**Министерство образования и науки Российской Федерации**

**Муниципальное автономное общеобразовательное учреждение**

**"Лицей №14 имени Заслуженного учителя Российской Федерации**

**А.М. Кузьмина"**

**Проектная работа**

**Тема: Создание многопользовательской игры с использованием локальной сети**

**Выполнил:**

Учащийся 11 класса «К»

Маняхин Михаил Владиславович

**Руководители:**

Вязовов Сергей Михайловчи

Слезин Кирилл Анатольевич

Оглавление

[Введение 3](#_Toc3807449)

[Цель и задачи 4](#_Toc3807450)

[Выбор подхода к реализации локальной сети 5](#_Toc3807451)

[Разработка идеи и плана приложения 7](#_Toc3807452)

[Начальное меню и поиск оппонента 8](#_Toc3807453)

[Расстановка кораблей 9](#_Toc3807454)

[Процесс сражения 11](#_Toc3807455)

[Процесс написания приложения. 13](#_Toc3807456)

[Проблемы в процессе выполнения проекта 17](#_Toc3807457)

[Заключение 18](#_Toc3807458)

[Список использованной литературы 19](#_Toc3807459)

# Введение

Сетевые технологии всегда были чрезвычайно важны в современном обществе. Возможность создавать игры с серверами. Осуществлять многопользовательские приложения и игры. Различные связи и чаты с огромным количеством человек. Возможности сетевых технологий нельзя недооценить. И их изучение в современном мире имеет важное значение. Многопользовательские игры заполонили современный рынок. И их существование без локальных сетей и сетевых технологий попросту невозможно. Зачем же именно она нужна?

* Передачи данных между устройствами без участия внешних накопителей (флешек, дисков и пр.);
* Открытия доступа к интернету для всех участников сети, если он подключен только к одному компьютеру;
* Управления портативными устройствами с разных компов (например, в пределах офиса с любого железа можно печатать на одном принтере);
* Организации голосовых и видеоконференций;
* Игр по сети

# Цель и задачи

**Цель**

Создание многопользовательской игры с локальной сетью

**Задачи**

1. Выбор игры для написания
2. Выбор подходящего типа сети
3. Создание локальной сети для общения между несколькими пользователями
4. Реализация связи между пользователями напрямую
5. Создание игры и внедрение в неё локальной сети

# Выбор подхода к реализации локальной сети

Можно было реализовать игру через общение нескольких пользователей с отдельным сервером. Проблема этого подхода заключается в том, что он очень прост в условиях выбранной игры и ненадежен. Для двух пользователей проще разобраться друг с другом и играть на компьютерах в локальной сети, чем надеяться на хорошую работу независимого сервера.

Одноранговая, децентрализованная, или пиринговая ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *peer-to-peer, P2P* — равный к равному) сеть — это [оверлейная](https://ru.wikipedia.org/wiki/%D0%9E%D0%B2%D0%B5%D1%80%D0%BB%D0%B5%D0%B9%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C) [компьютерная сеть](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C), основанная на равноправии участников. Часто в такой сети отсутствуют выделенные [серверы](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80_(%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5)), а каждый узел (peer) является как [клиентом](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B8%D0%B5%D0%BD%D1%82_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D1%8B%D0%B9)), так и выполняет функции сервера. В отличие от архитектуры [клиент-сервера](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B8%D0%B5%D0%BD%D1%82-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80), такая организация позволяет сохранять работоспособность сети при любом количестве и любом сочетании доступных узлов. Участниками сети являются все [пиры](https://ru.wikipedia.org/wiki/Peer).

В сети присутствует некоторое количество машин, при этом каждая может связаться с любой из других. Каждая из этих машин может посылать запросы другим машинам на предоставление каких-либо ресурсов в пределах этой сети и, таким образом, выступать в роли клиента. Будучи сервером, каждая машина должна быть способной обрабатывать запросы от других машин в сети, отсылать то, что было запрошено. Каждая машина также должна выполнять некоторые вспомогательные и административные функции (например, хранить список других известных машин-«соседей» и поддерживать его актуальность).

Любой член данной сети не гарантирует своего присутствия на постоянной основе. Он может появляться и исчезать в любой момент времени. Но при достижении определённого критического размера сети наступает такой момент, что в сети одновременно существует множество серверов с одинаковыми функциями.

