

ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ПРОФЕССИОНАЛЬНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ИРКУТСКОЙ ОБЛАСТИ  
«АНГАРСКИЙ ТЕХНИКУМ СТРОИТЕЛЬНЫХ ТЕХНОЛОГИЙ»

Курсовая работа

Тема: Камень Ножницы Бумага Онлайн

Разработал \_\_\_\_\_ Кабанов А.П.

Руководитель \_\_\_\_\_ Денисюк А.В.

Ангарск, 2025

# СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	5
I. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ.....	8
1.1 Анализ предметной области .....	8
1.2 Архитектурная диаграмма взаимодействия компонентов .....	9
1.3 Выбор программного обеспечения .....	11
1.3.1 Язык программирования.....	11
1.3.2 Интегрированная среда разработки (IDE).....	13
1.3.3 Графическая библиотека.....	14
1.3.4 Работа с сетью.....	15
1.3.5 Система сборки.....	17
1.3.6 Система тестирования.....	18
1.4 Техническое задание.....	19
1.4.1 Общее описание функционала продукта .....	20
1.4.2 Требования к программной среде .....	21
1.4.3 Требования к аппаратной среде .....	22
1.4.4 Системные требования для пользователя .....	23
1.4.5 План разработки.....	24
1.5. Макет GUI интерфейса .....	25
1.5.1. Схематичное представление окон.....	25
1.5.2. Расположение элементов .....	26
1.5.3. Визуальные элементы и стиль .....	27
1.5.4. Ответственность за дизайн.....	27
1.6. Руководство по использованию приложения для администратора/разработчика .....	28
1.6.1. Запуск сервера и клиента .....	29
1.6.2. Логика игрового процесса.....	30
1.7. Руководство конечного пользователя.....	31

						КР-09.02.07-К-214-25ПЗ					
Изм.	Кол.уч	Лист	№ док.	Подпись	Дата	Камень Ножницы  Бумага Онлайн			<u>Лит.</u>	Лист	
Разраб.	Кабанов А.П.										
Пров.	Денисюк А.В.										
Н.контр											
Утв.						ГАПОУ ИО АТСТ					



## ВВЕДЕНИЕ

Современное общество невозможно представить без информационных технологий, которые проникают во все сферы человеческой жизни. Особое место в этой системе занимают интерактивные приложения и игры, способствующие не только развлечению, но и развитию логики, стратегического мышления и взаимодействия между пользователями. В условиях бурного роста цифровизации и популярности удалённых сервисов особое значение приобретают сетевые игры, позволяющие пользователям из разных точек мира участвовать в игровом процессе в реальном времени.

Одной из наиболее простых и в то же время популярных интеллектуальных игр является «Камень, ножницы, бумага». Это классическая игра, в которую играют миллионы людей по всему миру. Её простота делает её универсальной для всех возрастов и уровней подготовки. Несмотря на минималистичные правила, игра содержит элементы случайности и стратегии, что делает её интересной для реализации в цифровом формате. Особенно актуально создание сетевой версии игры, когда игроки могут состязаться друг с другом, находясь на расстоянии, используя при этом общую игровую платформу.

Основная цель данного курсового проекта — разработка сетевой игры «Камень, ножницы, бумага» с графическим пользовательским интерфейсом на языке Java, реализованной по архитектуре клиент-сервер. Для построения пользовательского интерфейса используется технология JavaFX, а взаимодействие между клиентами осуществляется посредством сетевых сокетов. Такой подход обеспечивает гибкость и масштабируемость приложения, а также позволяет реализовать понятную и отзывчивую игровую логику.

Проект демонстрирует ключевые аспекты разработки клиент-серверных приложений, включая сетевое взаимодействие, многопоточность,

						КР-09.02.07- К-214-23ПЗ	Лист
							5
Изм.	Кол.уч	Лист	№ докум.	Подпись	Дата		

синхронизацию данных и реализацию пользовательского интерфейса. В рамках проекта будут рассмотрены принципы построения графического интерфейса, особенности работы с потоками данных, передающихся через сокеты, а также способы обеспечения корректной синхронизации ходов игроков и отображения результатов.

Выбор языка программирования Java обусловлен его широким распространением, богатым набором библиотек и встроенной поддержкой сетевого программирования. Кроме того, Java предоставляет средства кроссплатформенной разработки, что позволяет запускать разработанное приложение на различных операционных системах без модификации исходного кода. Использование JavaFX, как современного фреймворка для построения GUI, даёт возможность создавать интуитивно понятный и визуально привлекательный интерфейс, что особенно важно в игровых приложениях.

Разработка проекта ведётся с учётом принципов модульности, удобства расширения и повторного использования кода. Программа разделяется на клиентскую и серверную части, каждая из которых отвечает за выполнение определённых функций и взаимодействует с другой частью по установленному протоколу.

В процессе выполнения проекта планируется:

- Спроектировать и реализовать серверную часть, принимающую подключения от двух клиентов и координирующую игровой процесс;
- Разработать клиентское приложение с графическим интерфейсом, позволяющее пользователю выбирать ход и получать результат игры;
- Обеспечить корректный обмен сообщениями между клиентами через сервер;
- Провести тестирование приложения в различных условиях, выявить и устранить возможные ошибки;

						КР-09.02.07- К-214-23ПЗ	Лист 6
Изм.	Кол.уч	Лист	№ докум.	Подпись	Дата		

- Сформировать полную документацию, включающую теоретическую часть, руководство пользователя и описание архитектуры системы.

Таким образом, данный курсовой проект не только позволяет закрепить теоретические знания в области программирования, сетевых технологий и проектирования интерфейсов, но и получить практические навыки реализации полноценных программных решений с использованием современных инструментов разработки. Проект может быть использован как основа для более сложных многопользовательских игр или как учебный пример для изучения принципов построения клиент-серверных приложений на языке Java.

						КР-09.02.07- К-214-23ПЗ	Лист
							7
Изм.	Кол.уч	Лист	№ докум.	Подпись	Дата		

# I. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

## 1.1 Анализ предметной области

Игра «Камень, ножницы, бумага» — это классическая игра с элементами случайности и тактики, используемая как в развлекательных, так и в обучающих целях. В основе игры лежат три возможных хода: камень, ножницы и бумага, которые сопоставляются по круговому принципу: камень побеждает ножницы, ножницы побеждают бумагу, бумага побеждает камень. При совпадении ходов у обоих игроков — фиксируется ничья.

Несмотря на свою простоту, игра используется в самых разных контекстах — от детских развлечений до принятия случайных решений. Именно эта универсальность делает её идеальной кандидатурой для цифровой реализации в виде сетевого приложения, особенно с минималистичным дизайном и быстрой сессией игры, которая требует лишь короткого времени взаимодействия между двумя игроками.

В условиях удалённой работы, дистанционного обучения и стремительного развития онлайн-коммуникаций сетевые игры становятся всё более популярными. Пользователи предпочитают приложения, не требующие сложной установки, быстро подключающиеся и понятные с первого взгляда. Это делает разработку сетевой версии такой игры, как «Камень, ножницы, бумага», актуальной задачей, особенно в контексте учебных и исследовательских проектов, направленных на закрепление знаний в области клиент-серверной архитектуры.

Разработка игры также позволяет студенту изучить и реализовать:

- работу с сокетами для организации сетевого взаимодействия;
- принципы построения графического интерфейса пользователя на JavaFX;
- концепции многопоточности и синхронизации данных в многопользовательских приложениях;

- обработку сетевых сообщений и реализацию протокола обмена;
- базовую игровую механику и реализацию логики в модели MVC (Model-View-Controller).

Таким образом, предметная область охватывает как аспекты гейм-дизайна, так и принципы программной инженерии, сетевых взаимодействий и человеко-машинного интерфейса. Особое внимание в ходе реализации приложения уделяется удобству пользователя, минимальной задержке между действиями и ясной визуальной обратной связи.

## 1.2 Архитектурная диаграмма взаимодействия компонентов

В отличие от типичных информационных систем, в которых используется модель баз данных и диаграмма «сущность-связь» (ER-диаграмма), приложение «Камень, ножницы, бумага онлайн» реализует сетевое взаимодействие между двумя игроками без хранения данных в базе. Поэтому в данном разделе целесообразно представить архитектурную диаграмму клиент-серверного взаимодействия и логическую структуру компонентов приложения.

