

Министерство науки и высшего образования РФ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
Кафедра вычислительной техники

УТВЕРЖДАЮ

Заведующий кафедрой

_____ О.В. Непомнящий
подпись

«___» _____ 2023 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 — Информатика и вычислительная техника

Игра «Монополия». Клиентская часть

Руководитель	_____	ст. преподаватель	В.С. Васильев
	подпись, дата		
Выпускник	_____		А.Ю. Лапунов
	подпись, дата		
Нормоконтролер	_____	ст. преподаватель	В.С. Васильев
	подпись, дата		

Красноярск 2023

Министерство науки и высшего образования РФ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра вычислительной техники

УТВЕРЖДАЮ

Заведующий кафедрой

_____ О.В. Непомнящий

« ____ » _____ 2023 г.

**ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
в форме бакалаврской работы**

Студенту Лапунову Алексею Юрьевичу
фамилия, имя, отчество

Группа КИ19-07Б Направление (специальность) 090301
номер код

Информатика и вычислительная техника
наименование

Тема выпускной квалификационной работы: Игра «Монополия».

Клиентская часть

Утверждена приказом по университету № _____ от _____

Руководитель ВКР: В.С. Васильев, старший преподаватель каф. ВТ ИКИТ
инициалы, фамилия, учёная степень, должность, место работы

СФУ

Исходные данные для ВКР:

1) Ларман, К. Применение UML 2.0 и шаблонов проектирования / К. Ларман.
– Москва: Вильямс, 2014. – 624 с.

2) Фаулер, М. UML. Основы, 3-е издание. – Перевод с англ. / М. Фаулер. –
Санкт-Петербург: Символ-Плюс, 2004. – 192 с., ил.

3) Рекомендации руководителя.

Перечень разделов ВКР:

1) Спецификация требований к системе

2) Проектирование

3) Реализация и тестирование

Перечень графического материала: демонстрационное видео, презентация.

Руководитель ВКР _____ В.С. Васильев
подпись инициалы, фамилия

Задание принял к исполнению _____ А.Ю. Лапунов
подпись инициалы, фамилия

« » _____ 20 г.
дата

РЕФЕРАТ

Выпускная квалификационная работа по теме «Игра «Монополия». Клиентская часть» содержит 45 страниц текстового документа, 26 иллюстраций, 3 таблицы, 2 приложения, 14 использованных источников.

Ключевые слова: КЛИЕНТСКОЕ ПРИЛОЖЕНИЕ, ПРОЦЕСС ПРОЕКТИРОВАНИЯ ICONIX, РАЗРАБОТКА ИГР.

Целью работы является программная реализация клиентской части игры «Монополия» для настольных компьютеров под управлением ОС Windows и Linux.

Выпускная квалификационная работа состоит из введения, основной части из трёх глав и заключения. Структура работы отражает решаемые задачи.

Во введении определяется цель работы, на её основе выделяются задачи.

В первой главе рассматриваются аналоги системы, формируется спецификация требований к разрабатываемому приложению на основе прецедентов.

Во второй главе определяется модель предметной области, описывается итеративный процесс проектирования с демонстрацией разработанных диаграмм. Описывается метод взаимодействия с серверным приложением.

В третьей главе описываются: применённые в ходе реализации приложения инструменты разработки, анализ структуры проекта, процесс модульного тестирования. Написана инструкция по сборке, также описан способ создания документации.

В заключении формулируются итоги по выполненной работе, и приводится список возможных улучшений приложения.

СОДЕРЖАНИЕ

Введение.....	3
1 Спецификация требований к системе	4
1.1 Существующие аналоги.....	4
1.2 Разработка прецедентов.....	5
1.3 Выводы по главе	9
2 Проектирование.....	10
2.1 Модель предметной области	10
2.2 Разработка диаграмм пригодности	11
2.3 Разработка диаграмм последовательности	15
2.4 Формат сообщений к серверу и от сервера.....	21
2.4.1 HTTP-запросы	21
2.4.2 Запросы во время игровой сессии	24
2.5 Диаграмма классов	24
2.6 Выводы по главе	25
3 Реализация и тестирование	26
3.1 Инструменты разработки.....	26
3.2 Тестирование приложения.....	27
3.3 Инструкция по сборке.....	28
3.4 Документация.....	28
3.5 Выводы по главе	29
Заключение	30
Список использованных источников	31
ПРИЛОЖЕНИЕ А Правила и особенности игры	33
ПРИЛОЖЕНИЕ Б Диаграмма потока экранов.....	45

ВВЕДЕНИЕ

Одной из широко известных настольных игр является Монополия. **Целью** работы является программная реализация клиентской части. Структура работы отражает решаемые **задачи**.

В рамках **первой главы** работы рассмотрены сильные и слабые стороны существующих реализаций игры, в том числе решений с открытым исходным кодом. Отсутствие на рынке готовых клиент-серверных решений, обладающих такими возможностями как сохранение статистики, ведение рейтинговой системы, сохранение пользовательских настроек игры обуславливает **актуальность** разработки. С их учетом разработана спецификация требований к системе, в работе приводится часть спецификации, связанная с клиентской частью. Спецификация требований выполнена в формате прецедентов, которые также выполняют роль документации пользователя.

Вторая глава посвящена проектированию клиентской части системы. Сложность работы заключается в осуществлении корректного отображения происходящих процессов игры в реальном времени (в соответствии с приёмом информационных сообщений от сервера), а также в создании интуитивно понятного и отзывчивого интерфейса для пользователя.

В **третьей главе** приводятся инструкции разработчика по сборке приложения и его тестированию.

1 Спецификация требований к системе

Клиентская часть представляет собой приложение для настольных компьютеров, работающих под управлением операционных систем Windows и Linux.

