

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ГРИ В ШАХИ
SOFTWARE FOR CHESS GAMES

Виконав(ла): студент(ка) 4 курсу, групи КНТ-229

Спеціальності 122 Комп'ютерні науки

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Комп'ютерні науки

КАБАНЕНКО Є.О.

(ПРИЗВИЩЕ та ініціали)

Керівник ОЛІЙНИК А.О.

(ПРИЗВИЩЕ та ініціали)

Рецензент КОЦУР М.І.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет КНТ
Кафедра програмних засобів
Ступінь вищої освіти бакалавр
Спеціальність 122 Комп'ютерні науки
(код і найменування)

Освітня програма (спеціалізація) Комп'ютерні науки
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
“ ” 2023 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

КАБАНЕНКА Євгена Олександровича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Програмне забезпечення для гри в шахи. Software for Chess Games

керівник проєкту (роботи) д.т.н., професор, ОЛІЙНИК Андрій Олександрович,
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “ 04 ” квітня 2023 року № 87

2. Строк подання студентом проєкту (роботи) 01 червня 2023 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Матеріали і методи. 3. Опис програми. 4. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів) _____

Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРІЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	ОЛІЙНИК А.О., професор		
Нормоконтроль	ЛИПОВЕЦЬ М.В., асистент		

7. Дата видачі завдання “4” квітня 2023 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки.	2 тиждень	Розділ 2
4	Розробка структури програми.	2 тиждень	Розділ 3
5	Розробка програми.	3-4 тижні	Розділи 3, 4
6	Тестування та експериментальне дослідження програмного забезпечення.	5 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	6 тиждень	Додатки
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка)

(підпис) Євген КАБАНЕНКО
(Імя ПРІЗВИЩЕ)

Керівник проєкту (роботи)

(підпис) Андрій ОЛІЙНИК
(Імя ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра:
86 с., 8 табл., 32 рис., 3 дод., 17 джерел.

С#, КОМП'ЮТЕРНА ГРА, БАЗА ДАНИХ, КЛІЄНТ, ПРОГРАМНЕ
ЗАБЕЗПЕЧЕННЯ, СЕРВЕР.

Об'єкт дослідження – процес розробки програмного забезпечення для гри в шахи.

Предмет дослідження – програмні засоби для гри в шахи.

Метою роботи є розробка програмного забезпечення для гри в шахи.

Матеріали, методи та технічні засоби: мова програмування С#, середовище розробки Visual Studio, фреймворк .NET Framework, система керування базами даних Sql Server Management Studio, ігровий двигун Unity3d.

Результати. Розроблено програмне забезпечення для гри в шахи користувачів локальної мережі за допомогою мови програмування С#, що дозволяє забезпечити підтримку мережевої гри між декількома учасниками.

Практична цінність роботи полягає у розробці програмного забезпечення для підтримки процесу гри в шахи.

Висновки. Виконано проектування програмного забезпечення для гри в шахи. Розроблено програмне забезпечення для гри в шахи користувачів локальної мережі. Здійснено тестування розробленого програмного забезпечення для гри в шахи користувачів локальної мережі.

Галузь використання – ігрові застосунки.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 86 pages, 8 tables, 32 figures, 8 tables, 3 appendixes, 17 sources.

C#, COMPUTER GAME, DATABASE, CLIENT, SOFTWARE, SERVER.

The object of research is the process of developing chess game software.

The subject of the research is software for playing chess.

The purpose of this work is to develop the software for playing chess.

Materials, methods and technical tools: C# programming language, Visual Studio development environment, .NET Framework, Sql Server Management Studio database management system, Unity3d game engine.

Results. The software tools for chess game of local network users using C# programming language, which allows to support network game between several participants have been developed.

The practical value of the work lies in the development of software for playing chess.

Conclusions. The software for chess game has been designed. The software for playing chess for local network users has been developed. The developed software for chess game has been tested.

Scope of use – game applications.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок	8
Вступ	9
1 Аналіз предметної області	10
1.1 Поняття про комп'ютерні ігри та їх розробку	10
1.2 Дослідження програмних засобів для гри в шахи	24
1.3 Висновки за розділом 1	33
2 Матеріали і методи	35
2.1 Вибір мови програмування та інших технологій розробки	35
2.2 Вибір середовища розробки для створення програмного забезпечення для гри в шахи	37
2.3 Висновки за розділом 2	40
3 Опис програми	42
3.1 Структура процесів програмного забезпечення для гри в шахи	42
3.2 Опис основних модулів та компонентів програмного забезпечення для гри в шахи	43
3.3 База даних програмного забезпечення для гри в шахи	45
3.4 Розробка графічного інтерфейсу програмного забезпечення для гри в шахи	47
3.5 Функціонування програмного забезпечення для гри в шахи	48
3.6 Висновки за розділом 3	49
4 Експлуатація, тестування та експериментальне дослідження програми	51
4.1 Призначення й умови застосування програми	51
4.2 Характеристики програми та її встановлення	52
4.3 Інструкція по експлуатації програми	52
4.3.1 Звернення до програми	52
4.3.2 Вхідні й вихідні дані	52
4.3.3 Повідомлення	53
4.4 Виконання програмного забезпечення для гри в шахи	53

4.5 Тестування програмного забезпечення для гри в шахи.....	55
4.6 Висновки за розділом 4	56
Висновки	57
Перелік джерел посилань	60
Додаток А Технічне завдання	62
Додаток Б Фрагмент тексту програми	66
Додаток В Слайди презентації	78

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

ASP.NET – active server pages for .NET;

CLI – call level interface;

IDE – інтегрована середовище розробки;

SQL – structured query language;

БД – база даних;

ПЗ – програмне забезпечення;

ПК – персональний комп'ютер;

СКБД – систем керування базою даних.

ВСТУП

У наш час широко використовуються комп'ютерні ігри, що являють собою програмне забезпечення для розважання чи навчання людей. Сьогодні комп'ютерні ігри – це великий бізнес, існують мільйони різних комп'ютерних ігор, якими користуються люди будь-якого віку. Функціональність гри передбачає наявність декількох атрибутів, включаючи умови перемоги та поразки, перешкоди та нагороди. Гравці також повинні мати можливість керувати гравцем за допомогою пристрою введення, наприклад клавіатури та миші або контролера. Крім цих мінімальних вимог, сучасні ігри зазвичай оснащені іншими функціями та механізмами, щоб зробити гру більш захоплюючою [1]-[5].

Розумові логічні комп'ютерні ігри дозволяють користувачу не тільки отримати задоволення від гри, а і покращити власні вміння приймати вірні та ефективні рішення у складних ситуаціях.

У дипломній кваліфікаційній роботі бакалавра розв'язується актуальне завдання розробки програмного забезпечення для підтримки процесу гри в шахи.

Для досягнення поставленої мети у кваліфікаційній роботі бакалавра необхідно розв'язати такі задачі:

- виконати аналіз предметної області та програмних засобів для гри в шахи;
- здійснити проектування програмного забезпечення для гри в шахи;
- створити програмне забезпечення для підтримки процесу гри в шахи;
- виконати тестування розробленого програмного забезпечення для гри в шахи.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Поняття про комп'ютерні ігри та їх розробку

Комп'ютерні ігри являють собою програмне забезпечення для розважання чи навчання людей. Індустрія комп'ютерних ігор у наш час являє собою великий бізнес, існують мільйони різних комп'ютерних ігор, якими користуються люди будь-якого віку. Функціональність комп'ютерної гри передбачає наявність декількох атрибутів, включаючи умови перемоги та поразки, перешкоди та нагороди. Гравці також повинні мати можливість керувати гравцем за допомогою пристрою введення, наприклад клавіатури та миші або контролера. Крім цих мінімальних вимог, сучасні ігри зазвичай оснащені іншими функціями та механізмами, щоб зробити гру більш захоплюючою [1]-[6].

Часто створені окремою командою розробників програмні ресурси для гри також є важливою частиною ігрового дизайну. Середовища, персонажі, об'єкти та анімація повинні бути створені, щоб надати грі узгодженого вигляду та відчуття. У комп'ютерних іграх графіку часто створюють за допомогою редактора зображень. Якщо гра тривимірна, використовується програмне забезпечення для моделювання. У будь-якому випадку програмне забезпечення для анімації використовується для імітації руху. Аудіо створюється за допомогою програми цифрової аудіо станції і може залучати відповідних фахівців для створення звукових ефектів, озвучення персонажів і музичні композиції для саундтреку гри [1]-[6].

Коли програмне забезпечення перебуває в робочому стані, гра вважається альфа-стадією. На цьому етапі активні об'єкти гри, зокрема персонажі, інші об'єкти та середовище, часто є неповними, а тимчасові заповнювачі використовуються для перевірки та вдосконалення механіки гри. Після впровадження цих активів гра вважається бета-версією, доки гру не буде завершено. Тестування проводиться на цих етапах розробки, щоб гарантувати, що гра принесе користь будь-якому гравцеві [1]-[6].

Сьогодні ігри створюються практично для будь-яких пристроїв. Деякі останні ігри є кросплатформними, що дозволяє гравцям на будь-якій платформі приєднуватися один до одного. Комп'ютерні ігри випускаються для функціонування на персональних комп'ютерах, більшість комп'ютерних ігор грають у Windows, вони також можуть бути доступні на комп'ютерах під керуванням macOS або Linux. Консольні ігри випускаються для фірмових розважальних консолей, таких як Xbox, PlayStation, Nintendo та ін. Мобільні ігри випускаються на смартфонах і планшетах Android і iOS. Мобільні ігри зазвичай безкоштовні, але включають мікротранзакції та рекламу для отримання прибутку [1]-[6].

Завдяки своїй розповсюдженості Інтернет-ігри є ефективним способом провести час, поспілкуватися з друзями та, можливо, знайти нових друзів. Існує велика різноманітність ігор, які з'єднують гравців один з одним через Інтернет. Однак, якщо ви хочете потрапити в онлайн-ігри, варто врахувати кілька речей. У наступних розділах описано, що потрібно для гри через Інтернет [1]-[6].

До функціонування комп'ютерних ігор на стороні клієнта (користувача) часто висувуються певні мінімальні вимоги до апаратного забезпечення користувача [1]-[6].

Комп'ютерні ігри – це програми, які дозволяють гравцеві взаємодіяти з віртуальним ігровим середовищем для розваги. Існує багато типів комп'ютерних ігор, починаючи від традиційних карткових ігор і закінчуючи більш складними відеоіграми, такими як рольові та пригодницькі ігри [1]-[6].

Хоча комп'ютерні ігри в основному забезпечують розважальні функції, вони також покращують координацію рук та очей і навички вирішення проблем. Кожна гра має власну стратегію, екшн і фантазію, які роблять кожну гру унікальною та цікавою. Загалом можна класифікувати комп'ютерні ігри на такі типи: карткові ігри, настільні ігри, головоломки, лабіринти, бійки, бойовики, пригоди, рольові ігри, стратегії, спортивні ігри та симулятори. Однак ця класифікація є нечітким поняттям, оскільки багато ігор є гібридами,

які належать до кількох класів. Наприклад, Doom можна класифікувати як гру в лабіринт або екшен, тоді як Monopoly можна класифікувати як настільну гру або стратегію [1]-[6].

Комп'ютерна гра, як правило, складається з наступних компонентів: пристрої та програмні компоненти для введення, ігрова логіка, підтримка графіки/звуку, засоби виведення гри та засоби підтримки мережі [1]-[6]:

- пристрої та програмні компоненти для введення. Користувачі взаємодіють з ігровою програмою через пристрої введення. Звичайні пристрої введення включають клавіатуру, мишу або джойстик [1]-[6];

- ігрова логіка реалізує ігрову логіку або ігровий код, який обробляє більшість основних механізмів гри. Загалом, перш ніж розробити логіку гри, слід спроектувати сюжетну лінію про те, як грають у гру та як гравці мають взаємодіяти. Проста фізика, мережева підтримка та анімація повинні бути сплановані. У деяких просунутих іграх на цьому етапі також реалізовано штучний інтелект і виявлення зіткнень [1]-[6];

- механізм візуалізації графіки – він має складний код для ефективної ідентифікації та візуалізації ігрових об'єктів і фону з двовимірної або тривимірної моделі оточення. Він підтримує перетворення об'єктів, які переміщуються, обертаються та масштабуються, коли це необхідно [1]-[6];

- графічні/звукові драйвери – графічні драйвери отримують запити від механізму рендерингу до графічної бібліотеки за допомогою API. Windows API і Microsoft FoundationClasses (MFC) забезпечують підтримку двовимірної графіки для комп'ютера. Для підтримки двовимірної та тривимірної графіки OpenGL і DirectX є двома найпопулярнішими графічними бібліотеками. DirectX також надає бібліотеки для підтримки музики та звуку [1]-[6];

- виведення гри – створена двовимірною або тривимірною графікою виводиться на дисплей. Згенерований звуковий ефект або музика виводяться на звукову карту [1]-[6];

– мережева підтримка – забезпечує підтримку мережевого протоколу, який дозволяє кільком користувачам у віддалених місцях грати та взаємодіяти в одному ігровому середовищі. У мережевому ігровому середовищі сервер потрібен для підтримки інформації, яку підтримує світ віртуальної гри, зв'язується з ігровими клієнтами, які використовуються гравцями для надання їм інформації про спільне середовище. Сервер також повинен синхронізувати інформацію та підтримувати послідовність сцен віртуального ігрового світу серед мережевих клієнтів [1]-[6].

Коли ігрова програма починає виконуватися, вона повинна спочатку ініціалізувати пам'ять, завантажити зображення та звукові файли, запустити графіку та налаштувати такі змінні, як бали. Після ініціалізації запускається логіка гри. Після завершення ігрового сеансу також потрібно прибирання для оновлення звукових ефектів, оновлення та відображення результатів, оновлення структур даних тощо. Користувач може завершити цей сеанс або повернутися до логіки гри, щоб почати гру знову [1]-[6].

Щоб розробити успішну комп'ютерну гру, розробнику важливо мати гарну сюжетну лінію разом із гарною двовимірною чи тривимірною графікою та звуковими ефектами, щоб комп'ютерна гра виглядала реалістичною. Сюжетну лінію слід розробити перед тим, як почати розробку гри. Щоб розробити сюжетну лінію та алгоритми функціонування комп'ютерної гри, програмісту потрібно визначити таке [1]-[6]:

– тип гри – необхідно класифікувати характер гри за однією з категорій. Різні типи ігор мають різні вимоги, наприклад, рольові ігри вимагають реалістичності графіки та звукових ефектів, тоді як деякі стратегічні ігри наголошують на стратегії, а не на реалістичності, і для таких ігор буде достатньо спрощеного відображення [1]-[6];

– мета гри – нам потрібно скажіть гравцеві, що робити, щоб досягти успіху в грі. Мета гри в космічні війни – збити якомога більше космічних загарбників [1]-[6];

– ефективність гравця в грі – нам потрібно вирішити, як оцінювати гравців за їхні результати. Це має вказувати, наскільки близько вони досягли мети гри. Загалом використовується загальний бал як показник ефективності гравця. Крім того, коли гравець досягає мети, також потрібно визначити, як винагородити гравця [1]-[6];

– правила гри – необхідно розповісти гравцям, як грати в гру. Необхідно пояснити інструкції щодо роботи з клавіатурою чи мишею, щоб грати в гру [1]-[6].

Візуальні ефекти графіки та спеціальні звукові ефекти важливі для створення сюжетної лінії та надання ілюстрацій, які змушують гравця відчувати себе частиною гри. Багато простих ігор покладаються лише на двовимірну графіку, яка малює точки, лінії та форми з заливкою, наприклад прямокутники та багатокутники на площині. Графічні функції, надані з Windows API (інтерфейс прикладного програмування) або бібліотеки Microsoft Foundation Class (MFC) у середовищі Windows, підтримують двовимірну графіку. Просунуті комп'ютерні ігри часто потребують роботи з тривимірними об'єктами [1]-[6].

Крім двовимірної та тривимірної графіки, відображення зображень і маніпуляції є ще одним важливим прийомом для розробки ігор. Розробник також може маніпулювати зображеннями за допомогою масштабування та обертання. Анімація зображень є ще одним популярним прийомом, який використовується в багатьох іграх. Windows API і бібліотека MFC надають функції для відображення зображень і керування ними [1]-[6].

Звук – ще один важливий елемент, завдяки якому комп'ютерна гра виглядає реалістично. Windows і бібліотека MFC підтримують лише генерацію лише одного тону за раз. Однак за допомогою додаткової звукової карти більшість комп'ютерних ігор можуть створювати складніші звукові ефекти. Для підтримки розробки ігор Microsoft DirectX надає підсистему DirectXAudio, яка підтримує музику та стереозвукові ефекти. Бібліотека OpenGL не підтримує звукові ефекти [1]-[6].

У наш час при реалізації комп'ютерної гри використовується два основних типи підтримки мультиплеєра – одноранговий та клієнт-серверний [1]-[6].

У одноранговому режимі кожен гравець одночасно є клієнтом і сервером. Однією з переваг такого підходу є те, що розробнику гри не потрібно витрачати гроші на обслуговування сервера. Недоліком є те, що нечесні гравці з серверними модулями в іграх отримують більше можливостей обдурити інших гравців. Деякі гравці можуть отримати ефекти безсмертя, здібності нескінченної шкоди та щось відмінне від початкової ідеї гри, просто змінивши мережевий пакет. Найчастіше однорангові ігри використовуються в іграх, призначених для соціальних мереж [1]-[6].

