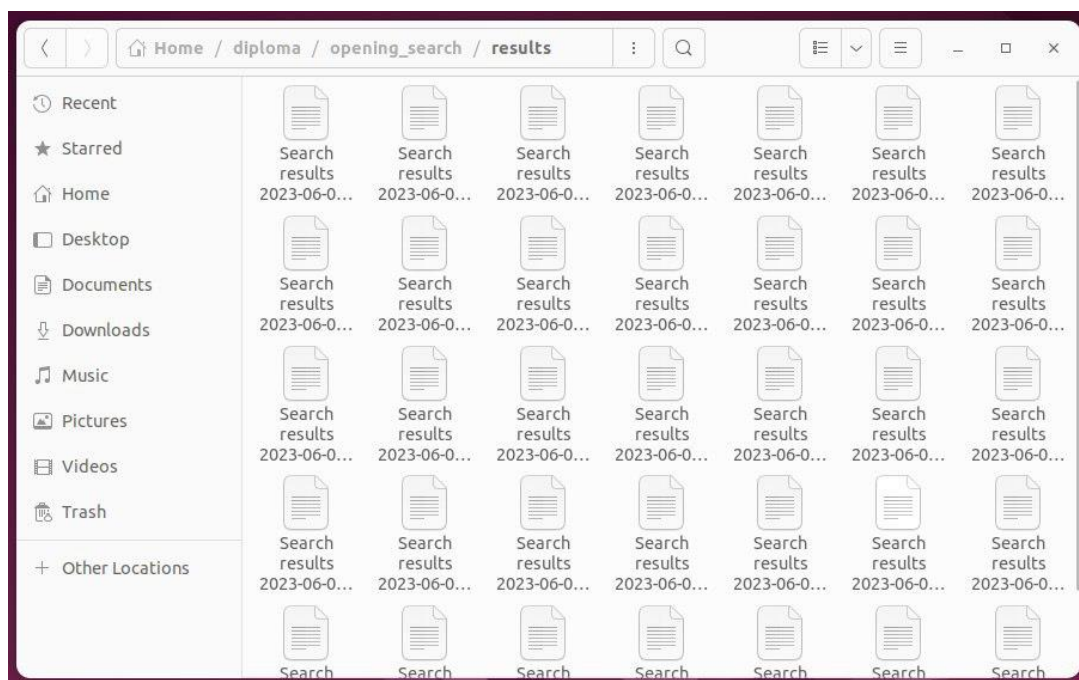


Рисунок 5.3 – Файли результатів пошуку, що зберігаються до папки results

Результат можна проглянути прямо в текстовому форматі чи імпортувати у будь-який шаховий графічний користувацький інтерфейс (приклад на рисунку 5.4). Користувачу постає повний шаховий дебютний репертуар для обраного кольору, що відповідає усім визначеним ним параметрам. Далі алгоритм можливо виконати повторно, якщо цікавить результат для інших параметрів пошуку.