1.1 Существующие аналоги

На GitHub доступны 8 тысяч проектов [1], связанных с запросом «monopoly», а по запросу «monopoly client server» выдаётся лишь 15 проектов [2], большая часть из которых являются незаконченными проектами, но при этом имеют описание реализуемых возможностей клиент-серверного приложения. Параметры наиболее популярных из них приведены в таблице 1.1.

Таблица 1.1 — Проекты с открытым кодом по запросу «monopoly client server»

Название	monopoly [3]	CA314_monopoly [4]	Tyranny [5]	MonopolyProject [6]
Код клиентского приложения	Java, JavaScript	Python	C++	Java
Код серверного приложения	Java	Java	C++	Java
Строк кода	11935	9877	17852	2342
Наличие сервера	Да	Да	Да	Да
Возможность создания лобби	Нет	Нет	Да	Нет
Рейтинговая игра	Нет	Нет	Нет	Нет
Дата последнего коммита	28.11.2014	13.12.2019	31.08.2011	6.01.2019

В связи с отсутствием в открытом доступе клиент-серверных проектов в стиле настольной Монополии, совмещающих в себе систему лобби и рейтинговых матчей, было принято решение реализовать в разрабатываемой в рамках данной ВКР системе указанные функции. В приложении А представлены основные правила и особенности разрабатываемого приложения.

1.2 Разработка прецедентов

На рисунке 1.1 представлена диаграмма вариантов использования, отражающая действия пользователя.

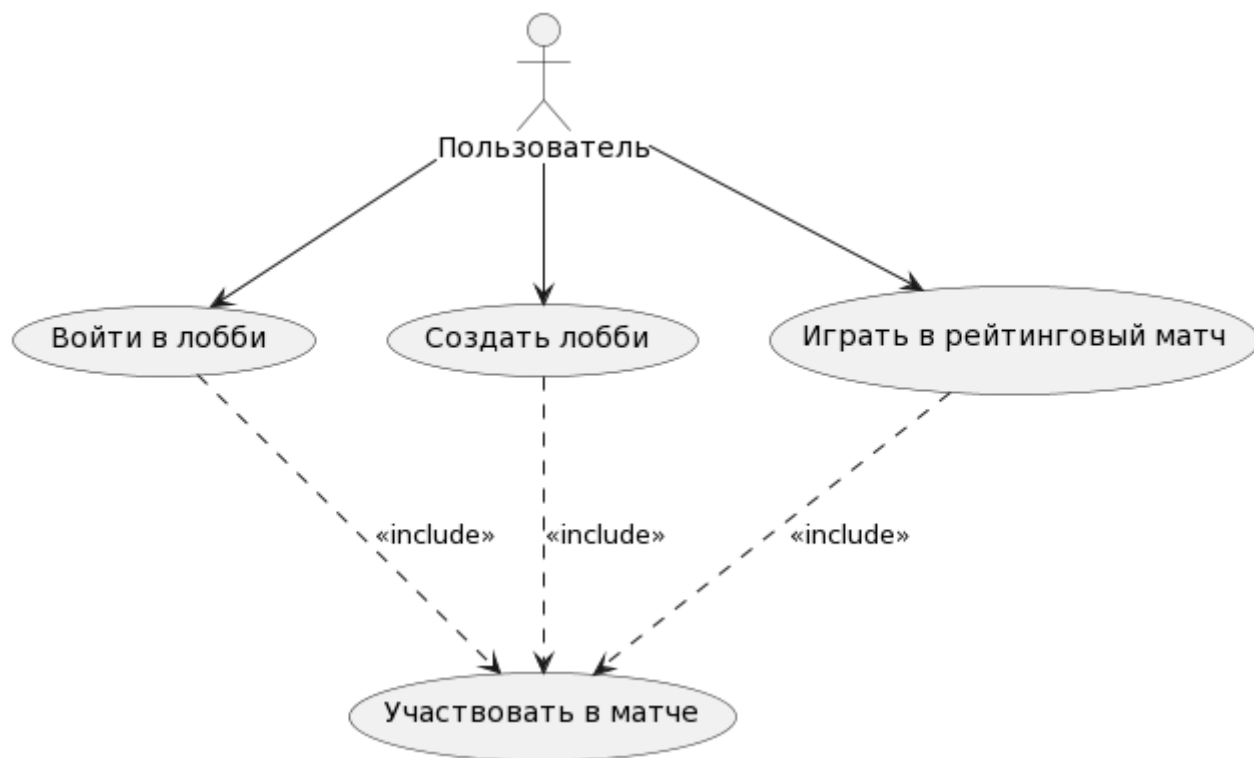


Рисунок 1.1 — Диаграмма вариантов использования

Для реализации вариантов использования пользователь взаимодействует с программой с помощью графического интерфейса, макет которого в форме диаграммы потока экранов [7] приведён в приложении Б.

Название прецедента: Войти в лобби.

Предусловие: Пользователь находится в окне «Главное меню».

Основная последовательность:

1. Пользователь выбрал лобби в таблице доступных комнат и дважды нажал по нему (либо нажал кнопку «Подключиться»).

2. Пользователь успешно подключился к выбранной комнате и был выполнен переход к окну «Лобби».

3. Пользователь переключает свой флаг готовности в состояние «Готов».

4. Пользователь ожидает подключения оставшихся участников для начала игры.

5. Все пользователи подтвердили готовность.

6. Создатель лобби нажал на кнопку «Запустить игру».

Постусловие: Ввод прецедента «Участвовать в матче».

Условия для ввода в действие альтернативного сценария:

Условие 1: Выбранное лобби является закрытым (защищённым паролем).

