

# Оглавление

– Функциональная спецификация.....	3
– Руководство пользователя.....	4
– Описание архитектуры игры.....	8
– Описание наиболее важных классов.....	10
– Тестирование.....	12

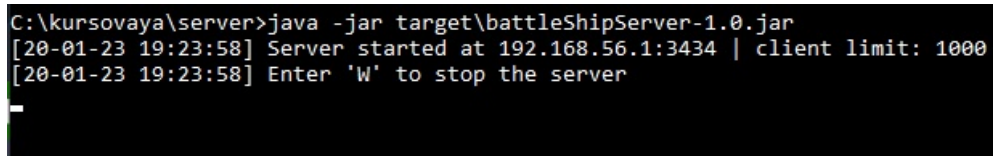
# Функциональная спецификация

## Функции, доступные пользователю

- Игрок может создать новую доску и ожидать подключение соперника;
- Присоединиться к готовой доске, если на ней только один игрок;
- Поддержание нескольких досок на сервере.

# Руководство пользователя

Данная программа делится на 2 части: клиент и сервер. Чтобы программа работала полностью, вы должны сначала запустить файл Server с помощью команды `java -jar target/battleShipServer-1.0.jar` в командной строке.



```
C:\kursovaya\server>java -jar target\battleShipServer-1.0.jar
[20-01-23 19:23:58] Server started at 192.168.56.1:3434 | client limit: 1000
[20-01-23 19:23:58] Enter 'W' to stop the server
```

Рисунок 1 — Запуск сервера

После этого в отдельной командной строке запустите файл клиента с помощью команды

java -jar target/battleShipClient-1.0.jar или открыв файл.

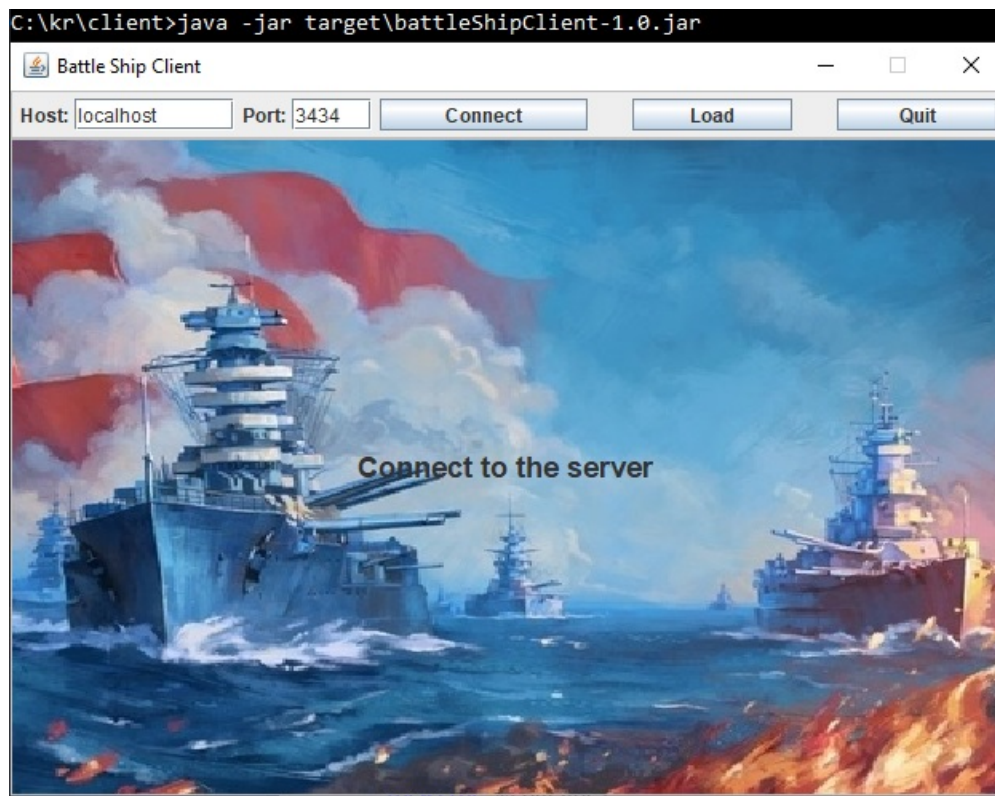


Рисунок 2 — Запуск клиента

После запуска программы, на экран будет выведено меню, в котором можно выбрать:

1 - Connect;

2 - Load;

3 - Exit

Выбрав Connect, вы будете ожидать игрока, который подключится к вашей доске. Выбрав Load, ваша доска будет считана из файла и играть вы будете с ней.

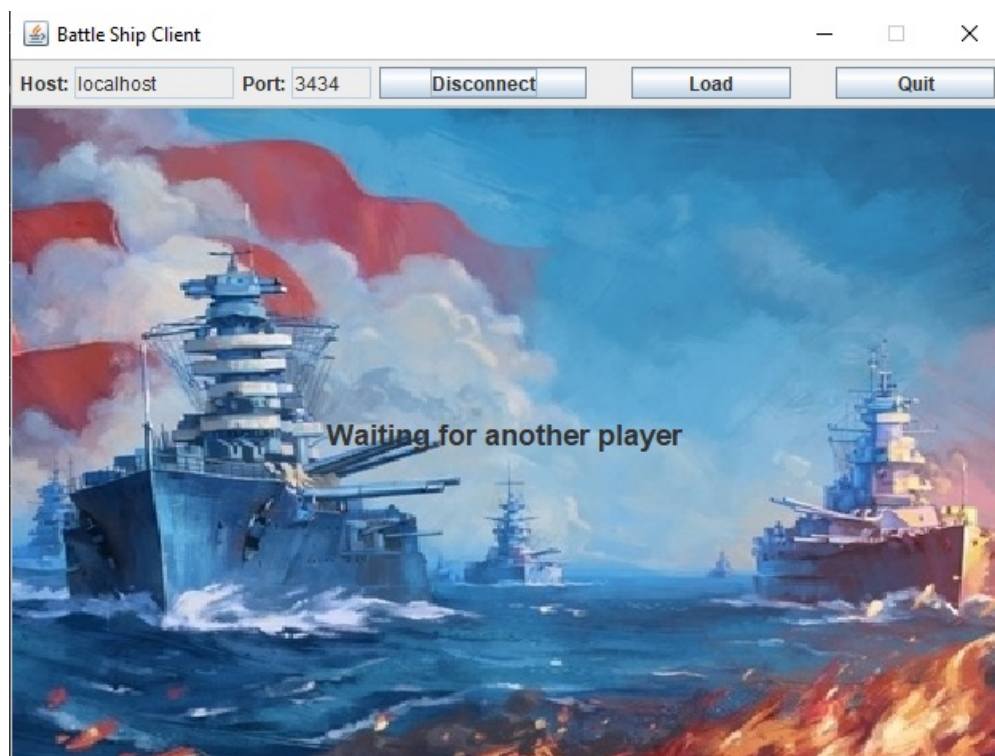


Рисунок 3 — Ожидание игрока

Когда подключится новый игрок, если есть игрок, который ожидает оппонента, у них запустится игра, если ожидающих игроков нет - ожидание оппонента.

