

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра САПР

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
по дисциплине «Сети ЭВМ»
Тема: Игра «Морской Бой»

Студенты гр. 9309

Аль Сайед А.З.

Серов А.В.

Юшин Е.В.

Преподаватель

Горячев А.В.

Санкт-Петербург

2022

ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ

Студентов Аль Сайеда А.З., Серова А.В., Юшина Е.В.

Группа 9309

Тема работы: Игра «Морской Бой»

Исходные данные:

Реализовать игру «Морской Бой» на языке программирования C# с
возможностью игры двух пользователей по локальной сети.

Содержание пояснительной записки:

Перечисляются требуемые разделы пояснительной записки (обязательны
разделы «Содержание», «Введение», «Заключение», «Список использованных
источников»)

Предполагаемый объем пояснительной записки:

Не менее 12 страниц.

Дата выдачи задания: 08.02.2022

Дата сдачи реферата: 30.05.2022

Дата защиты реферата: 30.05.2022

Студенты гр. 9309

Аль Сайед А.З.

Серов А.В.

Юшин Е.В.

Аннотация

В данном курсовом проекте реализована программа «Морской Бой» на языке программирования C# с использованием Windows Forms Applications. Это сетевая игра, использующая сокет, протокол TCP и клиент-серверную архитектуру, предназначена программа для использования по локальной сети, геймплей происходит между двумя синхронными игроками через асинхронный сервер.

SUMMARY

In this course project, the Sea Battle program was implemented by C# programming language using Windows Forms Applications. This is a network game using sockets, TCP protocol and client-server architecture, the program is designed to be used in local network, the gameplay between two synchronous players through an asynchronous server.

Содержание

Аннотация	3
Введение	5
1. Описание серверной части программы.....	6
2. Описание работы программы и пользовательский интерфейс	6
3. Проверка доступности порта	11
4. Описание основных моментов сетевого кода программы.	12
5. Заключение	12
6. Список использованных источников	12

Введение

Кратко описать цель работы, основные задачи и методы их решения.

Целью данного курсового проекта является реализация игры «Морской Бой» на языке программирования C# с возможностью игры по локальной сети.

Основными задачами курсового проекта стояла реализация стандартной игры морской бой для двух игроков с игровым полем 10x10 клеток. Необходимо было реализовать правильную расстановку кораблей на своём поле, таким образом, чтобы корабли не соприкосались, то есть минимальное расстояние между кораблями составляло хотя бы 1 клетку.

После реализации игрового процесса необходимо было продумать возможность игры по локальной сети, обеспечить интерфейс для входа на сервер с вводом никнейма, выбором соперника, а также интерфейс непосредственно самой игры(расстановка кораблей, перестрелка с оппонентом).