Альтернативная последовательность при условии 1:

1. Для пользователя отображается окно «Подключение к закрытому лобби».

2. Пользователь вводит правильный пароль к закрытой комнате.

Постусловие при условии 1: Продолжение основной последовательности прецедента «Войти в лобби» с пункта 3.

Условие 1.1: Пользователь ввёл неправильный пароль.

Альтернативная последовательность при условии 1.1:

1. Для пользователя отображается окно «Подключение к закрытому лобби».

2. Пользователь вводит неправильный пароль к закрытой комнате.

Постусловие при условии 1.1: Для пользователя отображается информационное окно, сообщающее о том, что он ввёл неправильный пароль для подключения к закрытой комнате.

Условие 2: Лобби заполнилось на момент подключения

Альтернативная последовательность при условии 2:

1. Пользователь запросил подключение к выбранному лобби.

2. Комната оказалась заполненной на момент подключения.

Постусловие при условии 2: Для пользователя отображается информационное окно, сообщающее о том, что он не сможет подключиться к выбранному лобби, так как оно заполнено.

Название прецедента: Создать лобби.

Предусловие: Пользователь находится в окне «Главное меню».

Основная последовательность:

1. Пользователь нажал на кнопку «Создать лобби».
2. Пользователь переключается на окно «Лобби».
3. Пользователь выбирает настройки комнаты.
4. Пользователь выбирает настройки матча.
5. Пользователь включает отображение для других пользователей своей комнаты в списке доступных комнат для присоединения.
6. Пользователь ожидает подключения оставшихся участников для начала игры.
7. Все участники подтвердили готовность.
8. Пользователь нажал на кнопку «Запустить игру».

Постусловие: Ввод прецедента «Участвовать в матче».

Условие для ввода в действие альтернативного сценария:

Условие 1: Не все пользователи подняли флаг готовности.

Альтернативная последовательность при условии 1:

1. Пользователь нажал на кнопку «Запустить игру».
2. Для пользователя-создателя лобби отображается диалоговое окно, сообщающее о том, что не все игроки готовы, и предлагающее ему две опции: всё равно начать игру, либо пока не начинать матч.
3. Пользователь начинает игру, несмотря на то, что не все игроки готовы.

Постусловие при условии 1: Ввод прецедента «Участвовать в матче».

Условие для ввода в действие альтернативного сценария:

Условие 1.1: Пользователь решает пока не начинать матч, так как не все пользователи готовы.

Альтернативная последовательность при условии 1.1:

1. Пользователь не начинает матч, нажав на соответствующую кнопку в диалоговом окне.
2. Пользователь ожидает поднятия флага готовности всеми участниками.

3. Все пользователи готовы.
4. Пользователь нажимает на кнопку «Запустить игру»

Постусловие при условии 1.1: Ввод прецедента «Участвовать в матче».

Название прецедента: Играть в рейтинговый матч.

Предусловие: Пользователь находится в окне «Главное меню» и не авторизовался как гость.

Основная последовательность:

1. Пользователь нажал на кнопку «Поиск рейтинговой игры».
2. Пользователь подключается к скрытому рейтинговому лобби.
3. Пользователь ожидает подключения оставшихся игроков.
4. Необходимое количество пользователей подключилось к комнате.
5. Запускается отсчёт времени до запуска рейтингового матча.

Постусловие: Ввод прецедента «Участвовать в матче».

Название прецедента: Участвовать в матче.

Предусловие: Пользователь находится в окне «Монополия».

Основная последовательность:

1. Пользователь получает свой цвет фишки.
2. Пользователь ожидает свою очередь хода.
3. При наступлении очереди пользователя его фишка перемещается по игровой карте согласно правилам игры.

4. Если поле, на которое переместилась фишка игрока, является активируемым, пользователь может нажать кнопку действия, чтобы воспользоваться эффектом особой клетки согласно правилам игры.

5. Последовательность 1-4 повторяется до тех пор, пока игра не завершится.

Постусловие: Игровой матч завершается с выводом статистики.

Условие ввода в действие альтернативного сценария:

Условие 1: Идёт обычный (не рейтинговый) матч и пользователь решил сдаться.

Альтернативная последовательность при условии 1:

1. Пользователь нажал на кнопку «Сдаться».
2. Для пользователя отображается диалоговое окно, требующее подтвердить данное действие.
3. В диалоговом окне пользователь подтверждает свой выбор.
4. Пользователь отключается от матча, его фишка исчезает с поля игры, занятые клетки освобождаются, а его имущество разделяется поровну между оставшимися участниками матча.

Постусловие при условии 1: Пользователь возвращается в окно «Главное меню».

1.3 Выводы по главе

1. Рассмотрены аналоги разрабатываемой системы.
2. Разработаны макеты пользовательского интерфейса приложения.
3. Разработана спецификация требований к клиентской части игры «Монополия» на основе прецедентов.

2 Проектирование

Проектирование приложения выполнено на основе итеративного процесса проектирования ICONIX [8]. В работе приведены результаты последней итерации:

- модель предметной области;
- диаграммы пригодности, используемые для идентификации потенциальных объектов или обязанностей объектов, необходимых для реализации прецедентов;
- диаграммы последовательности, визуализирующие отношения объектов в различных условиях;
- формат сообщений к серверу и от сервера;
- диаграмма классов.

2.1 Модель предметной области

Модель предметной области игры «Монополия» выполнена в нотации Мартина, приведена на рисунке 2.1. Игровое поле («FieldsGrid») состоит из клеток («Cell»). Фишки игроков («GamingPiece») размещаются на клетках игрового поля.

