

ГУАП
КАФЕДРА № 51

ПРЕПОДАВАТЕЛЬ

доц., канд. техн. наук		Е.М. Линский
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ О КУРСОВОЙ РАБОТЕ
СЕТЕВАЯ ИГРА КРЕСТИКИ-НОЛИКИ

по курсу: ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

РАБОТУ ВЫПОЛНИЛА

СТУДЕНТКА ГР. № 5723		Глушенкова А. Ю.
	подпись, дата	инициалы, фамилия

Санкт-Петербург 2020

Оглавление

– Функциональная спецификация.....	3
– Руководство пользователя.....	4
– Описание архитектуры игры.....	12
– Описание наиболее важных классов.....	14
– Тестирование.....	17

Функциональная спецификация

Базовая версия

Сетевая игра крестики-нолики будет две реализации: серверную и клиентскую, которые обе используют TCP соединение. Клиент будет подключаться к серверу с определенным ip. Игра имеет графический интерфейс. Клиенту при заходе на сервер будет предложена аутентификация или регистрация, если у него нет аккаунта. Каждый игрок должен иметь уникальный логин.

Игрок может либо создать новую доску, либо присоединиться к уже созданной. Игра ведется, пока у одного из игрока не будет 5 одинаковых символов в ряд или по диагонали. Для каждого игрока ведется подсчет очков: 3 - победа, 2 - ничья, 1 - проигрыш. После окончания игры, игроку будет предложено начать еще одну или выйти.

Руководство пользователя

Данная программа делится на 2 части: клиент и сервер. Чтобы программа работала полностью, вы должны сначала запустить файл Server с помощью Maven в командной строке.



Рисунок 1 — Запуск сервера

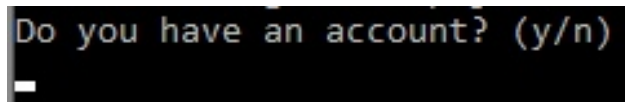
После этого в отдельной командной строке запустите файл клиента с помощью Maven.

The image shows a file named "MAVENCLIENT.jpg" in a monospaced font, likely from a terminal or a file explorer window. The file name is enclosed in quotes.

Рисунок 2 — Запуск клиента

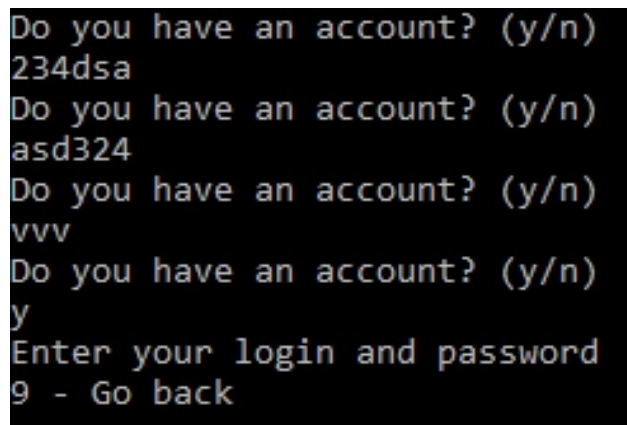
После запуска программы игрока (клиента), на экран будет выведен вопрос, имеет ли клиент учетную запись на сервере в следующей форме:

Необходимо ответить на вопрос y(yes) или n(no). При вводе иных символов, будет также выводиться это сообщение.



```
Do you have an account? (y/n)
_
```

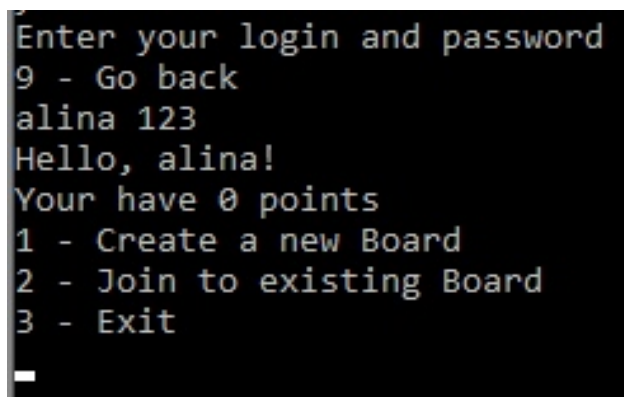
Рисунок 3 — Вход пользователя



```
Do you have an account? (y/n)
234dsa
Do you have an account? (y/n)
asd324
Do you have an account? (y/n)
vvv
Do you have an account? (y/n)
y
Enter your login and password
9 - Go back
```

