**Санкт-Петербургский государственный университет**

**Факультет прикладной математики и процессов управления**

**ОТЧЁТ**

**по дисциплине “Системное программирование в Linux”**

**о создании шахматной программы с поиском**

**лучших ходов**

Автор работы: Леонов А.А. Группа: 22.Б16-пу

Преподаватель: Киямов Ж. У.

Санкт-Петербург

2023 г.

**Аннотация**

Программа представляет собой реализацию шахматной игры, разработанную на языке C++. Она позволяет пользователям узнать лучшие ходы, возможные из самых разнообразных позиций на доске. Результаты представлены в виде трех последовательностей ходов.

**Содержание**

Оглавление

[**Аннотация** 2](#_Toc147532013)

[**1. Введение** 4](#_Toc147532014)

[**2. Цель работы** 4](#_Toc147532015)

[**3. Задачи** 4](#_Toc147532016)

[**4. Техническое описание** 5](#_Toc147532017)

[4.1 Используемые инструменты и библиотеки 5](#_Toc147532018)

[**5 Описание программы** 6](#_Toc147532019)

[5.1 Описание ключевых компонентов программы 6](#_Toc147532020)

[5.2 Общий ход программы 7](#_Toc147532021)

[**6. Рекомендации пользователю** 8](#_Toc147532022)

[**7. Рекомендации программисту** 8](#_Toc147532023)

[**8. Вывод** 9](#_Toc147532024)

[**Библиография** 11](#_Toc147532025)

**1. Введение**

Шахматы - одна из старейших и наиболее популярных интеллектуальных настольных игр, которая требует от игроков стратегического мышления и логических навыков. Программа создана с целью предоставить пользователям возможность исследовать оптимальные стратегии для завершения игры в пользу белых фигур. Это позволяет понять, какие ходы и варианты могут привести к наилучшим результатам в шахматах.

**2. Цель работы**

Целью данной работы является создание программы, которая вычисляет оптимальное завершение игры для белых фигур и выводит 3 возможных лучших хода, максимизируя количество очков.

**3. Задачи**

Для достижения цели работы были поставлены следующие задачи:

1. Создать шахматную доску и расставить фигуры на начальные позиции.
2. Реализовать правила движения различных фигур (пешек, коней, слонов, ладей, королей).
3. Предоставить возможность пользователям вводить количество ходов в игре.
4. Имитировать ходы белых и черных фигур в случайном порядке до завершения игры.
5. Вычислить наилучшее завершение игры для белых фигур.

**4. Техническое описание**

**4.1 Используемые инструменты и библиотеки**

Для разработки программы были использованы следующие инструменты и библиотеки:

Язык программирования: C++;

Библиотеки: iostream, string, vector, random, algorithm.

**4.2 Структура программы**

Программа состоит из следующих основных компонентов:

1. Piece - Структура, представляющая фигуру на шахматной доске. Содержит информацию о названии фигуры, цвете и ее позиции на доске.
2. ChessBoard - Класс, представляющий шахматную доску. Содержит вектор векторов для хранения фигур на доске, а также методы для их перемещения, проверки правил и вывода доски на экран.
3. Основная функция main() - В этой функции создается объект ChessBoard, инициализируется начальное состояние доски, и пользователь вводит количество ходов. Затем программа имитирует ходы фигур, вычисляет оптимальное завершение игры и выводит результат на экран.

**5** **Описание программы**

**5.1 Описание ключевых компонентов программы**

В таблице 1 представлено описание ключевых компонентов программы.