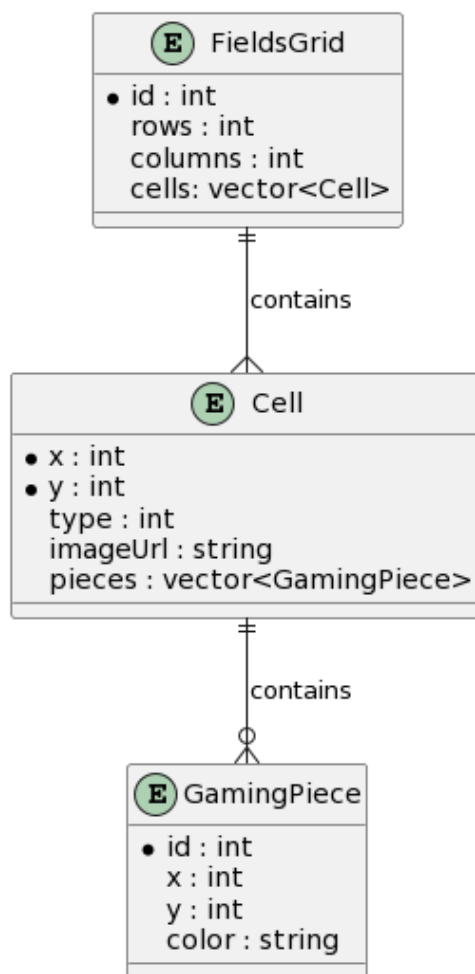


Рисунок 2.1 — Модель предметной области

2.2 Разработка диаграмм пригодности

В изображённой на рисунке 2.2 диаграмме пригодности прецедента «Войти в лобби» показано, что пользователь взаимодействует с двумя окнами: окно «Лобби» и окно «Главное меню». В «Главном меню» пользователь может активировать ячейку с лобби в таблице (выполнить двойное нажатие, либо нажать клавишу Enter, при условии, что строка в таблице выделена), чтобы присоединиться к комнате. Теперь, в окне «Лобби» он может переключить свой флаг готовности и ожидать, когда владелец комнаты начнёт игру. При необходимости пользователь может выйти из лобби, что вернёт его в окно «Главное меню».

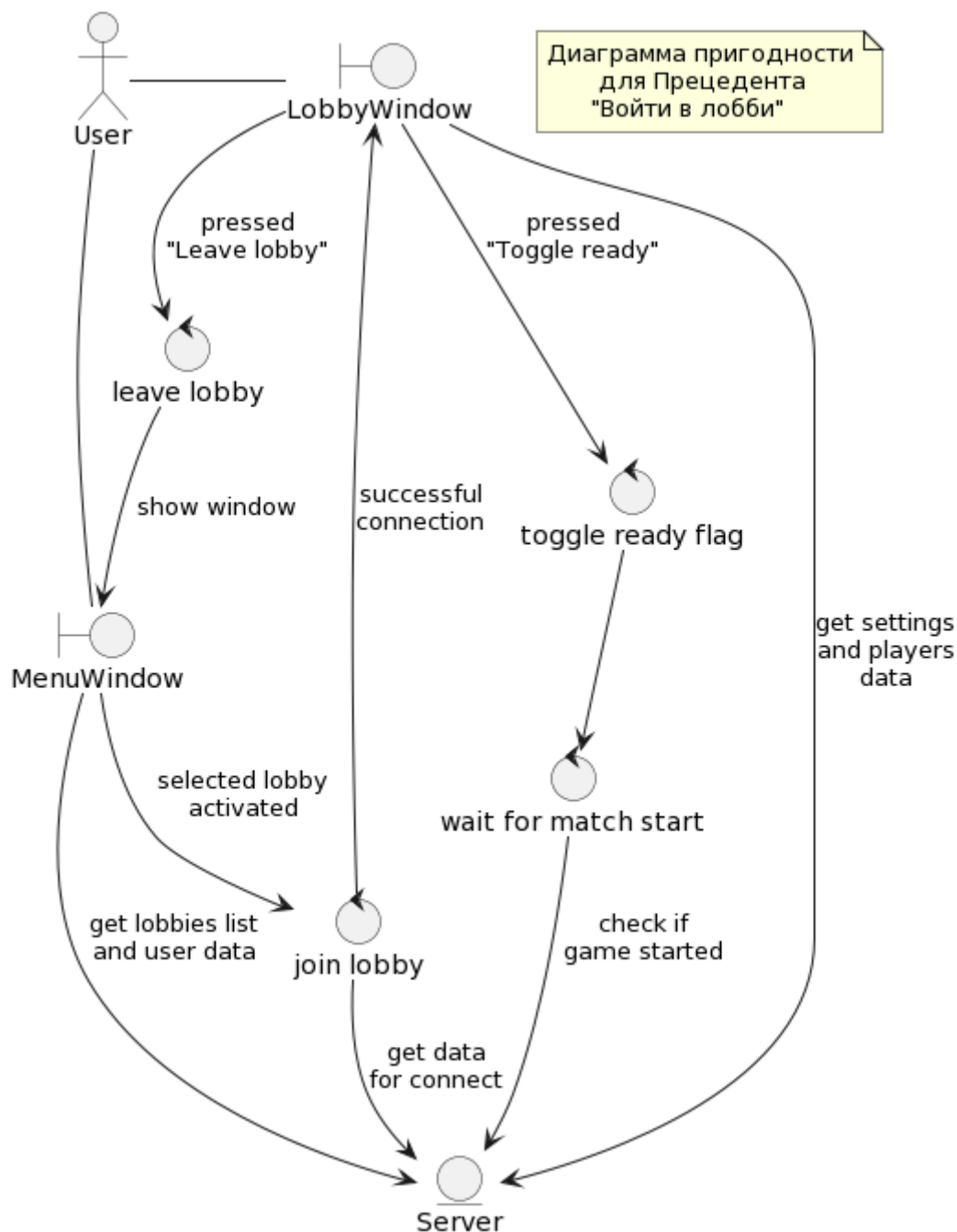


Рисунок 2.2 — Диаграмма пригодности для прецедента «Войти в лобби»