Помимо чистых P2P-сетей, существуют так называемые гибридные сети, в которых существуют [серверы](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80_(%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5)), используемые для координации работы, поиска или предоставления информации о существующих машинах сети и их статусе (on-line, off-line и т. д.). Гибридные сети сочетают скорость централизованных сетей и надёжность децентрализованных благодаря гибридным схемам с независимыми индексационными серверами, [синхронизирующими](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BD%D1%85%D1%80%D0%BE%D0%BD%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F_(%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)) информацию между собой. При выходе из строя одного или нескольких серверов сеть продолжает функционировать. К частично децентрализованным сетям относятся например [eDonkey](https://ru.wikipedia.org/wiki/EDonkey2000_(%D1%81%D0%B5%D1%82%D1%8C)), [BitTorrent](https://ru.wikipedia.org/wiki/BitTorrent_(%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB)), [Direct Connect](https://ru.wikipedia.org/wiki/Direct_Connect), [TOR](https://ru.wikipedia.org/wiki/Tor).

Как было сказано выше, во многих программах отдельный сервер используется как помощник в установлении peer to peer соединения. Но я решил пойти дальше и отойти от упрощающего работу гибридного устройства P2P соединений сервера, устанавливая соединение между двумя пользователями, находящимся в одной локальной сети без него. В моей сети происходит некий обмен информацией пользователей. Это сообщение между ними реализовано с оглядкой на реальное сообщение для красоты и удобства написания.

# Разработка идеи и плана приложения

Главной идеей было создание локальной сети для сообщения двух пользователей. Так как мне близка игровая тематика, решено было сделать игру, в которой с помощью сети будет разработано общение между пользователями. На этом этапе появляется несколько важных вопросов:

* какую игру написать
* какой язык программирования выбрать для написания приложения
* как реализовать общение двух пользователей
* как наладить между ними надежную связь для передачи данных в процессе игры

Решено было выбрать Java, так как с его помощью весьма удобно писать приложения, которые легко можно изменить и расширить, наметив «каркас». В качестве игры рассматривалось несколько вариантов:

* полноценная стратегия
  + раскрывает многие возможности для работы с сетью
  + сложность разработки
  + требует серьезных затрат на элементы интерфейса и графические
  + большие затраты времени на реализацию логики игры
* экшн с видом сверху
* большие затраты времени на разработку логики игры
* слишком ветвистое общение между пользователями
* настольные стратегии
  + позволяет доступно и в некоторой степени легко раскрыть и осознать общение внутри сети
  + простота реализации логики (правила таких игр в основном известны всем или интуитивно ясны)
  + интерфейс приложения не требует лишних усилий и объяснений

В качестве настольной стратегии было решено выбрать морской бой. Так как почленить эту игру на отдельные части мне показалось очень простым и удобным для написания. Для удобства разработки игра условно разделялась на части:

1. Начальное меню
2. Поиск оппонента для игры
3. Расстановка кораблей на поле
4. Процесс самой игры
5. Конец

# Начальное меню и поиск оппонента

Само меню должно включать в себя кнопки начала и завершения игры. Была идея создания кнопки для комнаты и самой комнаты, в которой можно было находить или выбирать оппонента. Но это не было выполнено в результате выбора подхода к созданию сети. Так как было решено работать без центрального сервера. Сохранять данные о каждом устройстве вошедшем в сеть и запустившим игру оказалось сложным занятием. В результате чего от идеи решено было отказаться. Начальное меню выглядит так:

При нажатии на exit осуществляется выход из приложения.

При нажатии PLAY! Запускается поиск оппонента по локальной сети, занимающий определенное количество времени для сканирования сети и нахождения оппонента, нажавшего PLAY!

Значок, который вы видите в углу «скриншота», означает ожидание оппонента. Когда оппонент найдется и установит связь, значок пропадет и экран переключится. Игра перейдет в состоянии расстановки кораблей обоими игроками.