1. Описание серверной части программы

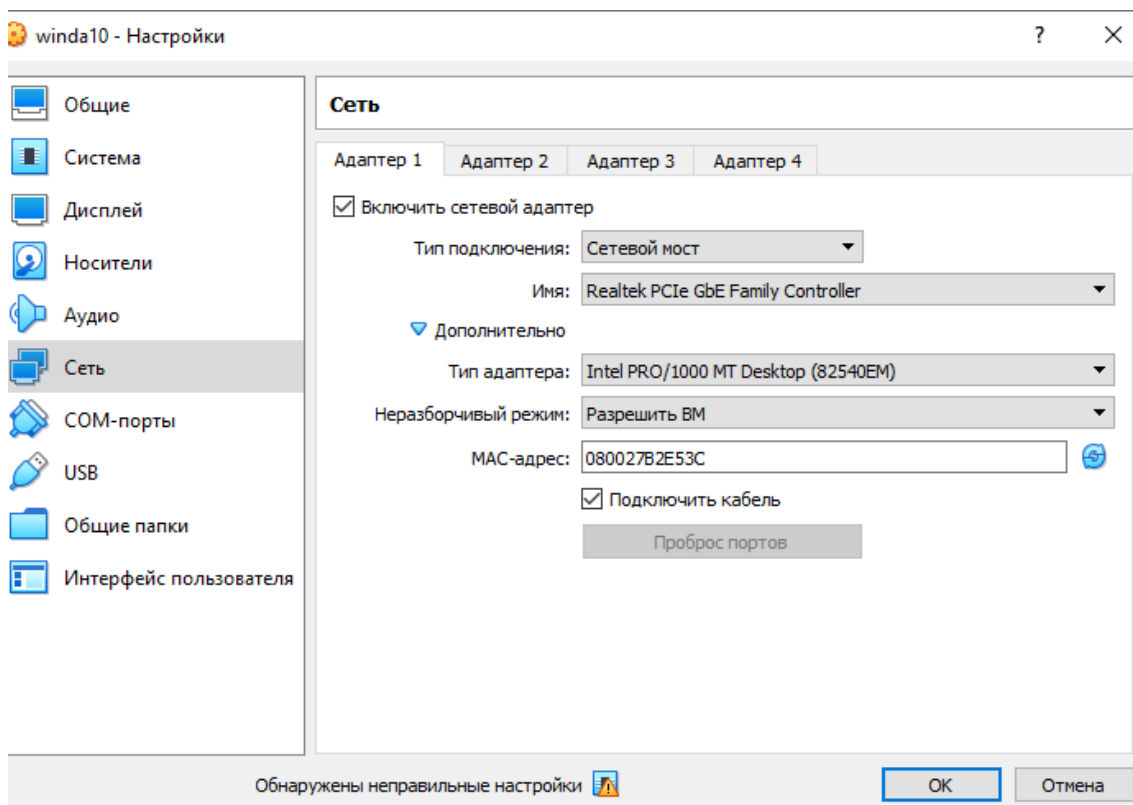
Данная программа для своей работы по сети использует сокеты, протокол TCP и клиент-серверную архитектуру, а также обеспечивает проверку доступности конкретного порта в текущий момент. Геймплей происходит между двумя синхронными игроками через асинхронный сервер.

Консольное серверное приложение получает и отправляет сообщения в асинхронном режиме и является посредником между двумя пользователями.

Программа создаёт сервер, который принимает запросы на соединение от клиентов. Сервер создаётся с использованием асинхронного сокета, поэтому выполнение серверного приложения не приостанавливается, пока оно ожидает подключения от клиента. Приложение получает строку от клиента, выводит ее в консоли, а затем отправляет строку обратно клиенту. Строка, полученная от клиента, должна содержать строку «<EOF>», чтобы сообщить об окончании сообщения.

2. Описание работы программы и пользовательский интерфейс

Для проверки функционала серверной части нам потребуется запустить Windows с помощью VirtualBox со следующими настройками сети:



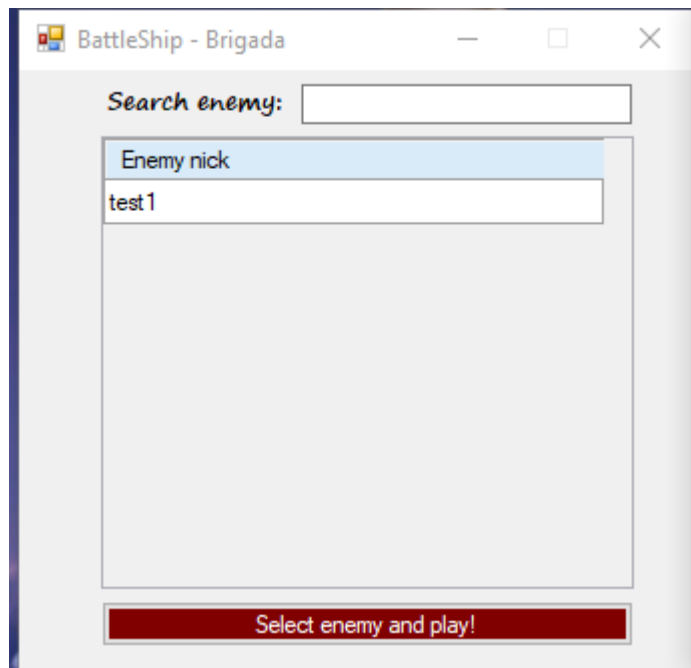
Данная программа позволяет пользователю сыграть в игру «Морской Бой» по локальной сети. Для начала пользователю необходимо запустить асинхронный сервер, открыв файл BattleShipServer.exe и выбрав желаемый сервер. На выбор у нас есть все адреса, доступные нашей машине. В данном случае 0(192.168.56.1) – это адаптер VirtualBox Host-Only Network, а 1(192.168.0.104) – это основной адаптер Ethernet.

```

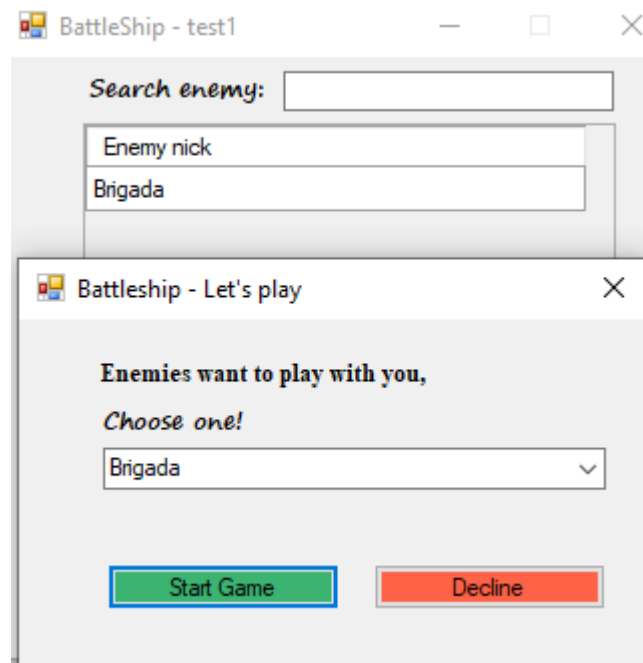
C:\Users\Adam\Desktop\BattleShipServer.exe
Choose server IPv4 address:
0 - 192.168.56.1
1 - 192.168.0.104
Type: 0
Server is running (192.168.56.1)

