МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение   
высшего образования

«КРЫМСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ им. В. И. ВЕРНАДСКОГО»

ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ

Кафедра компьютерной инженерии и моделирования

**Разработка многопользовательской игры "Тетрис"**

Курсовая работа

по дисциплине «Программирование»

студента 1 курса группы ПИ-б-о-201

Юращика Николая Александровича

направления подготовки 09.03.04 «Программная инженерия»

|  |  |  |
| --- | --- | --- |
| Научный руководитель  старший преподаватель кафедры компьютерной инженерии и моделирования | \_\_\_\_\_\_\_\_\_\_\_\_  (оценка)  \_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | Чабанов В.В. |

Симферополь, 2021

**РЕФЕРАТ**

Разработка многопользовательской игры "Тетрис". – Симферополь: ФТИ КФУ им. В. И. Вернадского, 2021. – 41 с., 17 ил., 7 ист.

Объектом исследования данной работы является игра «Тетрис».

Цель работы – создание рабочего игрового проекта с реализацией клиент-серверной системы. Сервер реализуется с использованием языка Python, а клиент с использованием языка C++.

Актуальность работы ­ обусловлена полным отсутствием аналогов на рынке, так как под «многопользовательским тетрисом» обычно понимают просто игру на выбывание, где у каждого игрока своё игровое поле и игра продолжается до тех пор, пока один из игроков не достигнет верхней границы – проиграет. Данный же проект реализует игру двух игроков на одном поле с последовательным управлением падающих фигур.

Было рассмотрено многочисленное количество вариантов реализации клиент-серверных систем, в ходе которых было принято решение использовать C++ для написания клиента и Python – для сервера, а для их взаимодействия использовать сокеты, что позволило обеспечить обмен данными в режиме достаточно приближенному к реальному времени.

ПРОГРАММИРОВАНИЕ, C++, SFML, PYTHON, СОКЕТЫ, КЛИЕНТ-СЕРВЕРНОЕ ПРИЛОЖЕНИЕ

**О**ГЛАВЛЕНИЕ

[СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ 4](#_Toc73560682)

[ВВЕДЕНИЕ 5](#_Toc73560683)

[ГЛАВА 1 ПОСТАНОВКА ЗАДАЧИ 6](#_Toc73560684)

[1.1 Цель проекта 6](#_Toc73560685)

[1.2 Существующие аналоги 6](#_Toc73560686)

[1.3 Основные отличия от аналогов 6](#_Toc73560687)

[1.4 Техническое задание 6](#_Toc73560688)

[ГЛАВА 2 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ПРИЛОЖЕНИЯ 10](#_Toc73560689)

[2.1 Анализ инструментальных средств 10](#_Toc73560690)

[2.2 Описание алгоритмов 11](#_Toc73560691)

[2.2.1 Алгоритм работы окон 11](#_Toc73560692)

[2.2.2 Алгоритм проверки ходов фигур 15](#_Toc73560693)

[2.3 Описание структур данных 16](#_Toc73560694)

[2.3.1 Клиент 16](#_Toc73560695)

[2.3.2 Сервер 18](#_Toc73560696)

[2.4 Описание основных модулей 19](#_Toc73560697)

[ГЛАВА 3 ТЕСТИРОВАНИЕ ПРОГРАММЫ 22](#_Toc73560698)

[3.1 Тестирование исходного кода 22](#_Toc73560699)

[3.2 Тестирование интерфейса пользователя и юзабилити 22](#_Toc73560700)

[ГЛАВА 4 ПЕРСПЕКТИВЫ ДАЛЬНЕЙШЕГО РАЗВИТИЯ ПРОЕКТА 25](#_Toc73560701)

[4.1 Перспективы технического развития 25](#_Toc73560702)

[4.2 Перспективы монетизации 25](#_Toc73560703)

[ЗАКЛЮЧЕНИЕ 26](#_Toc73560704)

[ЛИТЕРАТУРА 27](#_Toc73560705)

[ПРИЛОЖЕНИЕ 1 КОД ОСНОВНЫХ МОДУЛЕЙ ПРОЕКТА 28](#_Toc73560706)

# **СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ**

|  |  |
| --- | --- |
| ПО | Программное обеспечение |
| IDE | Интегрированная среда разработки |
| TCP | Transmission Control Protocol |
| IP | Internet Protocol |
| SFML | Simple and Fast Multimedia Library |

ВВЕДЕНИЕ

Игра «Тетрис» ориентирована на широкий возрастной диапазон пользователей. Согласно многим исследованиям, «Тетрис» является комплексной задачей для мозга. Он затрагивает множество когнитивных процессов: внимание, координацию между руками и зрением, память и пространственное восприятие.