## Расстановка кораблей

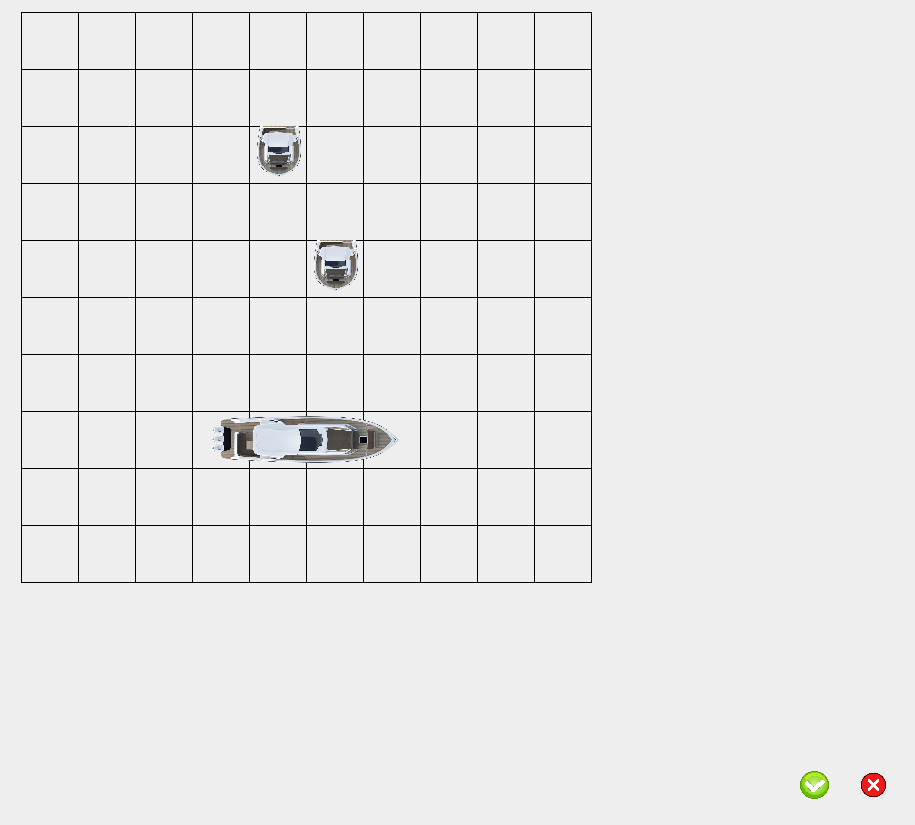
Появляется одно поле размером 10 на 10 клеток, в которые можно ставить корабли. Нажав мышью на то место, где хотите, чтобы располагался нос и, доведя мышью то места, где желаете иметь корму.

Всего кораблей 10, как и в оригинальной настольной игре:

* 4 однопалубных
* 3 двухпалубных
* 2 трехпалубных
* 1 четырехпалубный

Расстановка также соответствует каноничным правилам:

* Нельзя ставить на одну и ту же клетку
* Нельзя ставить за пределами 10 на 10 поля
* Нельзя ставить больше кораблей, чем есть
* Корабли также не должны стоять в смежных друг с другом клетках



Кнопка с изображением крестика стирает все корабли с поля и позволяет вам расставить их заново в нужном вам расположении.

Кнопка с изображением галочки подтверждает расстановку кораблей и переводит в режим ожидания расстановки оппонента, появляется значок загрузки

Когда ответ получен, поле изменяет свой статус и переходит в режим непосредственного сражения.

## Процесс сражения

После ответа от оппонента игра переходит в эту стадию. Игроки посредством «рандома» решают, кто ходит первым. И закрепляют за собой очередность.

Игрок, у которого высветилась табличка Your Turn ходит первым и продолжает ходить пока он не промахнется и надпись не исчезнет. Игрок может стрелять куда хочет и даже по уже сбитым кораблям, но это считаться не будет.



Также появляется 2 более маленьких поля. То, на котором расставлены корабли, ваше, другое, поначалу пустое принадлежит оппоненту.

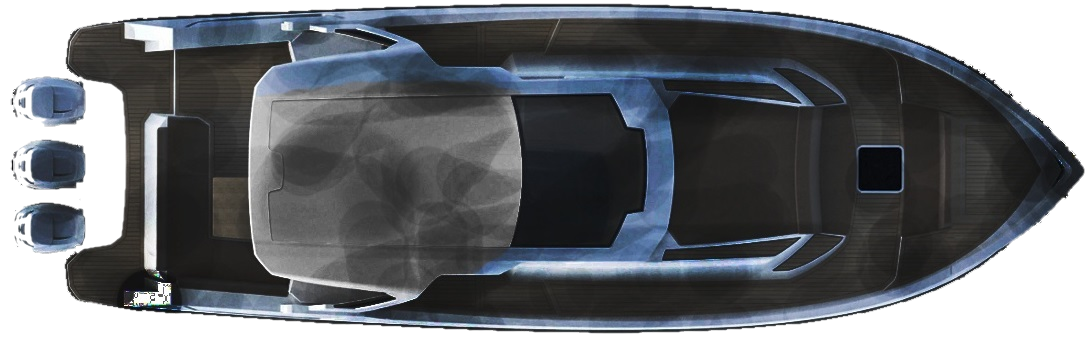
По нему надо производить выстрелы. Для этого, заметив значок очереди необходимо кликнуть на клетку на поле и ожидать результата, получаемого в результате обмена данными между 2мя компьютерами по сети.