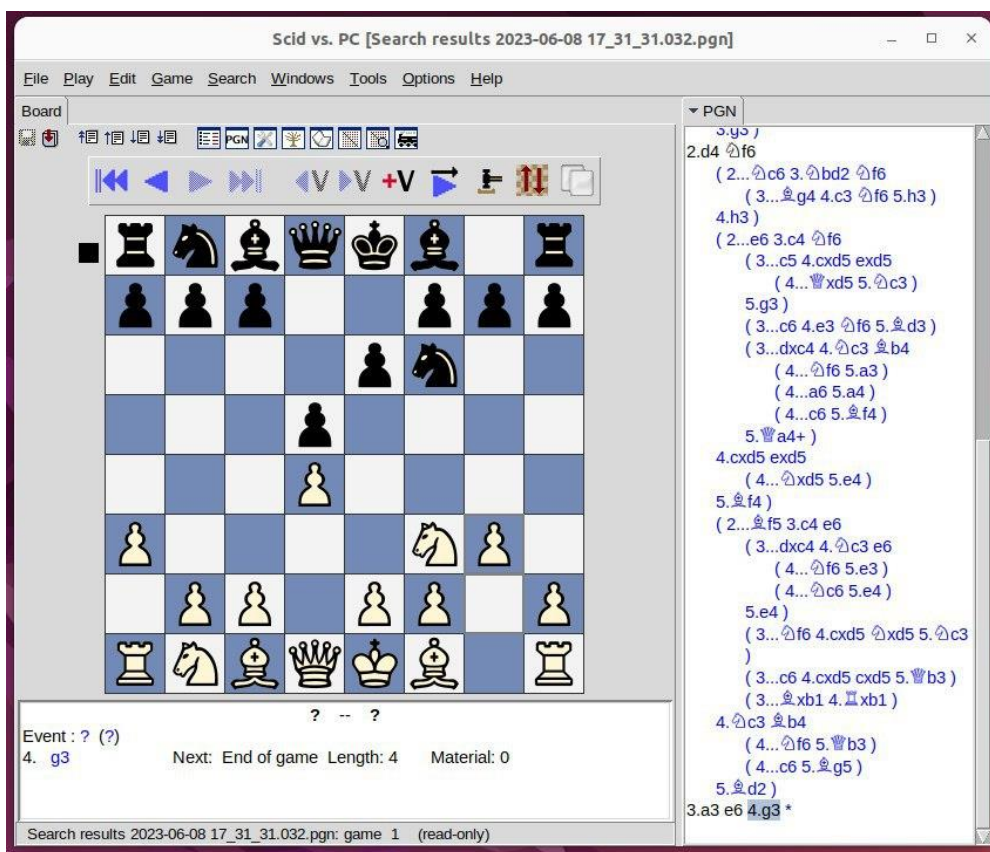


Рисунок 5.4 – Перегляд отриманого дерева ходів у шаховому графічному користувацькому інтерфейсі Scid vs. PC

5.2 Випробування програмного продукту

В цьому підрозділі описано тести та порядок тестування готового продукту для порівняння відповідності програмного забезпечення до задач, які були поставлені функціональним вимогам, розписаними у технічному завданні.

5.2.1 Мета випробувань

Метою випробувань – є перевірка відповідності функцій системи пошуку найкращих дебютів вимогам технічного завдання.

5.2.2 Загальні положення

Випробування проводяться на основі наступних документів:

- ГОСТ 34.603–92. Інформаційна технологія. Види випробувань автоматизованих систем;
- ГОСТ РД 50-34.698-90. Автоматизовані системи вимог до змісту документів.

5.2.3 Результати випробувань

Програмний продукт пройшов тестування на відповідність функціональним вимогам, а також на правильність його функціонування. Нижче наведено результати та зміст проведеного тестування. Тести включають в себе перевірку правильності роботи кожного параметра пошуку, коректність роботи інтерфейсу, а також стабільність та повноту виведення результатів пошуку, що продемонстровано у таблицях 5.1 – 5.9.

Таблиця 5.1 – Тестування коректності роботи параметрів початкової позиції пошуку та кольору, що має робити перший хід

Назва	Значення
Тест	Коректність роботи параметрів початкової позиції пошуку та кольору, що має робити перший хід
Початковий стан системи	Відкрита головна сторінка системи з формою пошуку дебютів. Вказані

Назва	Значення
	значення для параметрів середніх рейтингів та часових контролів.
Дія	Задаємо значення відносно популярної (сума змінних white, draws та black у відповіді сервера на запит з параметрами початку пошуку повинна бути не менше 5000), не початкової позиції до параметрів, що тестуються. Перевіряємо чи зберігаються зміни при діях з іншими віджетами. Тиснемо кнопку пошуку.
Очікуваний результат	Зміни при дії з іншими віджетами зберігаються. Система після виконання пошуку відкриває вікно з результатами пошуку. Дерево ходів має легальні продовження до заданої позиції початку пошуку. Продовження починаються з заданого кольору, що має робити перший хід.
Фактичний результат співпадає з очікуваним	Так

Таблиця 5.2 – Тестування коректності роботи параметру кольору, для якого здійснюється пошук

Назва	Значення
Тест	Коректність роботи параметру кольору, для якого здійснюється пошук
Початковий стан системи	Відкрита головна сторінка системи з формою пошуку дебютів. Вказані

Назва	Значення
	значення для параметрів середніх рейтингів та часових контролів.
Дія	Тиснемо кнопку пошуку. Після виконання пошуку на параметрах по замовчанню змінюємо значення параметру кольору, для якого здійснюється пошук. Перевіряємо чи зберігаються зміни при діях з іншими віджетами. Натискаємо кнопку пошуку. Порівнюємо отримані результати.
Очікуваний результат	Перший пошук здійснюється і дає результат. Зміни при дії з іншими віджетами зберігаються. Другий пошук здійснюється і дає результат. При порівнянні, дерева налічують різні варіації. Дерева мають ходи різних кольорів наприкінці варіацій.
Фактичний результат співпадає з очікуваним	Так

Таблиця 5.3 – Тестування коректності роботи параметру глибини пошуку

Назва	Значення
Тест	Коректність роботи параметру глибини пошуку
Початковий стан системи	Відкрита головна сторінка системи з формою пошуку дебютів. Вказані значення для параметрів середніх рейтингів та часових контролів.