### Архитектура приложения

Приложение построено по двухуровневой архитектуре:

1. Клиентская часть (интерфейс пользователя, отображение хода, отправка/получение сообщений).
2. Серверная часть (ожидание подключения клиента, обработка хода, передача данных).

Каждый экземпляр приложения может выступать либо сервером, либо клиентом, что определяется при запуске. Обмен данными осуществляется с помощью TCP-сокетов — стабильного сетевого протокола, обеспечивающего надёжную доставку сообщений.

Всё приложение состоит из следующих логических блоков:

						КР-09.02.07- К-214-23ПЗ	Лист 9
Изм.	Кол.уч	Лист	№ докум.	Подпись	Дата		



- GUI-компоненты (на JavaFX): интерфейс с кнопками, выпадающим списком для выбора хода, отображением результата, картинками.
- Контроллер (MainController.java): связывает графику с логикой, обрабатывает действия пользователя, управляет отображением и взаимодействием.
- Сетевые модули: создание и подключение сокетов, обмен сообщениями (в виде строк).
- Игровая логика: сравнение ходов, определение победителя, обработка ничьи.

#### Формат передаваемых данных

Обмен данными между игроками осуществляется в текстовом виде.

Каждое сообщение — это строка, содержащая выбранный ход на английском языке:

- "rock"
- "paper"
- "scissors"

Такая простая схема позволяет легко реализовать передачу и парсинг данных без дополнительных сериализаций или структур.

#### Обработка состояния игры

Каждый игрок может выбрать ход один раз за раунд. После получения хода противника приложение сравнивает оба значения и отображает результат. Обработка происходит асинхронно, с использованием многопоточности (через Thread) для ожидания данных и параллельной работы интерфейса.

Таким образом, архитектура приложения обеспечивает простую и надёжную систему, которую можно расширять: например, добавляя поддержку чата, рейтинговой системы, ботов или анимации.

### 1.3 Выбор программного обеспечения

Разработка сетевой игры «Камень, ножницы, бумага онлайн» требует тщательного подхода к выбору программного обеспечения, так как от него зависит не только корректная работа приложения, но и удобство разработки, сопровождения и возможного масштабирования проекта в будущем. Современные требования к прикладному программному обеспечению предъявляют высокие ожидания в отношении производительности, стабильности, кроссплатформенности и визуальной составляющей интерфейса. В данном разделе будет подробно рассмотрен выбор основных компонентов программного обеспечения, задействованных при реализации клиент-серверной архитектуры игры.

#### 1.3.1 Язык программирования

В качестве основного языка программирования был выбран Java. Этот выбор обусловлен рядом факторов, среди которых можно выделить следующие:

- Кроссплатформенность. Одним из ключевых преимуществ языка Java является его способность запускать приложения на различных операционных системах, таких как Windows, Linux и macOS, без необходимости вносить изменения в исходный код. Java-приложения могут быть запущены на любой системе, поддерживающей Java Virtual Machine (JVM), что обеспечивает универсальность и удобство при тестировании и развертывании на различных платформах. Это особенно важно для онлайн-игры, так как она должна работать на различных устройствах и операционных системах.
- Поддержка объектно-ориентированной парадигмы. Java активно использует объектно-ориентированный подход, который помогает грамотно структурировать код, организуя его в классы и методы с чётко определёнными задачами. Это облегчает понимание структуры

						КР-09.02.07- К-214-23ПЗ	Лист
							11
Изм.	Кол.уч	Лист	№ докум.	Подпись	Дата		

программы, упрощает её расширение и поддержку, а также способствует уменьшению числа ошибок.

- Мощные средства работы с сетью. Java предоставляет встроенные библиотеки для работы с сетевыми соединениями, такие как `java.net`, что позволяет удобно реализовывать как серверные, так и клиентские компоненты. В нашем случае важно было использовать низкоуровневые механизмы для сетевого взаимодействия, и Java идеально подходит для реализации сетевых приложений с использованием TCP/IP соединений.
- Поддержка многозадачности и многопоточности. Для реализации сетевой игры важно иметь возможность одновременно обрабатывать запросы от нескольких пользователей. Java предоставляет богатые средства для многозадачности и многопоточности, включая удобные API для работы с потоками, что даёт возможность эффективно управлять процессами ввода-вывода и сетевыми операциями без блокировки пользовательского интерфейса.
- Большое сообщество и множество готовых решений. Java имеет огромное сообщество разработчиков, что делает доступным множество библиотек и фреймворков. Это значительно ускоряет процесс разработки, так как можно использовать уже готовые решения для большинства задач, будь то обработка сетевых соединений, создание графического интерфейса или взаимодействие с базами данных.

Таким образом, язык Java полностью отвечает целям и требованиям поставленной задачи. Он позволяет создать надёжную, гибкую и масштабируемую архитектуру, что особенно важно в проекте, который потенциально будет расширяться или изменяться в будущем.

						КР-09.02.07- К-214-23ПЗ	Лист 12
Изм.	Кол.уч	Лист	№ докум.	Подпись	Дата		

### 1.3.2 Интегрированная среда разработки (IDE)

Для реализации проекта была выбрана IntelliJ IDEA — одна из самых популярных интегрированных сред разработки для языка Java. Данный выбор обусловлен рядом факторов, которые делают IntelliJ IDEA удобным и мощным инструментом для разработчиков, занимающихся созданием Java-приложений:

- Высокая производительность и стабильность. IntelliJ IDEA зарекомендовала себя как среда с высокой производительностью, что позволяет работать с проектами любого размера. Даже большие проекты с множеством зависимостей и классов не замедляют работу IDE.
- Интеллектуальные подсказки кода (code completion). IntelliJ IDEA предоставляет интеллектуальные подсказки и автозаполнение кода, что существенно ускоряет процесс написания программы, позволяет избежать множества ошибок и повышает продуктивность разработчика. Это особенно полезно при работе с большими кодовыми базами и сложными API.
- Удобная работа с JavaFX и Maven. Среда разработки полностью поддерживает работу с JavaFX — фреймворком для создания графических интерфейсов. Кроме того, IntelliJ IDEA идеально интегрируется с Maven, инструментом для управления зависимостями и автоматизации сборки проектов. Это позволяет легко управлять зависимостями проекта, автоматизировать сборку и тестирование.
- Поддержка отладки, тестирования и управления зависимостями. IntelliJ IDEA оснащена мощными средствами отладки, что позволяет эффективно устранять ошибки в коде. Также она поддерживает юнит-тестирование, что помогает в обеспечении качества кода и предотвращении багов на ранних стадиях разработки.

- Гибкость в настройке проекта и его структуры. В IntelliJ IDEA можно легко настроить структуру проекта, применить различные плагины, а также настроить интеграцию с системами контроля версий, такими как Git. Это важно для командной работы, а также для упрощения процесса тестирования и развертывания приложения.

Таким образом, IntelliJ IDEA является удобной и мощной средой разработки для реализации проекта, так как она поддерживает весь спектр инструментов, необходимых для работы с Java и JavaFX, и обеспечивает комфортную и стабильную работу при разработке и отладке.

### 1.3.3 Графическая библиотека

Для реализации интерфейса пользователя была использована библиотека JavaFX. Этот выбор был обусловлен рядом факторов, которые делают JavaFX идеальным инструментом для создания графических интерфейсов в Java-приложениях:

- Современный внешний вид интерфейсов. JavaFX предлагает гибкие возможности по стилизации и настройке элементов управления, что позволяет создавать эстетически привлекательные и современные пользовательские интерфейсы. Это особенно важно для игровых приложений, где внешний вид и взаимодействие с пользователем являются ключевыми аспектами.
- Простота в использовании. JavaFX обладает интуитивно понятной архитектурой и легко осваиваем, что позволяет быстро разрабатывать сложные интерфейсы. Также библиотека предоставляет большое количество готовых компонентов для создания кнопок, панелей, окон и других элементов управления.
- Поддержка FXML. Важной особенностью JavaFX является поддержка FXML — языка разметки, который позволяет отделить логику программы от её дизайна. Это даёт возможность разработчикам и

дизайнерам работать над проектом параллельно, а также упрощает поддержку кода, так как разделяет бизнес-логику и визуальные элементы.