Система клієнт-сервер дуже складна та більш важка для реалізації. За допомогою цього типу підтримку багатокористувальницьких ігор клієнти не розмовляють один з одним і надсилають пакети на сервер, який перевіряє повідомлення клієнта. Рівень контролю користувача може коливатися від простого тесту до повної симуляції ігрового світу. Чим глибший рівень контролю, тим більше ресурсів потрібно серверу і тим менше шансів клієнтів на обман [1]-[6].

У більшості випадків ігровий сервер є віртуальною клієнтською машиною, тому існує поняття виділеного сервера. Це стандартний ігровий сервер без жодних функцій. Усі несерверні функції, такі як графіка, обробка вхідних даних і фізика, завантажуються з цього сервера [1]-[6].

Насправді цей сервер може лише приймати гравців і надавати канал для гри між ними. У більшості випадків вони намагаються контролювати ситуацію, тому намагаються зменшити навантаження на сервер. Насправді хороший господар повинен просто перевірити правила гри [1]-[6].

Комп'ютерні ігри, окрім консолей і смартфонів, виконуються на комп'ютерах, особливо це стосується високоякісних ігор. Перевагами комп'ютерних ігор є такі [1]-[6]:

– концентрація. Однією з істотних переваг комп'ютерних ігор є те, що вони значною мірою покращують концентрацію користувачів. Відеоігри оснащені різними наборами рівнів, кожен із змагальними завданнями. Щоб пройти ці рівні, гравці повинні володіти деякими надзвичайними навичками уважності, які також застосовуються в повсякденному житті [1]-[6];

– аналітичні здібності. Деякі відеоігри вимагають високих аналітичних навичок для вирішення проблем, що допомагає удосконалити навички логічного мислення, які в майбутньому можна застосовувати у реальному житті. Дослідження показали, що комп'ютерні ігри можуть ефективно знімати стрес [1]-[6];

– координація команди. Деякі онлайн-ігри та багатокористувацькі ігри потребують координації команди для досягнення мети. Гра за допомогою Інтернету та мікрофона може ефективно спілкуватися з іншими користувачами по всьому світу. Це не лише зробить людину кращим командним гравцем та надасть навичок *soft skills*, але й дасть можливість отримати нові контакти [1]-[6];

– прийняття рішень. Для просування місій у комп'ютерній грі гравцям потрібно приймати рішення за частки секунди. Це покращує присутність розумових навичок для швидкого прийняття рішень. Для цієї мети існують спеціально розроблені ігри для тренування мозку, які містять кілька форм стимуляції [1]-[6].

Недоліками комп'ютерних ігор є такі [1]-[6]:

– залежність, оскільки існують користувачі, які не контролюють себе і часто стають занадто залежними від комп'ютерних ігор [1]-[6];

– порушення соціалізації. Надаючи занадто багато значення комп'ютерним іграм, користувач може віддалитися від оточуючих, хоча й онлайн-ігри допомагають гравцям спілкуватися з людьми по всьому світу;

– негативний академічний вплив. Хоча й комп'ютерні ігри покращують концентрацію та навички прийняття рішень, вони також можуть становити

загрозу для навчання. Більший інтерес користувачів до комп'ютерних ігор може серйозно вплинути на їх освіту [1]-[6];

– турбота про здоров'я. Крім рухів руками, комп'ютерні ігри повністю позбавлені будь-яких фізичних навантажень. Отже, комп'ютерні ігри сприяють численним проблемам зі здоров'ям, таким як біль у плечах, спині та головний біль [1]-[6].

У наш час існує багато жанрів відеоігор, багато з яких є взаємопов'язаними. Проаналізуємо жанри комп'ютерних ігор [1]-[6].

Екшн ігри, до яких відносяться бойовики, роботи та стрілялки, є одними з найпопулярніших жанрів відеоігор за останні кілька років. З усіх жанрів відеоігор ці ігри є одними із найрізноманітнішими. У цих типах відеоігор користувач керує діями й часто стикається із реакційними або фізичними перешкодами [1]-[6].

Пригодницькі ігри являють собою квінтесенцію ігор про героїв. Вони вирізняються серед інших подібних жанрів відеоігор тим, що вони не зосереджені на боях чи протистоянні. Натомість користувачу доручено досліджувати та розкривати таємниці загальної історії та навколишнього середовища [1]-[6].

Замкові ігри можуть мати відображення в різних жанрах, від стратегічних оборонних ігор до звичайних будівельників замків. Щоб виділитися серед інших жанрів відеоігор, гра про замок має підкреслити місце користувача на троні, звідки він будує та розвиває свій замок, відкриває торговельні шляхи та покращує його захист [1]-[6].

У банері казуальних ігор жанрів відеоігор є багато категорій ігор. Однак усі вони об'єднані легким ігровим процесом, простим керуванням і чіткою системою прогресу [1]-[6].

Деякі з провідних жанрів відеоігор значною мірою спираються на тему та всі пов'язані з нею середовища. Це стосується середньовічних ігор, у яких можна побачити, як користувач комп'ютерної гри грає в старі часи з мечами, щитами, стрільбою з луків верхи на конях і битвами в полі [1]-[6].

Багатокористувацькі онлайн-ігри стали популярними з тих пір, як онлайн-ігри стали можливими завдяки пропускній здатності мережі. Зараз існує багато жанрів відеоігор, які завдячують своїм існуванням першим комп'ютерним іграм подібного жанру. Такі ігри охоплюють велику кількість піджанрів, а основним елементом є те, що в них грає онлайн величезна кількість людей одночасно. Королівська битва – один із найновіших типів відеоігор, які потрапили у жанр багатокористувацьких онлайн-ігор [1]-[6].

Багатокористувацькі ігри не обов'язково мають бути такими великими, як інші жанри комп'ютерних ігор, які покладаються на підключення до Інтернету. Багатокористувацькі ігри з сюжетом вимагають величезної аудиторії, тоді як багатокористувацька гра може проводити лише окремі ігри з десятима, чотирма або навіть двома гравцями. Користувач може опинитися в бойовій грі із суперником один на один або вирушити в космічну одісею з чотирма іншими в грі на виживання для кількох гравців [1]-[6].

Комп'ютерні ігри типу головоломок є одним із найстаріших і найулюбленіших жанрів відеоігор. Існує аргумент, що багато жанрів відеоігор насправді є іграми-головоломками. У цьому списку жанрів відеоігор можуть бути виділені ігри-головоломки як ті, які мають естетику головоломки. Розробнику при створенні таких ігор необхідно запрограмувати способи переміщення набірної дошки або всередині налаштування, щоб дійти до кінця, зі скиданням головоломки під час перезапуску, якщо користувач не впорається з грою. Ігри-головоломки часто вибирають люди, які хочуть розв'язати більш інтелектуальні завдання, а не ті, що керуються ігровими навичками [1]-[6].

Деякі комп'ютерні ігри по типу «гравець проти гравця» можуть бути у формі багатьох інших жанрів відеоігор, але всі вони виставляють користувача онлайн проти іншого гравця-людини, щоб змагатися. У іграх «гравець проти гравця» користувач може зіткнутися з одним гравцем, щоб битися зі своїми командами, або разом з іншими гравцями битися з іншою групою у подібній грі [1]-[6].

Ігри з роботами продовжують набувати все більшої популярності. Як і багато інших жанрів, ігри про роботів можуть бути віднесені до кількох ігрових жанрів. Ігри про роботів часто можна зустріти як стратегії чи дії [1]-[6].

Рольові комп'ютерні ігри є одним з найбільших з усіх жанрів відеоігор, рольові ігри мають величезні продажі. Такі ігри ставлять користувача у роль головного героя. З цього моменту користувачу потрібно досліджувати світ, виконувати квести, знаходити предмети, створювати, покращувати, будувати, битися та дійти до кінця однієї чи кількох сюжетних ниток [1]-[6]. Основою рольових ігор є система підвищення рівня через очки досвіду, а також механізми прийняття рішень [1]-[6].

Комп'ютерні ігри типу пасьянсів є доступними як у режимі офлайн із фізичною стопкою карток, так і як складніші, візуально привабливі ігри, настільки доступні та відомі, що всі користувачі комп'ютерів можуть в них грати [1]-[6].

Багато жанрів відеоігор мають купу підкатегорій, але стратегічні ігри насправді мають лише дві. Існують покрокові стратегічні ігри, де користувач рухається, а потім чекає, поки всі інші гравці або фракції зроблять свої кроки по порядку. Або є стратегічні ігри в реальному часі, в яких користувачу потрібно коригувати свою тактику та плани в момент, який зазвичай є битвою [1]-[6].

Ігри про виживання, які зазвичай використовують елементи RPG та пригодницьких ігор, останніми роками стали надзвичайно популярними. Зазвичай користувач потрапляє у все більш вороже середовище, маючи не більше ніж деякі несуттєві можливості. Потім користувачу потрібно знайти місце, де можна облаштуватися, здобувати ресурси, створювати інструменти, будувати захист і досліджувати світ навколо [1]-[6].

Військові ігри є одними з найпопулярніших жанрів бойовиків. Незалежно від того, чи відбуваються вони у сучасний період, в епоху

середньовіччя чи в якийсь невизначений час у майбутньому, військові ігри викликають дію та стратегію [1]-[6].

Жанр відеоігор на платформі (також відомий як ігри на платформі) зосереджується переважно на стрибках і скелелазінні. В платформних іграх персонаж користувача переміщується між місцевостями, платформами, драбинами та іншими головоломками, намагаючись дійти до кінця рівня живим. Двовимірні платформери є платформерами з боковою прокруткою та двовимірною графікою. Двовимірний платформер був дуже популярним жанром у двадцятому столітті, коли ігри дозволяли гравцям перестрибувати через перешкоди та простори на землі [1]-[6].

Жанр відеоігор шутер, який часто називають шутерами, є піджанром відеоігор-екшенів, у яких головна мета гравця полягає у перемозі персонажів-супротивників, використовуючи доступну зброю. Ця зброя зазвичай є гвинтівками або іншою далекобійною зброєю, яка може бути оснащена аксесуарами, такими як гранати для непрямих атак, бронежилети для додаткового захисту або телескопічні приціли для зміни поведінки зброї. Оновлення, які покращують зброю, боєприпаси, броню та здоров'я гравця, є звичайними ресурсами багатьох шутерів. В ізольованих одиночних і багатокористувацьких мережевих режимах шутер перевіряє просторове усвідомлення, рефлексивність та швидкість гравця. Стрілялки поділяються на кілька піджанрів, усі з яких засновані на аватарах, що борються з супротивниками, керованими кодом, або інших аватарах, керованих іншими гравцями [1]-[6].

Файтинг (іноді званий екшн) є типом відеоігри, у якій двоє (чи більше) гравців борються один проти одного. Блокування, боротьба, протидія та зв'язування є поширеними бойовими прийомами, які використовуються в бойових іграх. У більшості битв персонажі використовують рукопашний бій, який зазвичай передбачає певну форму бойових мистецтв. Жанр бойових ігор є спорідненим, але іншим жанром, у якому один або кілька персонажів протистоять численним ворогам, керованим комп'ютером. Така гра передбачає битву між двома бійцями, які надмірно використовують бойові

мистецтва. Зазвичай він присвячений єдиноборствам або єдиноборствам, але деякі різновиди містять зброю. Бійці на екрані часто дивляться збоку, і навіть у тривимірні бойові ігри часто грають у двовимірній площині руху. У той час як деякі ігри дозволяють гравцям подорожувати між паралельними площинами руху, більшість ігор обмежують персонажів рухами вліво та вправо та стрибками. Останні ігри, як правило, є тривірними і дозволяють прокручувати збоку, але грають як у двовимірному просторі [1]-[6].

На відміну від більшості екшн-ігор, стелс-ігри вимагають від гравця за будь-яку ціну уникати попередження зловмисника. Основні елементи ігрового процесу сучасних стелс-ігор — уникати конфліктів, створювати якомога менше шуму та стріляти по ворогам із тіні та ззаду. Досягнення цілей, непомічених ворогами, іноді називають привидами, є популярною стратегією в іграх зі стелс-іграми. Єдиний спосіб завершити гру — уникнути виявлення, але ви можете використовувати кілька шляхів або стилів гри, щоб досягти своєї мети. Гравці можуть ховатися за предметами або в тіні та бити або пробігати повз ворогів, які дивляться в протилежний бік. Якщо зловмисники помітять, поки вороги не перестануть їх шукати, гравець може сховатися. У результаті планування, проб і помилок стають вирішальними. Деякі стелс-ігри більше зосереджуються на фізичних бойових здібностях під час виявлення гравця. У деяких іграх ви можете вбивати своїх ворогів або просто бити їх. Навіть якщо привиди не обов'язкові або погано підтримуються грою, гравці можуть намагатися уникати бою з моральних причин або щоб продемонструвати свої навички. Ці ігри називалися стелс-іграми на початку жанру стелс [1]-[6].

Ігри на виживання вважаються розширенням загальної теми відеоігор, де персонаж гравця опиниться на мілині або ізольований і повинен працювати сам, щоб вижити та досягти своїх цілей. Як правило, гравець починає з мінімальним обладнанням і повинен намагатися вижити якомога довше, будуючи інструменти, зброю, бункери та ресурси. Ігри на виживання призначені для гри в багатокористувацькому режимі, де постійний світ

зазвичай розташований на ігровому сервері, а також для гри в одиночному режимі [1]-[6].

Отже, у наш час є понад ста жанрів відеоігор, у яких користувач може грати та досліджувати, включаючи всі піджанри та додаткові жанри. Багато ігор підходять до кількох жанрів відеоігор, дозволяючи отримати найкращі аспекти від обох і насолодитися незабутніми враженнями.

Список популярних жанрів відеоігор включає стрілялки, бойовики, пригодницькі ігри, рольові ігри, багатокористувальницькі онлайн ігри, спортивні симулятори, гоночні ігри і казуальні ігри. Звичайно, у кожному з цих жанрів відеоігор також будуть популярні піджанри. До найпопулярніших жанрів відеоігор можна віднести такі: екшн-ігри (зокрема, файтинг і платформери), пригодницькі ігри (включно з пригодницькими іграми), рольові комп'ютерні ігри (включаючи багатокористувальницькі онлайн ігри), стратегічні ігри, симулятори (включно зі спортовними симуляторами).

Гра у шахи є однією з найстаріших настільних ігор у світі, а також одна з найпопулярніших. Незалежно від того, чи хочете ви навчитися грати в шахи, щоб грати в звичайні ігри з друзями чи брати участь у змаганнях, у цій статті ви можете дізнатися всю необхідну інформацію про королівську гру [7].

Шахи є настільною грою між двома гравцями, що імітує війну між двома королівствами. Це одна з найпопулярніших ігор у світі. Мільйони людей грають заради розваги та змагання [7].

Шахи – це ігрова гра без прихованої інформації. З цієї причини елемент удачі не включається в гру. Шахи – складна гра з багатьма іграми і правилами. Щоб грати в шахи, гравцям потрібен принаймні один набір шахів. Шаховий годинник також корисний, коли гравці вирішують зіграти в партію на час і гравці використовують таблицю лідерів для запису своїх ходів [7].

Шахи складаються з двох окремих армій, кожна з яких складається з восьми пішок, двох коней, двох слонів, двох турів, ферзя та короля. Гравці можуть сортувати свої війська за кольором за допомогою чорних і сірих одиниць. Незалежно від виробу фактичного кольору, світла сторона

називається білою, а темна – чорною, що нагадує слонову кістку та чорне дерево, які використовувалися в минулому [7].

На шахівниці 64 поля, розташовані вісім на вісім. Панель розділена на частини (рядки) і стовпці [7].

Існує шість видів шахових фігур. Це пішак, кінь, слон, тура, ферзь і король. Кожна з цих фігур відрізняється і має свою цінність. Пішаки оцінюються в одне значення і не вважається одиницею. Лицарі та єпископи варті трьох очок, але лицарі та імператори часто вважають єпископів трохи більш цінними. Тури мають п'ять очок, а королі — дев'ять очок. Тури та ферзі – найважливіші частини. У шахах важливий король, оскільки його втрата призводить до програшу у грі. Відзначимо, що вартість одиниці є лише приблизною і не визначає результат гри [7].

Початкова конфігурація. Обидва гравці починають з вісьмома пішаками, двома конями, двома слонами, двома турами, ферзем і королем. Білі пішаки розташовано на другому рядку дошки, а чорні – на сьомому рядку. Лицарі, слони і єпископи починають заповнювати перший ряд білих і восьмий ряд чорних. Після формування ферзя він завжди переходить на поле, яке відповідає його кольору (білий ферзь на світлому полі, чорний ферзь на темному полі) [7].

Кожна шахова фігура має свої правила ходу. Пішаки рухаються вгору на один квадрат, перший раз можуть переміщатися на два квадрати. Пішак може ловити фігури на одне поле по діагоналі вперед. Кінь пересувається на дві клітинки по горизонталі й одну по вертикалі або дві клітинки по вертикалі й одну по горизонталі [7].

Слон може переміститися на будь-яку кількість квадратів. Через цей хід слон ніколи не може наступити на поле іншого кольору, з якого він починав свій рух. Гравці починають гру з двома слонами: світлим і темним [7].

Тура може рухатися на будь-яку кількість квадратів вертикально або горизонтально. Тура є єдиною фігурою, яка може бути рокована із королем.

