1 Описание программы

Программа позволяет сыграть в логическую игру "Шашки"с другим человеком.

Игра ведется на шахматной доске 8x8. Смысл игры в том, чтобы "съесть"все шашки соперника. Всего на доске присутвуют 24 фигуры, 12 белых и 12 черных (рис. 1). Первым всегда ходит тот, кто играет за белые фигуры. Шашки могут ходить только по диагонали на одну клетку вверх, назад шашка ходить не может. Бить чужие фигуры шашка может только по диагонали, если за шашкой противника не стоит другая шашка. Бить назад шашка может. Одной шашкой можно побить несколько шашек противника за один ход. Как только один игрок сделал свой ход, очередь переходит к другому игроку. Если игрок довел свою шашку до первой линии противника, то она автоматически превращается в "дамку". Дамка может ходить по диагонали на любое количество клеток в любом направлении. Игра продолжается до тех пор, когда у кого-то из игроков не кончатся все фигуры. Побеждает тот, кто "съел"все фигуры соперника.

Рис. 1: Пример будущей игры

2 Пошаговая инструкция пользователя

1. Запустив программу, пользователь оказывается в главном меню. В нем присутствует кнопка "Play поле для ввода своего имени, поле для ввода имени игрока, с которым пользователь хочет сыграть и кнопка выхода "Exit"(рис. 2).

2. Чтобы изменить свое имя, пользователь должен ввести имя в первое текстовое поле и нажать кнопку "Change name". После этого в верхней части экрана появится ваше текущее имя (рис. 3).

3. Чтобы сыграть с другим человеком, пользователь должен ввести имя противника во второе текстовое поле и нажать кнопку "Play"(рис. 4). Если противник с заданным

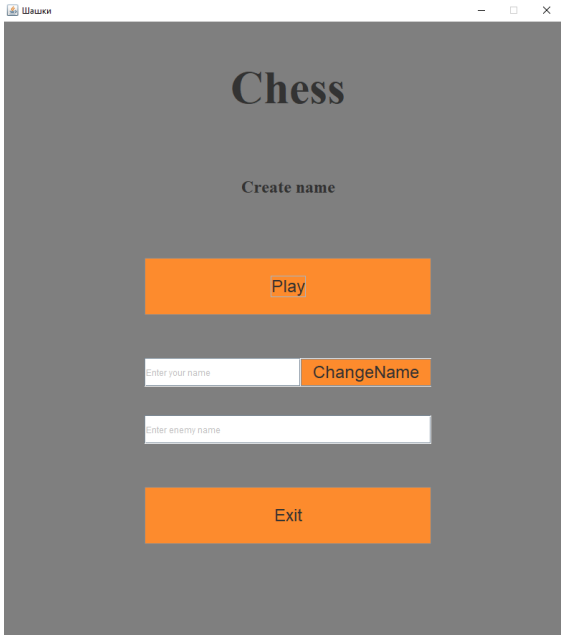

Рис. 2: Главное меню

именем находится в сети, то ему придет уведомление, что пользователь хочет с ним сыграть. Противник может принять предложение и начнется игра, а может отказаться, тогда пользователю придет сообщение, что противник отказался с ним играть (рис. 5, рис. 6).

4. Если другой игрок согласился начать игру, то между игроками случайным образом определяется то, за какой цвет шашек он будет играть. Далее игра проходит по правилам игры "Шашки описанным выше (рис. 7). Игрок может досрочно выйти из игры, нажав кнопку "leave тогда он выйдет в главное меню и ему будет засчитано, а у другого игрока завершится игра и выведется надпись, что он выиграл (рис. 8). Далее он может перейти в главное меню, нажав на кнопку "leave". Если игроки доиграли до конца и кто-то из них выиграл, то у победителя выведется надпись, что он выиграл, у проигравшего надпись, что он проиграл (рис. 9). Далее игрокам нужно нажать на кнопку "leave чтобы выйти в главное меню. Там они могут начать новую игру или выйти из игры.

3 Описание архитектуры программы

Используемая архитектура программы - MVC (model, view, controller).

3.1 Контроллер (Controller)

Контроллер управляет запросами пользователя, получаемые в виде запросов на сервер и ответов с него, когда пользователь нажимает на элементы управления для выполнения различных действий. Контроллер вызывает соответствующую модель для задачи и выбирает подходящий вид.