Если вы попали, то увидите:



Иначе:



Уничтожив же весь корабль: 

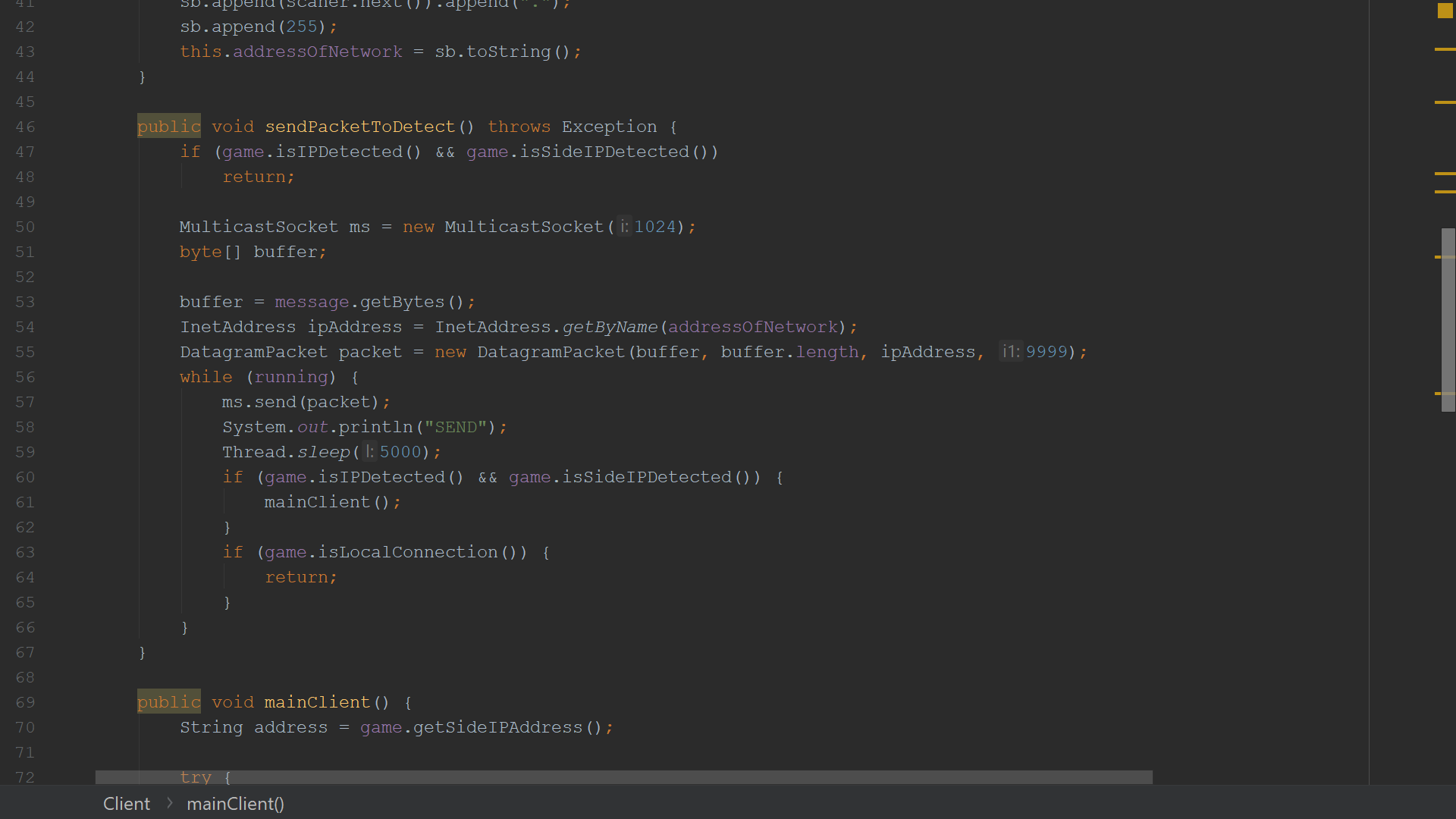
После окончания очереди человек для совершения хода опять ждет промаха от противника и надписи и очереди.

