

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ

ОТЧЕТ
по лабораторной работе
по дисциплине «Системное программирование в Linux»
на тему «Алгоритм нахождения лучшего шахматного хода»

Студент гр. 22.Б15-пу

Осовский Н.С

Преподаватель

Киямов Ж.У.

Санкт-Петербург
2023 г.

Оглавление

1. Цель работы	3
2. Задача	3
3. Описание алгоритма	3
4. Описание программы	4
4.1. Описание структуры	4
5. Рекомендации пользователя	4
6. Рекомендации программиста	4
7. Вывод	4
Приложение	5

1. Цель работы

Данная работа направлена на создание программы, которая способна определить наилучшие ходы для белых игроков, чтобы максимизировать общее количество очков на шахматной доске. Это важно как для улучшения стратегического мышления в шахматах, так и для развития игровых алгоритмов.

2. Задача

1. Инициализация шахматной доски и размещение фигур на начальных позициях для белых и черных игроков.
2. Определение возможных ходов для каждой фигуры, учитывая правила шахмат.
3. Оценка веса каждой фигуры на доске в соответствии с их значимостью: пешка - 1, слон и конь - 3, ладья - 5, король - 1.
4. Поиск оптимальных последовательностей ходов для обоих игроков с учетом максимизации суммарного веса фигур на доске.
5. Отображение результатов в виде последовательностей ходов и общего количества очков.

3. Описание алгоритма

1. Создание экземпляра класса ChessBoard и инициализация шахматной доски и фигур в методе make().
2. Вывод начального расположения фигур на доске с помощью метода printBoard().
3. Ввод пользователем количества ходов для выполнения в основном игровом цикле.
4. В цикле происходит перемещение фигур поочередно для белых и черных, фигуры перемешиваются случайным образом и выбирают один из доступных ходов с помощью метода move(). Если фигура была взята противником, она снимается с доски.
5. Для оптимизации ходов белых фигур после случайных ходов происходит анализ последовательности ходов на основе весовых характеристик фигур. Оптимальная последовательность ходов и оценка их эффективности сохраняются.
6. Анализ текущего состояния доски и фигур производится в конце игры. Если на доске не осталось фигур обоих цветов, игра завершается. Если короли противников не находятся под боем, игра также завершается.

7. Выводится оптимальная последовательность ходов для белых фигур и оценка их эффективности.

4. Описание программы

4.1. Описание структуры

Таблица 5.1. Описание

Имя	Описание
Piece (Структура)	Представляет фигуру на шахматной доске.
ChessBoard (Класс)	Управляет состоянием шахматной доски и фигурами на ней.
ChessBoard.make()	Инициализирует фигуры на доске
ChessBoard.printBoard()	Выводит текущее состояние доски
ChessBoard.move()	<p>На вход принимает фигуру</p> <p>Функция move() принимает объект Piece в качестве входного параметра и возвращает вектор векторов, содержащий все возможные ходы для этой фигуры на текущей доске. Сначала она проверяет тип фигуры (пешка, ладья, конь, слон, ферзь), а затем генерирует все возможные ходы для этой фигуры на основе ее положения и текущего состояния доски.</p> <p>Для пешек она проверяет возможные ходы вперед, а также диагональные захваты.</p> <p>Для ладей она проверяет возможные ходы по строкам и столбцам.</p> <p>Для коней она генерирует все возможные ходы в форме буквы L.</p> <p>Для слонов она проверяет возможные ходы по диагоналям.</p> <p>Для ферзей она объединяет возможные ходы ладей и слонов.</p> <p>Функция возвращает вектор всех возможных ходов.</p>
main (Главная функция)	<p>Данный код отвечает за управление выполнением программы. Он создает экземпляр класса ChessBoard, который инициализирует игровую доску и фигуры. Затем происходит выполнение ходов фигур в случайном порядке. После этого производится анализ и оптимизация последовательности ходов для белых фигур.</p>

	Оптимизирует ходы для белых фигур учитывая веса и выводит лучшие ходы найденные на доске
--	---------------------------------------------------------------------------------------------

5. Рекомендации пользователя

Изучите вывод программы и поэкспериментируйте с количеством ходов.

6. Рекомендации программиста

Для запуска программы необходим C++ 17.

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

7. Вывод

Была разработана функциональная система для анализа игры в шахматы, которая моделирует игру и анализирует ходы на основе стратегии. Программа реализована на языке C++ и использует стандартную библиотеку STL. Она позволяет пользователю ввести количество ходов и выполняет их с учетом случайных факторов. В результате программа выводит оптимальные ходы и стратегии для выигрыша. Это полезный инструмент для людей, желающих изучить различные стратегии игры в шахматы или использовать ее как образовательный инструмент.

Приложение

<https://github.com/naelxd/linuxspbu>