- Интеграция с Java. JavaFX полностью совместим с языком Java, что позволяет работать с ним без использования дополнительных языков программирования или фреймворков. Это упрощает разработку, так как вся логика приложения остаётся в одном языке, а взаимодействие с графическим интерфейсом не требует дополнительных настроек или библиотек.
- Поддержка работы с изображениями и анимациями. JavaFX поддерживает работу с изображениями, анимациями и событиями, что является необходимым для игрового интерфейса. Для отображения хода пользователя, таких как изображения камня, ножниц и бумаги, JavaFX предлагает удобные механизмы для загрузки и отображения графических объектов, а также анимацию, которая делает интерфейс более живым и динамичным.
- Поддержка событий. JavaFX предоставляет мощную систему обработки событий, что позволяет реализовывать интерактивность в приложении. К примеру, действия игрока могут вызывать изменение состояния интерфейса, отображение всплывающих окон, изменение кнопок и другие визуальные эффекты.

Таким образом, JavaFX является отличным выбором для создания графического интерфейса для сетевой игры, так как она предоставляет все необходимые инструменты для быстрой и эффективной разработки современных интерфейсов с поддержкой анимаций, изображений и событий.

#### 1.3.4 Работа с сетью

Для реализации сетевого взаимодействия использовался пакет `java.net`, предоставляющий базовые средства для работы с TCP/IP-соединениями. В

						КР-09.02.07- К-214-23ПЗ	Лист
							15
Изм.	Кол.уч	Лист	№ докум.	Подпись	Дата		

этом проекте важно обеспечить корректное и быстрое взаимодействие между клиентами и сервером, что достигается через использование сокетов. Пакет `java.net` включает в себя следующие ключевые элементы:

- Серверные сокет (ServerSocket). Серверная часть игры использует `ServerSocket`, который прослушивает заданный порт и принимает подключения от клиентов. Это позволяет серверу ожидать соединения от клиентов и обрабатывать запросы каждого игрока. Важно, чтобы сервер мог обрабатывать несколько подключений одновременно, так как возможно взаимодействие множества игроков. Для этого используется многозадачность (многопоточность), чтобы каждому клиенту был выделен отдельный поток обработки данных.
- Клиентские сокет (Socket). Клиентская часть игры использует `Socket` для подключения к серверу по IP-адресу и порту. При установлении соединения клиент может отправлять и получать данные с сервера, включая информацию о сделанных ходах и результатах игры. Использование сокетов позволяет создать устойчивое соединение между клиентом и сервером, которое активно используется в процессе игры.
- Потоки ввода-вывода (`InputStream`, `OutputStream`). Для обмена данными между клиентом и сервером используются потоки ввода и вывода. Это позволяет передавать и получать информацию о ходах игроков, а также другие данные, необходимые для игры. Потоки играют ключевую роль в передаче данных в реальном времени, что особенно важно для многопользовательских онлайн-игр, где задержки могут существенно повлиять на качество взаимодействия.

Работа с сетевыми соединениями через сокет является основой для создания клиент-серверной архитектуры игры. Важно отметить, что использование сокетов в Java предоставляет низкоуровневый контроль над

процессом передачи данных, что позволяет оптимизировать скорость работы сети и гарантировать надежность соединения.

Для обеспечения синхронной работы игры, сервер и клиент должны обмениваться данными в реальном времени, что требует грамотной реализации многозадачности. Сервер, в свою очередь, должен корректно обрабатывать сообщения от нескольких клиентов и следить за состоянием игры для каждого игрока.

### 1.3.5 Система сборки

Для управления зависимостями и сборкой проекта использовалась система Maven. Maven является одним из самых популярных инструментов для автоматизации сборки и управления зависимостями в Java-проектах. В рамках нашего проекта Maven выполняет несколько ключевых функций:

- Управление зависимостями. Maven позволяет легко добавлять внешние библиотеки и фреймворки, необходимые для работы проекта. Например, JavaFX и другие необходимые компоненты можно подключить через Maven, что упрощает их использование и автоматическую загрузку в проект. Это позволяет избежать проблем с несовместимостью версий и облегчает поддержку проекта, так как все зависимости управляются централизованно.
- Автоматизация сборки. Maven автоматически компилирует исходный код, создает исполняемые файлы и упаковывает проект в архивы (например, JAR-файлы). Это ускоряет процесс сборки, особенно когда проект растет и включает множество файлов и зависимостей. Также автоматизация сборки позволяет избежать ошибок, связанных с некорректным компилированием или пропуском важных файлов.
- Управление проектами и версиями. Maven поддерживает управление версиями, что позволяет легко обновлять зависимости и следить за изменениями в проекте. Также Maven помогает в организации проекта,



разделяя его на модули, что особенно полезно для больших и сложных приложений.

- Интеграция с системами контроля версий. Maven интегрируется с популярными системами контроля версий, такими как Git, что позволяет удобно управлять версиями исходного кода и синхронизировать изменения в проекте между различными разработчиками. Это облегчает командную работу и управление различными этапами разработки.

Использование Maven для управления зависимостями и сборки проекта ускоряет процесс разработки и позволяет избежать множества проблем, связанных с ручным подключением библиотек, настройкой компиляции и сборкой проектов. Это важный инструмент, который помогает в организации и поддержке кода, особенно в крупных проектах с множеством внешних зависимостей.

### 1.3.6 Система тестирования

Для обеспечения качества разработки и минимизации количества ошибок в проекте использовалась система тестирования. Одним из популярных инструментов для тестирования в Java является JUnit. Система тестирования позволяет разрабатывать и проводить автоматические проверки различных частей программы, что важно для поддержки качества кода на всех этапах разработки.

- Юнит-тесты. С помощью JUnit были созданы юнит-тесты для проверки работы основных компонентов игры, таких как логика обработки ходов игроков, сетевое взаимодействие и отображение графического интерфейса. Юнит-тесты позволяют автоматически проверять корректность работы каждого метода и компонента, что значительно снижает вероятность ошибок в коде.

- Интеграционные тесты. Для проверки взаимодействия между сервером и клиентом были написаны интеграционные тесты. Эти тесты проверяют работу сетевого взаимодействия, отправку и получение данных между клиентом и сервером. Интеграционные тесты помогают удостовериться в том, что вся система работает как единое целое и нет проблем с взаимодействием компонентов.
- Тестирование производительности. Также была проведена серия тестов для оценки производительности игры, чтобы убедиться в том, что она работает без задержек, даже при значительном числе подключений. Эти тесты позволили выявить возможные узкие места в работе сети и интерфейса, которые могли бы ухудшить качество игры.

Тестирование играет важную роль в поддержке надежности и стабильности приложения. Оно позволяет вовремя обнаруживать и исправлять ошибки, а также помогает поддерживать код в актуальном и работоспособном состоянии на протяжении всей разработки.

Этот подход к выбору программного обеспечения позволяет создать качественное и надёжное приложение для реализации сетевой игры «Камень, ножницы, бумага онлайн». В дальнейшем, при масштабировании игры или добавлении новых функций, выбранные инструменты и подходы обеспечат гибкость, расширяемость и стабильность проекта.

#### 1.4 Техническое задание

Техническое задание (ТЗ) является важным этапом разработки программного обеспечения, так как в нем закрепляются все требования к проекту, а также основные принципы работы системы. Для создания сетевой игры «Камень, ножницы, бумага онлайн» был составлен ряд требований, которые будут описаны в данном разделе. ТЗ включает в себя описание функционала продукта, требования к программному и аппаратному обеспечению, а также системные требования для конечных пользователей.

						КР-09.02.07- К-214-23ПЗ	Лист 19
Изм.	Кол.уч	Лист	№ докум.	Подпись	Дата		

### 1.4.1 Общее описание функционала продукта

Игра «Камень, ножницы, бумага онлайн» представляет собой многопользовательскую сетевую игру, которая реализуется с использованием клиент-серверной архитектуры. Основной функционал игры включает в себя:

Регистрация и авторизация пользователей. Игроки должны иметь возможность создать учётную запись или войти в существующую, чтобы участвовать в игре. В процессе регистрации необходимо указать никнейм, который будет отображаться во время игры. Авторизация обеспечит доступ к игровым сессиям.