На рисунке 2.3 показана диаграмма пригодности для прецедента «Создать лобби». Пользователь также взаимодействует с двумя окнами: окно «Лобби» и окно «Главное меню». Нажатием на кнопку «Создать лобби» пользователь переключается на окно «Лобби», где может настроить как параметры комнаты, так и параметры матча. Когда минимально необходимое количество игроков собралось, владелец может запустить матч нажатием на кнопку «Начать игру». Удаление лобби происходит в случае, если владелец выйдет из комнаты.

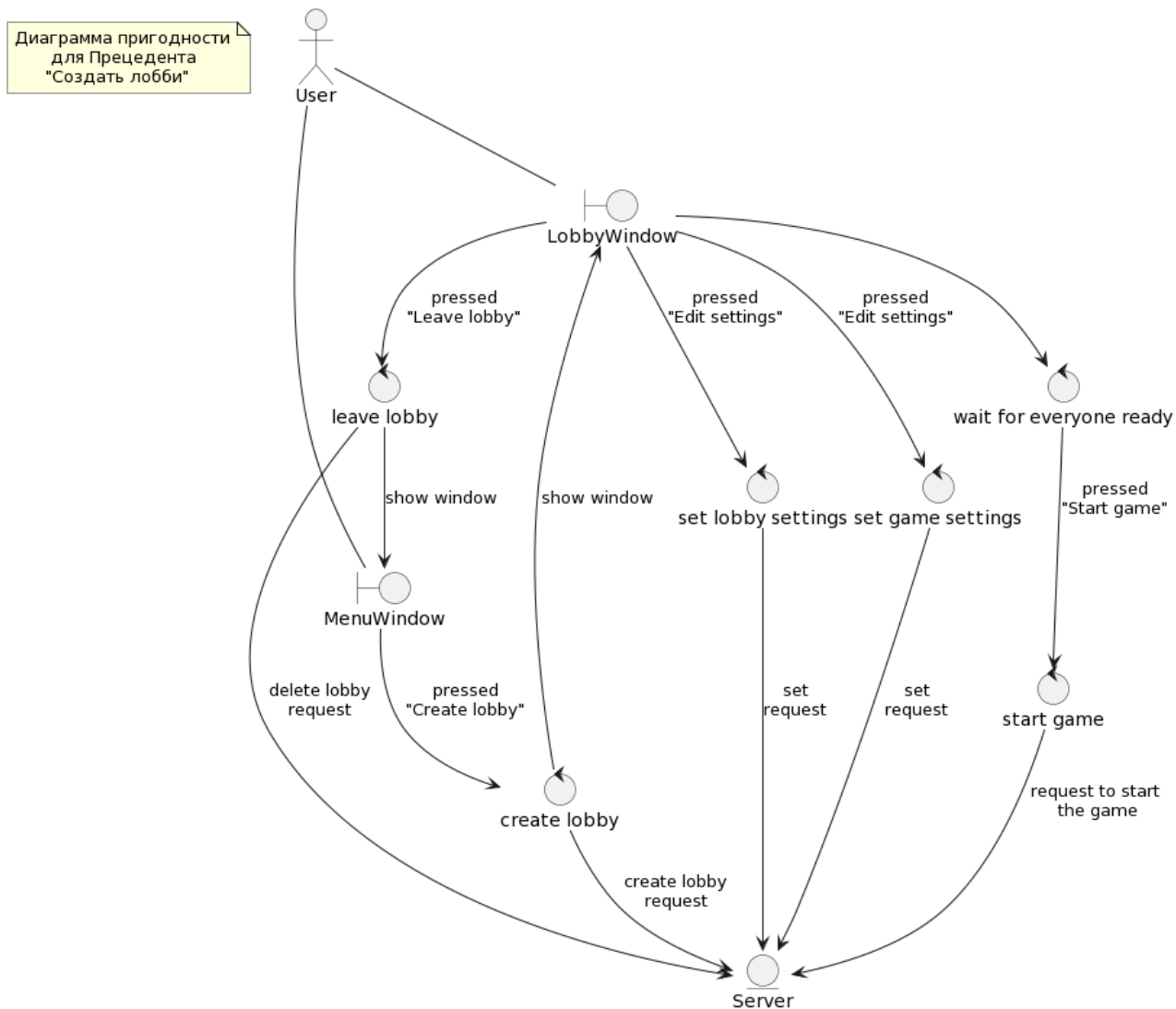


Рисунок 2.3 — Диаграмма пригодности для прецедента «Создать лобби»

На рисунке 2.4 показана диаграмма пригодности для прецедента «Играть в рейтинговый матч». Пользователь взаимодействует с окнами «Лобби» и «Главное меню». Посредством нажатия кнопки «Найти рейтинговый матч» пользователь вступает в очередь на присоединение в рейтинговое лобби. После подключения в рейтинговую комнату пользователь ожидает, пока наберётся достаточное количество игроков для автоматического старта игры.