```

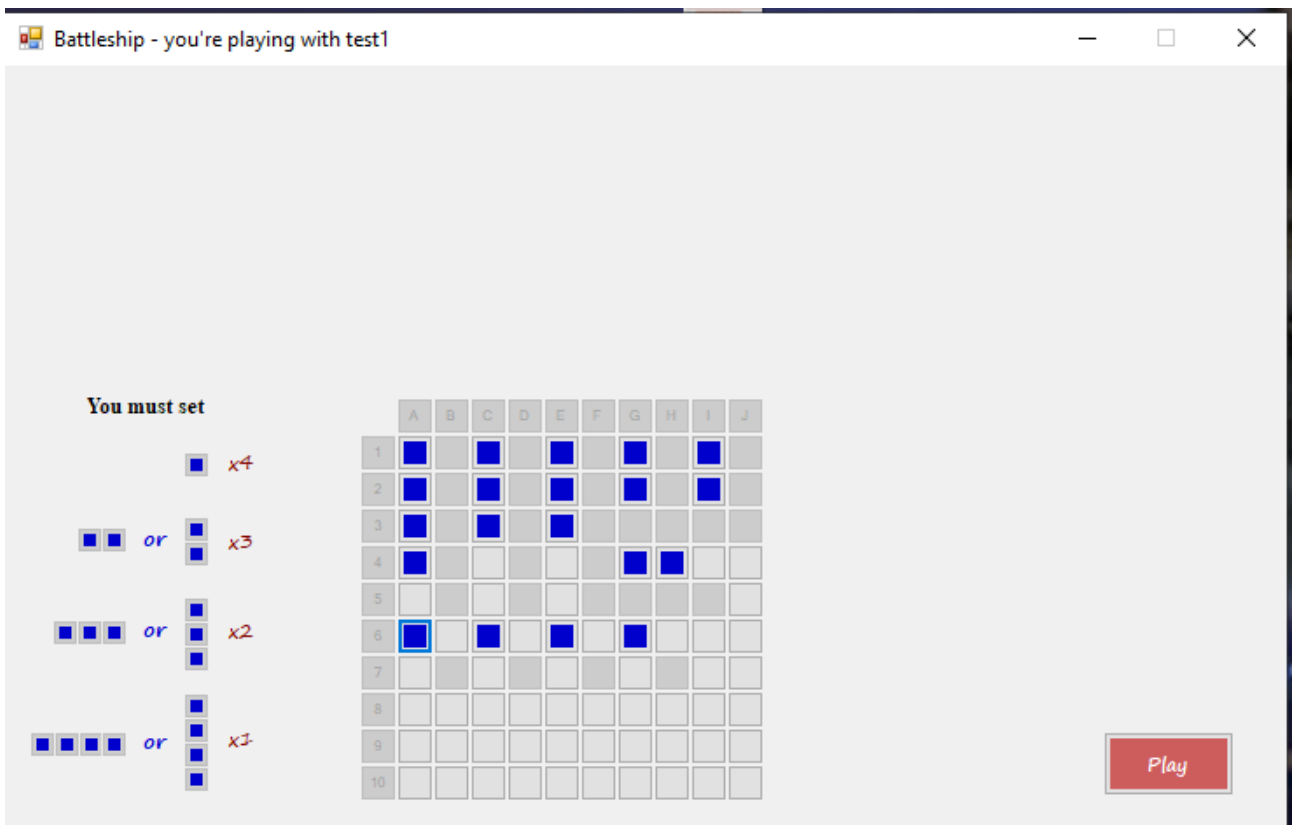
После запуска сервера, пользователю необходимо запустить файл BattleShipCleint.exe и ввести желаемый никнейм и выбранный IP сервера.