Королева або ферзь є найсильнішою фігурою на дошці. Ця фігура може рухатися горизонтально або вертикально. Король може рухатися на одне поле в будь-якому напрямку [7].

Коли ворог атакує короля гравця, король знаходиться під шахом. Гравець, чий король піддається атаці, повинен захистити короля однією зі своїх фігур, видалити короля з зони дій атакуючого або захопити фігуру, що атакує власного короля. Якщо атакований гравець не має можливість нічого з цього, король отримає мат, і гравець програє гру [7].

Визначено, що у наш час широко використовуються комп'ютерні ігри, що являють собою програмне забезпечення для розважання чи навчання людей, індустрія комп'ютерних ігор у наш час являє собою великий бізнес, існують мільйони різних комп'ютерних ігор, якими користуються люди будь-якого віку. Виявлено, що комп'ютерна гра, як правило, складається з таких програмних та апаратних компонентів: пристрої та програмні компоненти для введення, ігрова логіка, підтримка графіки/звучу, засоби виведення гри та засоби підтримки мережі. Визначено, що при реалізації комп'ютерної гри використовується два основних типи підтримки мультимедіа – одноранговий та клієнт-серверний. Виконано класифікацію комп'ютерних ігор, показано, що зараз існує понад ста жанрів відеоігор, у яких користувач може грати та досліджувати, включаючи всі піджанри та додаткові жанри. Відзначено, що гра у шахи є однією з найстаріших та найпопулярніших настільних ігор у світі, яка є настільною грою між двома гравцями, що імітує війну між двома королівствами, та яка у наш час програмно реалізована у вигляді багатьох програмних застосунків.

1.2 Дослідження програмних засобів для гри в шахи

Зараз існує значна кількість програм для гри в шахи. Проведемо огляд певних програм та вебсайтів, що підтримують процес гри у шахи [8]-[11].

Програмне забезпечення Chess Arena (рис. 1.1) являє собою вебсайт, що дозволяє користувачам грати в шахи між собою. На цьому сайті підтримується можливість багатокористувальницької гри між користувачами (грає два користувачі один проти одного), є ресурси для навчання грі у шахи. Програмне забезпечення Chess Arena надає можливість користувачам дивитися в онлайн-режимі відеотрансляції ігор інших користувачів [8].



Рисунок 1.1 – Програмне забезпечення Chess Arena [8]

Програмне забезпечення Chess Arena підтримує ведення рейтингу користувачів, а також надає можливість зареєстрованим користувачам придбати шахові дошки та інші товари, пов'язані з грою у шахи, через свій власний онлайн-магазин [8]. Chess Arena є популярною шаховою програмою. Може використовуватися в якості інтерфейсу для багатьох шахових движків, корисна в аналізі шахових партій. Має невелику вбудовану бібліотеку найбільш поширених варіантів [8].

Серед особливостей даного сервісу, що надає можливості для гри в шахи, можна виділити такі [8]:

- можливість для гри в шахи через зручний вебінтерфейс;

- підтримка можливості організації багатокористувальницької гри між користувачами (грає два користувачі один проти одного);
- наявність ресурсів для навчання гри у шахи;
- можливість користувачам дивитися в онлайн-режимі відеотрансляції ігор інших користувачів;
- підтримка ведення рейтингу користувачів;
- можливість придбати шахові дошки та інші товари, пов'язані з грою у шахи, через свій власний онлайн-магазин;
- можливість використання як движок в іншому програмному забезпеченні;
- ефективна служба підтримки;
- вбудована бібліотека найбільш поширених варіантів гри в шахи;
- засоби допомоги користувачу;
- наявність безкоштовного тарифного плану;
- використання хмарних технологій.

Серед недоліків програмного забезпечення Chess Arena виділяється те, що ця програма не дозволяє підтримувати гру зі штучним суперником, стратегія дій якого заснована на методах штучного інтелекту [8]. Крім того, у цьому програмному забезпеченні обмежена можливість зміни мови (за замовченням встановлено мову країни-агресора, є можливість встановлення англійської мови), у т.ч. відсутня україномовна версія цього програмного забезпечення [8].

Програмне забезпечення BabasChess (рис. 1.2) являє собою застосунок, що дозволяє користувачам грати в шахи під керуванням операційної системи Windows. У цьому програмному забезпеченні підтримується можливість гри між користувачами та комп'ютерною програмою, є ресурси для навчання гри у шахи. Програмне забезпечення BabasChess надає можливість користувачам практикуватися у грі в шахи [9].



Рисунок 1.2 – Програмне забезпечення BabasChess [9]

Програмне забезпечення BabasChess є популярною шаховою програмою. Має невелику вбудовану бібліотеку найбільш поширених варіантів [9].

Серед особливостей даного сервісу, що надає можливості для гри в шахи, можна виділити такі [9]:

- можливість для гри в шахи через зручний інтерфейс;

- підтримка можливості організації гри в шахи між користувачами та комп'ютером;
- наявність ресурсів для навчання гри у шахи;
- ефективна служба підтримки;
- вбудована бібліотека найбільш поширених варіантів гри в шахи;
- засоби допомоги користувачу;
- безкоштовна;
- використання хмарних технологій.

Серед недоліків програмного забезпечення BabasChess виділяється те, що ця програма не дозволяє підтримувати багатокристувальницьку гру між різними суперниками [9]. Крім того, у цьому програмному забезпеченні обмежена можливість зміни мови, у т.ч. відсутня україномовна версія цього програмного забезпечення [9].

Програмне забезпечення DecodeChess є шаховим програмним забезпеченням, яке містить навчальний розділ, що навчає користувача основним стратегіям, гри в дебюті та ендшпілі, тактичним ходам, вдосконаленій тактиці та розробляє тактику контролю за центром і ендшпілем. Користувач також може отримати гарне уявлення про різні відкриті поля на шахівниці, включаючи пішаків, тур і коней, а також про те, як контролювати центр. Користувач може транслювати свою гру на відеоплатформах за допомогою програмного забезпечення потокового передавання YouTube [10].

Крім того, посібник програмного забезпечення DecodeChess містить коротке пояснення того, як використовувати деякі складніші тактики в грі. Він також охоплює різні відкриття, які можна використовувати в грі, включно з відкриттям тури, відкриттям ферзя та варіацією відкриття тури та ферзя. Оскільки програмне забезпечення DecodeChess (рис. 1.3) є безкоштовним, будь-хто може випробувати програмне забезпечення, перш ніж придбати його [10].

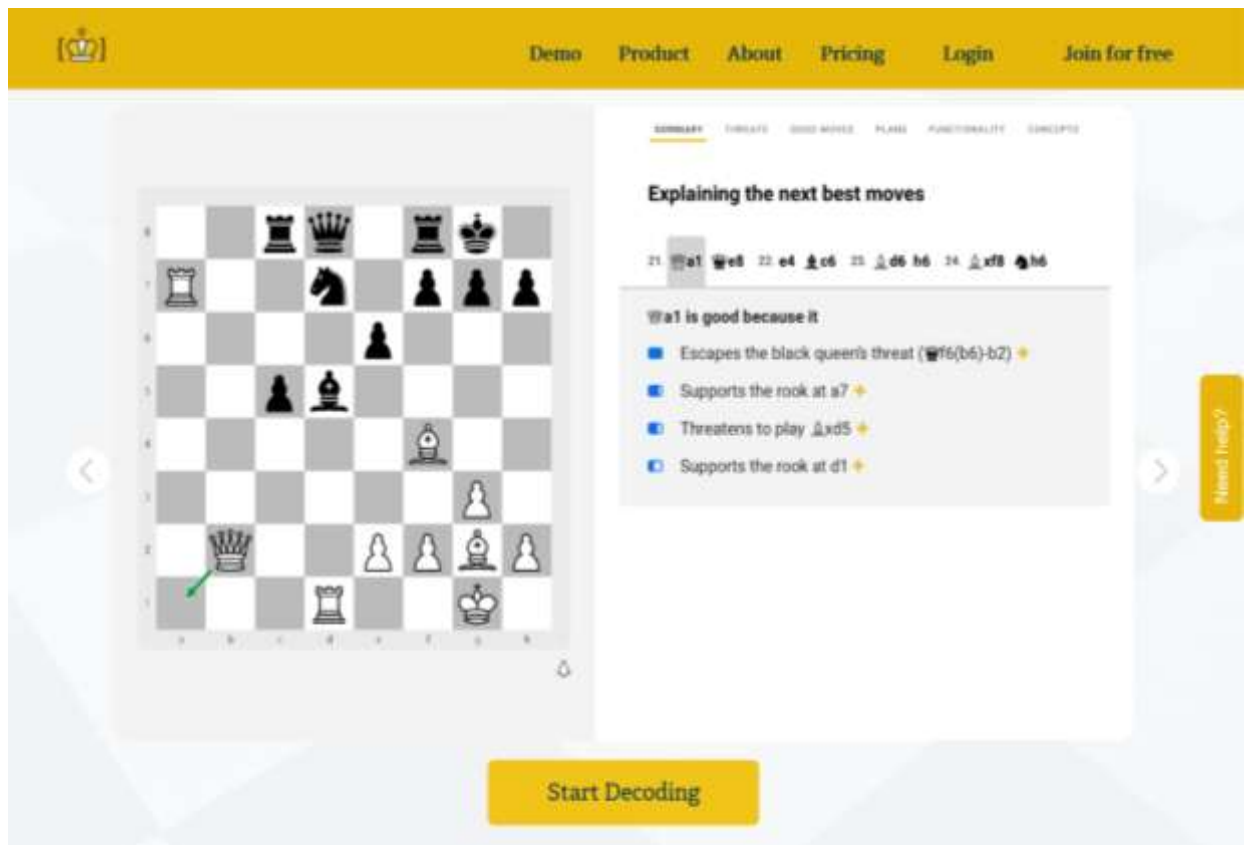


Рисунок 1.3 – Програмне забезпечення DecodeChess [10]

Програмне забезпечення DecodeChess підтримує ведення рейтингу користувачів, а також надає можливість зареєстрованим користувачам придбати шахові дошки та інші товари, пов'язані з грою у шахи, через свій власний онлайн-магазин [10]. DecodeChess є популярною шаховою програмою. Може використовуватися в якості інтерфейсу для багатьох шахових движків, корисна в аналізі шахових партій. Має невелику вбудовану бібліотеку найбільш поширених варіантів [10].

Серед особливостей даного сервісу, що надає можливості для гри в шахи, можна виділити такі [10]:

- можливість для гри в шахи через зручний вебінтерфейс;
- підтримка можливості організації багатокористувальницької гри між користувачами (грає два користувачі один проти одного);
- наявність ресурсів для навчання грі у шахи;
- можливість користувачам дивитися в онлайн-режимі відеотрансляції ігор інших користувачів;

- підтримка ведення рейтингу користувачів;
- можливість придбати шахові дошки та інші товари, пов'язані з грою у шахи, через свій власний онлайн-магазин;
- можливість використання як движок в іншому програмному забезпеченні;
- ефективна служба підтримки;
- вбудована бібліотека найбільш поширених варіантів гри в шахи;
- засоби допомоги користувачу;
- наявність безкоштовного тарифного плану;
- використання хмарних технологій.

Серед недоліків програмного забезпечення DecodeChess виділяється те, що ця програма не дозволяє підтримувати гру зі штучним суперником, стратегія дій якого заснована на методах штучного інтелекту [10]. Крім того, у цьому програмному забезпеченні обмежена можливість зміни мови, у т.ч. відсутня україномовна версія цього програмного забезпечення [10].

Програмне забезпечення Komodo Chess Engine є шаховим двигуном, що є ефективним інструментом для гравців у шахи, щоб краще зрозуміти гру. Після того, як користувач дізнається про тонкощі шахової системи, він зможе почати грати в шахи краще. Ця програма має кнопку автоматичного повтору гри. Але якщо користувачу потрібно записати гру, знадобиться стороннє програмне забезпечення для запису гри. Однією з найкращих переваг безкоштовного шахового програмного забезпечення є можливість завантажити його прямо на свій комп'ютер, щоб продовжити гру в будь-який час. Той факт, що є миттєвий доступ до програмного забезпечення, дозволяє грати в будь-який час і без будь-яких обмежень [11].

Програмне забезпечення Komodo Chess Engine (рис. 1.4) підтримує ведення рейтингу користувачів, а також надає можливість зареєстрованим користувачам придбати шахові дошки та інші товари, пов'язані з грою у шахи, через свій власний онлайн-магазин [11].

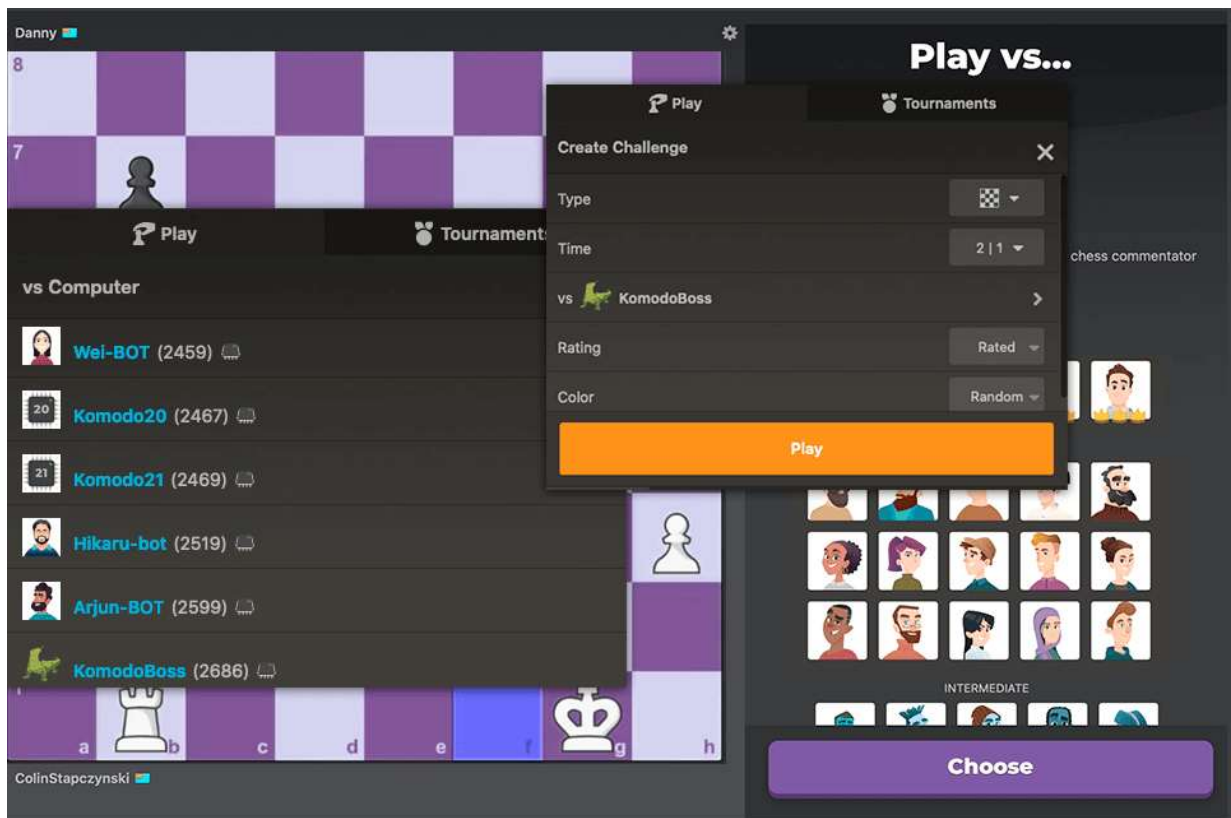


Рисунок 1.4 – Програмне забезпечення Komodo Chess Engine [11]

Серед особливостей даного сервісу, що надає можливості для гри в шахи, можна виділити такі [11]:

- можливість для гри в шахи через зручний вебінтерфейс;
- підтримка можливості організації багатокористувальницької гри між користувачами (грає два користувачі один проти одного);
- наявність ресурсів для навчання грі у шахи;
- можливість користувачам дивитися в онлайн-режимі відеотрансляції ігор інших користувачів;
- підтримка ведення рейтингу користувачів;
- можливість придбати шахові дошки та інші товари, пов'язані з грою у шахи, через свій власний онлайн-магазин;
- можливість використання як движок в іншому програмному забезпеченні;
- ефективна служба підтримки;
- вбудована бібліотека найбільш поширених варіантів гри в шахи;

- засоби допомоги користувачу;
- наявність безкоштовного тарифного плану;
- використання хмарних технологій.

Серед недоліків програмного забезпечення Komodo Chess Engine виділяється те, що ця програма не дозволяє підтримувати гру зі штучним суперником, стратегія дій якого заснована на методах штучного інтелекту [11]. Крім того, у цьому програмному забезпеченні обмежена можливість зміни мови, у т.ч. відсутня україномовна версія цього програмного забезпечення [11].

Порівняльну характеристику програмного забезпечення для гри в шахи наведено у таблиці 1.1.

Таблиця 1.1 – Порівняння програмного забезпечення для гри в шахи

Критерій порівняння	Arena [8]	BabasChess [9]	DecodeChess [10]	Komodo Chess Engine [11]
Онлайн версія	+	–	+	+
Необхідність оплати	+–	–	–	–
Можливість грати з іншими суперниками	+	–	+	+
Україномовна версія	–	–	–	–
Платформонезалежність	+	–	+	+

За результатами проведеного аналізу можна зробити висновок, що у наш час існує досить багато програмних систем для гри в шахи. Проте деякі з них не надають можливості грати з іншими суперниками, інші є платформозалежними та платними для користувачів. Тому актуальною є розробка програмного забезпечення для гри в шахи.