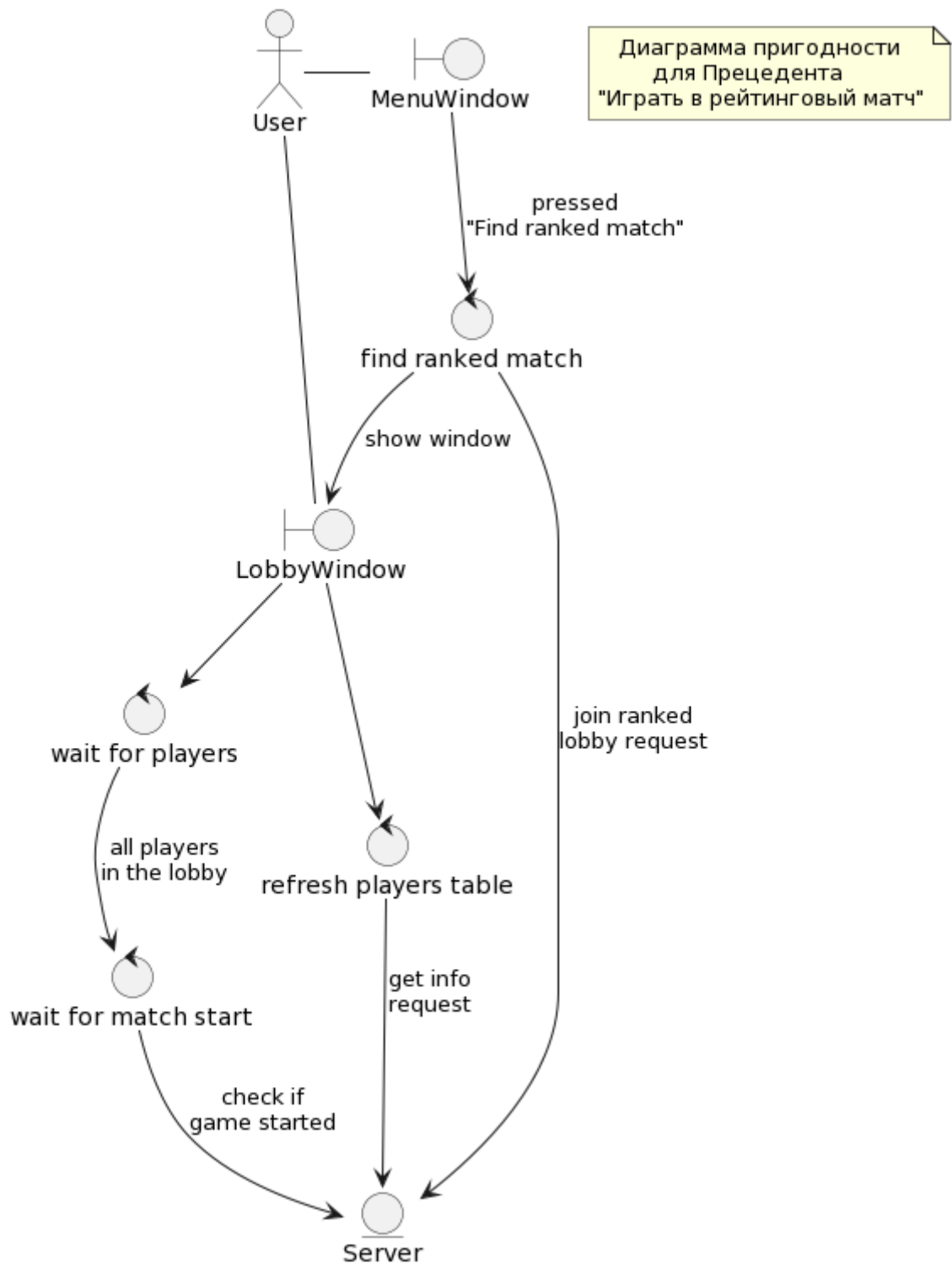


Рисунок 2.4 — Диаграмма пригодности для прецедента «Играть в рейтинговый матч»

На рисунке 2.5 изображена диаграмма пригодности для прецедента «Участвовать в матче». Пользователь взаимодействует с окном игры «Монополия». В процессе игры он может ожидать свой ход, активировать

специальное поле за отведённое на ход время, досрочно завершить свой ход при необходимости, а также сдаться (если матч не рейтинговый).