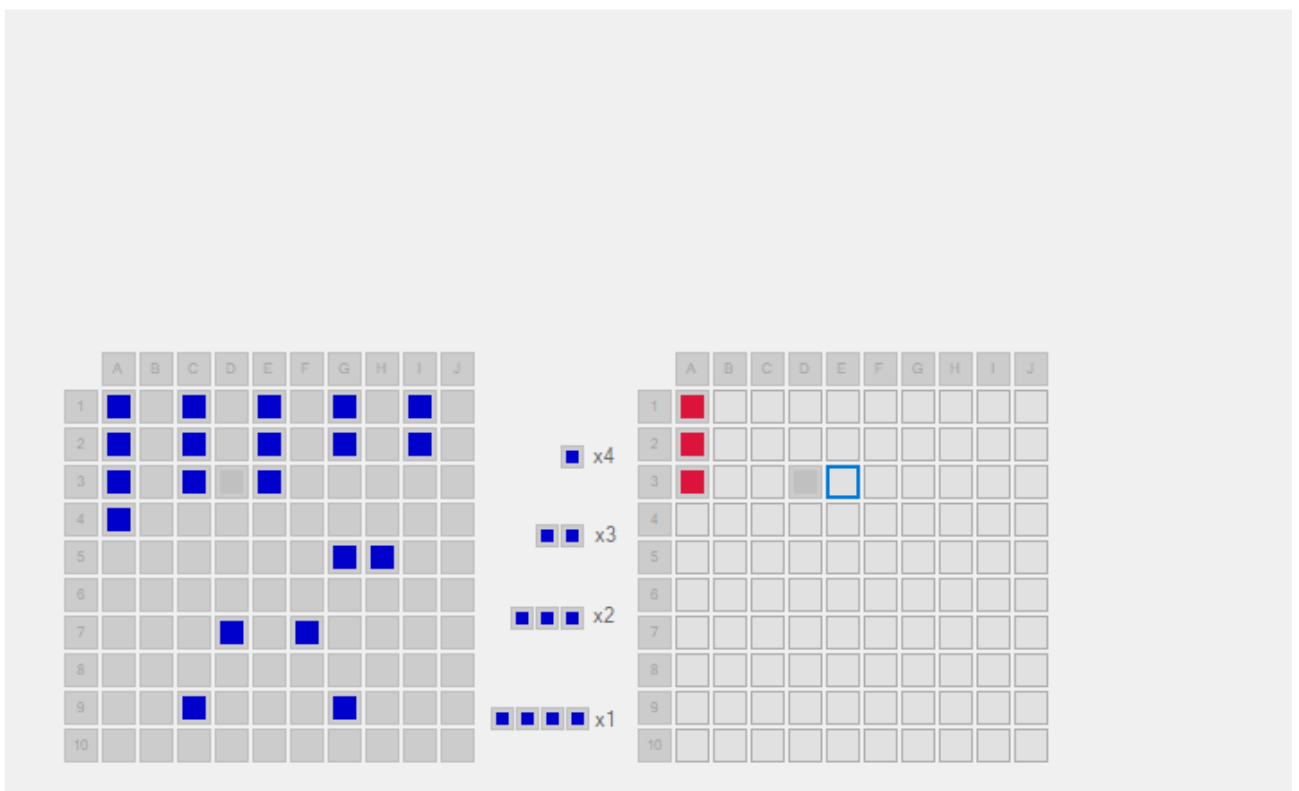
После этого пользователь попадает в поле выбора соперника, в котором будут отображаться все пользователи, которые сейчас находятся на этом сервере в данный момент. Также доступен поиск соперника по никнейму, но так как в данном случае у нас всего 1 доступный соперник, выбираем его и предлагаем начать игру.