При розробці програми для гри в шахи необхідно забезпечити такі функціональні вимоги:

- підтримка процесу гри користувача у шахи;
- можливість пересувати фахові фігури та робити ходи згідно правил (перевірка ходів користувача на відповідність правилам гри у шахи);
- надання підказок щодо можливих ходів;
- реалізовувати гру шляхом запитів на сервер;
- підтримка звернень та збережень даних у базі даних;
- підтримка багатокористувальницької гри між різними користувачами;
- виведення повідомлень користувачу у випадку виникнення критичних ситуацій.

1.3 Висновки за розділом 1

Визначено, що у наш час широко використовуються комп'ютерні ігри, що являють собою програмне забезпечення для розважання чи навчання людей, індустрія комп'ютерних ігор у наш час являє собою великий бізнес, існують мільйони різних комп'ютерних ігор, якими користуються люди будь-якого віку. Виявлено, що комп'ютерна гра, як правило, складається з таких програмних та апаратних компонентів: пристрої та програмні компоненти для введення, ігрова логіка, підтримка графіки/звuku, засоби виведення гри та засоби підтримки мережі. Визначено, що при реалізації комп'ютерної гри використовується два основних типи підтримки мультимедіа – одноранговий та клієнт-серверний. Виконано класифікацію комп'ютерних ігор, показано, що зараз існує понад ста жанрів відеоігор, у яких користувач може грати та досліджувати, включаючи всі піджанри та додаткові жанри. Відзначено, що гра у шахи є однією з найстаріших та найпопулярніших настільних ігор у світі, яка є настільною грою між двома гравцями, що імітує війну між двома королівствами, та яка у наш час програмно реалізована у вигляді багатьох програмних застосунків.

За результатами проведеного аналізу зроблено висновок про те, що у наш час існує досить багато програмних систем для гри в шахи. Проте деякі з

них не надають можливості грати з іншими суперниками, інші є платформозалежними та платними для користувачів. Тому актуальною є розробка програмного забезпечення для гри в шахи.

Сформульовано функціональні вимоги до програмного забезпечення для гри в шахи.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Вибір мови програмування та інших технологій розробки

При розробці програмного забезпечення для гри в шахи було використано мову програмування C# [12] – [14].

Мова програмування C# є об'єктно-орієнтованою мовою програмування. Він був розроблений групою інженерів як основна мова для створення програм на платформі Microsoft .NET. Компілятор C# самостійно інтегрується зі стандартною платформою .NET, тому програми можна створювати та скомпільовувати без таких інструментів, як Visual Studio [12] – [14].

C# належить до сімейства мов із синтаксисом C, і його синтаксис близький до C++ і Java. Ця мова має фіксований шрифт, підтримує поліморфізм і перевантаження, а також визначає дії членів групи, властивості, події, атрибути, параметри та визначення у форматі XML. C# успадковує багато від своїх попередників, за винятком деяких моделей, які виявилися проблематичними для розробки систем, C++, Delphi, Modula та Smalltalk на основі реалізацій програм: так, C# не підтримує кілька класів успадкування (в C++) або типи вказівників [12] – [14].

Хоча C# є об'єктно-орієнтованою мовою, він підтримує компонентно-орієнтоване програмування. Сучасні розробники програмного забезпечення все більше цікавляться створенням програмних компонентів як самостійних пакетів, що реалізують певні функції. Ключовою особливістю відділів є характер, стиль і подієвість програмування. Кожен елемент має функцію, яка містить інформацію про компонент і текстовий опис. C# надає вбудовану мову, яка повністю підтримує цю функціональну концепцію. Це робить його потужним для створення та модифікації програмних компонентів C# [12] – [14].

Ця мова програмування має строгу типізацію, підтримує поліморфізм і перевантаження, а також визначає дії членів групи, властивості, події, атрибути, параметри та визначення у форматі XML. C# успадковує багато від

своїх попередників, за винятком деяких моделей, які виявилися проблематичними для розробки систем, C++, Delphi, Modula та Smalltalk на основі реалізацій програм: так, C# не підтримує кілька класів. успадкування (в C++) або типи вказівників [12] – [14].

Обґрунтування вибору мови програмування для розробки програмного забезпечення для гри в шахи наведено у таблиці 2.1.

Таблиця 2.1 – Обґрунтування вибору мови програмування для розробки програмного забезпечення для гри в шахи

Критерій порівняння мов програмування	Мова програмування		
	C#	Java	Php
Зручність для створення програмного забезпечення для гри в шахи	+	+-	+-
Автоматичний збиральник сміття	+	+	-
Динамічна типізація	+	+	-
Зручність створення графічного інтерфейсу користувача	+	+-	+-
Категорії змінних	+	+	-
Простота використання	+	+-	+-

Отже, для реалізації програмного забезпечення для гри в шахи обрано мову програмування C#, яка є об'єктно-орієнтованою мовою програмування, був розроблена групою інженерів як основна мова для створення програм на платформі Microsoft .NET, має строгу типізацію, підтримує поліморфізм і перевантаження, є зручною для створення графічного інтерфейсу для взаємодії з користувачем.

2.2 Вибір середовища розробки для створення програмного забезпечення для гри в шахи

В якості середовища розробки для створення програмного забезпечення для гри в шахи було обрано Visual Studio [15], [16].

Microsoft Visual Studio являє собою набір інструментів розробки програмного забезпечення. Visual Studio підтримує C#, C і C++, JavaScript, F# і Visual Basic, а також при необхідності може підтримувати і інші мови програмування. Visual Studio добре працює та інтегрується зі сторонніми програмами, такими як Unity та Apache Cordova, через плагіни. Visual Studio також дозволяє налаштовувати інструменти для виконання певних завдань [15], [16].

Visual Studio включає вихідний код і підтримку технології IntelliSense, а також можливості перегляду коду. Вбудована архітектура може діяти як рішення на рівні ресурсів, так і як машина високого класу. Інші інструменти включають редактор форм, вебредактор, дизайнер класів і редактор бази даних для спрощення графічного інтерфейсу користувача програми. Visual Studio дозволяє створювати та інтегрувати сторонні додатки (плагіни) для покращення функціональності майже на будь-якому рівні, незалежно від того, додається підтримка оригінальної системи контролю версій, додаються нові інструменти або додаються інструменти до інших областей розробки програмного забезпечення [15], [16].

Інтерфейс Visual Studio забезпечує більше взаємодії, ніж будь-який інший редактор. Для спрощення взаємодії з користувачем код Visual Studio поділено на основні області: панель завдань, бічна панель, група сценаріїв, стандартна панель [15], [16].

Visual Studio — це безкоштовний, простий, але потужний інструмент, який працює на комп'ютері та в Інтернеті та може використовуватися в різних операційних системах [15], [16].

Visual Studio швидко та легко запустити та містить повний код для змінних, методів та імпортованих модулів, багатоточкове редагування, індексування та інші потужні операції редагування, управління вихідним кодом, включаючи підтримку систем контролю версій [15], [16].

За потреби надається додаткова інформація. Він забезпечує різноманітну підтримку для різних мов програмування та розширень, від простих лінійних виразів і масивів до обробки та рекурсії [15], [16].

Код у репозиторії Visual Studio є відкритим кодом. Продукти в Studio містять невеликий відсоток ексклюзивного контенту, тому вони мають стандартну ліцензію. Це середовище розробки має комерційну ліцензію, але є безкоштовним [15], [16].

Обґрунтування вибору середовища розробки для створення програмного забезпечення для гри в шахи наведено у таблиці 2.2.

Таблиця 2.2 – Обґрунтування вибору середовища розробки для створення програмного забезпечення для гри в шахи

Критерій порівняння середовищ розробки	Середовище розробки		
	Visual Studio	SharpDevelop	CodeLite
Необхідність оплати за користування	–	–	+
Наявність візуального конструктору	+	+	–
Зручність використання при розробці програми для гри у шахи	+	+–	+–
Багатоплатформеність	+	–	+
Підсвічування коду	+	–	+

Таким чином, для створення програмного забезпечення для гри в шахи обрано середовище розробки Visual Studio, що містить набір інструментів розробки програмного забезпечення, підтримує мову програмування C#, добре працює та інтегрується зі сторонніми програмами, зокрема з Unity.

Для розробки програмного забезпечення для гри в шахи як ігровий рушій було обрано Unity [17].

Unity — це кросплатформний інструмент для розробки 2D і 3D програм, зокрема комп'ютерних ігор. Програмне забезпечення було розроблено з Unity та працює на різних операційних системах та ігрових приставках.

Unity Editor має простий інтерфейс Drag and Drop для легкого редагування та сумісності з різними операційними системами Windows, тому ви можете легко завантажити свою гру з редактора. Rushi підтримує (редагувати) C# і JavaScript#. Об'єднаний проєкт розділений на розділи (рівні) - різні файли, що містять ігровий світ з власними предметами, об'єктами та описами. Зображення може містити як об'єкти (моделі), так і вільні ігрові об'єкти, тобто без моделей. Об'єкти також містять текстові елементи.

Переваги Unity такі: движок гри повністю сумісний із середовищем розробки, що дозволяє тестувати гру безпосередньо; робота з інструментами можлива за допомогою стандартного Drag and Drop; система підтримує попередні версії; підтримка імпорту кількох форматів файлів; бібліотечна підтримка.

Обґрунтування вибору ігрового рушія для створення програмного забезпечення для гри в шахи наведено у таблиці 2.3.

Отже, для розробки програмного забезпечення для гри в шахи як ігровий рушій було обрано Unity, що є кросплатформним інструментом для розробки двовимірних та тривимірних комп'ютерних ігор.

Таблиця 2.3 – Обґрунтування вибору ігрового рушія для створення програмного забезпечення для гри в шахи

Критерій порівняння ігрових рушіїв	Ігрові рушії		
	Unity	Project Anarchy	Unreal Engine
Легкість використання	+	–	+
Підтримка розробників	+	–	+
Зручність налаштування	+	+–	+–
Підтримувані скриптові мови	C#, JavaScript	–	C++
Зручність для створення програмного забезпечення для гри в шахи	+	–	–

2.3 Висновки за розділом 2

Для реалізації програмного забезпечення для гри в шахи обрано мову програмування C#, яка є об'єктно-орієнтованою мовою програмування, був розроблена групою інженерів як основна мова для створення програм на платформі Microsoft .NET, має строгу типізацію, підтримує поліморфізм і перевантаження, є зручною для створення графічного інтерфейсу для взаємодії з користувачем.

В якості середовища розробки для створення програмного забезпечення для гри в шахи обрано Visual Studio, що містить набір інструментів розробки програмного забезпечення, підтримує мову програмування C#, добре працює та інтегрується зі сторонніми програмами, зокрема з Unity.

Для розробки програмного забезпечення для гри в шахи як ігровий рушій було обрано Unity, що є кросплатформним інструментом для розробки двовимірних та тривимірних комп'ютерних ігор.

3 ОПИС ПРОГРАМИ

3.1 Структура процесів програмного забезпечення для гри в шахи

Відповідно до технічного завдання було обрано структуру програми для гри в шахи. Структура процесів програми для гри в шахи зображена на рисунку 3.1.

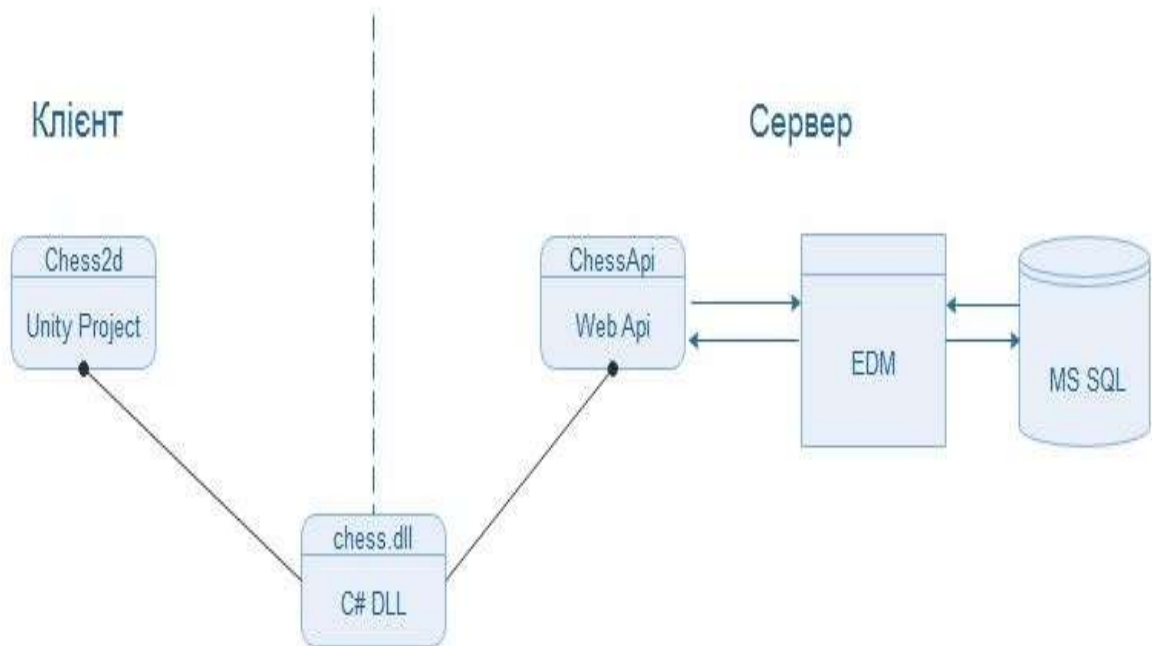


Рисунок 3.1 – Структура процесів програмного забезпечення для гри в шахи

Як видно з рис. 3.1 програмне забезпечення складається з клієнтської частини (модуль Chess2d) та серверної частини (модуль ChessApi та модулі для роботи з базою даних). Взаємозв'язок клієнтської та серверної частин забезпечується за допомогою бібліотеки chess.dll.

Призначення основних модулів та компонентів програмного забезпечення для гри в шахи наведено у табл. 3.1.

Таблиця 3.1 – Призначення основних модулів та компонентів програмного забезпечення для гри в шахи

Модуль	Опис призначення
Chess.dll	Бібліотека класів, що реалізує усі шахові правила
Chess2d	Графічний інтерфейс для гри у шахи
ChessBD	Сервер та база даних для мережевої гри
ChessAPI	Модель та контролери для взаємодії з сервером
ConsoleChess	Консольний додаток для тестування шахової бібліотеки

3.2 Опис основних модулів та компонентів програмного забезпечення для гри в шахи

Для програмної реалізації, відповідно до технічного завдання, логіки гри в шахи, перевірки коректності виконуваних ходів було створено допоміжну бібліотеку Chess.dll.

Діаграму класів модуля Chess.dll зображено на рис. 3.2.

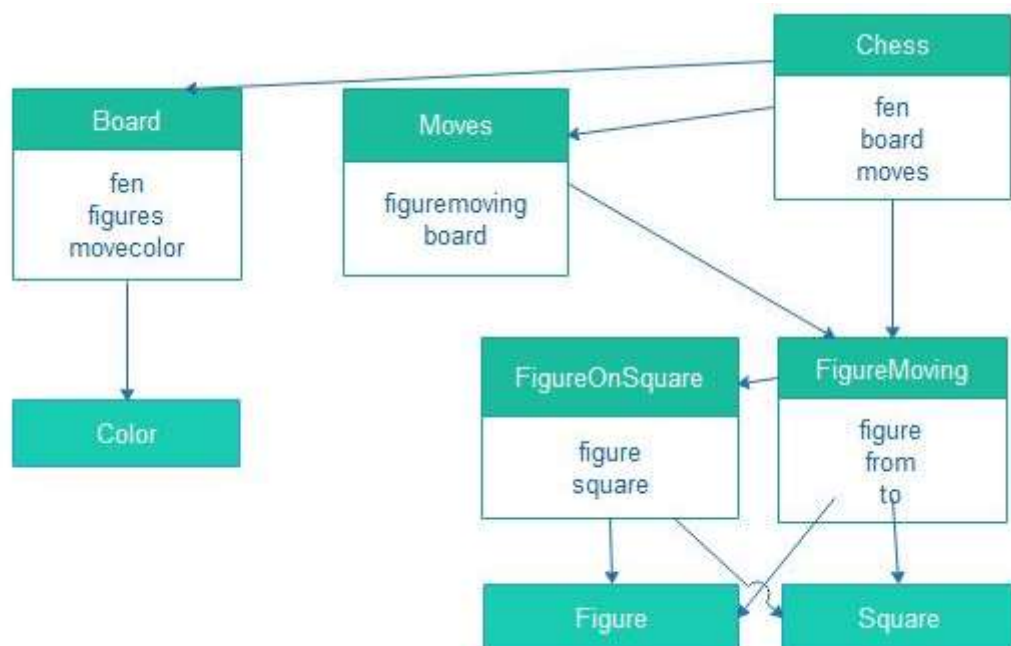


Рисунок 3.2 – Діаграма класів модуля Chess

Опишемо основні класи модуля Chess.dll. В класі Board описана шахова дошка та функції для розміщення фігур. В класі Moves описані усі правила, функції перевірки можливості виконання ходу та генерації списку можливих ходів. Клас FigureMoving описує переміщення фігур по клітинам шахової дошки. В класі FigureOnSquare описані шахові фігури та допоміжні функції для роботи з ними.