Данный проект выводит старую игру на новый уровень, ведь теперь, после каждого Вашего хода, следующей фигурой – так называемой тетрамино – будет управлять другой человек. При таком подходе старые тактики уже не принесут должного эффекта и придётся искать новые подходы. Теперь продолжительность игры зависит не только от Ваших решений, но и от решений соперника, хотя в данном случае правильнее сказать напарника. Ведь в конце нет ни победителя, ни проигравшего, каждый игрок получает столько баллов, сколько линий он смог собрать.

Для реализации данного проекта были выбраны такие языки программирования как C++ и Python.

При планировании данного проекта была поставлена следующая цель – реализовать клиент-серверный вариант игры «Тетрис».

Для достижения цели в данной работе были поставлены следующие задачи:

1. изучить «старые» правила игры и обновить их для использования в многопользовательском варианте;
2. изучить библиотеку SFML для языка программирования C++;
3. изучить возможности языка SQL, установку СУБД MySQL, а также библиотеку для работы с MySQL на Python - pymysql;
4. изучить работу с сокетами;
5. реализация программного кода клиента и сервера;
6. отладка и тестирование.

Объектом исследования является игра «Тетрис».

Предметом исследования является реализация клиент-серверного многопользовательского варианта игры «Тетрис».

Методы исследования:

* моделирование – построение и изучение моделей с целью приобретения новых знаний, улучшение характеристик объектов исследований или управление ими;
* сравнение – сравнение предметов между собой на выявление сходства и различия, преимуществ и недостатков.

ГЛАВА 1  
ПОСТАНОВКА ЗАДАЧИ

**1.1 Цель проекта**

По итогу разработки планируется создать многопользовательский вариант игры «Тетрис» для двух игроков. Игра будет отличаться правилами – теперь на одном поле будут поочерёдно ходить оба игрока, что сделает бессмысленным использование старых тактик и заставит пользователей адаптироваться и находить новые стратегии для успешной игры.

**1.2 Существующие аналоги**

Прямых аналогов данного проекта не существует, есть лишь варианты, где у каждого игрока своё поле, а игра продолжается пока один из них не достигнет верхнего предела, то есть проиграет. В пример можно привести такие проекты, как:

* min2win Тетрис – браузерная онлайн версия классического тетриса с возможностью игры вдвоём на разных полях. Нет ведения статистики после покидания страницы с игрой;
* **Playmap Тетрис на двоих – похожая на предыдущий проект реализация, однако правила игры изменены – при собирании линии на одном поле, на втором образуются лишние фигуры снизу, которые поднимают основные наверх, что значительно усложняет игру. Также нет ведения статистики игроков после покидания страницы и неудобное управление, которое при входе необходимо изменять, что создаёт неудобства;**
* HoopGame Tetris – классический тетрис без особых нововведений, браузерный проект.

**1.3 Основные отличия от аналогов**

* Наличие сервера для ведения статистики игроков;
* Возможность получения монет за игру для осуществления покупок во встроенном магазине (например, текстур);
* Многопользовательский вариант исполнения для игры на одном поле.

## 1.4 Техническое задание

Программа должна соответствовать следующему техническому заданию:

1. Главное меню программы должно содержать несколько кнопок для:
   1. Перехода в подменю с игровыми комнатами, с возможностью подключения к уже существующей, а также создания собственной комнаты;
   2. Перехода в подменю «Магазин», с возможностью покупки новых текстур и их последующей установки;
   3. Перехода в подменю «Статистика» для просмотра топ 10 ведущих по очкам игроков;
   4. Перехода в подменю «Помощь».
2. Программа должна корректно интерпретировать и соблюдать следующие основные правила игры.

Для начала игры необходимо два игрока: один создаёт комнату, а второй подключается к ней. Право первого хода определяется случайным образом и не зависит от игроков и их действий.

После начала игры появляется игровое поле с размерами 20 на 10 клеток, а также первая фигура – тетрамино.

Тетрамино может быть 7 видов, описанных в программе (таблица 1.1).