Таблица 1. Ключевые компоненты

|  |  |
| --- | --- |
| Компонент | Описание |
| Piece | Структура, представляющая шахматную фигуру с полями для имени, цвета, а также координаты по горизонтали и вертикали на доске. |
| ChessBoard | Класс, представляющий шахматную доску и логику игры. Включает в себя векторы фигур для белых и черных игроков, методы для расстановки фигур, проверки возможных ходов и вывода доски на экран. |
| main | Главная функция программы, которая создает объект ChessBoard, инициализирует начальное состояние доски, имитирует ходы игроков, оптимизирует завершение игры для белых фигур и выводит результат на экран. |
| std::random\_device и std::mt19937 | Библиотеки для генерации случайных чисел, используемые для перемешивания фигур и ходов. |
| winmoves1, winmoves2, winmoves3 | Векторы для хранения оптимальных завершений игры для белых фигур. |
| t | Флаг для управления циклом игры и определения завершения. |
| steps | Переменная для хранения количества ходов, введенных пользователем. |

**5.2 Общий ход программы**

Общий ход программы можно описать следующим образом:

1. Начало игры:
   * Создается объект ChessBoard, представляющий шахматную доску.
   * Доска инициализируется начальным состоянием, где фигуры расставлены на своих начальных позициях.
2. Ввод количества ходов:
   * Пользователю предлагается ввести количество ходов (steps), которое определяет, сколько ходов будет имитировано в игре.
3. Имитация ходов:
   * Запускается цикл игры (while (t)), который продолжается до завершения игры. Флаг t управляет циклом.
   * Игра чередует ходы между белыми и черными фигурами. Каждый ход имитируется в соответствии с правилами шахмат.
   * Если наступает мат или пат (когда одному из игроков не остается ходов), игра завершается.
4. Оптимизация завершения для белых фигур:
   * После завершения игры программа вычисляет оптимальное завершение для белых фигур. Оптимизация осуществляется путем анализа различных вариантов ходов.
   * Вычисляется счет (количество очков) для каждого завершения игры.
   * Завершение, приносящее наибольший счет, считается оптимальным.
5. Вывод результатов:
   * Результаты игры выводятся на экран, включая текущее состояние доски и оптимальное завершение для белых фигур.
   * Информация о завершении игры и ходах выводится пользователю для анализа.
6. **Рекомендации пользователю**

**Ввод количества ходов:** При запуске программы введите желаемое количество ходов для имитации. Оно будет определять продолжительность игры.

**Отслеживайте ход игры:** Программа автоматически имитирует ходы игроков, и вам не требуется вводить собственные ходы. Однако, изучите на каких позициях стоят фигуры, чтобы удостовериться в лучших ходах.

**Изучайте оптимальное завершение:** После завершения игры, обратите внимание на оптимальное завершение для белых фигур. Это может дать вам представление о том, как можно стратегически завершить игру.

**7.** **Рекомендации программисту**

**Запуск программы:** Для запуска программы на Linux, убедитесь, что у вас установлен компилятор C++.

**Компиляция программы:** Откройте терминал в вашей системе Linux и перейдите в директорию, где находится файл с исходным кодом программы с помощью команды cd.

**Компилируйте программу:** Когда вы скомпилируете вашу программу, у вас появится исполняемый файл, который необходимо запустить.

**Ввод данных:** При выполнении программы у вас будет возможность ввести количество ходов (steps), которое определит продолжительность имитации игры.

**Отслеживайте выполнение программы:** Программа автоматически имитирует ходы игроков и выводит на экран результаты. Обратите внимание на вывод программы и результаты оптимизации завершения игры.

**8. Вывод**

В ходе выполнения данной работы была разработана программа, имитирующая игру в шахматы с целью узнать лучшие возможные ходы из самых разнообразных позиций. Программа включает в себя структуры данных и классы, представляющие шахматную доску и фигуры, а также функции для генерации и оптимизации ходов виртуальных игроков. Используется случайный выбор ходов и вычисление оптимальных последовательностей для определенных сценариев игры.

Основной целью программы было найти оптимальные ходы для завершения партии с минимальным количеством ходов при заданных правилах шахмат. Для достижения этой цели был использован жадный алгоритм, который исследует различные комбинации ходов и выбирает наилучшие из них.

Данная работа демонстрирует важность алгоритмов оптимизации в контексте игровой интеллектуальной обработки данных и может служить отправной точкой для дальнейших исследований в области оптимизации шахматных стратегий.

**Библиография**

GitHub с реализацией игры - https://github.com/Krikker/Chess