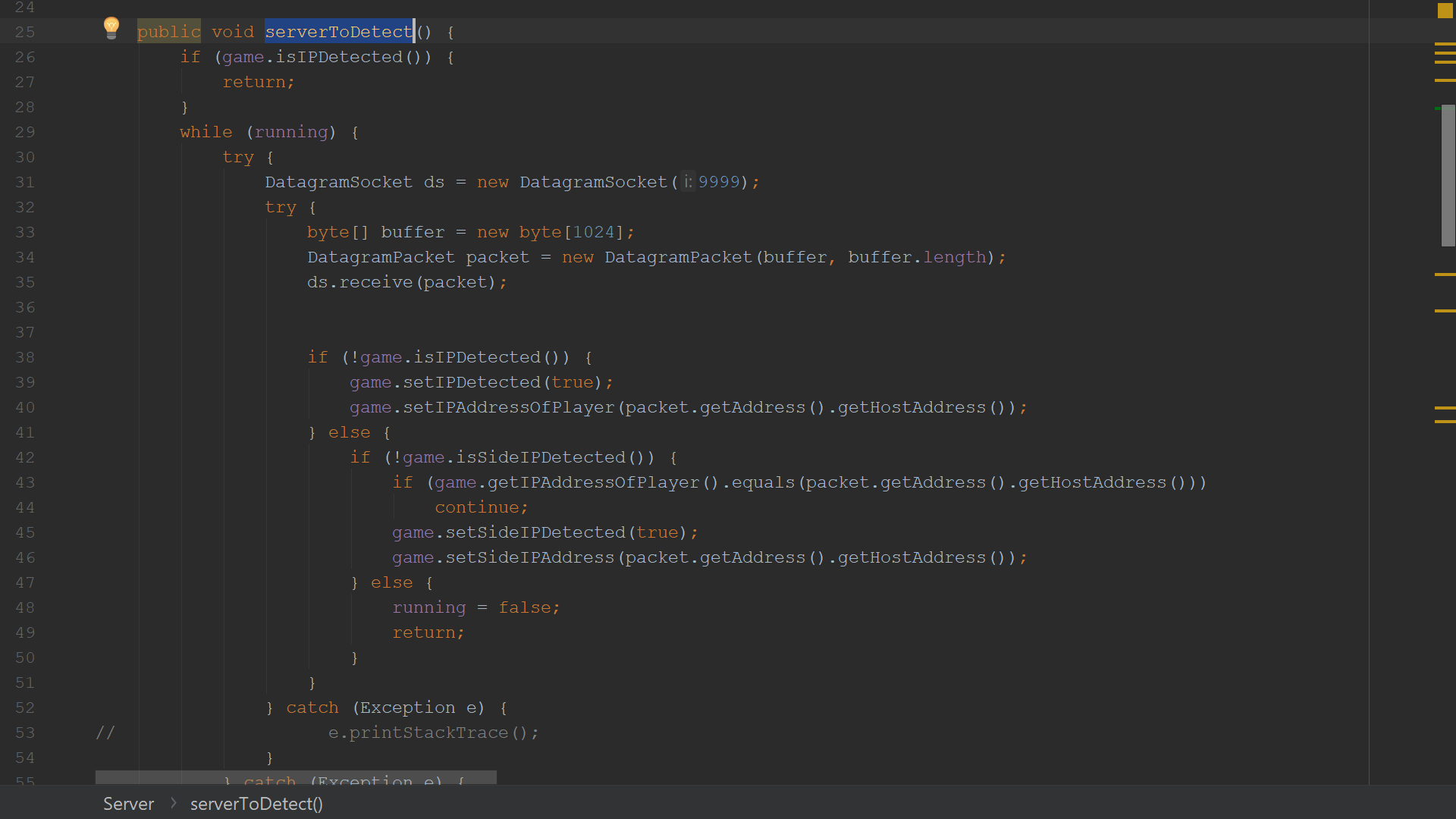
Сбив все 10 кораблей у вас появиться значок победы, иначе поражения. И автоматически программа перейдет в главное меню.