Таблица 1.1. Возможные виды тетрамино

|  |  |
| --- | --- |
| **O** |  |
| **I** |  |
| **L** |  |
| **S** |  |
| **Z** |  |
| **J** |  |
| **T** |  |

Тетрамино опускается вниз на одну клетку каждые 0.5 секунды. Управление осуществляется с помощью стрелок вверх, вправо и влево тем игроком, который в данный момент обладает правом хода. Клавиши «влево» и «вправо» соответственно меняют положение тетрамино по горизонтали. Клавиша «вверх» осуществляет вращение фигуры относительно ее центра. При этом тетрамино не может выходить за пределы игрового поля. Когда оно касается боковых граней последующее смещение в сторону края не происходит. Если фигура касается нижней границы игрового поля или любой другой фигуры, которая уже упала, то она считается упавшей и управление ей более невозможно. Право хода передаётся другому игроку, а также создаётся новая фигура сверху, которая аналогично начинает сдвигаться вниз на клетку по таймеру.

Целью каждого игрока является собирание как можно большего количества линий – когда тетрамино полностью заполняют горизонтальную линию, то есть все 10 клеток. При этом сама линия удаляется, а все фигуры, которые были выше нее, опускаются вниз на одну клетку. Игрок, у которого на данный момент было право хода, получает 1 балл за каждую собранную линию.

Игра заканчивается на моменте, когда какая-либо тетрамино после падения пересекает верхнюю границу поля. При этом каждому игроку начисляется то количество очков и монет, сколько баллов они накопили за данную игру. После окончания игры игроков переносит в главное меню.

ГЛАВА 2  
ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ПРИЛОЖЕНИЯ

## 2.1 Анализ инструментальных средств

Одной из самых масштабных задач в подобных проектах является реализация графической составляющей. В данном решении необходимо описать: главное меню, меню комнат, меню магазина, меню статистики, меню помощи, игровое поле и непосредственно тетрамино. Для этого было решено использовать библиотеку SFML – Simple and Fast Multimedia Library – это библиотека для представления мультимедийных данных. SFML обеспечивает простой интерфейс для разработки игр и прочих мультимедийных приложений. Состоит она из пяти модулей:

* system – управление временем и потоками, он является обязательным, так как все модули зависят от него;
* window – управление окнами и взаимодействием с пользователем;
* graphics – делает простым отображение графических примитивов и изображений;
* audio – предоставляет интерфейс для управления звуком;
* network – для сетевых приложений.

Из аналогов были также рассмотрены такие библиотеки как SDL и Allegro. Allegro – это кроссплатформенная библиотека, в основном предназначенная для видеоигр и мультимедийных программ. Она выполняет общие, низкоуровневые задачи, такие как создание окон, прием пользовательского ввода, загрузка данных, рисование изображений, воспроизведение звуков. SDL – это свободная кроссплатформенная мультимедийная библиотека, реализующая единый программный интерфейс к графической подсистеме, звуковым устройствам и средствам ввода для широкого спектра платформ.

Несмотря на то, что выбор всё равно остался за SFML, у данного решения есть некоторые недостатки, например, невозможность создания поля для ввода, которое придётся создавать другими средствами.

Также был выбран ряд других инструментов, таких как:

* Среда Microsoft Visual Studio 2019 – была выбрана в пользу знакомого интерфейса, возможности создания проекта с использованием языка C++ и подключения библиотеки SFML;
* Текстовый редактор Sublime Text 3 для редактирования кода на языке Python;
* Среда IDLE для отладки кода на языке Python – была выбрана так как идет «в комплекте» со стандартной установкой Python в ОС Windows;
* C++ и Python – как изучаемые языки программирования;
* Модуль WinSock2 – сетевое взаимодействие со стороны клиента;
* Библиотека socket – сетевое взаимодействие со стороны сервера;
* Библиотека PyMySQL для работы с СУБД MySQL в Python.