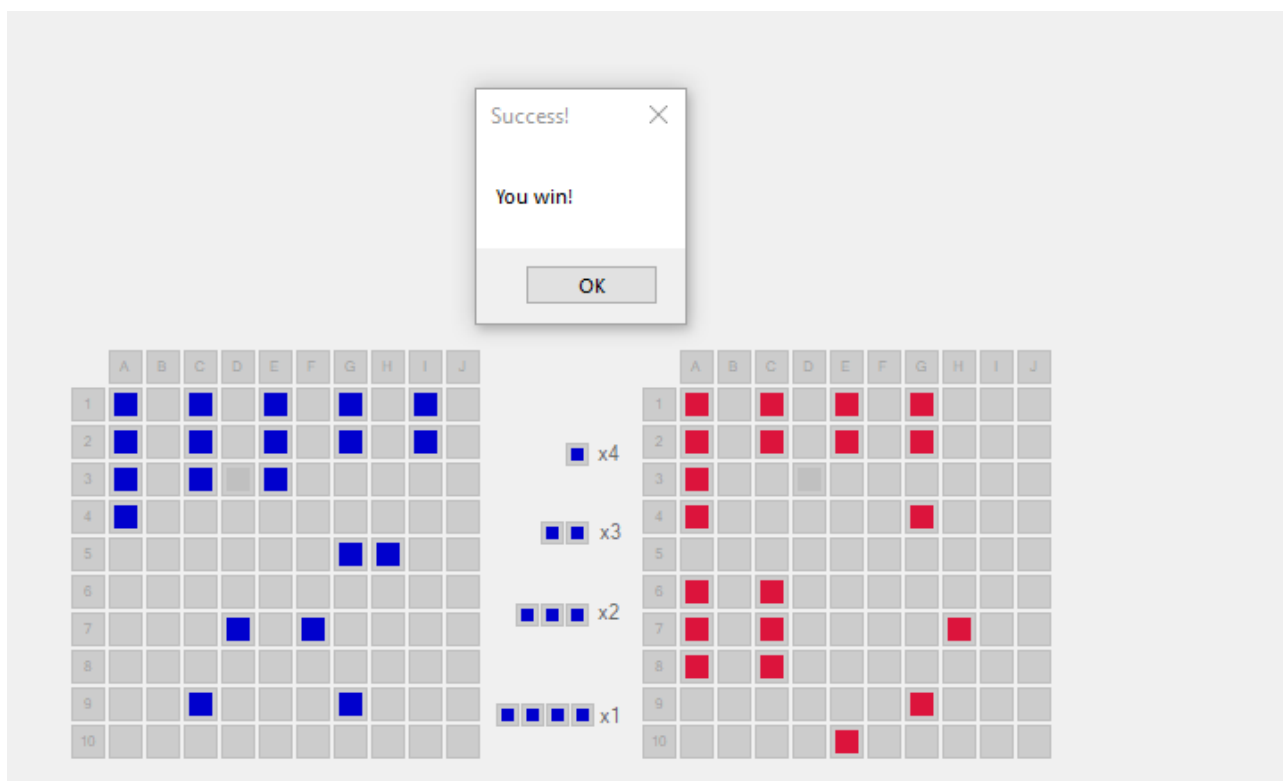
Второму пользователю придёт уведомление о том, что с ним хотят сыграть. Принимаем предложение и начинаем партию.



Расставляем свои корабли по правилам Морского Боя, слева от нашего поля видим подсказку о необходимом количестве кораблей каждого типа, и затем нажимаем кнопку «Play». Игра не будет начата, пока второй пользователь не расставит свои корабли и не подтвердит начало игры.



После этого начинается игра, слева находится наше поле, справа – поле соперника. Наши попадания обозначаются красным цветом, промахи – затемнённым серым.