3.2 Модель (Model)

Модель - это данные и правила, которые используются для работы с данными, которые представляют концепцию управления приложением. В данном приложении в моделе

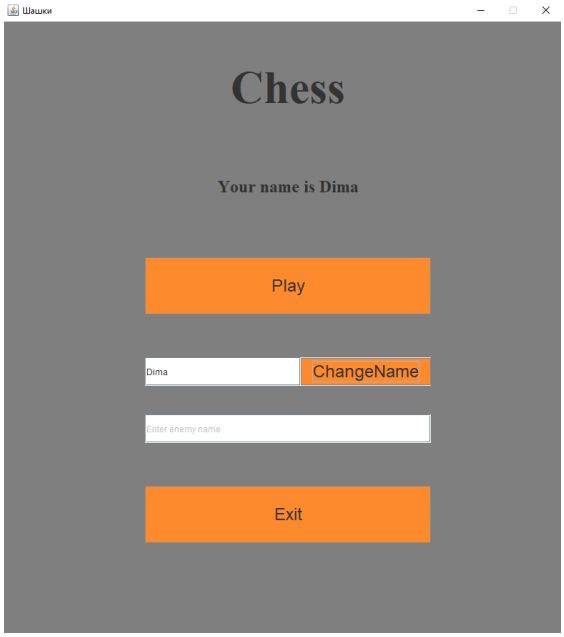

Рис. 3: Смена имени пользователя

хранится игровая доска и положение шашек на ней. Так же хранит или позволяет добыть информацию о том, как может ходить игрок, исходя из положения фигур на доске.

3.3 Вид (View)

Вид обеспечивает различные способы представления данных, которые получены из модели. В данном случаем вид реализован с помощью графической библиотеки swing и сотоит из отображения игральной доски, элементов находящихся на ней и главное меню игры.

4 Описание классов и наиболее важных методов

1. Класс Client служит для хранения данных, полученных от сервера и запросов на сервер.
2. Класс Server служит для вызова потоков для каждого нового подключившегося клиента.
3. Класс ServerThread служит для получения запросов от клиента и посылки ответа на эти запросы.
4. Класс UserList служит вспомогательным классом для ServerThread для хранения списка всех подключенных к серверу клиентов.
5. Класс MainWindow служит для создания нового окна приложения.
6. Класс MainMenu служит для отображения главного меню игры.
7. Класс Field служит для отрисовки игрового поля и реагирования на действия игрока.
   (a) Метод clickOnSquare отвечает за то, что делать при нажатии на клетку.
   (b) Метод doMakeMove отвечает за выполнение хода.
8. Класс BoardMenu служит для создания игрового поля (экземпляра класса Field) и задает ему настойки внешнего вида.