## 2.2 Описание алгоритмов

## 2.2.1 Алгоритм работы окон

В данном проекте присутствуют понятия окон и перехода между ними. В оконной логике приложения есть 4 основных блока:

1. Блок управления – основной цикл, в котором последовательно строятся основные окна приложения в зависимости от готовности перейти на следующий блок;
2. Блок авторизации/регистрации – первый оконный блок приложения, в котором можно зарегистрироваться или войти в существующий аккаунт. Если вход возможен, то осуществляется переход на следующий блок;
3. Блок главное меню – второй блок, содержащий в себе кнопки:
   1. «Играть» – отправляет запрос на сервер для формирования открытых комнат и осуществляет переход на следующий блок – подменю комнат;
   2. «Магазин» – отправляет запрос на сервер для формирования доступных текстур и осуществляет переход на блок – магазин;
   3. «Статистика» – отправляет запрос на сервер для формирования десяти лучших игроков и отображает их;
   4. «Помощь» – отображает краткую информацию о проекте.
4. Блок меню комнат – подменю для создания собственной комнаты и подключения к уже существующим, содержит в себе кнопки:
   1. «Создать игру» - отправляет запрос о создании новой комнаты и циклично проверяет наличие второго игрока, после чего осуществляет переход к блоку игрового поля;
   2. Таблица комнат – таблица, содержащая в себе номер комнаты и имя игрока, который ее создал, при нажатии на любую запоминается ее номер;
   3. «Подключиться» - отправляет запрос о подключении к выбранной комнате на сервер, после чего осуществляет переход к блоку игрового поля.
5. Блок магазина – подменю для покупки и установки текстур для тетрамино, содержит в себе таблицу текстур, а также кнопки:
   1. «Купить» – отправляет запрос на сервер, где определяется возможность покупки выбранной текстуры, после получения положительного ответа делает эту текстуру возможной для установки;
   2. «Установить» – проверяет куплена ли текстура, в зависимости от проверки, если положительная – делает данную текстуру активной, если отрицательная – ничего не изменяет.
6. Блок игрового поля – отображение игрового поля, падающих фигурок тетрамино. Право хода определяется сервером в случайном порядке. Игрок, который ходит, осуществляет управление фигурой и после каждого ее падения на одну клетку отправляет данные на сервер для синхронизации второго клиента. После окончания хода отправляет информацию об этом на сервер, чтобы ожидающий игрок синхронизировался и начал свой ход.

Данные блоки взаимодействует с сервером. их взаимодействие показано на рисунке 2.1 в виде UML диаграммы.