Создание и подключение к серверу. Пользователи могут создавать серверы для игры или подключаться к уже существующим. Каждый сервер будет поддерживать подключение нескольких игроков одновременно. В случае создания сервера, игрок должен иметь возможность указать порт и IP-адрес, а также выбрать режим игры (например, количество раундов).

Выбор хода. Во время игрового процесса каждый игрок может выбрать один из трёх возможных ходов: камень, ножницы или бумага. Выбор хода должен отображаться на экране игрока и быть передан через сеть на сервер.

Определение победителя. После того как оба игрока сделают свои ходы, сервер анализирует комбинации и определяет победителя. Результат раунда отображается для обоих игроков.

Обмен сообщениями. Игроки должны иметь возможность обмениваться сообщениями в чате во время игры. Это может быть полезно для общения между игроками или для получения уведомлений от сервера.

						КР-09.02.07- К-214-23ПЗ	Лист
							20
Изм.	Кол.уч	Лист	№ докум.	Подпись	Дата		

Подсчёт очков. Каждый раунд игры будет сопровождаться подсчётом очков, которые сохраняются на сервере. В зависимости от выбранных ходов, одному из игроков может быть присуждена победа, а другому — поражение.

Завершение игры. Игра завершается, когда один из игроков набирает требуемое количество очков или когда оба игрока соглашаются завершить игру. При этом игроки могут увидеть итоговый счёт и узнать, кто стал победителем.

#### 1.4.2 Требования к программной среде

Для разработки игры был выбран набор технологий, подходящих для создания сетевого приложения с графическим интерфейсом. Важно, чтобы программная среда обеспечивала поддержку всех необходимых библиотек и фреймворков, а также позволяла эффективно управлять проектом.

Язык программирования: Java 17, так как этот язык поддерживает объектно-ориентированную парадигму, имеет богатую экосистему библиотек и является кроссплатформенным. Использование Java также позволяет эффективно работать с многопоточностью, что критично для реализации клиент-серверной архитектуры.

Среда разработки: IntelliJ IDEA — это популярная интегрированная среда разработки (IDE), которая поддерживает работу с Java, JavaFX и Maven. IntelliJ IDEA включает в себя средства для отладки, анализа и тестирования кода, что существенно ускоряет процесс разработки.

Библиотеки и фреймворки: Для реализации графического интерфейса используется JavaFX, а для сетевой части проекта — пакет `java.net`, который предоставляет средства для работы с сокетами и TCP/IP-соединениями.

Система сборки: Maven, используемая для управления зависимостями, сборки проекта и организации структуры. Maven автоматизирует процесс компиляции и упаковки проекта, что делает его удобным для работы в команде и управления зависимостями.

### 1.4.3 Требования к аппаратной среде

Для нормальной работы игры «Камень, ножницы, бумага онлайн» игрокам потребуется оборудование, соответствующее минимальным системным требованиям. Серверная часть игры также должна быть размещена на сервере, который будет обеспечивать стабильное подключение и обработку запросов от клиентов.

Минимальные системные требования для сервера:

Процессор: 2 ядра, частота не менее 2.0 GHz.

Оперативная память: 2 GB.

Жёсткий диск: 100 MB свободного места для размещения серверных файлов.

Сетевое соединение: стабильное соединение с интернетом с пропускной способностью не менее 1 Mbit/сек.

Операционная система: Windows Server, Linux или macOS, с поддержкой Java.

Минимальные системные требования для клиентов:

Процессор: 1 ядро, частота не менее 1.5 GHz.

Оперативная память: 1 GB.

Видеокарта: с поддержкой OpenGL 2.1 и выше.

Жёсткий диск: 50 MB свободного места для игры.

Операционная система: Windows, macOS, Linux.

Необходимая версия JDK (Java Development Kit) — JDK 17 или выше.

#### 1.4.4 Системные требования для пользователя

Игроки должны иметь устройства с соответствующими характеристиками для комфортной игры. Помимо аппаратных требований, пользователи должны иметь стабильное интернет-соединение для того, чтобы подключиться к игровому серверу и поддерживать актуальные данные о ходе игры в реальном времени.

Системные требования для пользователя:

Операционная система: Windows 7 или новее, macOS 10.11 или новее, Linux.

Программное обеспечение: установлена последняя версия JDK (Java 17).

						КР-09.02.07- К-214-23ПЗ	Лист
							23
Изм.	Кол.уч	Лист	№ докум.	Подпись	Дата		

Интернет: стабильное подключение к интернету с минимальной пропускной способностью 1 Mbit/c.

#### 1.4.5 План разработки

Разработка игры «Камень, ножницы, бумага онлайн» включает несколько этапов, каждый из которых включает в себя задачи по проектированию, реализации, тестированию и оптимизации. Разработка будет проходить поэтапно с учётом потребностей заказчика и пользователей.

Этап 1: Проектирование системы. На данном этапе создаются все необходимые диаграммы (включая диаграмму классов и диаграмму последовательности), определяются требования и архитектура системы. Разрабатывается подробное техническое задание и описываются все ключевые функциональные возможности.

Этап 2: Разработка серверной и клиентской части. Реализуются основные компоненты клиент-серверной архитектуры, включая создание серверной логики и клиентского интерфейса. Клиенты должны иметь возможность подключаться к серверу, делать ходы, видеть результаты и общаться друг с другом.

Этап 3: Тестирование и отладка. Проводятся юнит-тесты и интеграционные тесты для проверки функциональности игры. Также тестируется производительность игры, чтобы она стабильно работала.

Этап 4: Развертывание и внедрение. Сервер разворачивается на хостинге или локальной машине для обеспечения работы игры в сети. Проводятся финальные проверки перед запуском.

Этап 5: Поддержка и улучшения. После запуска игры продолжается её поддержка, исправление ошибок и добавление новых функций в соответствии с отзывами пользователей.

						КР-09.02.07- К-214-23ПЗ	Лист
							24
Изм.	Кол.уч	Лист	№ докум.	Подпись	Дата		

## 1.5. Макет GUI интерфейса

Интерфейс игры "Камень, ножницы, бумага" был спроектирован с учётом удобства и интуитивно понятного взаимодействия с пользователем. Все элементы управления, такие как кнопки, изображения и текстовые поля, расположены так, чтобы создать простоту использования и минимизировать вероятность ошибок. Основной акцент был сделан на визуальной доступности всех элементов и логике их расположения.

### 1.5.1. Схематичное представление окон

1. Главное меню: Главное меню является начальной точкой игры. Здесь пользователи могут:

- Создать сервер: Это позволяет одному игроку стать хостом игры, предоставляя сервер для подключения других игроков.
- Подключиться: Эта кнопка позволяет игроку подключиться к уже работающему серверу, введя его IP-адрес и порт.
- Поле для ввода никнейма: Перед началом игры игрок должен ввести свой никнейм, который будет отображаться во время игрового процесса.
- Информация о статусе подключения: Под кнопками отображается текущий статус соединения, например: "Ожидание подключения" или "Подключено".

2. Окно выбора хода: После подключения к серверу и начала раунда игроки попадают в окно выбора хода:

- Три кнопки для выбора хода (камень, ножницы, бумага): Эти кнопки крупные и расположены в центре экрана для легкости выбора.
- Изображения выбранных ходов: После того как игрок выбрал свой ход, его выбор отображается в виде изображения (камень, ножницы или бумага).

						КР-09.02.07- К-214-23ПЗ	Лист 25
Изм.	Кол.уч	Лист	№ докум.	Подпись	Дата		



- Информация о противнике: В верхней части окна отображается имя противника и его выбор (если он уже сделал его).
- Кнопка "Играть": После того как оба игрока выбрали свои ходы, эта кнопка становится активной, и нажатие на неё отправляет ходы на сервер.