Назва	Значення
Дія	Тиснемо кнопку пошуку. Після виконання пошуку на параметрах по замовчанню зменшуємо значення параметру глибини на 1. Перевіряємо чи зберігаються зміни при діях з іншими віджетами. Натискаємо кнопку пошуку. Порівнюємо отримані результати.
Очікуваний результат	Перший пошук здійснюється і дає результат. Зміни при дії з іншими віджетами зберігаються. Другий пошук здійснюється і дає результат. Перше дерево має варіації, що налічують більш пізні ходи.
Фактичний результат співпадає з очікуваним	Так

Таблиця 5.4 – Тестування коректності роботи параметру ширини пошуку

Назва	Значення
Тест	Коректність роботи параметру ширини пошуку
Початковий стан системи	Відкрита головна сторінка системи з формою пошуку дебютів. Вказані значення для параметрів середніх рейтингів та часових контролів.
Дія	Тиснемо кнопку пошуку. Після виконання пошуку на параметрах по замовчанню зменшуємо значення параметру ширини до 1. Перевіряємо чи зберігаються зміни

Назва	Значення
	при діях з іншими віджетами. Натискаємо кнопку пошуку. Порівнюємо отримані результати.
Очікуваний результат	Перший пошук здійснюється і дає результат. Зміни при дії з іншими віджетами зберігаються. Другий пошук здійснюється і дає результат помітно швидше. Деревя відрізняються.
Фактичний результат співпадає з очікуваним	Так

Таблиця 5.5 – Тестування коректності роботи параметру середніх рейтингів

Назва	Значення
Тест	Коректність роботи параметру середніх рейтингів
Початковий стан системи	Відкрита головна сторінка системи з формою пошуку дебютів. Вказане значення для параметру часових контролів, не вказане значення для параметру середніх рейтингів.
Дія	Тиснемо кнопку пошуку. Задаємо значення для параметру середніх рейтингів. Перевіряємо чи зберігаються зміни при діях з іншими віджетами. Тиснемо кнопку пошуку. Змінюємо значення цього параметру. Натискаємо кнопку пошуку. Порівнюємо отримані результати.

Назва	Значення
Очікуваний результат	При першій спробі виконати пошук видається попередження, що не вибрано значення для параметру середніх рейтингів. Зміни при дії з іншими віджетами зберігаються. При другій спробі пошук здійснюється і дає результат. При третій спробі пошук здійснюється і дає результат. Дерева відрізняються.
Фактичний результат співпадає з очікуваним	Так

Таблиця 5.6 – Тестування коректності роботи параметру часових контролів

Назва	Значення
Тест	Коректність роботи параметру часових контролів
Початковий стан системи	Відкрита головна сторінка системи з формою пошуку дебютів. Не вказане значення для параметру часових контролів, вказане значення для параметру середніх рейтингів.
Дія	Тиснемо кнопку пошуку. Задаємо значення для параметру часових контролів. Перевіряємо чи зберігаються зміни при діях з іншими віджетами. Тиснемо кнопку пошуку. Змінюємо значення цього параметру. Натискаємо кнопку пошуку. Порівнюємо отримані

Назва	Значення
	результати.
Очікуваний результат	При першій спробі виконати пошук видається попередження, що не вибрано значення для параметру часових контролів. Зміни при дії з іншими віджетами зберігаються. При другій спробі пошук здійснюється і дає результат. При третій спробі пошук здійснюється і дає результат. Деревя відрізняються.
Фактичний результат співпадає з очікуваним	Так

Таблиця 5.7 – Тестування коректності роботи параметру мінімальної популярності відповідей суперника для розгляду

Назва	Значення
Тест	Коректність роботи параметру мінімальної популярності відповідей суперника для розгляду
Початковий стан системи	Відкрита головна сторінка системи з формою пошуку дебютів. Вказані значення для параметрів середніх рейтингів та часових контролів.
Дія	Задаємо значення 7 для параметру мінімальної популярності відповідей суперника для розгляду. Перевіряємо чи зберігаються зміни при діях з іншими віджетами. Тиснемо кнопку пошуку.