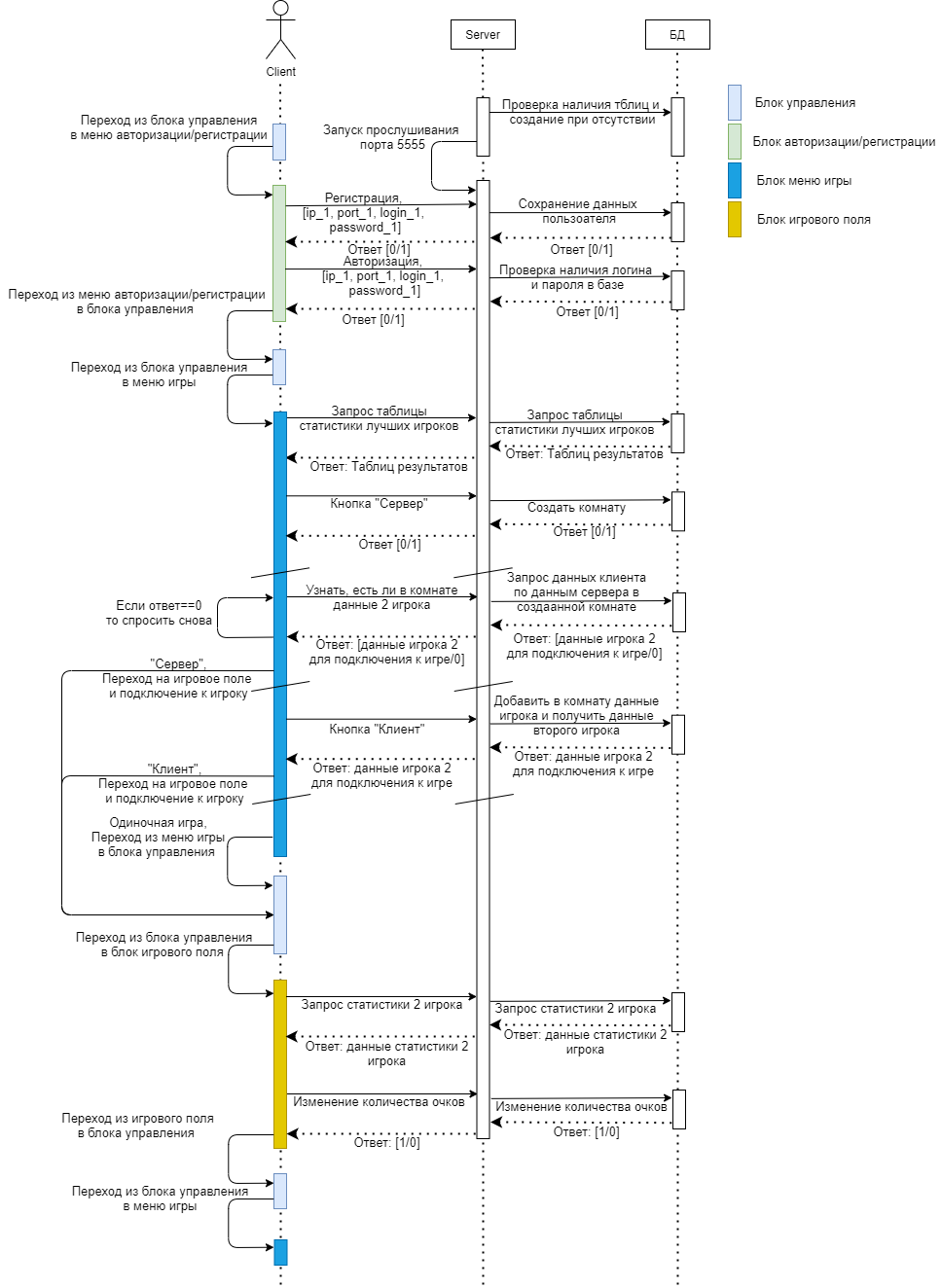


Рисунок 2.1. Схема взаимодействия Клиент – Сервер.

При запуске программы появляется окно входа и регистрации пользователя. Пользователь вводит данные логин и пароль, клиент проверяет все ли поля заполнены корректно, после этого отправляет запрос на сервер для проверки существования учетной записи данного пользователя (рисунок 2.2).



Рисунок 2.2. Проверка авторизации или регистрация пользователя.

После этого сервер проверяет данные в базе данных и возвращает результат (рисунок 2.3). Если такая учетная запись есть в базе данных пользователь попадет в игровое меню, а если ее не существует, тогда пользователю придаться зарегистрироваться.



Рисунок 2.3. Получение данных об авторизации или регистрации на сервере и их обработка.

## 2.2.2 Алгоритм проверки логики игры

После создания комнаты происходит запись в базу данных в таблицу rooms об имени пользователя и номере комнаты. Далее, после подключения второго игрока к комнате, запись обновляется – дописывается имя второго игрока, а также создается запись во временном хранилище сервера. Запись в базе данных нужна для синхронизации временного хранилища на обоих потоках. Во временном хранилище находится информация о праве хода, а также текущее положение тетрамино. После выполнения всех вышеперечисленных действий на обоих потоках запускается функция game (рисунок 2.4), отвечающая за приём и передачу информации о положении тетрамино, праве хода, а также синхронизацию между двумя потоками с помощью временного хранилища.

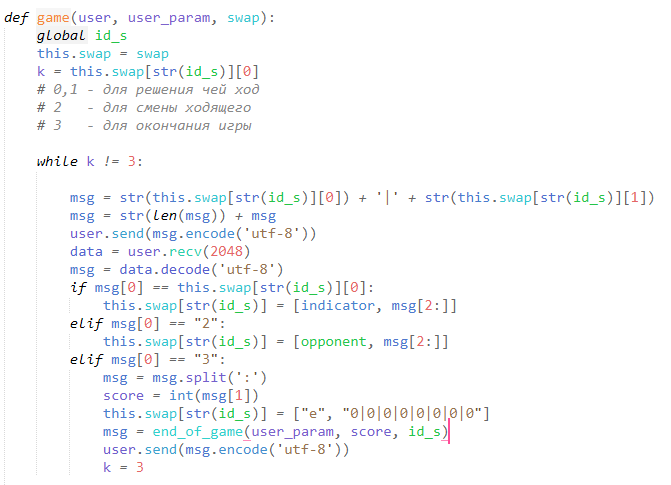


Рисунок 2.4. Функция game.