Модуль ChessApi створений для взаємодії клієнтської та серверної частин програм. Даний модуль реалізує шаблон MVC для вебзастосунків. На рис. 3.3 зображено діаграму класів ChessApi.

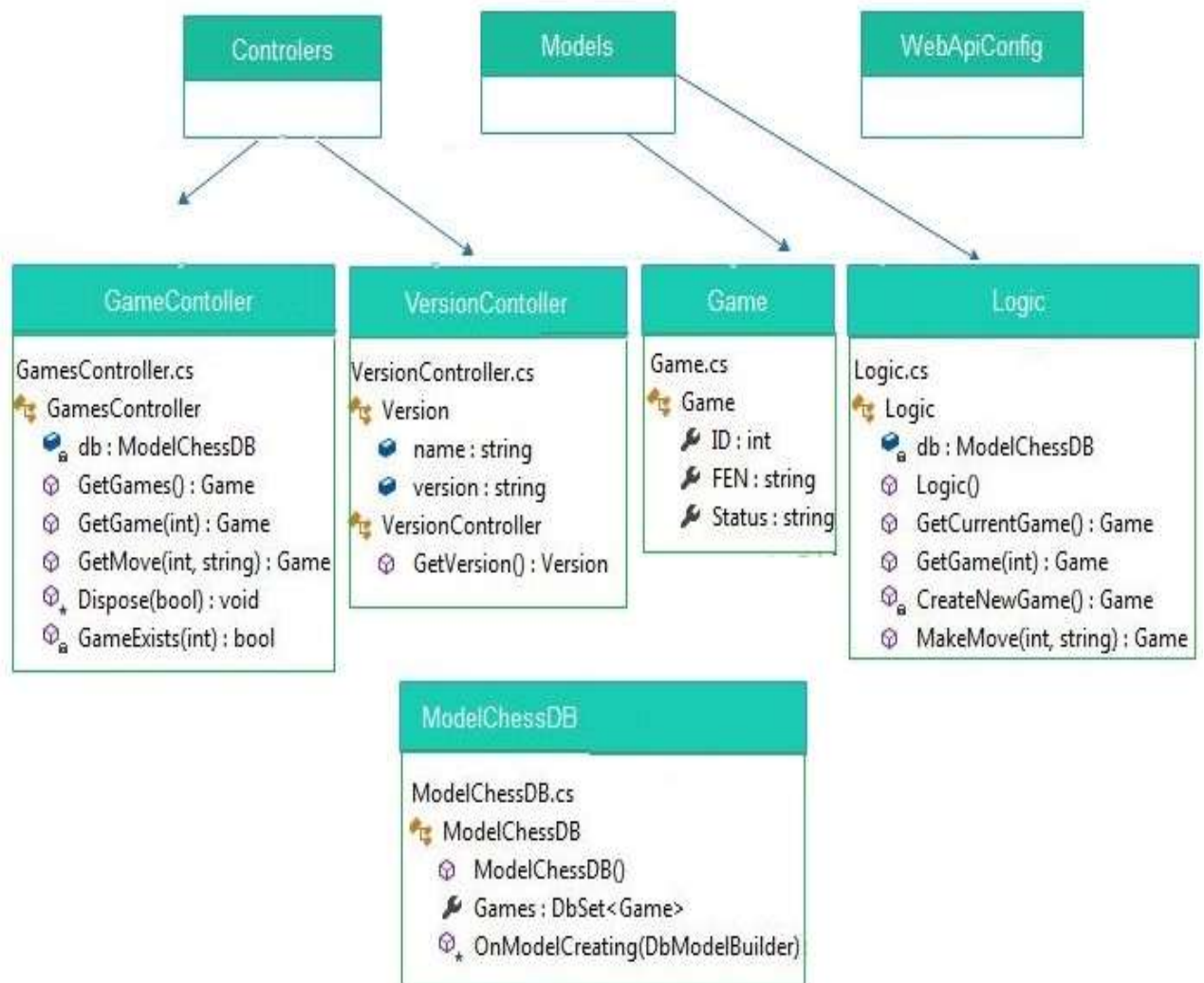


Рисунок 3.3 – Діаграма класів модуля ChessApi

В якості контролера виступають класи GameController та VersionController. Класи Game та Logic реалізують модель гри. Клас ModelChessDB виконує роль представлення бази даних всередині програми.

3.3 База даних програмного забезпечення для гри в шахи

Проектування бази даних було виконано засобами Sql Server Management Studio.

Концептуальну модель бази даних програмного забезпечення для гри в шахи зображено на рисунку 3.4.

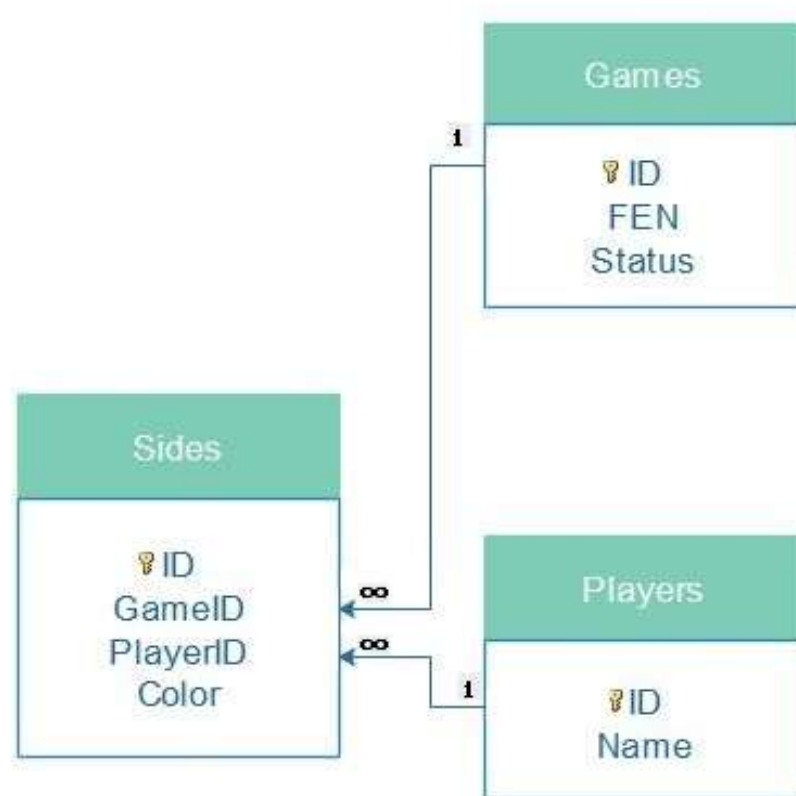


Рисунок 3.4 – Концептуальна модель бази даних програмного забезпечення для гри в шахи

Основною сутністю бази даних програмного забезпечення для гри в шахи є сутність Games (Id, FEN, Status), що містить інформацію про ігрові партії та їх статуси (табл. 3.2). Поля сутності: Id – унікальний ідентифікатор

партії, FEN – загальноприйнята нотація для відображення порядку фігур на дошці; Status – поле для відображення статусу гри.

Таблиця 3.2 – Опис полів сутності Games

№	Назва поля	Тип даних	Розмір
1	ID	uniqueidentifier	255
2	FEN	String	255
3	Status	String	50

Сутність Sides(ID, GameID, PlayerID, Color) – сутність, що містить інформацію про сторони ігрової партії. Поля сутності: ID – унікальний ідентифікатор сторони партії; GameID – ідентифікатор партії, PlayerID – ідентифікатор гравця, Color – сторона, за яку грає користувач.

Опис полів сутності Sides та їх типу наведено у таблиці 3.3.

Таблиця 3.3 – Опис полів сутності Sides

№	Назва поля	Тип даних	Розмір
1	ID	uniqueidentifier	255
2	GameID	int	255
3	PlayerID	int	255
4	Color	String	10

Сутність Players(ID, Name) – сутність, що містить інформацію про гравців. Поля сутності: ID – унікальний ідентифікатор гравця; Name – ім'я користувача.

Опис полів сутності Players та їх типу наведено у таблиці 3.4.

Таблиця 3.4 – Опис полів сутності Players

№	Назва поля	Тип даних	Розмір
1	ID	uniqueidentifier	255
2	Name	string	30

3.4 Розробка графічного інтерфейсу програмного забезпечення для гри в шахи

Графічний інтерфейс створено засобами ігрового движку Unity. Для реалізації гри було необхідно створити об'єкти у вигляді гральної дошки, фігур, та спрайтів для прорисовування фігур на дошці. На рисунку 3.5 зображена схема інтерфейсу гри.

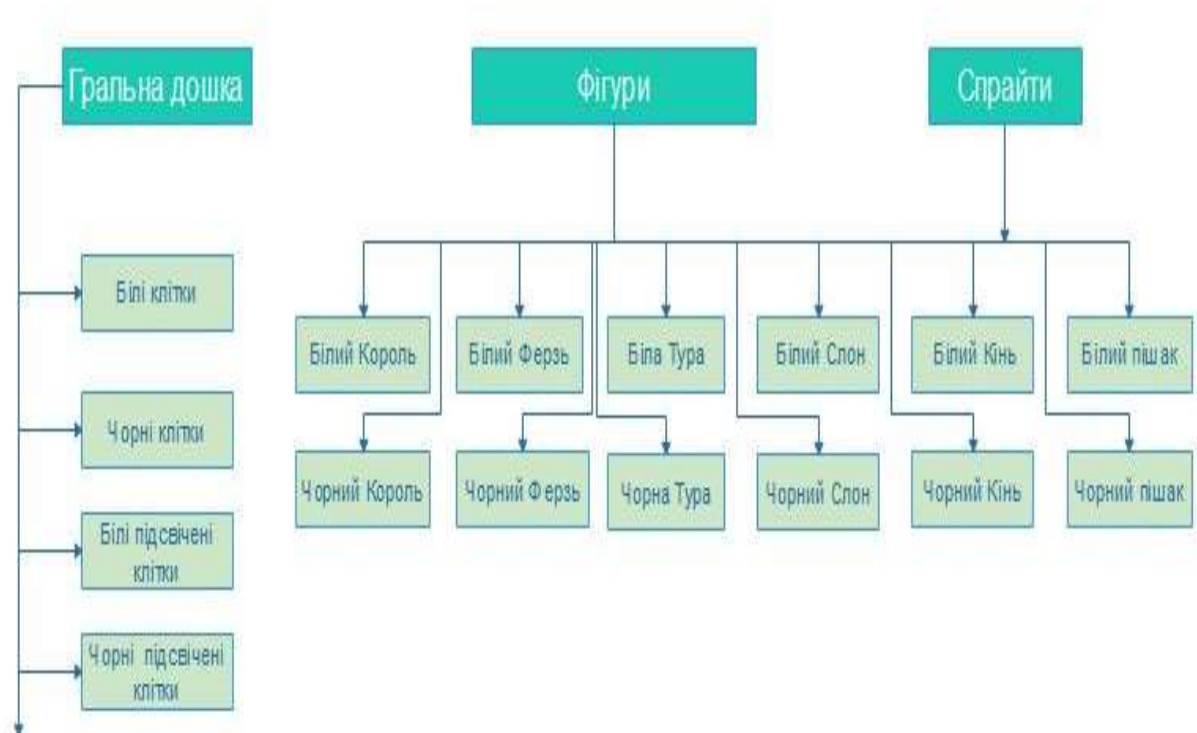


Рисунок 3.5 – Схема графічного інтерфейсу програмного забезпечення для гри в шахи

Зовнішній вигляд розробленого інтерфейсу програмного забезпечення для гри в шахи зображено на рисунку 3.6.

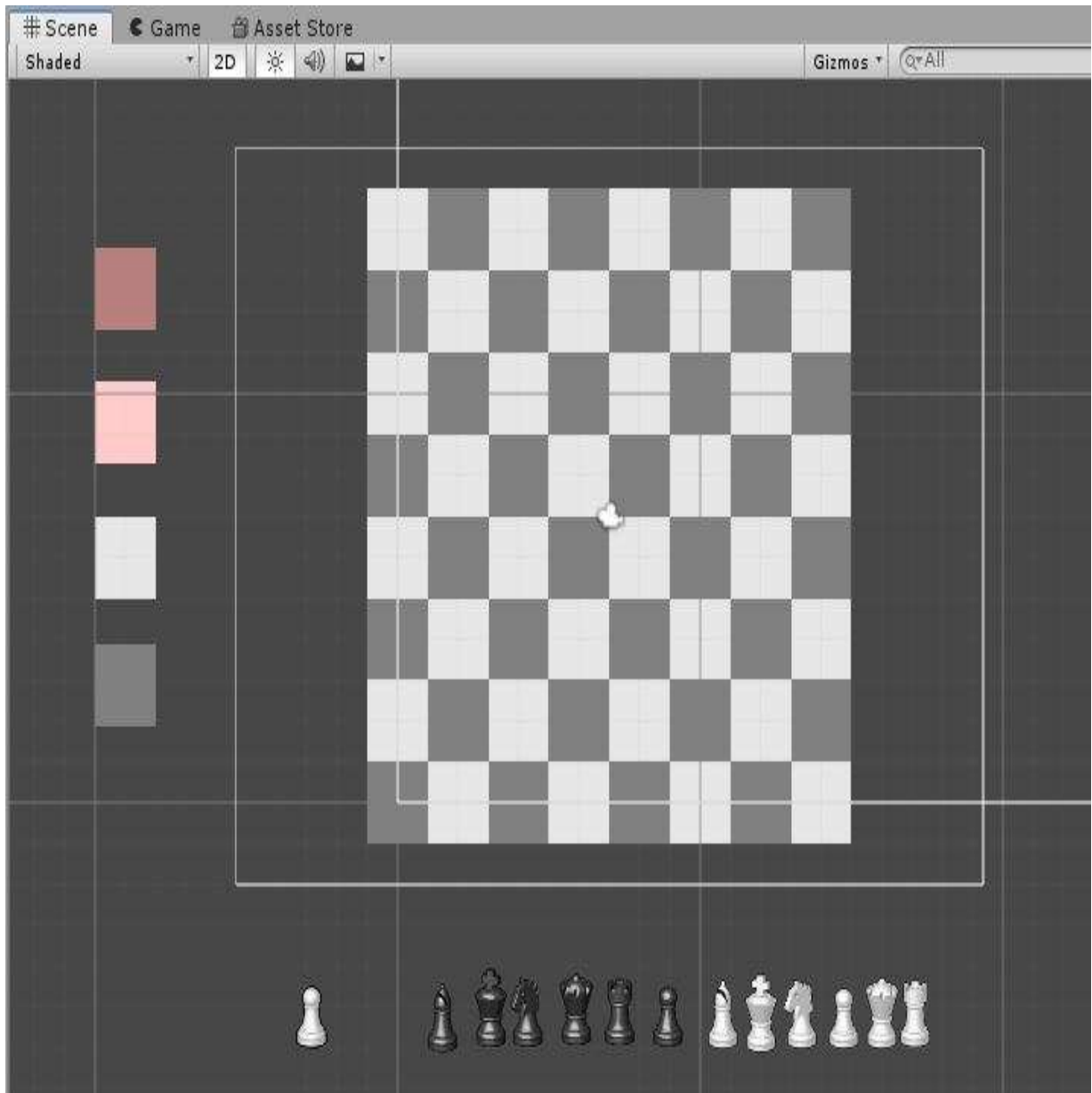


Рисунок 3.6 – Інтерфейс програмного забезпечення для гри в шахи

3.5 Функціонування програмного забезпечення для гри в шахи

Процес функціонування програмного забезпечення для гри в шахи представимо за допомогою діаграми послідовностей (рис. 3.7).