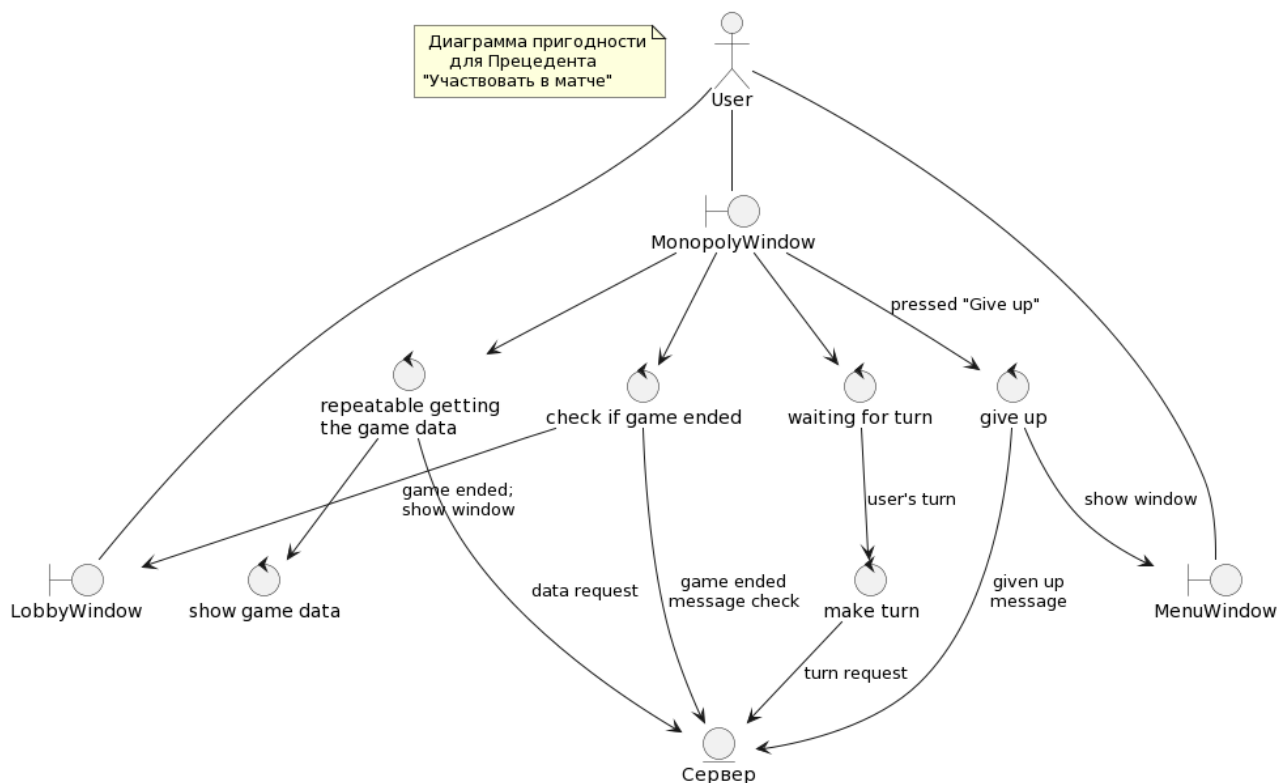


Рисунок 2.5 — Диаграмма пригодности для прецедента «Участвовать в матче»

2.3 Разработка диаграмм последовательности

На рисунке 2.6 показана диаграмма последовательности, основанная на диаграмме пригодности для прецедента «Войти в лобби».

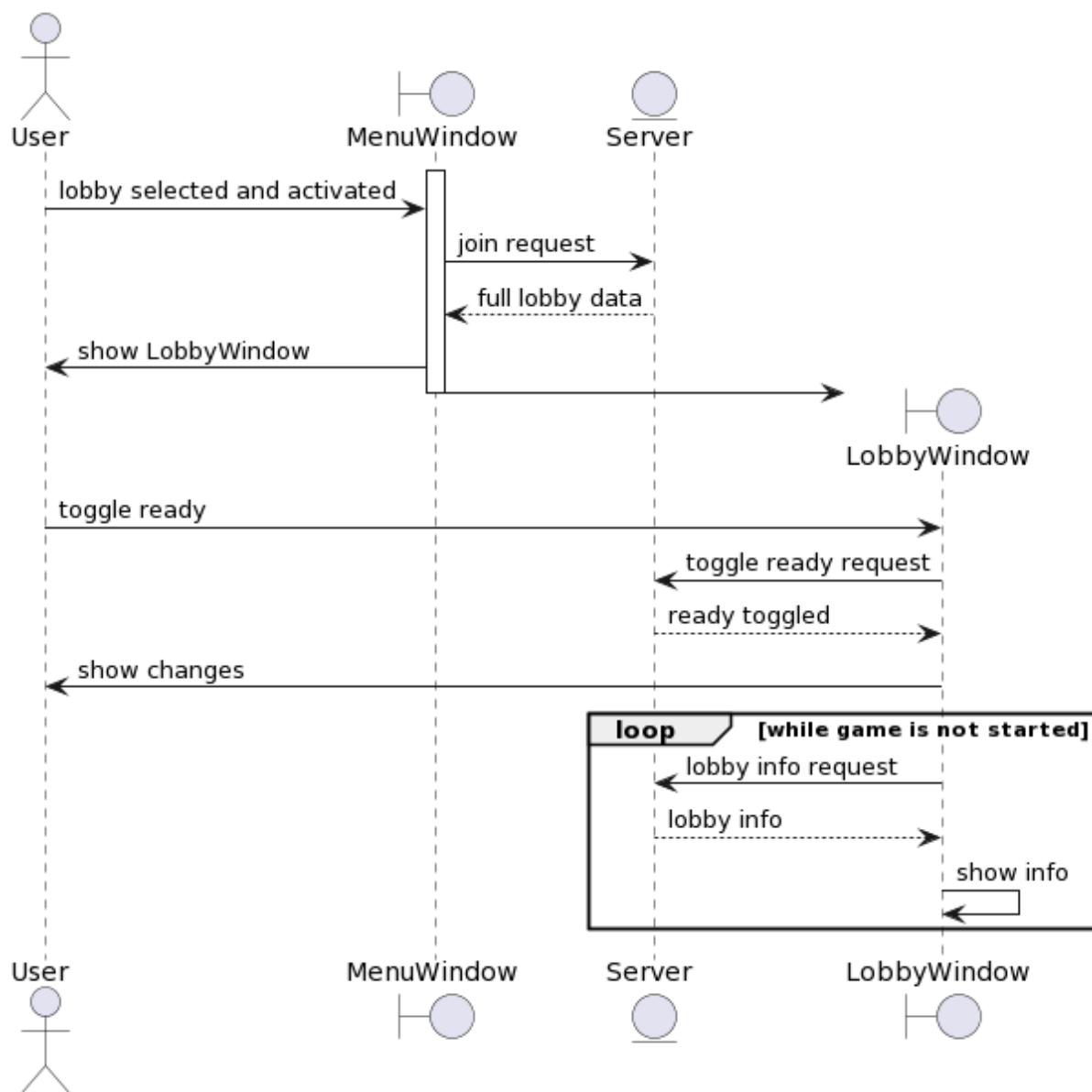


Рисунок 2.6 — Диаграмма последовательности «Войти в лобби»

Пользователь активирует строку лобби в таблице со списком всех доступных комнат, либо нажимает кнопку «Подключиться к лобби» при выделенной в таблице комнате. Активация подразумевает двойное нажатие левой клавиши мыши по строке в таблице, либо нажатие клавиши «Enter» на выделенной строке. Отправляется запрос к серверу для подключения. При успешном подключении пользователя окно главного меню «MenuWindow» получает основной контекст комнаты, который наполняет данными отображённое для пользователя окно лобби «LobbyWindow». При нажатии на кнопку «Переключить готовность» отправляется запрос к серверу для выполнения соответствующего действия. Сервер отправляет ответ на данный

запрос в виде кода. Пока игра ещё не начата, к серверу отправляется запрос о текущих данных комнаты с определённой периодичностью. Такие данные соответствуют информации о присутствующих игроках и о настройках.