Рисунок 4 — Вход пользователя (вариации ответов)

При вводе у пользователю необходимо ввести данные своей учетной записи. Если же он ошибся, то он может вернуться обратно и зарегистрироваться. Все учетные записи хранятся на сервере.



```
Enter your login and password
9 - Go back
alina 123
Hello, alina!
Your have 0 points
1 - Create a new Board
2 - Join to existing Board
3 - Exit
_
```

Рисунок 5 — Вход пользователя (вариации ответов)

При отсутствии учетной записи, клиент может зарегистрироваться:

```
Do you have an account? (y/n)
n
Enter login and password (format (only english alphabet and numbers)):
login password
dog 789
Enter your login and password
9 - Go back
dog 789
Hello, dog!
Your have 0 points
1 - Create a new Board
2 - Join to existing Board
3 - Exit
```

Рисунок 6 — Регистрация

Зайдя в учетную запись, клиент увидит перед собой меню следующего вида:

```
1 - Create a new Board
2 - Join to existing Board
3 - Exit
```

Рисунок 7 — Главное меню

Если нажать 1, то вы создадите новую доску и будете ожидать подключение другого игрока. При этом вы будете играть крестиком. Вы ходите первым. Если нажать 2, то вы присоединитесь к уже существующей созданной доске и будете играть ноликом. Вы ходите вторым. После выбора действия, первый игрок должен ввести ход, а второй игрок ждет окончания хода первого игрока.

```
1 - Create a new Board
2 - Join to existing Board
3 - Exit
1
Server: glhf
Server: Enter correct coord
```

Рисунок 8 — Первый игрок

```
Your have 0 points
1 - Create a new Board
2 - Join to existing Board
3 - Exit
2
Server: glhf
```

Рисунок 9 — Второй игрок

Необходимо ввести координаты в формате x y:

[illegible]

Рисунок 10 — Ход первого игрока

Второй игрок также получает отрисовку доски и просьбу ввести координаты:

[illegible]

Рисунок 11 — Второй игрок

После хода второго игрока:

```
Server: Enter correct coord
4 2
dog: move 2
-----
  x
  o
-----
Server: glhf
C:\WINDOWS\system32\cmd.exe - java -jar target/Player.jar
-----
  x
  o
-----
Server: glhf
Server: Enter correct coord
```