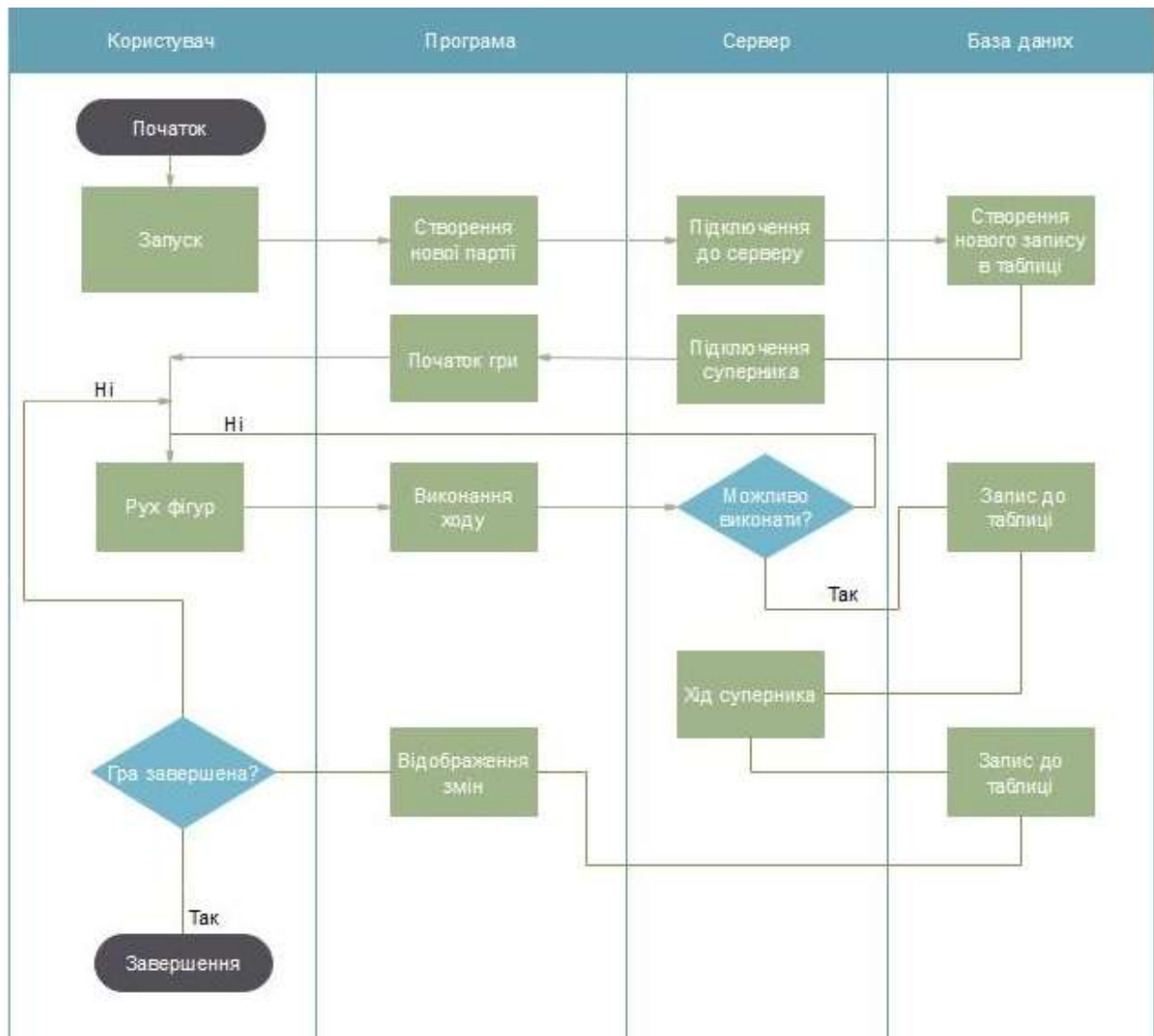


Рисунок 3.7 – Діаграма послідовностей програмного забезпечення для гри в шахи

3.6 Висновки за розділом 3

Розроблено програмне забезпечення для гри в шахи.

Запропоновано структуру процесів програмного забезпечення для гри в шахи. Визначено, що програмне забезпечення складається з клієнтської частини (модуль Chess2d) та серверної частини (модуль ChessApi та модулі для роботи з базою даних). Взаємозв'язок клієнтської та серверної частин забезпечується за допомогою бібліотеки chess.dll.

Для програмної реалізації, відповідно до технічного завдання, логіки гри в шахи, перевірки коректності виконуваних ходів було створено

допоміжну бібліотеку Chess.dll, в якій реалізовано декілька класів. В класі Board описана шахова дошка та функції для розміщення фігур. В класі Moves описані усі правила, функції перевірки можливості виконання ходу та генерації списку можливих ходів. Клас FigureMoving описує переміщення фігур по клітинам шахової дошки. В класі FigureOnSquare описані шахові фігури та допоміжні функції для роботи з ними.

Модуль ChessApi створений для взаємодії клієнтської та серверної частин програм. Даний модуль реалізує шаблон MVC для вебзастосунків. В якості контролера виступають класи GameController та VersionController. Класи Game та Logic реалізують модель гри. Клас ModelChessDB виконує роль представлення бази даних всередині програми.

Розроблено базу даних програмного забезпечення для гри в шахи. Основними сутностями бази даних є: Games, що містить інформацію про ігрові партії та їх статуси; Sides, що містить інформацію про сторони ігрової партії; Players, що містить інформацію про гравців.

Розроблено графічний інтерфейсу програмного забезпечення для гри в шахи, основним елементом якого є рисунок, що відображає шахову дошку.

Процес функціонування програмного забезпечення для гри в шахи представлено за допомогою діаграми послідовностей.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Призначення й умови застосування програми

Програма призначена для підтримки процесу гри у шахи. Програма написана на мові C#, в якості середовища розробки для створення програмного забезпечення для гри у шахи обрано Visual Studio, що містить набір інструментів розробки програмного забезпечення, підтримує мову програмування C#, добре працює та інтегрується зі сторонніми програмами, зокрема з Unity. Як ігровий рушій використано Unity, що є кросплатформним інструментом для розробки двовимірних та тривимірних комп'ютерних ігор.

Програмне забезпечення для підтримки процесу гри у шахи забезпечує виконання таких функцій:

- підтримка процесу гри користувача у шахи;
- можливість пересувати фахові фігури та робити ходи згідно правил (перевірка ходів користувача на відповідність правилам гри у шахи);
- надання підказок щодо можливих ходів;
- реалізовувати гру шляхом запитів на сервер;
- підтримка звернень та збережень даних у базі даних;
- підтримка багатокористувальницької гри між різними користувачами;
- виведення повідомлень користувачу у випадку виникнення критичних ситуацій.

Для експлуатації програмного забезпечення для підтримки процесу гри у шахи необхідно таке апаратне та програмне забезпечення:

- тактова частота мікропроцесора не менше 2,3 ГГц;
- оперативна пам'ять з мінімальним об'ємом 8 Гб;
- операційна система Windows 10 або вище;
- на стороні клієнта підтримка платформи .NET Framework, починаючи з версії 4.0.

Функціональні характеристики розробленого програмного забезпечення для підтримки процесу гри у шахи наведено у технічному завданні (додаток А). Фрагмент тексту програмного забезпечення для підтримки процесу гри у шахи наведено у додатку Б.

4.2 Характеристики програми та її встановлення

Програмне забезпечення для підтримки процесу гри у шахи працює відповідно до потреб користувача та в залежності від налаштувань та можливостей сервера може працювати цілодобово.

Програма підтримки процесу гри у шахи складається з таких програмних модулів:

- Chess.dll – бібліотека класів, що реалізує шахові правила;
- Chess2d – графічний інтерфейс для гри у шахи;
- ChessBD – сервер та база даних для мережевої гри;
- ChessAPI – модель та контролери для взаємодії з сервером;
- ConsoleChess – консольний додаток для тестування шахової бібліотеки.

4.3 Інструкція по експлуатації програми

4.3.1 Звернення до програми

Для запуску програми спочатку необхідно запустити серверну частину (ChessApi), а потім програмний застосунок Chess2d на стороні клієнта. Це можна зробити стандартними засобами операційної системи користувача.

4.3.2 Вхідні й вихідні дані

Вхідними даними до програмного забезпечення для гри в шахи є інформація про рух фігур на гральній дошці та інформація з таблиць бази даних.

Вихідними даними програмного забезпечення для гри в шахи є відображення можливих ходів, пересування фігур або неможливість їх пересування.

4.3.3 Повідомлення

Користувач програмного забезпечення для гри в шахи може отримати повідомлення про неможливість створення мережевої партії, яке виникає при відсутності підключення до серверу або несправностях у його роботі.

4.4 Виконання програмного забезпечення для гри в шахи

Після запуску програми для гри в шахи можна обрати розширення екрану, якість графіки та інші налаштування (рис. 4.1).

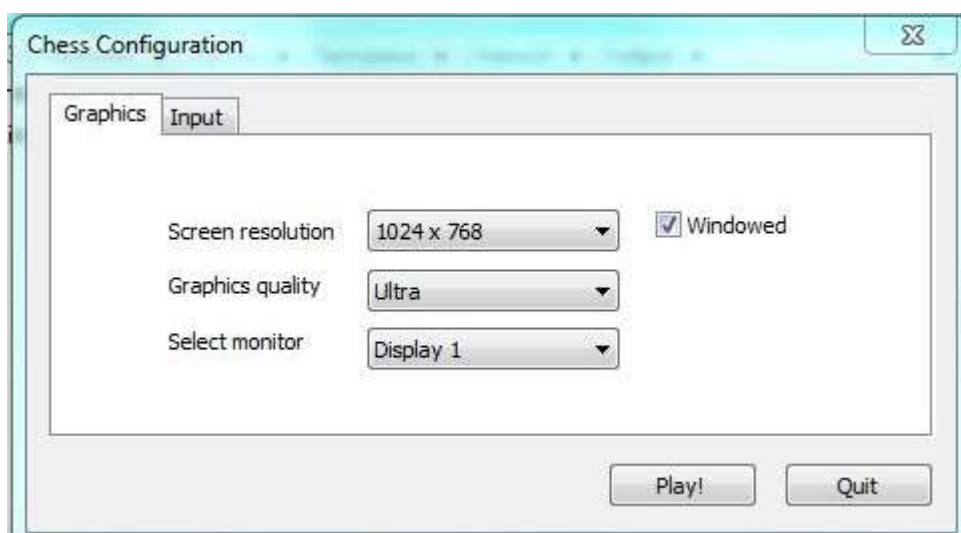


Рисунок 4.1 – Вікно налаштувань для роботи програмного забезпечення

Після вибору налаштувань з'являється шахова дошка, яка являє собою імітацію реальної дошки, де розташовано шахові фігури відповідно до правил гри (рис. 4.2).



Рисунок 4.2 – Стартове положення гри у шахи

У процесі гри (рис. 4.3) для підтримки користувача програма відображає фігури, які можуть виконати певний хід (відповідні фігури підсвічено рожевим кольором).

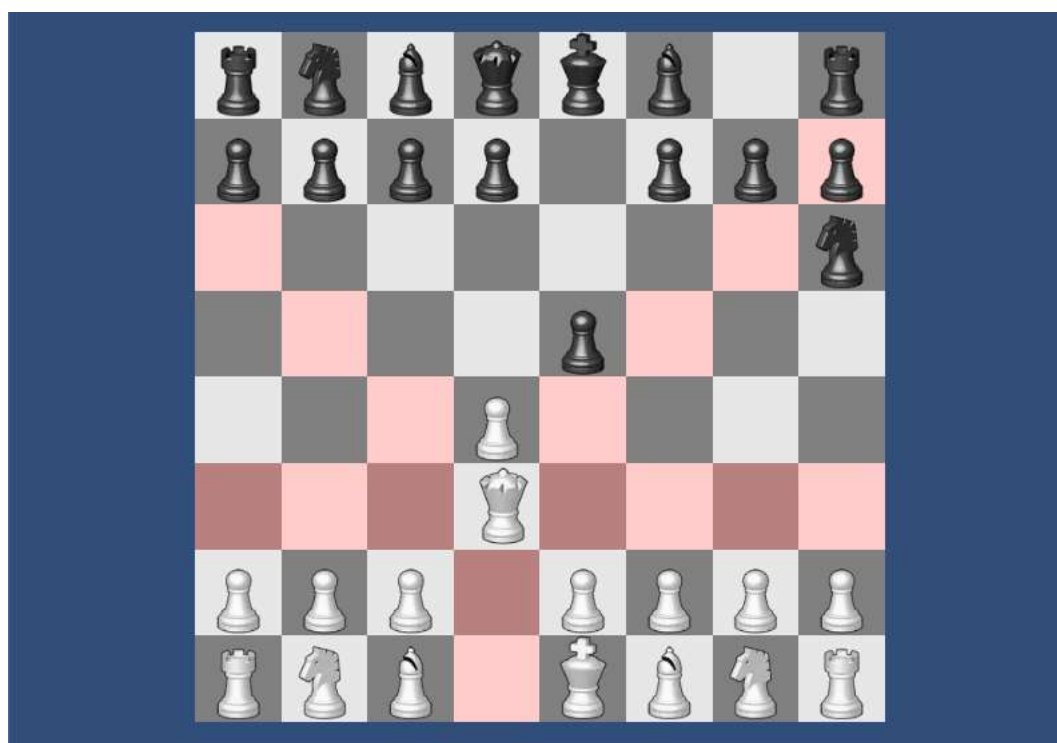
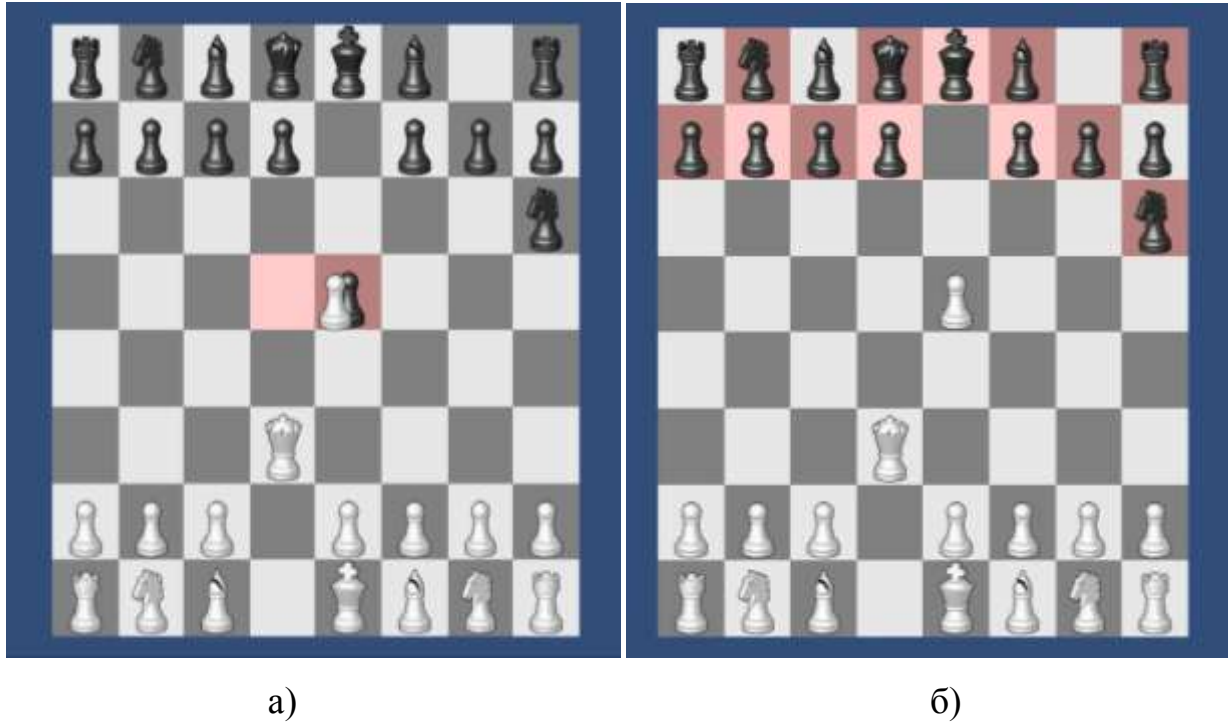


Рисунок 4.3 – Відображення можливих ходів ферзя

У випадку бажання користувача здійснити хід, який передбачає вбивство фігури суперника, програма відображає цей процес та його результат (рис. 4.4).



а- процес ходу; б - результат ходу.

Рисунок 4.4 – Процес та результат ходу користувача (гравець білими фігурами), що передбачає удар по фігурі суперника

4.5 Тестування програмного забезпечення для гри в шахи

Для тестування програмного забезпечення для гри в шахи було обрано такі функції програми: підключення до серверу; створення запису нової гри в базі; генерація можливих ходів; виконання ходу; оновлення бази даних; отримання поточної інформації про стан гри.

Тестування проводилося на комп'ютері з характеристиками, які відповідають технічному завданню.

Результати виконання модульних тестів наведено на рис. 4.5.

ChessConsole (тести: 7)	
✓ ChessConsoleTests (7)	439 мс
✓ ChessConsoleTests (7)	439 мс
✓ ChessConsoleTests (7)	439 мс
✓ ConnectionTest	15 мс
✓ createNewGame_Test	35 мс
✓ currentFigureMoves_Test	117 мс
✓ genereteAllMoves_Test	107 мс
✓ getGameInfo_Test	53 мс
✓ move_Test	67 мс
✓ updateDB_Test	58 мс
Сводка	
Last Test Run Пройден (Total Run Time 0:00:05,4276911)	
✓ Тести: 7 Пройден	

Рисунок 4.5 – Результати виконання модульних тестів

Результати модульного тестування програмного забезпечення для гри в шахи продемонстрували працездатність програми та відповідність отриманих даних очікуваним результатам.

Найбільш завантаженими функціями виявились генерація усіх можливих ходів та генерація ходів конкретної фігури, що вказує на необхідність їх оптимізації при подальшій доробці програми.

4.6 Висновки за розділом 4

Описано програмне забезпечення для гри в шахи. Виконано тестування розробленого програмного забезпечення для гри в шахи. Результати тестування програмного забезпечення для гри в шахи підтвердили працездатність програми та відповідність отриманих даних очікуваним результатам.

ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи бакалавра було проаналізовано та досліджено процес розробки програмного забезпечення для гри в шахи.

Визначено, що у наш час широко використовуються комп'ютерні ігри, що являють собою програмне забезпечення для розважання чи навчання людей, індустрія комп'ютерних ігор у наш час являє собою великий бізнес, існують мільйони різних комп'ютерних ігор, якими користуються люди будь-якого віку. Виявлено, що комп'ютерна гра, як правило, складається з таких програмних та апаратних компонентів: пристрої та програмні компоненти для введення, ігрова логіка, підтримка графіки/звучу, засоби виведення гри та засоби підтримки мережі. Визначено, що при реалізації комп'ютерної гри використовується два основних типи підтримки мультиплеєра – одноранговий та клієнт-серверний. Виконано класифікацію комп'ютерних ігор, показано, що зараз існує понад ста жанрів відеоігор, у яких користувач може грати та досліджувати, включаючи всі піджанри та додаткові жанри. Відзначено, що гра у шахи є однією з найстаріших та найпопулярніших настільних ігор у світі, яка є настільною грою між двома гравцями, що імітує війну між двома королівствами, та яка у наш час програмно реалізована у вигляді багатьох програмних застосунків.