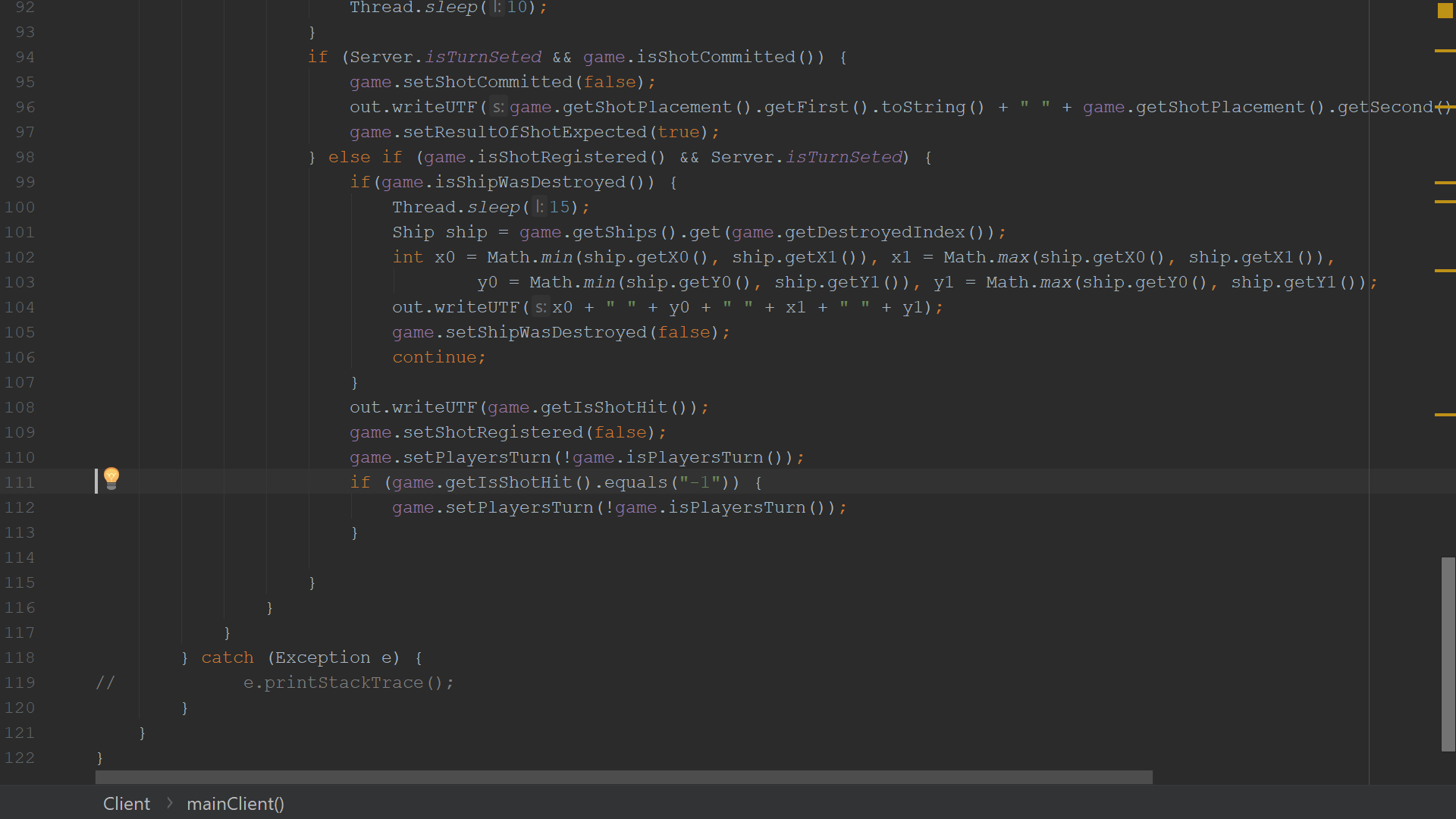
## Процесс написания приложения.