Назва	Значення
	Змінюємо значення цього параметру до 18. Натискаємо кнопку пошуку. Порівнюємо отримані результати.
Очікуваний результат	Зміни при дії з іншими віджетами зберігаються. Перший пошук здійснюється і дає результат. Другий пошук здійснюється і дає результат помітно швидше. Перше отримане дерево, в середньому, налічує більше варіантів ходів суперника, ніж друге.
Фактичний результат співпадає з очікуваним	Так

Таблиця 5.8 – Тестування коректності роботи параметру мінімальної кількості зустрічей партій у базі даних

Назва	Значення
Тест	Коректність роботи параметру мінімальної кількості зустрічей партій у базі даних
Початковий стан системи	Відкрита головна сторінка системи з формою пошуку дебютів. Вказані значення для параметрів середніх рейтингів та часових контролів.
Дія	Задаємо довільне значення для параметру мінімальної кількості зустрічей партій у базі даних. Перевіряємо чи зберігаються зміни при діях з іншими віджетами. Тиснемо кнопку пошуку.
Очікуваний результат	Зміни при дії з іншими віджетами

Назва	Значення
	зберігаються. Пошук здійснюється і дає результат. Позиція після останнього ходу у будь-якій варіації налічує не менше обраної кількості партій при ручній перевірці.
Фактичний результат співпадає з очікуваним	Так

Таблиця 5.9 – Тестування коректності роботи параметру мінімальної кількості зустрічей партій у базі даних

Назва	Значення
Тест	Коректність роботи методів виводу інформації
Початковий стан системи	Відкрита головна сторінка системи з формою пошуку дебютів. Вказані значення для параметрів середніх рейтингів та часових контролів.
Дія	Тиснемо кнопку пошуку. Після закінчення пошуку у вікні результату виділяємо та копіюємо текст. Зберігаємо його для подальшого порівняння. Копіюємо результат за допомогою кнопки копіювання та зберігаємо. Відкриваємо новий .pgn у папці results та порівнюємо його зміст з отриманим з двох попередніх методів.
Очікуваний результат	Файл формату .pgn з результатом пошуку створюється у папці results. Вікно

Назва	Значення
	результатів пошуку налічує в собі текстове поле з результатом та конпку для копіювання. При порівнянні усі отримані тексти збігаються.
Фактичний результат співпадає з очікуваним	Так

Висновки до розділу

У п'ятому розділі було створено керівництво користувача, що включає опис інтерфейсу, функціональних елементів та можливостей системи. Кожен параметр пошуку був розглянутий з описом його призначення, надані рекомендації з використання та описані вимоги для успішної роботи. Також було детально описано функціонал кожного вікна програми й надані знімки екрану, що наглядно демонструють зовнішній вигляд системи.

У розділі, присвяченому тестуванню програмного продукту, було встановлено мету та надано загальні положення для проведення тестів. Також були проведені самі тести, де для кожного з них було описано виконані дії та протестовані результати. Було перевірено, що усі фактичні результати відповідають очікуваним.

ВИСНОВКИ

У пояснювальній записці було детально описано дипломний проєкт, що присвячений створенню системи для пошуку найкращих дебютів за статистикою гравців різних рівнів. Розглядається сучасний стан предметного середовища, описано процес діяльності. А також було розписано потребу у такій системі. Сформовано функціональні вимоги, описано функціональну модель, що відповідає за варіанти використання системи, розписано можливості програми, які будуть реалізовані.

Подібні програми були розглянуті, і їхні недоліки були виявлені в першому розділі.

Цілі та завдання розробки були визначені після вивчення та опису мети розробки.

У розділі «Інформаційне забезпечення» розглядаються вхідні та вихідні дані. Також розроблено та впроваджено в проєкт структуру інформаційного масиву. Також враховуються приклади вхідних і вихідних даних. У розділі «Математичне забезпечення», описано математичну постановку задачі. Побудовано та обґрунтовано спосіб вирішення задачі.

У розділі «Програмне та технічне забезпечення», були описані методи розробки проєкту, а також причини вибору кожної технології та їх переваги.

Була описана архітектура продукту та надано обґрунтування вибору цієї конкретної архітектури в підрозділі про архітектуру програмного забезпечення. Також було включено опис чотирьох схем: 2 схеми алгоритмів, діаграму компонентів та діаграму послідовності. Кожну з цих схем можна знайти в графічному матеріалі дипломного проєкту.

Після цього були надані специфікації функцій. Опис всіх функцій був поділений на декілька частин.