3. Окно результата: После завершения раунда отображается окно с результатом игры:

- Текстовое сообщение о результате: В окне появляется сообщение о победе, поражении или ничьей.
- Информация о ходе второго игрока: Игроки могут увидеть, что выбрал их противник.
- Кнопка "Новый раунд": Эта кнопка позволяет перейти к следующему раунду игры. После её нажатия оба игрока снова выбирают свои ходы.

#### 1.5.2. Расположение элементов

Расположение элементов интерфейса было продумано для создания лёгкости восприятия и минимизации количества кликов для выполнения операций. Основные компоненты интерфейса размещены следующим образом:

1. Главное меню:

- Кнопки "Создать сервер" и "Подключиться" расположены в нижней части экрана.
- Кнопка "ИГРАТЬ" расположена между кнопок "Создать сервер" и "Подключиться".

2. Окно выбора хода:

- Кнопка для выбора хода (камень, ножницы, бумага) расположена выпадающим списком с последующим нажатием кнопки "ИГРАТЬ".

- Изображения выбранных ходов отображаются ниже выпадающего списка.
- Кнопка "Играть" находится внизу окна, и её активность зависит от того, сделал ли второй игрок свой выбор.

### 3. Окно результата:

- После выбора ходов двумя игроками появляется сообщение о результате (например, "Вы выиграли!" или "Ничья").

## 1.5.3. Визуальные элементы и стиль

Интерфейс игры включает в себя следующие визуальные элементы:

- Изображения для каждого хода (камень, ножницы, бумага): Эти изображения чётко обозначают выбранные ходы, что позволяет пользователям легко воспринимать информацию.
- Цветовая палитра: Основные элементы управления (кнопки) и весь остальной интерфейс оформлен в яркие, но не раздражающие цвета для лёгкости восприятия.
- Шрифты: Для текста использовались чёткие и читаемые шрифты, которые легко воспринимаются на любом экране. В частности, крупные и жирные шрифты для отображения результатов игры и информации о противнике.

## 1.5.4. Ответственность за дизайн

Ответственность за дизайн интерфейса лежит на разработчиках, обеспечивших удобство и доступность игры для всех пользователей. Важно, чтобы каждый элемент интерфейса выполнял свою функцию и был легко доступен в любой момент игры.

						КР-09.02.07- К-214-23ПЗ	Лист
							27
Изм.	Кол.уч	Лист	№ докум.	Подпись	Дата		

## 1.6. Руководство по использованию приложения для администратора/разработчика

Для корректного функционирования игры и её настройки администраторы и разработчики должны иметь понимание принципов работы приложения, а также уметь запускать сервер и клиента, обеспечивать правильное завершение работы игры. Это руководство охватывает основные шаги по настройке и эксплуатации игры.

						КР-09.02.07- К-214-23ПЗ	Лист
							28
Изм.	Кол.уч	Лист	№ докум.	Подпись	Дата		

### 1.6.1. Запуск сервера и клиента

#### Запуск сервера:

- Для начала работы игры необходимо запустить серверное приложение на машине, которая будет работать как хост. Сервер принимает подключения от клиентов.
- После запуска сервер ожидает подключений от клиентов. Приложение будет слушать указанный порт и готово принять запросы от игроков.

#### Подключение клиента:

- Для подключения клиент должен запустить клиентское приложение, которое будет обращаться к серверу по IP-адресу и порту.
- Пошаговая инструкция:
- Нажмите на кнопку «Создать сервер» слева снизу в приложении, далее введите IP-адрес сервера и порт в соответствующие поля в главном окне клиента.
- Нажмите кнопку "Подключиться". После этого клиент будет соединён с сервером, и игрок сможет начать игру.

#### Проверка соединения:

- После подключения клиента к серверу приложение проверяет успешность соединения. Если соединение установлено успешно, игроки смогут начать выбор хода. В случае неудачного соединения выводится ошибка с соответствующим сообщением.

## 1.6.2. Логика игрового процесса

Игровой процесс происходит по следующему сценарию:

### 1. Выбор хода:

- После подключения клиента к серверу оба игрока переходят в окно выбора хода. В это время каждый игрок должен выбрать один из трёх вариантов: камень, ножницы или бумага.

### 2. Ожидание второго игрока:

- Игра блокирует выбор следующего хода, пока второй игрок не сделает свой выбор. После того как оба игрока выбрали ход, кнопка снова можно будет нажать.

### 3. Вывод результата:

- Как только оба игрока сделали свой выбор, сервер обрабатывает данные и вычисляет победителя. После этого на экране каждого игрока появляется результат игры — кто победил или если была ничья.

### 4. Переход к новому раунду:

- По завершению раунда оба игрока могут начать новый раунд, нажав на кнопку "Играть" и снова выбрав ход. Важно, что выбор хода возможен только после завершения предыдущего раунда.

### 5. Ошибки и исключения:

- Если один из игроков попытается сделать ход до того, как второй игрок выберет, появится сообщение об ошибке, уведомляющее, что нужно подождать.
- В случае отключения одного из игроков от сервера в процессе игры, второй игрок получит сообщение о том, что игра завершена из-за потери соединения.

## 1.7. Руководство конечного пользователя

Это руководство предназначено для обычных пользователей, желающих поиграть в сетевую игру «Камень, ножницы, бумага». Оно объясняет, как подключиться к игре, сделать ход, интерпретировать результат и что делать в случае ошибок.

### 1.7.1. Подключение к игре

#### 1. Выбор роли:

- При запуске игры пользователю предлагается выбрать одну из двух ролей:
  - Создать сервер — если пользователь хочет быть хостом и ожидать подключения другого игрока.
  - Подключиться к серверу — если пользователь знает IP-адрес и порт сервера и хочет присоединиться к игре.

#### 2. Ввод IP и порта:

- В появляющемся диалоговом окне пользователю необходимо ввести:
  - IP-адрес — адрес компьютера, где запущен сервер.
  - Порт — числовое значение, на котором работает сервер (обычно указывается по умолчанию или задаётся администратором).

#### 3. Проверка соединения:

- После нажатия кнопки «Подключиться» происходит проверка соединения.
- При успешном подключении пользователь получает подтверждение и переходит в интерфейс выбора хода.
- При ошибке появляется сообщение об ошибке, например: «Не удалось подключиться к серверу» или «Неверный IP или порт».

### 1.7.2. Выбор хода и участие в игре

#### 1. Выбор одного из вариантов:

- На экране игрока отображаются три кнопки или изображения, соответствующие возможным ходам:
  - Камень
  - Ножницы
  - Бумага
- Игрок кликает на соответствующее изображение или кнопку, тем самым выбирая свой ход.

#### 2. Ожидание второго игрока:

- После выбора хода игроку нужно нажать кнопку «Играть».
- Если второй игрок ещё не сделал ход, выводится предупреждение: «Ожидается выбор второго игрока».
- Также, если игрок повторно нажимает кнопку «Играть», не изменяя ход, появляется сообщение: «Ход уже выбран».

#### 3. Получение и отображение результата:

- После того как оба игрока выбрали свои ходы, на экране появляется всплывающее окно с результатом:
  - Победа, поражение или ничья.
  - Указывается имя (ник) второго игрока, с которым проводилась партия.
- Это окно появляется синхронно у обоих игроков.

### 1.7.3. Повторный раунд

#### 1. Выберите ход и снова нажмите на кнопку «Играть»:

- После завершения текущего раунда игроки могут сыграть ещё раз.
- Интерфейс сбрасывается и позволяет выбрать новый ход.

						КР-09.02.07- К-214-23ПЗ	Лист 32
Изм.	Кол.уч	Лист	№ докум.	Подпись	Дата		

## 2. Ограничения:

- Пока не завершён предыдущий раунд, нельзя начать новый. Также нельзя выбрать ход дважды.
- Были устранены ошибки, при которых можно было многократно нажимать на кнопку «Играть» и отправлять ходы, что нарушало логику игры.
- Также была исправлена ошибка, при которой выбранный ход не отображался (не менялось изображение).

### 1.7.4. Работа с ошибками

Приложение разработано с учётом типичных пользовательских ошибок, и включает уведомления и подсказки для удобства.

#### 1. Ошибка подключения:

- Если не удаётся подключиться к серверу — появляется всплывающее сообщение с причиной (например, неправильный IP/порт или сервер не запущен).