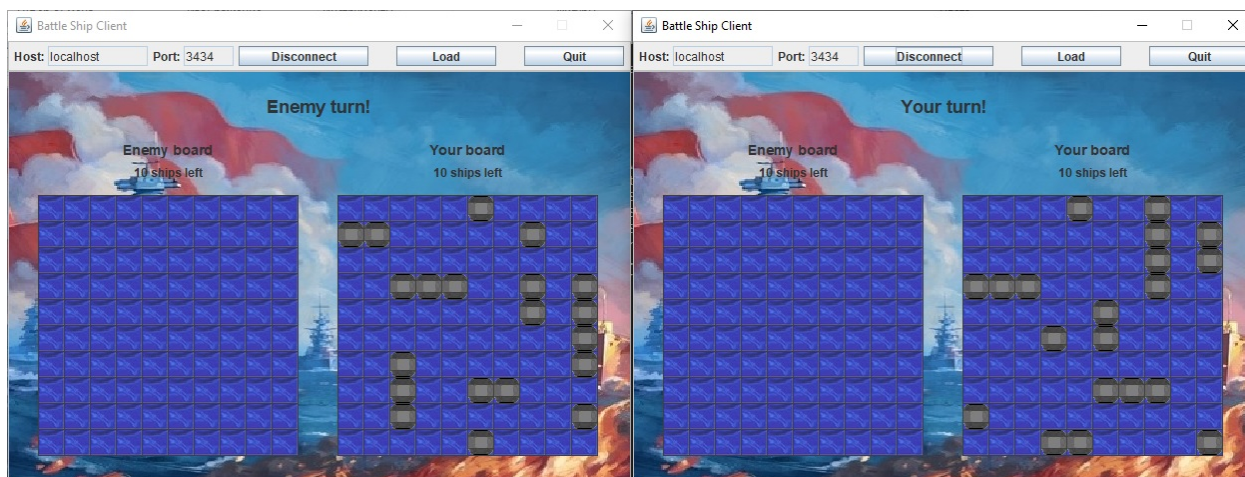


Рисунок 4 — Игровой процесс

Игра закончится когда кто-то из игроков уничтожит все корабли соперника.

После окончания игры, выводится сообщение о том, победили вы или проиграли, и высветиться предложение сыграть снова.

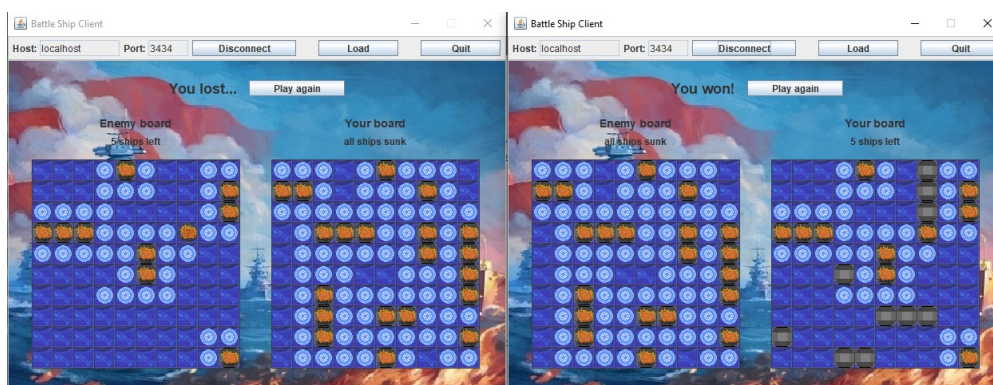


Рисунок 5 — Конец игры

## 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. 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:

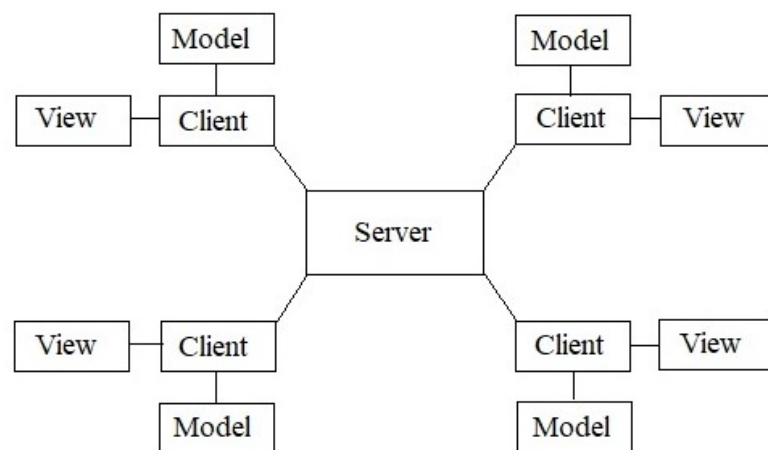


Рисунок 6 — Архитектуры игры



## Описание наиболее важных классов

Наиболее важные классы данной игры это:

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

Назначение класса Model заключается в управление логикой игры, а то есть за сделать ход, определить закончилась игра или продолжаем играть дальше и т.д. У этого класса есть подклассы Board и Ship. Board отвечает за расположение кораблей на доске и проверку закончилась ли игра. Ship отвечает за расстановку кораблей и выстрелы по ним.

Класс View отвечает за вывод доски на экран и за вывод главного игрового меню. Для вывода GUI использовались основные подклассы MainWindow, BoardPanel, GamePanel.

MenuPanel.

MenuPanel отвечает за ввод Port и Host, для присоединения к Серверу.

MainWindow отвечает за вывод фона, а так же всех действий, таких как соединение с сервером, ожидание игрока, новая игра.

BoardPanel отвечает за вывод доски и кораблей на экран, а также отслеживает активность мыши, и изменяет игровое поле в соответствии с щелчками мыши.

Класс Client отвечает за взаимодействие с пользователями. Он вызывает все меню, обрабатывает выбор пользователя в этих меню и осуществляет общение с сервером.

ClientAction отвечает за вверность ввода Port и Host и подключение к серверу.

ClientReaction за сам игровой процесс, позволяет делать пользователю ходы и пересылает их координаты на сервер, который в свою очередь отправляет их оппоненту и определяет конец игры.

Класс Server состоит из подкласса ConnectionClient и главный метод этого класса - connect, который запускает поток Clients. Метод run в подклассе отвечает за прием данных, которые присылает клиент, связь двух игроков, пересылку между ними данных и запись всех действий в logFile.

# Тестирование

Для проверки работы методов класса Model мы используем unit-тесты. С помощью Maven запускается класс BattleShipModelTest, он запускает методы этого класса, а эти методы вызывают методы из класса Model с соответствующими названиями. Он проверяет их, передавая различные параметры и смотрит все ли работает.

-isValidShotTest - данный метод проверяет правильность работы метода isValidShot, который возвращает некоторую константу, соответствующую результату выстрела. Тестирующий метод подаёт на вход тестируемого метода некоторые входные данные и сравнивает результат с константой, которая ожидается.

-getRemainingShipsTest - данный метод проверяет правильность работы метода getRemainingShips, который возвращает true, если количество кораблей еще не закончилось и возвращает false при возникновении ошибки. Тестирующий метод подаёт на вход тестируемого метода некоторые входные данные и сравнивает результат с true.

```

C:\kursovaya\server>mvn package
[INFO] Scanning for projects...
[INFO]
[INFO] -----< server:battleShipServer >-----
[INFO] Building battleShipServer 1.0
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ battleShipServer ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory C:\kursovaya\server\src\main\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.8.0:compile (default-compile) @ battleShipServer ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ battleShipServer ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory C:\kursovaya\server\src\test\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.8.0:testCompile (default-testCompile) @ battleShipServer ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-surefire-plugin:2.22.1:test (default-test) @ battleShipServer ---
[INFO]
[INFO] -----
[INFO] T E S T S
[INFO] -----
[INFO] Running battleShipServer.BattleShipModelTest
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.438 s - in battleShipServer.BattleShipModelTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- maven-jar-plugin:3.0.2:jar (default-jar) @ battleShipServer ---
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] -----
[INFO] Total time: 22.998 s
[INFO] Finished at: 2020-01-23T21:20:16+03:00
[INFO]
[INFO] -----

```

Рисунок 7 — Тест