После окончания игры получаем соответствующее окно, сообщающее нам о результате игры.

```

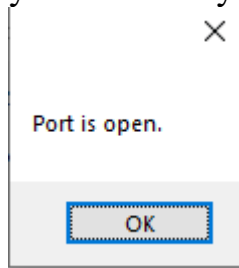
C:\Users\Adam\source\Kursa4\BattleShipServer\bin\Deb
ent 7 bytes to client.
ead 24 bytes from socket.
ata : ♠ 192.168.56.1 5 2 <EOF>
ent 11 bytes to client.
ead 11 bytes from socket.
ata : ♣ 123 <EOF>
ent 7 bytes to client.
ead 24 bytes from socket.
ata : ♠ 192.168.56.1 6 2 <EOF>
ent 11 bytes to client.
ead 11 bytes from socket.
ata : ♣ 123 <EOF>
ent 7 bytes to client.
ead 24 bytes from socket.
ata : ♠ 192.168.56.1 7 2 <EOF>
ent 11 bytes to client.
ead 24 bytes from socket.
ata : @ 192.168.56.1 123 <EOF>
ent 7 bytes to client.
ead 11 bytes from socket.
123 <EOF>
ent 39 bytes to client.
ead 12 bytes from socket.
ata : ♠ 123 <EOF>
ead 20 bytes from socket.
192.168.56.1 <EOF>
ent 7 bytes to client.
ead 21 bytes from socket.
ata : ♠ 192.168.56.1 <EOF>

```

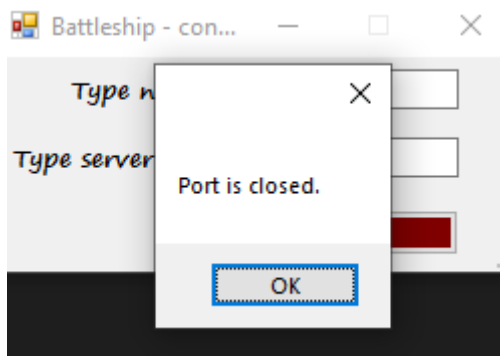
Работа сервера.

3. Проверка доступности порта

В данной программе также проверяется доступность заданного нами порта, в успешном случае получаем следующее сообщение:



Для теста, попробуем использовать закрытый порт и получим отрицательное сообщение:



4. Описание основных моментов сетевого кода программы.

1. Создание TCP/IP сокета.

```
Socket listener = new Socket(AddressFamily.InterNetwork,
    SocketType.Stream, ProtocolType.Tcp);
```

2. Получение данных с сокета клиента.

```
bytesRead = handler.EndReceive(ar);
if (bytesRead > 0)
{
    state.sb.Append(Encoding.ASCII.GetString(
        state.buffer, 0, bytesRead));
    content = state.sb.ToString();
    if (content.IndexOf("<EOF>") > -1)
    {
        Console.WriteLine("Read {0} bytes from socket. \nData : {1}",
            content.Length, content);
        string messageBits = Utilities.getBinaryMessage(content);
```

3. Фиксация основных действий игры происходит с помощью параметров игроков, далее с помощью оператора switch описываем различные ситуации, такие как: попадание по кораблю, промах, конец игры, один из игроков сдался и т.д.

```
string[] parameters = content.Split(' ');
string nick = String.Empty;
string IPport = String.Empty;
string port = String.Empty;
string enemyNick = String.Empty;
string players = String.Empty;
bool result = false;
string whoSent = "";
string whomSent = "";
```

5. Заключение

В течение семестра нами были изучены различные способы передачи данных по сети, в данной курсовой работе мы использовали протоколы TCP и клиент-серверную архитектуру для реализации сетевой игры между двумя пользователями с помощью асинхронного сервера.

6. Список использованных источников

1. Примеры кода сокетов - <https://docs.microsoft.com/ru-ru/dotnet/framework/network-programming/socket-code-examples>

2. Разработка сетевых приложений, многопользовательские игры -
[https://sd.blackball.lv/library/Mnogopoljzovateljskie_igry__Razrabotka_setevyh_prilozhenij_\(2017\).pdf](https://sd.blackball.lv/library/Mnogopoljzovateljskie_igry__Razrabotka_setevyh_prilozhenij_(2017).pdf)