#### 2. Повторный ход:

- При попытке выбрать ход несколько раз или нажать «Играть» повторно — отображается сообщение: «Ход уже выбран».

#### 3. Недоступность сервера:

- Если соединение с сервером было потеряно, игрок получает уведомление: «Подключение прервано. Игра завершена».

### 1.8. Перспективы развития проекта

Разработка сетевой игры «Камень, ножницы, бумага» в её текущем виде представляет собой функциональный и удобный инструмент для развлечения двух игроков в реальном времени. Однако проект обладает потенциалом для дальнейшего расширения и усовершенствования:

						КР-09.02.07- К-214-23ПЗ	Лист 33
Изм.	Кол.уч	Лист	№ докум.	Подпись	Дата		



- Добавление статистики игроков: ведение учёта количества побед, поражений и ничьих.
- Расширение интерфейса: включение анимации ходов, более детализированных визуальных эффектов.
- Чат между игроками: возможность общения в процессе игры.
- Поддержка нескольких раундов подряд с автоматическим счётом.
- Добавление новых режимов: например, режим с дополнительными элементами вроде «ящерица» и «Спок» (как в расширенной версии игры).
- Мобильная адаптация: создание Android- или iOS-приложения на основе уже готовой логики.

Этот пункт подчёркивает, что текущий проект — не конечный продукт, а гибкая основа, на которую можно настраивать новые функции и улучшения в будущем.

						КР-09.02.07- К-214-23ПЗ	Лист
Изм.	Кол.уч	Лист	№ докум.	Подпись	Дата		34

## II. РАЗРАБОТКА ИГРЫ

### 2.1 Структура проекта

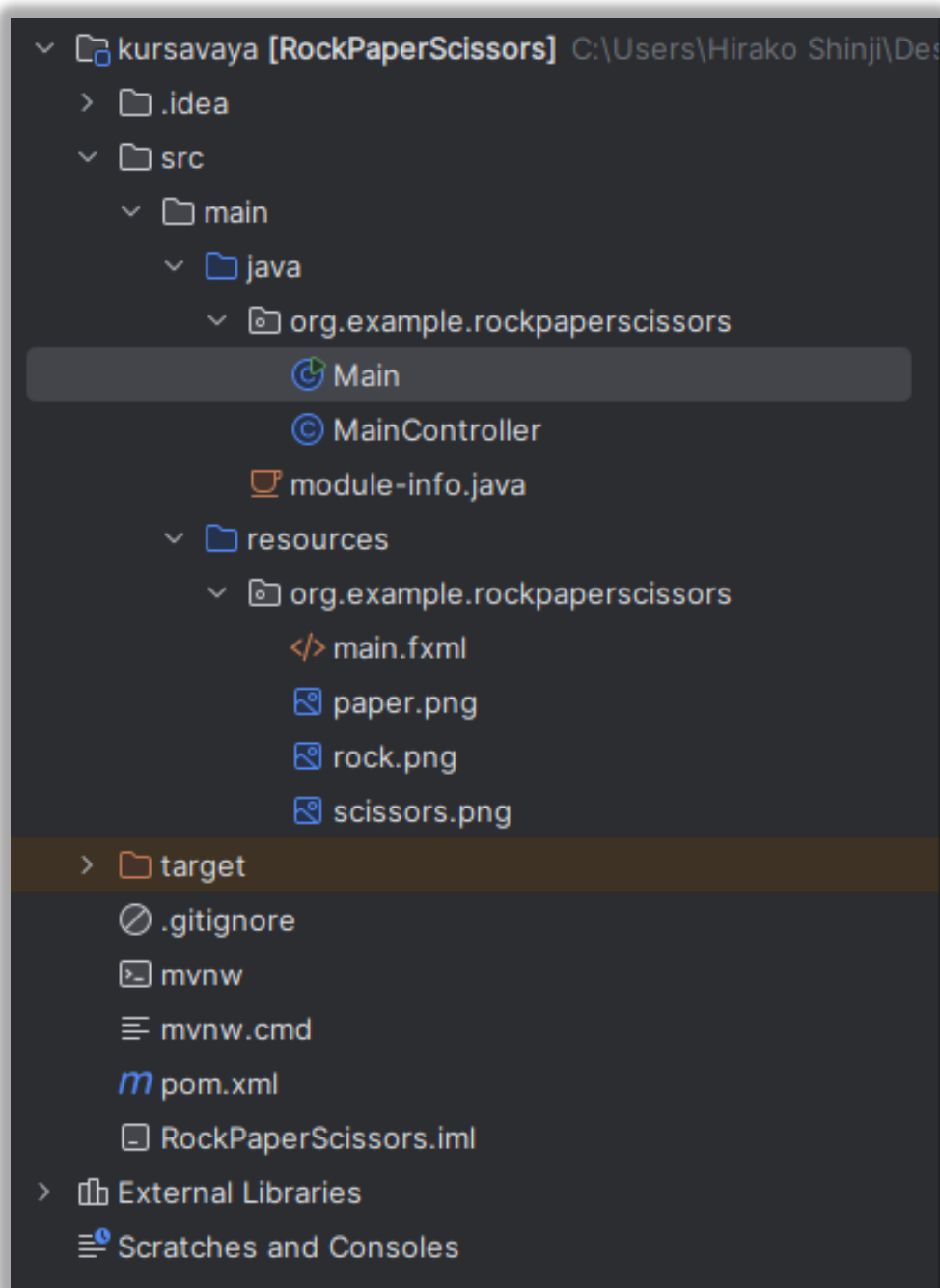


Рисунок 1

Проект состоит из следующих основных компонентов:

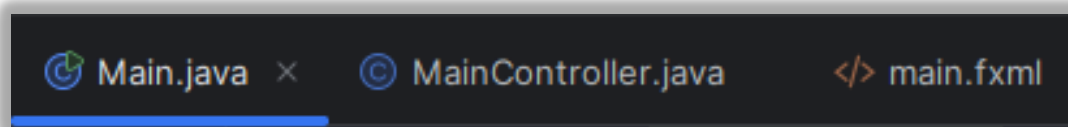


Рисунок 2

- Main.java: Главный класс приложения, запускающий интерфейс.
- MainController.java: Контроллер, обрабатывающий взаимодействие пользователя с интерфейсом.
- main.fxml: FXML-файл, описывающий графический интерфейс.

## 2.2 Интерфейс пользователя

Графический интерфейс выглядит следующим образом:

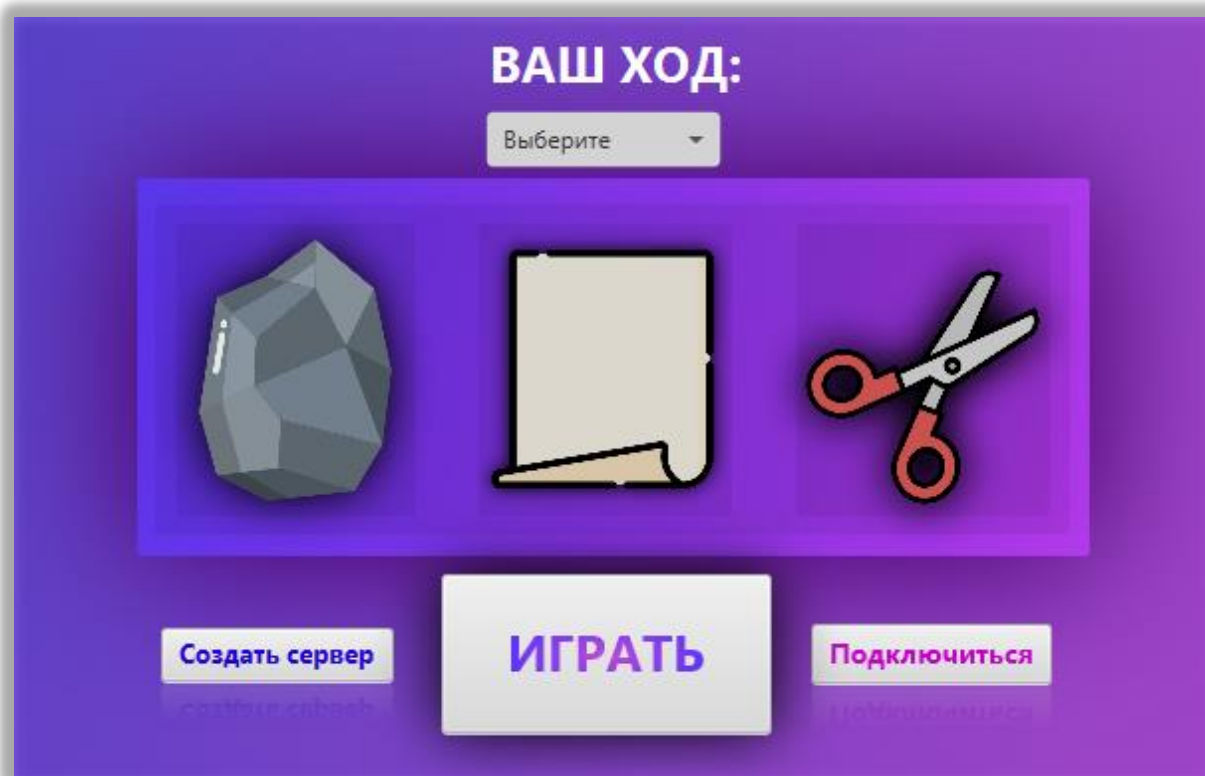


Рисунок 3

Также он включает такие элементы как:

- Выпадающий список для выбора хода (Камень, Ножницы, Бумага):

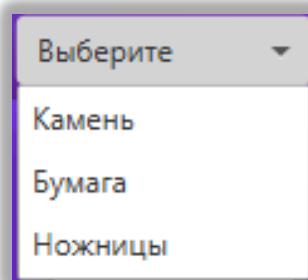


Рисунок 4

- Кнопки для создания сервера и подключения к нему:



Рисунок 5



Рисунок 6

- Кнопку "Играть" для отправки выбранного хода:



Рисунок 7

- Изображения, отображающие выбор игрока:



Рисунок 8

- Текстовое и графическое изображение выбранного хода:

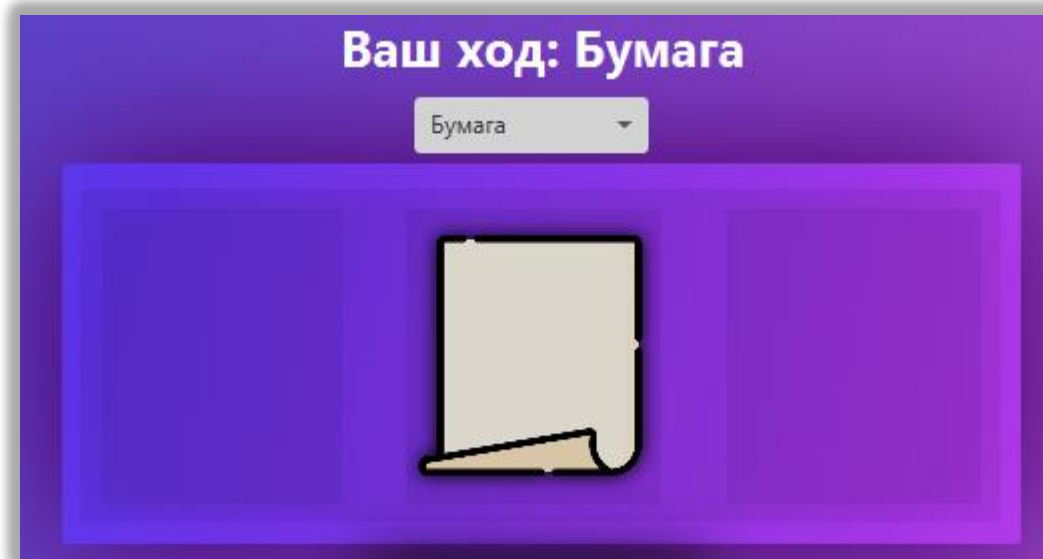


Рисунок 9

## 2.3 Логика игры

Логика игры реализована в классе MainController.java и включает:

- Выбор хода: Пользователь выбирает один из вариантов и нажимает кнопку "Играть".
- Ожидание выбора хода: Если пользователь не выбирает ход и нажимает кнопку "Играть", то программа покажет окно в котором попросит о выборе какого-то хода:

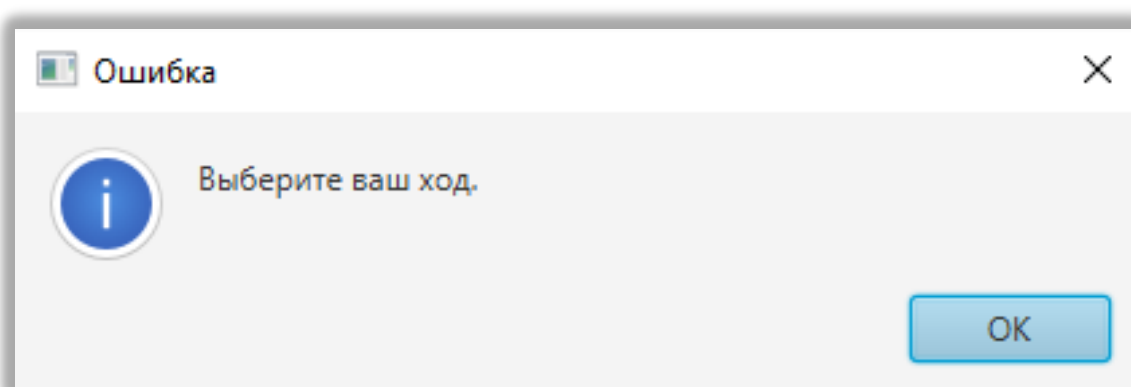


Рисунок 10

- Отправка хода: Выбранный ход отправляется через сокет на сервер или клиент.

- Получение хода противника: Программа ожидает ход второго игрока.
- Определение результата: Сравниваются ходы обоих игроков, и определяется победитель.
- Отображение результата: Результат раунда отображается в интерфейсе.

## 2.4 Сетевое взаимодействие

Сетевое взаимодействие реализовано с использованием Java Sockets:

- Создание сервера: Один из игроков запускает сервер, ожидающий подключения:

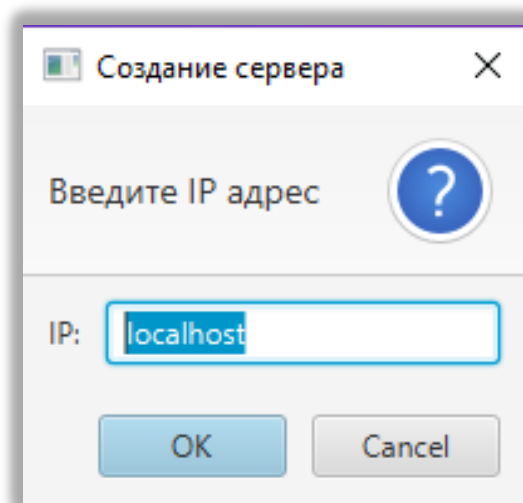


Рисунок 11

- Подключение к серверу: Второй игрок вводит IP-адрес и порт для подключения к серверу:

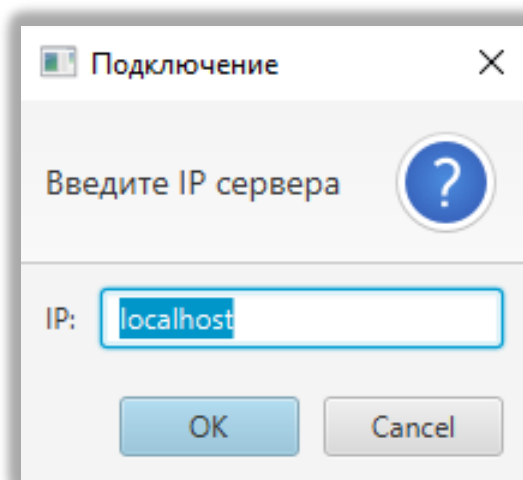


Рисунок 12

- Обмен данными: После установления соединения игроки обмениваются выбранными ходами.