Рисунок 12 — Отображение досок

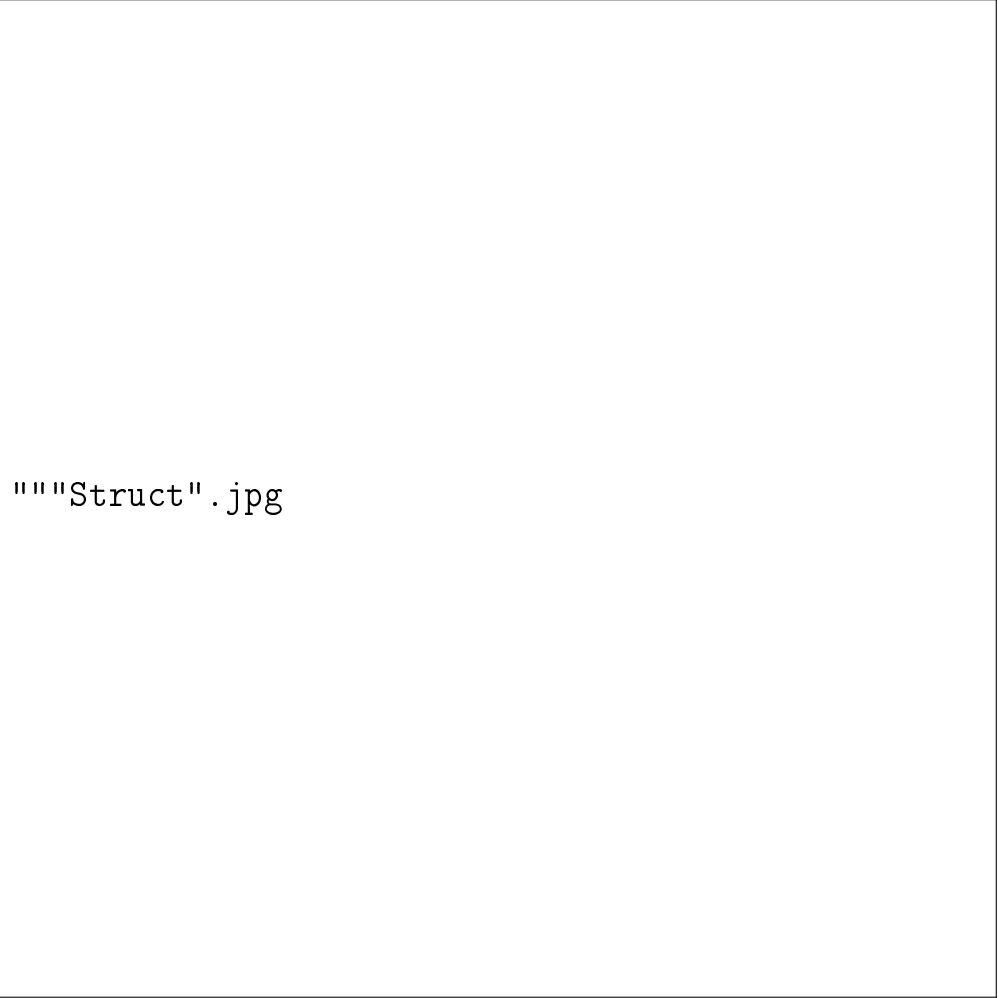
Описание архитектуры игры

Данная игра состоит из четырех блоков:

- Model.
- View.
- Client.
- Server.

Взаимодействие блоков между собой. Первым запускается Server. После, к нему подключаются Clients (Players). У каждого Client есть свои Model и View. Model отвечает за логику игры (сделать ход, определить закончилась игра или продолжаем играть дальше и т.д.), а View отвечает за отображение игры (вывод доски на экран, меню и общение с пользователем).

Между собой данные блоки взаимодействуют так:



""Struct".jpg

Рисунок 13 — Архитектуры игры

Описание наиболее важных классов

Наиболее важные классы данной игры это:

- Model.
- View.
- Client.
- Server.

Назначение класса Model заключается в управление логикой игры. Данный класс отвечает за постановку хода (соответствует введенный ход условиям или нет), определение конца игры и ее продолжения. Основные методы `move`, `checkWinner`.

Метод `move` делает ход, внутри проверяется правильность ввода. Метод `checkWinner` проверяет победителя, а следовательно проверяет конец игры.

Класс View отвечает за вывод доски на экран и за вывод главного игрового меню. Основные методы данного класса это: `printBoard`, `menu`, `login`.

Метод `printBoard` отвечает за вывод игрового поля на экран пользователя.

Методы `menu`, `login` отвечают за вывод меню и регистрацию.

Класс `Client` отвечает за взаимодействие с пользователями. Он вызывает все меню, обрабатывает выбор пользователя в этих меню и осуществляет общение с сервером. Основными методами данного класса являются: `createBoard`, `joinBoard`.

Метод `createBoard` создает доску. Метод `joinBoard` присоединяется к доске.

Класс `Server` состоит из подкласса `ClientTread`, который общается с клиентом. Также есть подкласс `AcceptThread`, который ловит присоединяемых клиентов.

Тестирование

Тестирующим классом является класс `ModelTest`. Он тестирует публичные методы класса `Model`. Данный класс содержит следующие методы:

- `checkMoveTrue` - данный метод проверяет правильность работы метода `move`, который возвращает `true`, если ход был поставлен успешно, и возвращает `false`, если координаты выходят за поля или уже стоит ход другого игрока. Тестирующий метод подаёт на вход тестируемого метода некоторые входные данные и сравнивает результат с `true`.
- `checkWinnerTrue` - данный метод проверяет правильность работы метода `checkWinner`, который возвращает некоторую константу, соответствующую победителю (1- первый, 2- второй). Тестирующий метод подаёт на вход тестируемого метода некоторые входные данные и сравнивает результат с константой, которая ожидается.