За результатами проведеного аналізу зроблено висновок про те, що у наш час існує досить багато програмних систем для гри в шахи. Проте деякі з них не надають можливості грати з іншими суперниками, інші є платформозалежними та платними для користувачів. Тому актуальною є розробка програмного забезпечення для гри в шахи.

Сформульовано функціональні вимоги до програмного забезпечення для гри в шахи.

Для реалізації програмного забезпечення для гри в шахи обрано мову програмування C#, яка є об'єктно-орієнтованою мовою програмування, був

розроблена групою інженерів як основна мова для створення програм на платформі Microsoft .NET, має строгу типізацію, підтримує поліморфізм і перевантаження, є зручною для створення графічного інтерфейсу для взаємодії з користувачем.

В якості середовища розробки для створення програмного забезпечення для гри в шахи обрано Visual Studio, що містить набір інструментів розробки програмного забезпечення, підтримує мову програмування C#, добре працює та інтегрується зі сторонніми програмами, зокрема з Unity.

Для розробки програмного забезпечення для гри в шахи як ігровий рушій було обрано Unity, що є кросплатформним інструментом для розробки двовимірних та тривимірних комп'ютерних ігор.

Розроблено програмне забезпечення для гри в шахи. Запропоновано структуру процесів програмного забезпечення для гри в шахи. Визначено, що програмне забезпечення складається з клієнтської частини (модуль Chess2d) та серверної частини (модуль ChessApi та модулі для роботи з базою даних). Взаємозв'язок клієнтської та серверної частин забезпечується за допомогою бібліотеки chess.dll.

Для програмної реалізації, відповідно до технічного завдання, логіки гри в шахи, перевірки коректності виконуваних ходів було створено допоміжну бібліотеку Chess.dll, в якій реалізовано декілька класів. В класі Board описана шахова дошка та функції для розміщення фігур. В класі Moves описані усі правила, функції перевірки можливості виконання ходу та генерації списку можливих ходів. Клас FigureMoving описує переміщення фігур по клітинам шахової дошки. В класі FigureOnSquare описані шахові фігури та допоміжні функції для роботи з ними.

Модуль ChessApi створений для взаємодії клієнтської та серверної частин програм. Даний модуль реалізує шаблон MVC для вебзастосунків. В якості контролера виступають класи GameController та VersionController.

Класи Game та Logic реалізують модель гри. Клас ModelChessDB виконує роль представлення бази даних всередині програми.

Розроблено базу даних програмного забезпечення для гри в шахи. Основними сутностями бази даних є: Games, що містить інформацію про ігрові партії та їх статуси; Sides, що містить інформацію про сторони ігрової партії; Players, що містить інформацію про гравців.

Розроблено графічний інтерфейсу програмного забезпечення для гри в шахи, основним елементом якого є рисунок, що відображає шахову дошку.

Виконано тестування розробленого програмного забезпечення для гри в шахи. Результати тестування програмного забезпечення для гри в шахи підтвердили працездатність програми та відповідність отриманих даних очікуваним результатам.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Video Game Genres [Electronic resource]. – Access mode: <https://plarium.com/en/blog/video-game-genres/>.
2. Game [Electronic resource]. – Access mode: <https://www.computerhope.com/jargon/g/game.htm>.
3. Computer Games [Electronic resource]. – Access mode: [https://www3.ntu.edu.sg/home/asschui/Computer Games.PDF](https://www3.ntu.edu.sg/home/asschui/Computer%20Games.PDF).
4. Drawbacks & Benefits of Computer Games [Electronic resource]. – Access mode: <https://www.hitechwhizz.com/2021/05/5-advantages-and-disadvantages-drawbacks-benefits-of-computer-games.html>.
5. 12 Types Of Computer Games [Electronic resource]. – Access mode: <https://thoughtcatalog.com/jane-hurst/2015/02/12-types-of-computer-games-every-gamer-should-know-about/>.
6. The Definitive Guide to Video Game Genres and Game Types [Electronic resource]. – Access mode : <https://codakid.com/video-game-genres/>.
7. Chess Terms [Electronic resource]. – Access mode: <https://www.chess.com/terms/chess>.
8. Chess Arena [Electronic resource]. – Access mode: <https://chessarena.com/>.
9. BabasChess [Electronic resource]. – Access mode: <https://prosoftpedia.com/babaschess/>.
10. DecodeChess [Electronic resource]. – Access mode: <https://fixthephoto.com/best-free-chess-software.html>.
11. Komodo Chess Engine [Electronic resource]. – Access mode: <https://komodochess.com/>.
12. C# programming guide [Electronic resource]. – Access mode: <https://learn.microsoft.com/en-us/dotnet/csharp/programming-guide/>.
13. C# Tutorial [Electronic resource]. – Access mode: <https://www.tutorialspoint.com/csharp/index.htm>.

14. Learn C# Programming [Electronic resource]. – Access mode: <https://www.tutorialsteacher.com/csharp>.

15. It's how you make software [Electronic resource]. – Access mode: <https://visualstudio.microsoft.com/>.

16. Basic Information And Overview Of Visual Studio [Electronic resource]. – Access mode: <https://www.c-sharpcorner.com/article/basic-information-and-overview-of-visual-studio-2017/>.

17. Unity [Electronic resource]. – Access mode : <https://unity.com>.

ДОДАТОК А
Технічне завдання

Вступ

Програмне забезпечення у вигляді комп'ютерної гри може використовуватися для підтримки процесу гри в шахи.

A.1 Підстава для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Програмне забезпечення для гри в шахи», затверджене наказом Національного університету «Запорізька політехніка» № 87 від 4 квітня 2023 р.

A.2 Призначення розробки

Програмний продукт призначений для підтримки процесу гри в шахи у вигляді комп'ютерної гри.

A.3 Основні вимоги до програми, що розробляється

A.3.1 Вимоги до функціональних характеристик

Програмне забезпечення для підтримки процесу гри забезпечує виконання таких функцій:

- підтримка процесу гри користувача у шахи;
- можливість пересувати фахові фігури та робити ходи згідно правил (перевірка ходів користувача на відповідність правилам гри у шахи);
- надання підказок щодо можливих ходів;
- реалізовувати гру шляхом запитів на сервер;
- підтримка звернень та збережень даних у базі даних;
- підтримка багатокористувальницької гри між різними користувачами;

– виведення повідомлень користувачу у випадку виникнення критичних ситуацій.

А.3.2 Вимоги до інтерфейсу програми

Інтерфейс програми для гри в шахи повинен бути зручним для користувачів та містити імітацію шахової дошки з шаховими фігурами, розташованими відповідно до правил гри. Взаємодія між програмою та користувачем повинна бути організована через зручні візуальні засоби.

А.3.3 Вимоги до надійності

Програмне забезпечення для гри в шахи повинно забезпечити надійне функціонування та передбачити можливість запобігання некоректних ситуацій.

А.3.4 Умови експлуатації

Для експлуатації програмного забезпечення для гри в шахи необхідна наявність персонального комп'ютеру.

А.3.5 Вимоги до складу та параметрів технічний засобів

Для експлуатації програмного забезпечення для гри в шахи необхідно таке апаратне та програмне забезпечення:

- тактова частота мікропроцесора не менше 2,3 ГГц;
- оперативна пам'ять з мінімальним об'ємом 8 Гб;
- операційна система Windows 10 або вище;

– на стороні клієнта підтримка платформи .NET Framework, починаючи з версії 4.0.

A.3.6 Вимоги до маркування і пакування

Програма для гри в шахи може бути записана на будь-якому носії інформації. На пакуванні повинна бути назва програми – «Програмне забезпечення для гри в шахи».

A.4 Вхідні дані до роботи

Вхідними даними до програмного забезпечення для гри в шахи є інформація про рух фігур на гральній дошці та інформація з таблиць бази даних.

Вихідними даними програмного забезпечення для гри в шахи є відображення можливих ходів, пересування фігур або неможливість їх пересування.

ДОДАТОК Б
Фрагмент тексту програми

```
namespace ChsRls
```

```
{
    enum Fggr
    {
        none,

        whtKng = 'K',
        whtQn = 'Q',
        whtRk = 'R',
        whtBshp = 'B',
        whtKnght = 'N',
        whtPwn = 'P',

        blkKng = 'k',
        blkQnn = 'q',
        blkRkk = 'r',
        blkBshp = 'b',
        blkKnght = 'n',
        blkPwn = 'p',
    }
}
```

```
struct Sqrrr
```

```
{
    public static Sqrrr none = new Sqrrr(-1, -1);

    public int xx { get; private set; }
    public int yy { get; private set; }

    public Sqrrr(int xx, int yy)
    {
        this.xx = xx;
        this.yy = yy;
    }

    public Sqrrr (string eE2)
    {
        if (eE2.Length == 2 &&
            eE2[0] >= 'a' && eE2[0] <= 'h' &&
            eE2[1] >= '1' && eE2[1] <= '8')
        {
            xx = eE2[0] - 'a';

```

```

        yy = eE2[1] - '1';
    }
    else
        this = none;
}

public bool OnBrrd ()
{
    return xx >= 0 && xx < 8 &&
        yy >= 0 && yy < 8 ;
}

public string Name { get { return ((char)('a' + xx)).ToString()
+ (yy + 1).ToString(); } }

public static bool operator ==(Sqrrr aa, Sqrrr bb)
{
    return aa.xx == bb.xx && aa.yy == bb.yy;
}

public static bool operator !=(Sqrrr aa, Sqrrr bb)
{
    return aa.xx != bb.xx || aa.yy != bb.yy;
}

public static IEnumerable<Sqrrr> YieldSqrrrs()
{
    for (int yy = 0; yy < 8; yy++)
        for (int xx = 0; xx < 8; xx++)
            yield return new Sqrrr(xx, yy);
}
}

class FggrMvng
{
    public Fggr Fggr { get; private set; }
    public Sqrrr Frmm { get; private set; }
    public Sqrrr to { get; private set; }
    public Fggr prmtn { get; private set; }

    public FggrMvng (FggrOnSqrrr ffss, Sqrrr tto, Fggr prmtn =
Fggr.none)
    {
        this.Fggr = ffss.Fggr;
    }
}

```

```

        this.Frmm = ffss.Sqrrr;
        this.tto = tto;
        this.prmtn = prmtn;
    }

    public FggrMvng(string mvv)
    {
        this.Fggr = (Fggr)mvv[0];
        this.Frmm = new Sqrrr(mvv.Substring(1, 2));
        this.tto = new Sqrrr(mvv.Substring(3, 2));
        this.prmtn = (mvv.Length == 6) ? (Fggr)mvv[5] : Fggr.none;
    }

    public int DltX { get { return tto.xx - Frmm.xx; } }
    public int DltY { get { return tto.yy - Frmm.yy; } }

    public int AbsDltX { get { return Math.Abs(DltX); } }
    public int AbsDltY { get { return Math.Abs(DltY); } }

    public int SgnX { get { return Math.Sgn(DltX); } }
    public int SgnY { get { return Math.Sgn(DltY); } }

    public override string ToString()
    {
        string txt = (char)Fggr + Frmm.Nmm + to.Nmm;
        if (prmtn != Fggr.none)
            txt += (char)prmtn;
        return txt;
    }
}

class FggrOnSqrrr
{
    public Fggr Fggr { get; private set; }
    public Sqrrr Sqrrr { get; private set; }

    public FggrOnSqrrr(Fggr Fggr, Sqrrr Sqrrr)
    {
        this.Fggr = Fggr;
        this.Sqrrr = Sqrrr;
    }
}

class mvvs

```

```

{
    FggrMvng fm;
    Brrd brrd;

    public mvvs(Brrd brrd)
    {
        this.brrd = brrd;
    }

    public bool CnnMvv(FggrMvng ffm)
    {
        this.ffm = ffm;
        return
            CnnMvvFrmm() &&
            CnnMvvTo() &&
            CnnFggrMvv();
    }

    bool CnnMvvFrmm()
    {
        return fm.Frmm.OnBrrd() &&
            fm.Fggr.GetColor() == brrd.MvvColor;
    }

    bool CnnMvvTo()
    {
        return fm.to.OnBrrd() &&
            fm.Frmm != fm.to &&
            brrd.GetFggrAt(fm.to).GetColor() != brrd.MvvColor;
    }

    bool CnnFggrMvv()
    {
        switch (fm.Fggr)
        {
            case Fggr.whtKng:
            case Fggr.blkKng:
                return CnnKngMvv();

            case Fggr.whtQn:
            case Fggr.blkQnn:
                return CnStrghtMv();
        }
    }
}

```

```

        case Fggr.whtRk:
        case Fggr.blkRkk:
            return (fm.SgnX == 0 || fm.SgnY == 0) &&
                CnStrghtMv();

        case Fggr.whtBshp:
        case Fggr.blkBshp:
            return (fm.SgnX != 0 && fm.SgnY != 0) &&
                CnStrghtMv();

        case Fggr.whtKnght:
        case Fggr.blkKnght:
            return CnnKnghtMvv();

        case Fggr.whtPwn:
        case Fggr.blkPwn:
            return CnnPwnMvv();

        default: return false;
    }
}

bool CnnPwnMvv()
{
    if (fm.Frmm.y < 1 || fm.Frmm.y > 6)
        return false;
    int stpY = fm.Fggr.GetColor() == Color.white ? 1 : -1;
    return
        CnnPwnGo(stpY) ||
        CnnPwnJmp(stpY) ||
        CnnPwnEat(stpY);
}

private bool CnnPwnGo(int stpY)
{
    if (brrd.GetFggrAt(fm.to) == Fggr.none)
        if (fm.DltX == 0)
            if (fm.DltY == stpY)
                return true;
    return false;
}

```

```

    }

private bool CnnPwnJmp(int stpY)
{
    if (brrd.GetFggrAt(fm.to) == Fggr.none)
        if (fm.DltX == 0)
            if (fm.DltY == 2 * stpY)
                if (fm.Frmm.y == 1 || fm.Frmm.y == 6)
                    if (brrd.GetFggrAt(new Sqrrr(fm.Frmm.x,
fm.Frmm.y + stpY)) == Fggr.none)
                        return true;
                return false;
    }

private bool CnnPwnEat(int stpY)
{
    if (brrd.GetFggrAt(fm.to) != Fggr.none)
        if (fm.DltX == 1)
            if (fm.DltY == stpY)
                return true;
        return false;
    }

private bool CnStrghtMv()
{
    Sqrrr att = fm.Frmm;
    do
    {
        att = new Sqrrr(att.xx + fm.SgnX, att.yy + fm.SgnY);
        if (att == fm.to)
            return true;
    } while (att.OnBrrd() &&
        brrd.GetFggrAt(att) == Fggr.none);
    return false;
}

private bool CnnKnghtMvv()
{
    if (fm.AbsDltX == 1 && fm.AbsDltY == 2) return true;
    if (fm.AbsDltX == 2 && fm.AbsDltY == 1) return true;
    return false;
}

```

```

    }

    private bool CnnKngMvv()
    {
        if (fm.AbsDltX <= 1 && fm.AbsDltY <= 1)
            return true;
        return false;
    }
}

class Brrd
{
    public string fnnn { get; private set; }
    Fggr[,] Fggrs;
    public Color mvvColor { get; private set; }
    public int mvvNmbr { get; private set; }

    public Brrd(string fnn)
    {
        this.fnn = fnn;
        Fggrs = new Fggr[8, 8];
        Init();
    }

    void Init()
    {
        string[] prts = fen.Split();
        if (prts.Length != 6) return;
        InitFggrs(prts[0]);
        mvvClr = (prts[1] == "b") ? Color.blk : Color.white;
        mvvNmbr = int.Parse(prts[5]);
    }

    void InitFggrs(string dtta)
    {
        for (int j = 8; j >= 2; j--)
            dtta = dtta.Replace(j.ToString(), (j - 1).ToString() +
"1");

        dtta = dtta.Replace("1", ".");
        string[] lines = dtta.Split('/');
        for (int yy = 7; yy >= 0; yy--)
            for (int xx = 0; xx < 8; xx++)

```

```

        Fggrs[x, y] = lines[7 - y][x] == '.' ? Fggr.none :
            (Fggr)lines[7 - y][x];
    }

    void GnrtFnn()
    {
        fnn = FnnFggrs() + " " +
            (MvvColor == Color.white ? "w" : "b") +
            " - - 0 " + mvvNumber.ToString();
    }

    public IEnumerable<FggrOnSqrr> YieldFggrs()
    {
        foreach (Sqrr Sqrr in Sqrr.YieldSqrrs())
            if (GetFggrAt(Sqrr).GetColor() == mvvColor)
                yield return new FggrOnSqrr(GetFggrAt(Sqrr),
Sqrr);
    }

    string FnnFggrs()
    {
        StringBuilder sbb = new StringBuilder();
        for(int yy = 7; yy >=0; yy--)
        {
            for (int xx = 0; xx < 8; xx++)
                sbb.Append(Fggrs[xx, yy] == Fggr.none ? '1' :
(char)Fggrs[xx, yy]);
            if (yy > 0)
                sbb.Append('/');
        }
        string egght = "11111111";
        for (int jj = 8; jj > 2; jj--)
            sbb.Replace(egght.Substring(0, jj), jj.ToString());
        return sbb.ToString();
    }

    public Fggr GetFggrAt (Sqrr Sqrr)
    {
        if (Sqrr.OnBrdd())
            return Fggrs[Sqrr.x, Sqrr.y];
        return Fggr.none;
    }

    void SetFggrAt(Sqrr Sqrr, Fggr Fggr)