## 2.5 Обработка исключений и завершение игры

Программа обрабатывает возможные ошибки, такие как:

- Невозможность создать сервер или подключиться к нему:

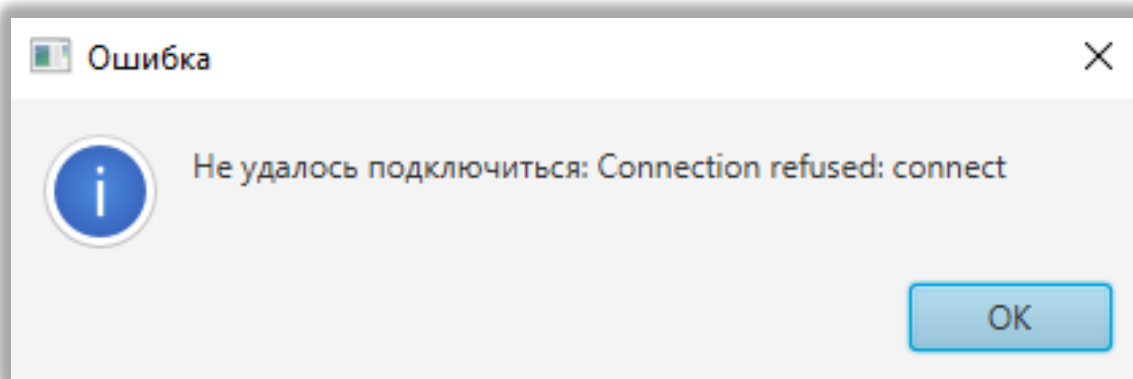


Рисунок 13

- Проблемы при отправке или получении данных:

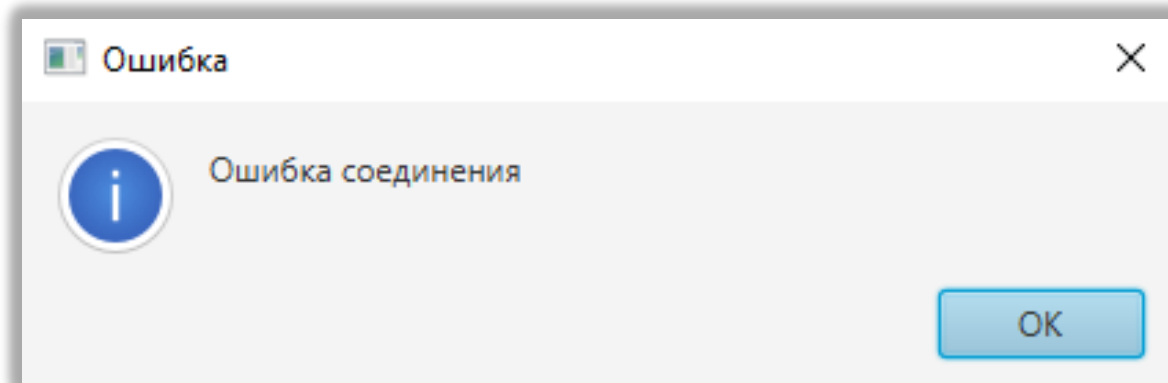


Рисунок 14

## 2.6 Тестирование приложения

Перед тем как начать полноценное использование игры, было проведено несколько этапов тестирования для проверки корректности работы всех функций.

1. Тестирование интерфейса:

- Проверка правильности отображения элементов интерфейса на различных разрешениях экранов.
- Тестирование всех кнопок, выпадающих списков и меток на предмет их реакции на действия пользователя.

Все элементы были протестированы на различных размерах окон. Особое внимание было уделено взаимодействию между пользователем и интерфейсом, включая проверку на наличие багов при изменении окна и переходах между экранами.

## 2.7 Отображение результата только у одного игрока:

На начальных этапах реализации игры, результат раунда отображался только у одного из игроков. Это создавало дисбаланс в восприятии игры, так как второй игрок не получал информации о текущем результате.

Чтобы устранить эту ошибку, была внедрена система, при которой результат отображается у обоих игроков одновременно в виде всплывающих окон, позволяя каждому увидеть результат игры.

## 2.8 Спам ходов:

Еще одной проблемой было то, что игроки могли многократно выбирать ход и отправлять его серверу. Это создавало ненужную нагрузку на сервер и приводило к потенциальным багам при обработке игровых данных.

Для решения этой проблемы была добавлена логика, которая проверяет, был ли уже выбран ход, и в случае повторного нажатия на кнопку "Играть" выводит предупреждение о том, что ход уже выбран. Это ограничивает возможность отправки нескольких ходов подряд и улучшает взаимодействие с пользователем.



## 2.9 Изменение картинки выбранного хода:

Еще одной важной ошибкой было то, что после выбора хода и нажатия на кнопку "Играть", изображение выбранного хода не обновлялось. Это сбивало игроков с толку и дезориентировало, так как они не могли видеть, какой именно ход был выбран.

Для исправления этой проблемы была внедрена логика, которая обновляет картинку в интерфейсе, показывая игроку его выбор.

## 2.10 Заключение

В ходе выполнения курсового проекта была разработана сетевая игра «Камень, ножницы, бумага» с графическим интерфейсом, реализованным на JavaFX, и механизмом сетевого взаимодействия через сокеты. Проект охватывает полный цикл разработки: от проектирования интерфейса и написания серверной и клиентской логики до обработки ошибок и обеспечения устойчивости приложения к сетевым сбоям.

В процессе реализации были достигнуты следующие цели:

- изучены и применены на практике основные принципы сетевого программирования на языке Java;
- создан графический интерфейс с использованием JavaFX, удобный и понятный для пользователя;
- реализована логика обмена данными между двумя игроками с помощью сокет-соединений;
- устранены ошибки, выявленные в процессе тестирования: предотвращена возможность многократного ввода хода, обеспечено отображение результатов на стороне обоих игроков, реализована смена изображения в зависимости от выбранного хода;
- добавлены механизмы защиты от некорректного поведения пользователя и автоматическая очистка ресурсов при завершении работы.

Проект позволяет двум пользователям запускать игру по сети, обмениваться ходами в реальном времени и получать результат в виде всплывающего окна. Создана прочная основа для дальнейшего развития проекта, включая расширение функционала, добавление анимации, реализацию чата и многопользовательского режима.

Таким образом, курсовая работа позволила не только закрепить знания по Java и работе с JavaFX и сокетами, но и приобрести практический опыт в разработке полноценных клиент-серверных приложений с графическим интерфейсом.

### 2.11 Список использованной литературы

1. Эккель Б. – Философия Java. 4-е издание. – СПб.: Питер, 2017.
2. Хорстманн К. – Основы программирования на Java. Том 1. – М.: Диалектика, 2021.
3. Oracle Documentation – Java SE Documentation.  
<https://docs.oracle.com/javase/>
4. Oracle JavaFX Documentation. <https://openjfx.io>
5. Knute, R. – Java Network Programming. – O'Reilly Media, 2020.
6. Sierra K., Bates B. – Head First Java. 3rd Edition. – O'Reilly, 2022.
7. Герберт Шилдт – Java. Полное руководство. 11-е издание. – М.: Вильямс, 2023.
8. Java Tutorials – The Java™ Tutorials.  
<https://docs.oracle.com/javase/tutorial/>
9. JavaFX Tutorial – Learn JavaFX with Examples.  
<https://code.makery.ch/library/javafx-tutorial/>
10. Vogella – Java Networking Tutorial.  
<https://www.vogella.com/tutorials/JavaNetworking/>
11. Baeldung – Java Networking Guides. <https://www.baeldung.com/java-networking>

- 12.Baeldung – JavaFX and UI tutorials. <https://www.baeldung.com/javafx>
- 13.Stack Overflow – ответы и обсуждения по Java и JavaFX.  
<https://stackoverflow.com>
- 14.GitHub – примеры клиент-серверных приложений на Java.  
<https://github.com>
- 15.GeeksForGeeks – Java Networking Basics.  
<https://www.geeksforgeeks.org/java-networking/>
- 16.Java Code Examples – Socket Programming in Java.  
<https://www.javatpoint.com/socket-programming>
- 17.Open Source Projects – JavaFX Game Demos and Examples.  
<https://github.com/topics/javafx-game>