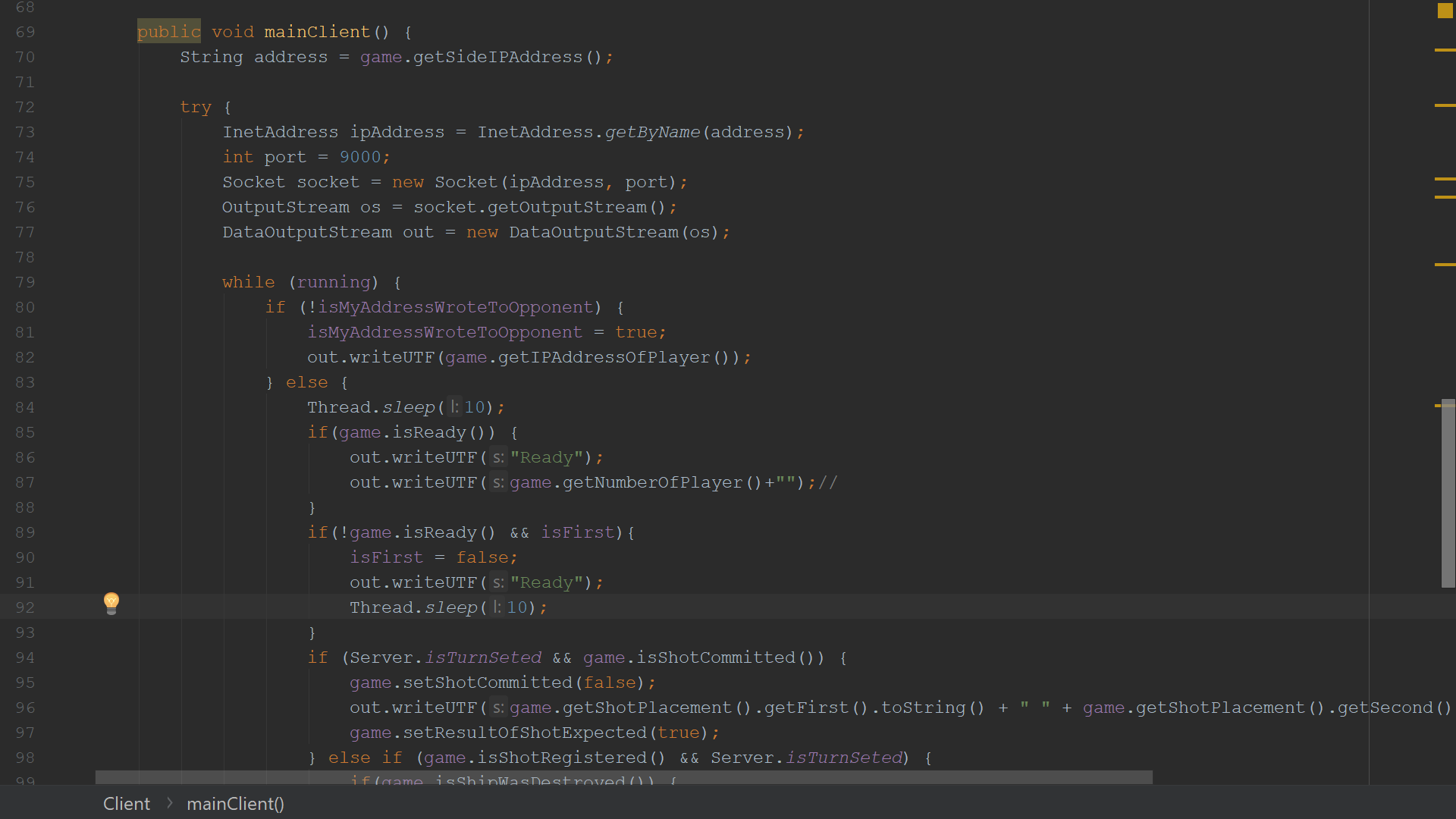
Вся программа и процесс отрисовки в частности написан компактно, чтобы в будущем иметь возможность к расширению и доработки.