```

```

{
    if (Sqrrr.OnBrrd())
        Fggrs[Sqrrr.x, Sqrrr.y] = Fggr;
}
public Brrd mvv (FggrMvng fm)
{
    Brrd nxt = new Brrd(fen);
    nxt.SetFggrAt(fm.Frmm, Fggr.none);
    nxt.SetFggrAt(fm.to, fm.prmtn == Fggr.none ? fm.Fggr :
fm.prmtn);
    if (MvvColor == Color.blk)
        nxt.MvvNumber++;
    nxt.MvvColor = mvvColor.FlipColor();
    nxt.GnrtFen();
    return nxt;
}
bool CnnEatKng()
{
    Sqrrr badKng = FindBadKng();
    mvvs mvvs = new mvvs(this);
    foreach (FggrOnSqrrr ffss in YieldFggrs())
    {
        FggrMvng fm = new FggrMvng(fs, badKng);
        if (Mvvs.CnnMvv(fm))
            return true;
    }
    return false;
}

private Sqrrr FndBdKng()
{
    Fggr bdKng = mvvColor == Color.blk ? Fggr.whtKng :
Fggr.blkKng;
    foreach (Sqrrr Sqrrr in Sqrrr.YieldSqrrrs())
        if (GetFggrAt(Sqrrr) == bdKng)
            return Sqrrr;
    return Sqrrr.none;
}

public bool IsChck()
{
    Brrd after = new Brrd(fen);
    after.MvvColor = mvvColor.FlipColor();

```

```

        return after.CnnEatKng();
    }

    public bool IsChckAftrMvv (FggrMvng fm)
    {
        Brrd aftr = mvv(fm);
        return aftr.CnnEatKng();
    }
}

public class Chs
{
    public string fnn { get; private set; }
    Brrd brrd;
    mvvs mvvs;
    List<FggrMvng> allMvvs;

    public Chs(string fen =
"rnbqkbnr/pppppppp/8/8/8/8/PPPPPPPP/RNBQKBNR w KQkq - 0 1")
    {
        this.fnn = fnn;
        brrd = new Brrd(fnn);
        mvvs = new mvvs(brrd);
    }

    Chs(Brrd brrd)
    {
        this.brrd = brrd;
        fnn = brrd.fnn;
        mvvs = new mvvs(brrd);
    }

    public Chs mvv (string mvv)
    {
        FggrMvng fm = new FggrMvng(Mvv);
        if (!Mvvs.CnnMvv(fm))
            return this;
        if(brrd.IsCheckAfterMvv(fm))
            return this;
        Brrd nxtBrrd = brrd.Mvv(fm);
        Chs nxtChs = new Chs(nxtBrrd);
        return nxtChs;
    }
}

```

```

    }

    public char GetFggrAt (int xx, int yy)
    {
        Sqrrr Sqrrr = new Sqrrr(xx, yy);
        Fggr ff = brrd.GetFggrAt(Sqrrr);
        return ff == Fggr.none ? '.' : (char)f;
    }

    void FndAllMvvs()
    {
        allMvvs = new List<FggrMvng>();
        foreach (FggrOnSqrrr ffss in brrd.YieldFggrs())
            foreach (Sqrrr to in Sqrrr.YieldSqrrrs())
            {
                FggrMvng fm = new FggrMvng(fs, to);
                if (Mvvs.CnnMvv(fm))
                    if(!brrd.IsCheckAfterMvv(fm))
                        allMvvs.Add(fm);
            }
    }

    public List<string> GtAlMvvs()
    {
        FindAllMvvs();
        List<string> lst = new List<string>();
        foreach (FggrMvng fmm in allMvvs)
            list.Add(fmm.ToString());
        return lst;
    }

}

```

ДОДАТОК В
Слайди презентації

Міністерство освіти і науки України
Національний університет «Запорізька політехніка»

Програмне забезпечення для гри в шахи

КАБАНЕНКО Євген

ст. групи КНТ-229

ОЛІЙНИК Андрій

д.т.н., професор НУ «Запорізька політехніка»

Рисунок В.1 – Слайд 1

Об'єкт, предмет та мета роботи

Об'єкт дослідження – процес розробки програмного забезпечення для гри в шахи.

Предмет дослідження – програмні засоби для гри в шахи.

Метою роботи є розробка програмного забезпечення для гри в шахи.

2

Рисунок В.2 – Слайд 2

Завдання роботи

Для досягнення мети роботи потрібно розв'язати такі завдання:

- виконати аналіз предметної області та програмних засобів для гри в шахи;
- здійснити проектування програмного забезпечення для гри в шахи;
- створити програмне забезпечення для підтримки процесу гри в шахи;
- виконати тестування розробленого програмного забезпечення для гри в шахи.

3

Рисунок В.3 – Слайд 3

Порівняння існуючих аналогів

Критерій порівняння	Arena	BabasChess	DecodeChess	Komodo Chess Engine
Онлайн версія	+	–	+	+
Необхідність оплати	+–	–	–	–
Можливість грати з іншими суперниками	+	–	+	+
Україномовна версія	–	–	–	–
Платформонезалежність	+	–	+	+

4

Рисунок В.4 – Слайд 4

Порівняння мов програмування

Критерій порівняння мов програмування	Мова програмування		
	C#	Java	Php
Зручність для створення програмного забезпечення для гри в шахи	+	+–	+–
Автоматичний збиральник сміття	+	+	–
Динамічна типізація	+	+	–
Зручність створення графічного інтерфейсу користувача	+	+–	+–
Категорії змінних	+	+	–
Простота використання	+	+–	+–

5

Рисунок В.5 – Слайд 5

Обґрунтування вибору середовища розробки

Критерій порівняння середовищ розробки	Середовище розробки		
	Visual Studio	SharpDevelop	CodeLite
Необхідність оплати за користування	–	–	+
Наявність візуального конструктору	+	+	–
Зручність використання при розробці програми для гри у шахи	+	+–	+–
Багатоплатформеність	+	–	+
Підсвічування коду	+	–	+

6

Рисунок В.6 – Слайд 6

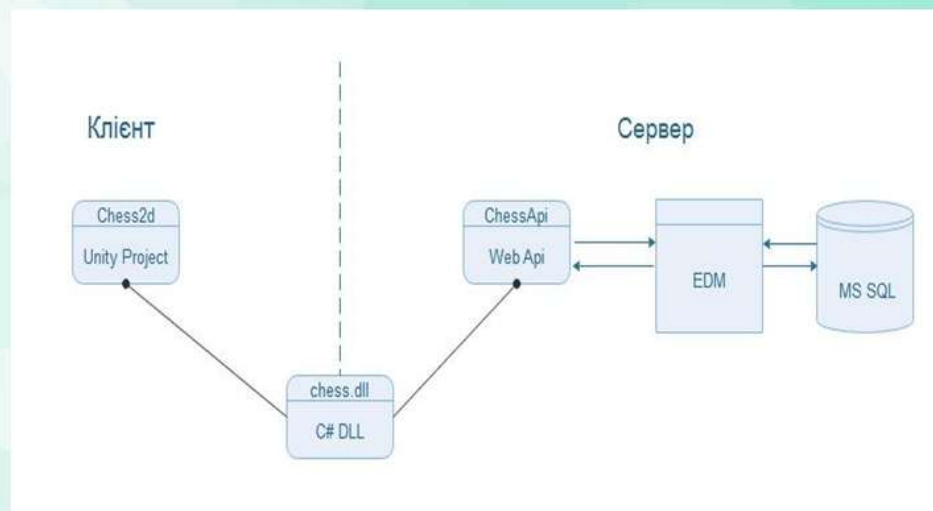
Обґрунтування вибору ігрового рушія

Критерій порівняння ігрових рушіїв	Ігрові рушії		
	Unity	Project Anarchy	Unreal Engine
Легкість використання	+	–	+
Підтримка розробників	+	–	+
Зручність налаштування	+	+–	+–
Підтримувані скриптові мови	C#, JavaScript	–	C++
Зручність для створення програмного забезпечення для гри в шахи	+	–	–

7

Рисунок В.7 – Слайд 7

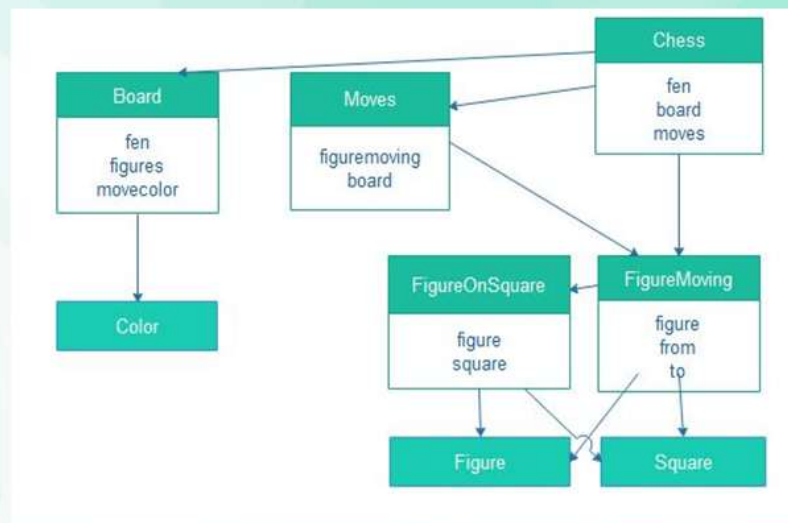
Структура процесів програмного забезпечення для гри в шахи



8

Рисунок В.8 – Слайд 8

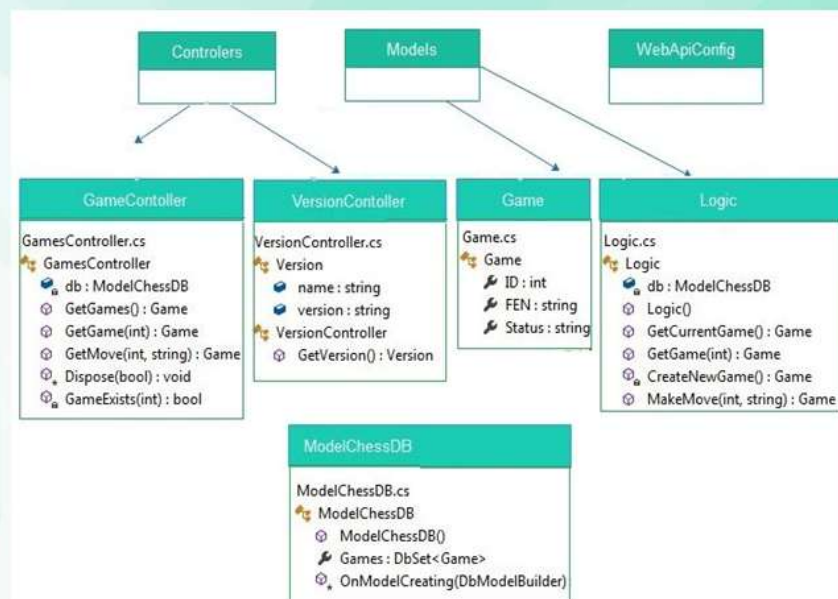
Діаграма класів модуля Chess



9

Рисунок В.9 – Слайд 9

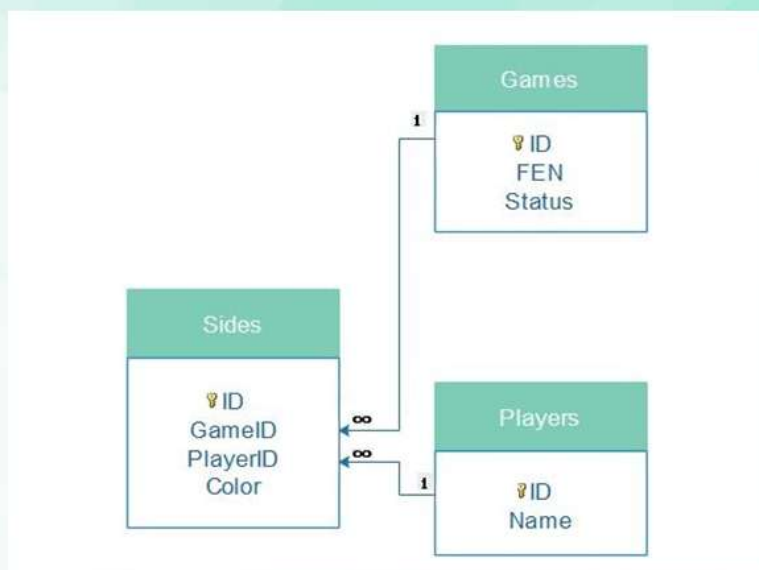
Діаграма класів модуля ChessApi



10

Рисунок В.10 – Слайд 10

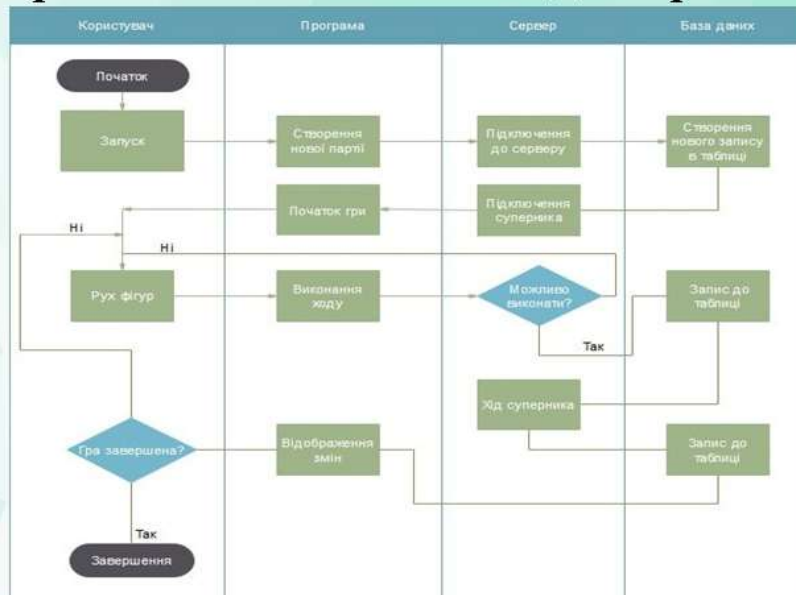
Схема бази даних



11

Рисунок В.11 – Слайд 11

Діаграма послідовностей програмного забезпечення для гри в шахи



12

Рисунок В.12 – Слайд 12

Робота з програмою



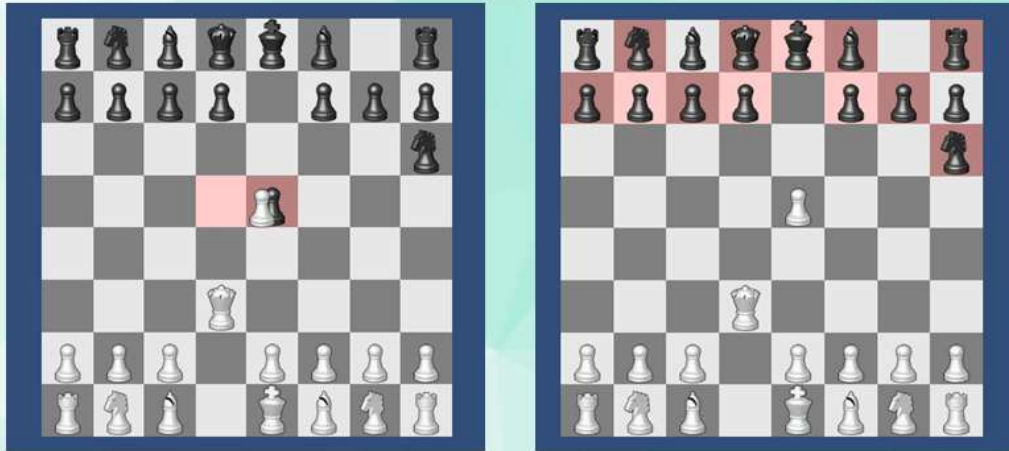
Рисунок В.13 – Слайд 13

Робота з програмою. Відображення можливих ходів



Рисунок В.14 – Слайд 14

Робота з програмою. Процес та результат ходу користувача



15

Рисунок В.15 – Слайд 15

Висновки

В ході виконання дипломної кваліфікаційної роботи бакалавра було проаналізовано та досліджено процес розробки програмного забезпечення для гри в шахи.

Розроблено програмне забезпечення для гри в шахи забезпечує виконання таких функцій:

- підтримка процесу гри користувача у шахи;
- можливість пересувати фахові фігури та робити ходи згідно правил ;
- надання підказок щодо можливих ходів;
- реалізовувати гру шляхом запитів на сервер;
- підтримка звернень та збережень даних у базі даних;
- підтримка багатокористувальницької гри між різними користувачами;
- виведення повідомлень користувачу у випадку виникнення критичних ситуацій.

Результати тестування програмного забезпечення для гри в шахи показали, що розроблена програма працює коректно та може використовуватися за призначенням.

16

Рисунок В.16 – Слайд 16