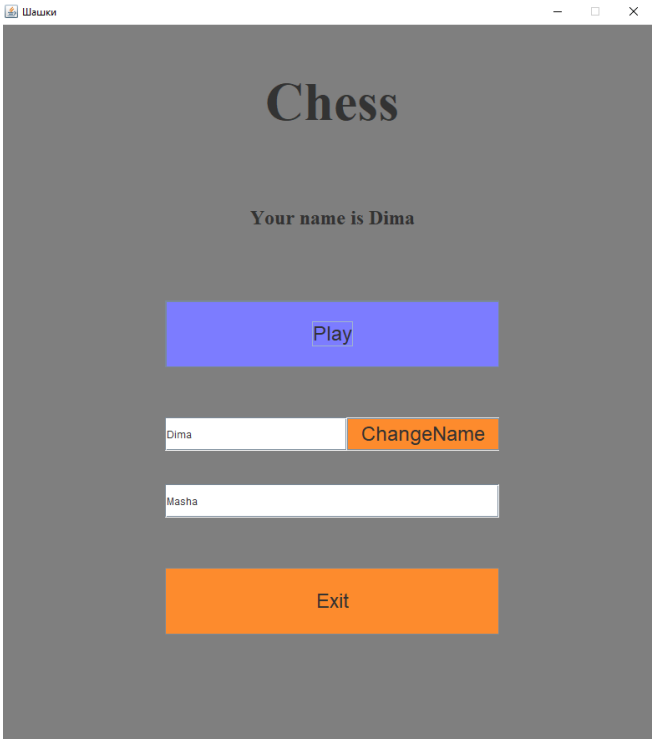

Рис. 4: Приглашение в игру

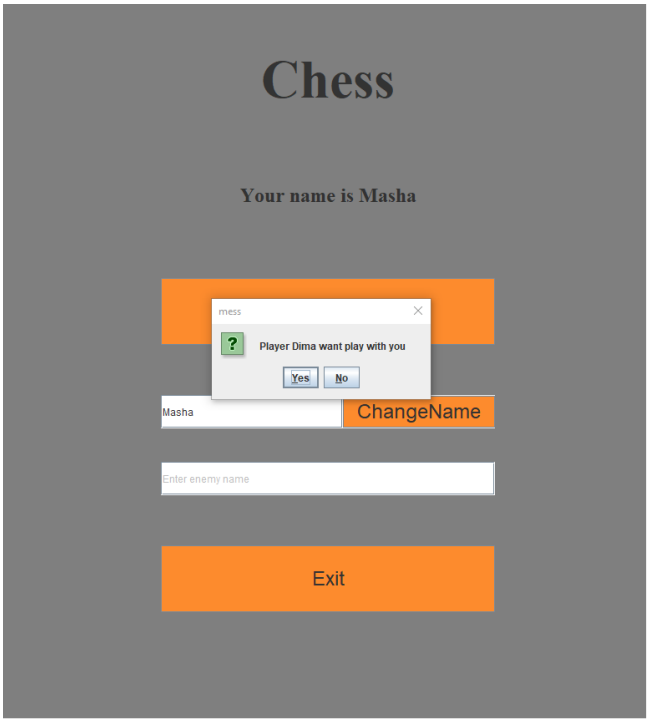

Рис. 5: Запрос на игру

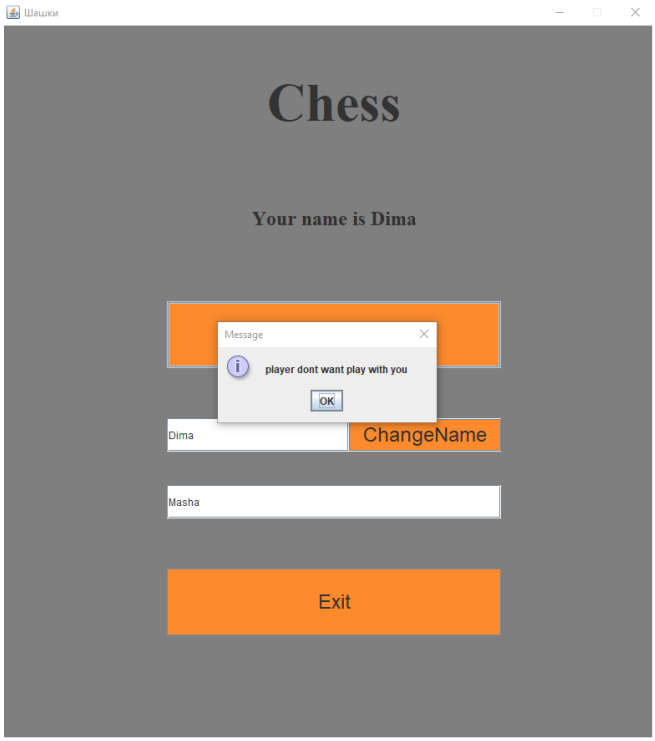

Рис. 6: Отказ от игры

9. Класс ButtonMenuItem служит для настройки внешнего вида кнопок в главном меню.
10. Класс VerticalLayout служит для настройки внешнего вида главного меню.
11. Класс Data отвечает за хранение информации о состоянии игрового поля.
    (a) Метод makeMove отвечает за выполнение хода и освобождение позиции, на которой стояла чужая шашка
    (b) Метод getMoves находит все возможные ходы текущего игрока
    (c) Метод getEats вызывается после того, как съели шашку, и возвращает дальнейшие ходы, куда может съесть текущая шашка
    (d) Метод canEat возвращает значение может ли есть пешка
    (e) Метод canMove говорит может ли пройти пешка.
12. Класс Move хранит в себе данные о передвижении фигуры.