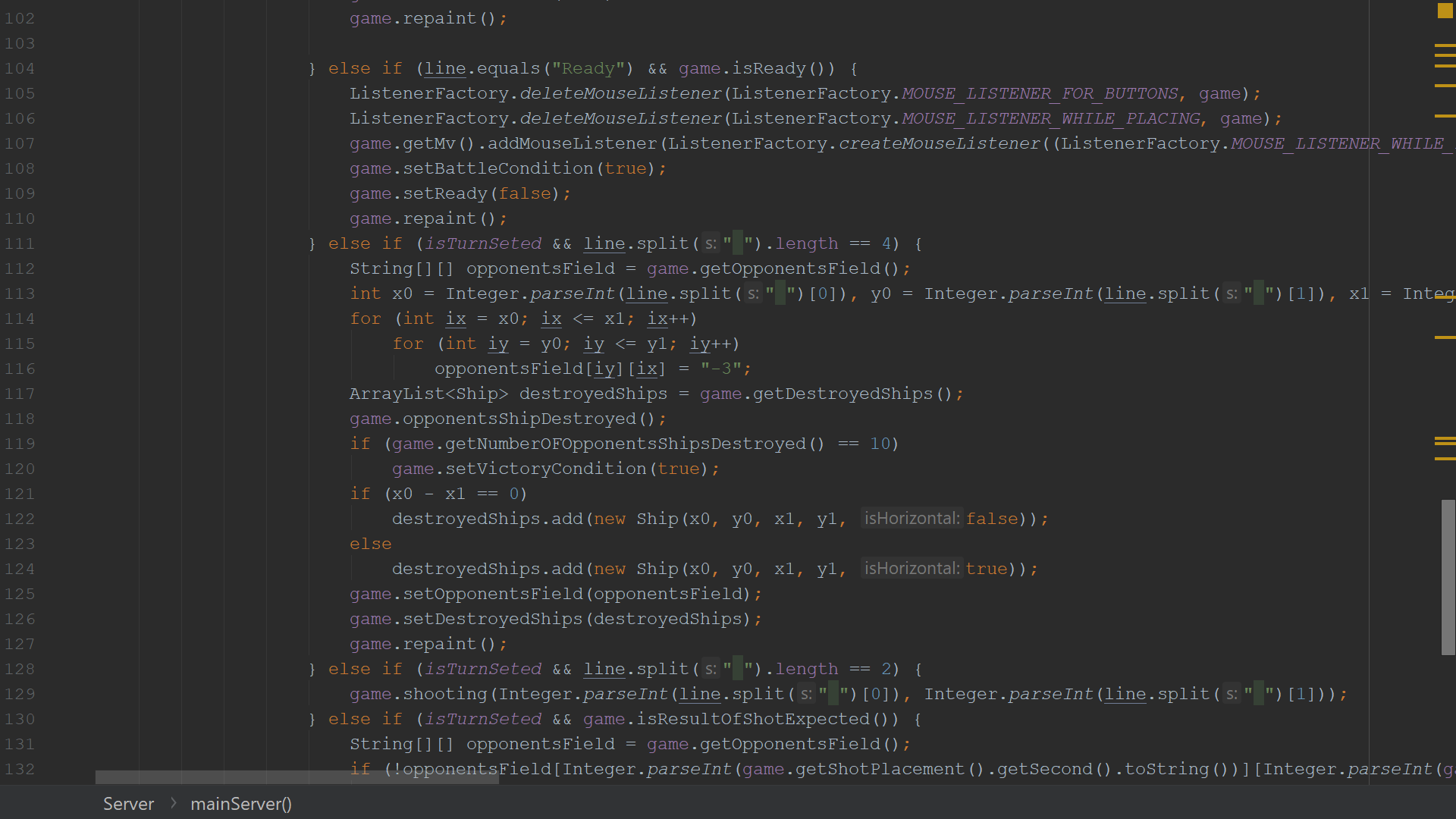
Самой сложной частью оказалось написание самой сети и отношений внутри неё.

Изначально программе-клиенте производится широковещание, где все участники сети получают одинаковые пакеты данных. Функция SendPacketToDetect производит рассылку пока соединение между 2мя пользователями не будет полностью установлено. 

Программа сервера serverToDetect производит сканирование данных и устанавливает адрес приславшего адреса. Именно эта функция производит чтение и запись двух адресов своего и оппонента для установления прямой связи. Когда связь между 2мя компьютерами установлена эти функции прекращают работу в целях экономии по времени и памяти. 

После установления полноценной связи осуществляется непосредственное общение между сервером и клиентом. 2 программы клиент-сервера подключаются на один порт. И осуществляют общение об очередности хода, готовности на переход к другой части игры, подбитых или уничтоженных кораблях. ****

****

****

## Проблемы в процессе выполнения проекта

* Информация о построении подобных сетей в интернете была крайне мала. Само построение этой сети и её отношений далось тяжелым методом проб и ошибок. Отсутствие команды для нахождения собственного IP и вероятность того что придет больше или меньше двух нужных мне IP
* Общение между двумя одинаковыми программами, которые должны вести себя немного по-разному
* Не четкое представление о том, как должны строится отношения в сети, из-за чего её пришлось переписывать заново около десятка раз.

## Заключение

В последнее время потребность в многопользовательских играх высока. А значит изучение сетевых технологий, построение сетей и их отношений играет ключевую роль в разработке игр.

Мой проект заключается в создании сети и её рациональном использовании. Эти наработки и знания, полученные благодаря проекту можно использовать для множества игр и приложений: каких-либо чатов, тех же самых шахмат и прочих стратегий и многого другого.

Была написана компактная программа с возможностью для расширения, это позволяет в готовый проект добавлять какие угодно нововведения.

## Список использованной литературы

<https://ru.wikipedia.org/wiki/Одноранговая_сеть>

[https://javarush.ru/groups/posts/](https://javarush.ru/groups/posts/654-klassih-socket-i-serversocket-ili-allo-server-tih-menja-slihshishjh)

<https://docs.oracle.com/javase/7/docs/api/java/net/Socket.html>

[https://stackoverflow.com/questions/21868956/](https://stackoverflow.com/questions/21868956/how-to-obtain-the-actual-packet-size-byte-array-in-java-udp)

<https://javatalks.ru/topics/19298>

<http://www.quizful.net/post/java-socket-programming>

[https://ru.stackoverflow.com/questions/533429/](https://ru.stackoverflow.com/questions/533429/Параллельная-обработка-запросов-клиент-сервер)