## 2.3 Описание структур данных

## 2.3.1 Клиент

Главными структурами данных выступают:

1. Двумерный массив fields, хранящий в себе игровое поле с фигурами в виде чисел (рисунок 2.5).
2. Двумерный массив видов тетрамино (рисунок 2.5);
3. Структура Point, описывающая текущий вид и местоположение падающей фигуры на поле (рисунок 2.6).

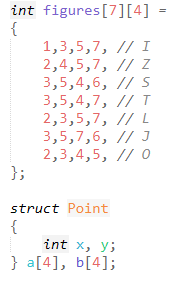


Рисунок 2.5. Двумерный массив видов тетрамино.

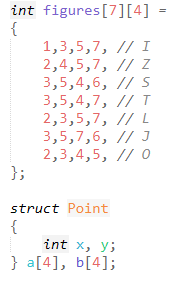


Рисунок 2.6. Структура Point.

К структуре Point обращается большинство функций логики игры. При каждом падении тетрамино на клетку вниз, а также изменении горизонтального положения, происходит изменение координат в структуре. Далее эти данные передаются на сервер для синхронизации второго клиента.

Запись координат в двумерный массив fields происходит только после падения фигуры. В этом случает выполняется проверка, касается ли фигура нижней границы поля или другой фигуры снизу, после чего происходит запись координат в массив поля, а также отправка данных на сервер, с указанием окончания хода.

Двумерный массив видов тетрамино используется для формирования новой фигуры в начале хода.

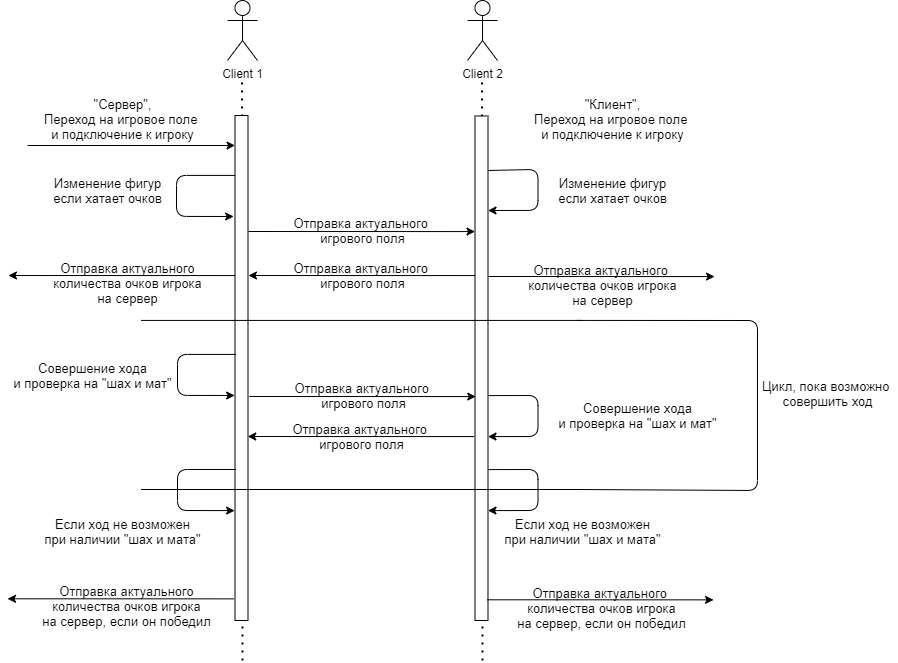


Рисунок 2.6. Схема взаимодействия Клиент – Клиент

## 2.3.2 Сервер

Для хранения информации об аккаунтах пользователей используется база данных под управлением MySQL. В базе данных присутствуют таблицы «users», «rooms» и «shop». Таблица «users» хранит в себе имя пользователя, пароль, статус, рейтинг, количество монет, а также приобретенные текстуры (рисунок 2.8). Таблица «rooms» хранит информацию о созданных комнатах, их номер, логин первого игрока, логин второго игрока и статус комнаты (рисунок 2.9). Таблица «shop» хранит информацию о доступных для приобретения текстур, а также их стоимости (рисунок 2.10).



Рисунок 2.8. Структура таблицы «users».



Рисунок 2.9. Структура таблицы «rooms».



Рисунок 2.10. Структура таблицы «shop».

## 2.4 Описание основных модулей