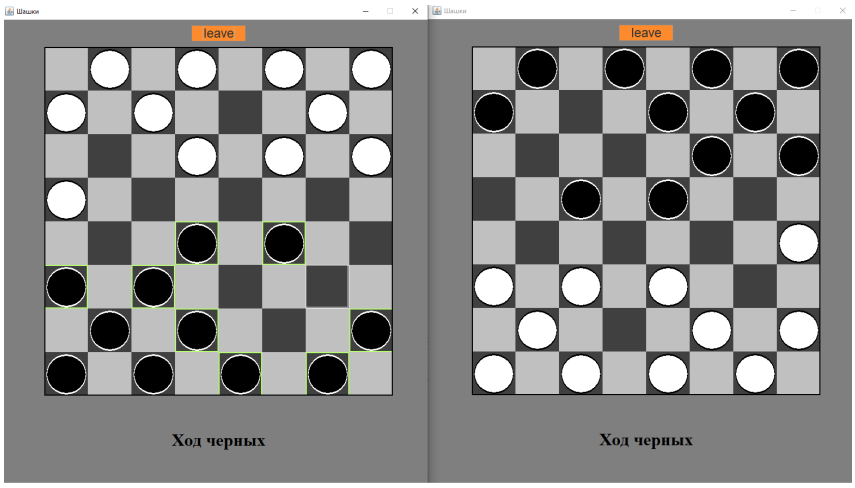

Рис. 7: Процесс игры

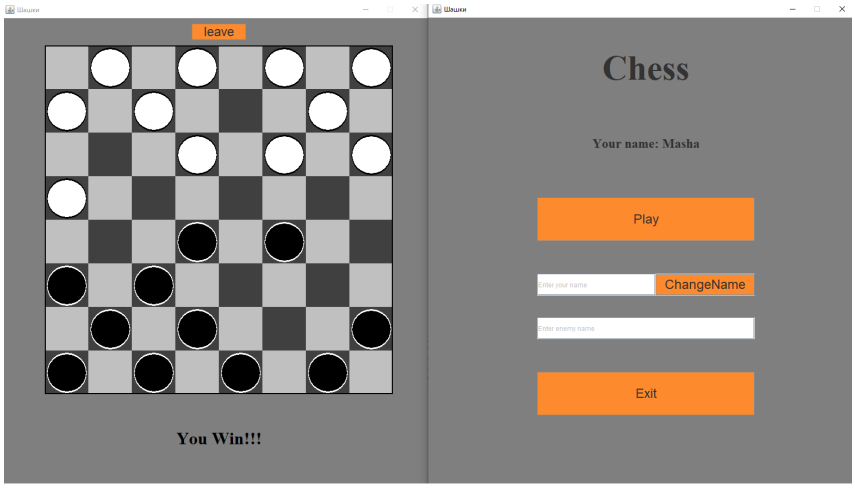

Рис. 8: Досрочный выход игрока

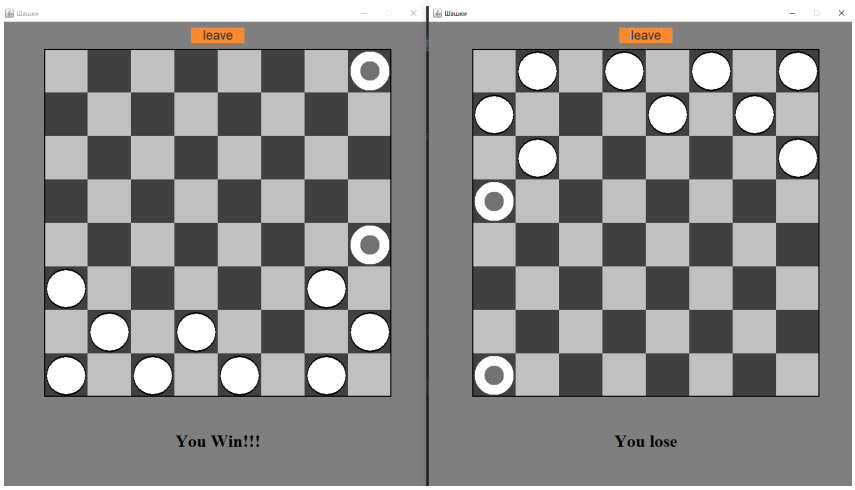

Рис. 9: Конец игры

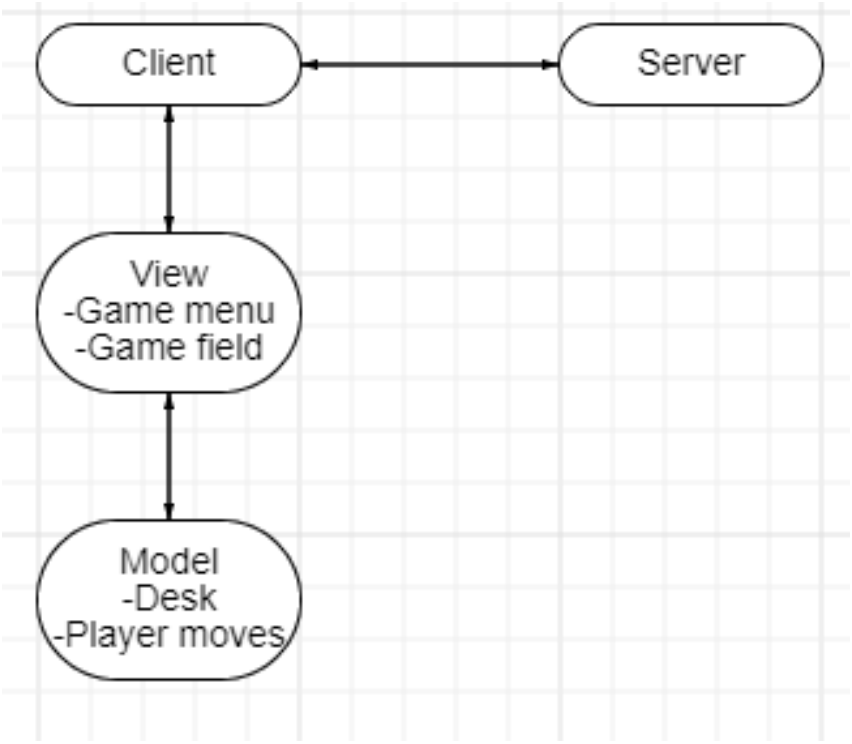

Рис. 10: Архитектура программы