Заключним розділом роботи було створено керівництво користувача, яке містить опис інтерфейсу, усіх функціональних елементів і всіх можливостей системи. У кожному вікні є інструкції з використання, які пояснюють, як вибрати

					ІС91.180БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		56

параметри пошуку, запусити пошук і переглянути результати.

Мета і загальні вимоги до тестів, а також самі тести описані в підрозділі про тестування програмного продукту.

Описано дії, які були перевірені та виконані під час кожного тесту. Було перевірено, що фактичний результат відповідає очікуваному.

					ІС91.180БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		57

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Chess.com Reaches 100 Million Members! URL: <https://www.chess.com/article/view/chesscom-reaches-100-million-members> (дата звернення: 24.04.2023).

2. Сервер-дослідник дебютів «lila-openingexplorer». URL: <https://github.com/lichess-org/lila-openingexplorer> (дата звернення: 31.04.2023).

3. Офіційний сайт Stockfish. URL: <https://stockfishchess.org> (дата звернення: 31.04.2023).

4. John Shaw. The King's Gambit, Quality Chess, 2013, 680 p.

5. Portable Game Notation Specification and Implementation Guide. URL: <https://www.chessclub.com/help/PGN-spec> (дата звернення: 10.05.2023).

6. FEN Made Easy. URL: <https://www.chessgames.com/fenhelp.html> (дата звернення: 10.05.2023).

7. Документація для Stockfish. URL: <https://github.com/official-stockfish/Stockfish/wiki/Commands> (дата звернення: 11.05.2023).

8. HTTP request methods. URL: <https://developer.mozilla.org/enUS/docs/Web/HTTP/Methods> (дата звернення: 22.05.2023).

9. ГОСТ 19.701-90. Unified system for program documentation. Data, program and system flowcharts, program network charts and system resources charts. Documentation symbols and conventions for flowcharting.

10. ГОСТ 19.003-80. United system for program documentation. Graphical flowchart symbols.

11. What is Sequence Diagram? URL: <https://www.visual-paradigm.com/guide/uml-unifiedmodeling-language/what-is-sequence-diagram> (дата звернення: 25.05.2023).

12. What is Component Diagram? URL: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-component-diagram> (дата звернення: 25.05.2023).

					ІС91.180БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		58

13. Документація Python по роботі з JSON: json — JSON encoder and decoder. URL: <https://docs.python.org/3/library/json.html> (дата звернення: 16.05.2023).
14. Документація бібліотеки python-chess. URL: <https://python-chess.readthedocs.io/en/latest> (дата звернення: 17.05.2023).
15. Документація бібліотеки requests. URL: <https://requests.readthedocs.io/en/latest> (дата звернення: 17.05.2023).
16. Документація бібліотеки tkinter. URL: <https://docs.python.org/3/library/tk.html> (дата звернення: 23.05.2023).
17. Lichess.org API reference. URL: <https://lichess.org/api> (дата звернення: 22.05.2023).
18. Understanding Recursion Using Python. URL: <https://understanding-recursion.readthedocs.io/en/latest> (дата звернення: 22.05.2023).
19. Відкрита база даних партій Lichess. URL: <https://database.lichess.org> (дата звернення: 26.04.2023).
20. Савченко А. С., Синельников О. О. Методи та системи штучного інтелекту, Київ: НАУ, 2017. 190 с.
21. Methods of Search. URL: <http://www.cse.unsw.edu.au/~billw/Justsearch.html> (дата звернення: 22.05.2023).
22. Stefan Edelkamp, Stefan Schrödl. Heuristic search: theory and applications, Morgan Kaufmann Publishers, 2012, 712 p.
23. What is Constrained Optimisation? URL: <https://www.baeldung.com/cs/constrained-optimization> (дата звернення: 22.05.2023).
24. Richard E. Korf. Depth-first iterative-deepening: An optimal admissible tree search, 1985.
25. Dimitri P. Bertsekas. Constrained Optimization and Lagrange Multiplier Methods, Academic Press, 1982, 400 p.
26. Philip R. Bevington. Data Reduction and Error Analysis for the Physical Sciences, 1969.
27. Трохимчук Р. М. ТЕОРІЯ ГРАФІВ Навчальний посібник для

студентів факультету кібернетики, Київ: РВЦ «Київський університет», 1998, 43 с.

28. Зиков О. О. Енциклопедія кібернетики, 1973, 584 с.

					ІС91.180БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		60