На рисунке 2.7 изображена диаграмма последовательности «Войти в лобби».

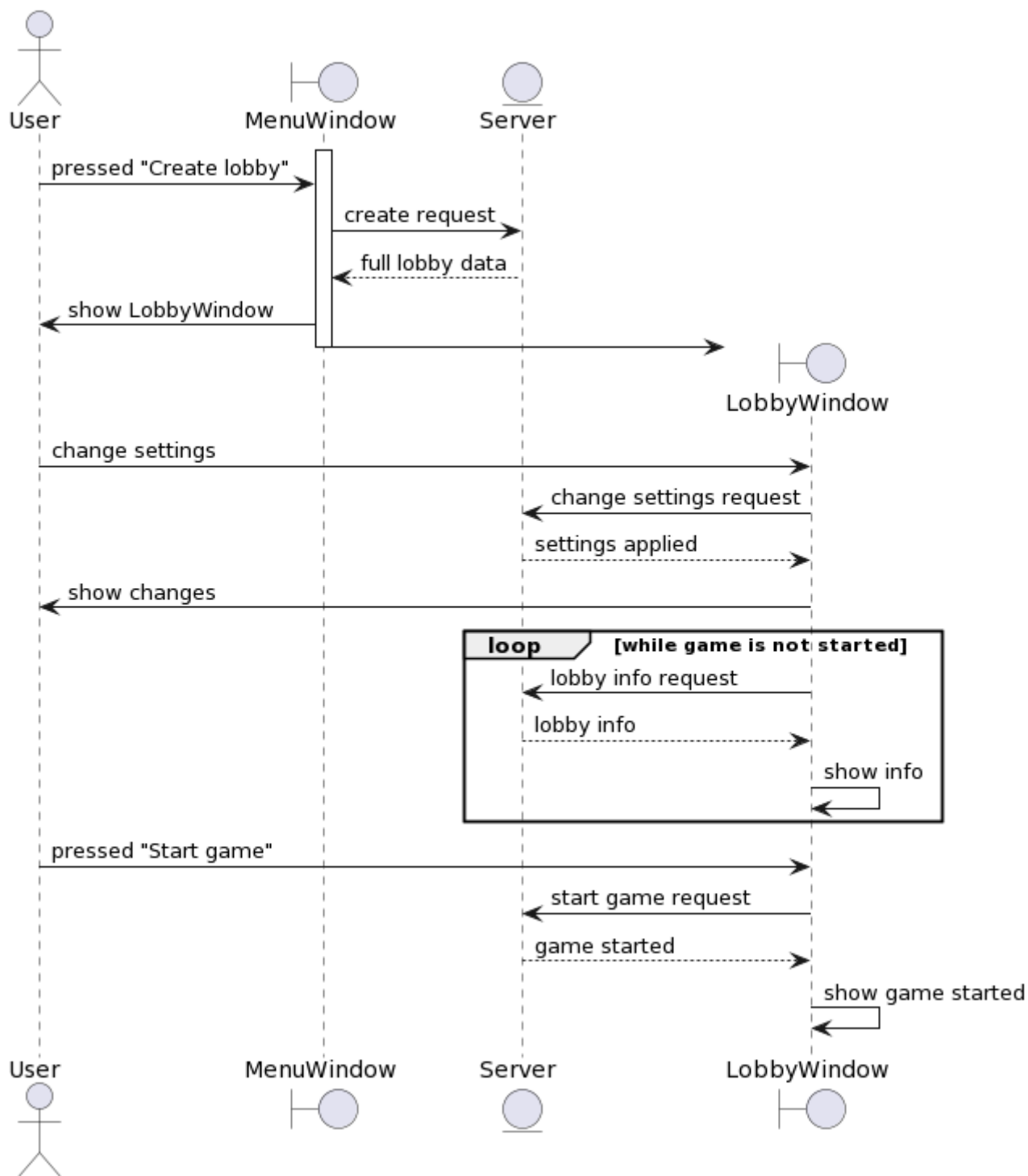


Рисунок 2.7 — Диаграмма последовательности «Создать лобби»

Пользователь нажимает кнопку «Создать лобби» в главном окне «MenuWindow». Отправляется запрос к серверу для создания комнаты. Сервер возвращает основной контекст лобби для наполнения им окна комнаты «LobbyWindow», которое отображается пользователю. Пользователь, являющийся владельцем комнаты, может изменять настройки комнаты посредством внесения изменений в элементы управления окна лобби «LobbyWindow» и нажатия кнопки «Применить изменения». Это действие отправляет запрос к серверу на изменение настроек. Пока игра не начата, окно лобби «LobbyWindow» обновляет информацию, запрашивая с определённой периодичностью данные у сервера. Пользователь, являющийся владельцем комнаты, нажимает на кнопку «Начать игру», что соответствует отправке запроса к серверу для старта игры.

На рисунке 2.8 изображена диаграмма последовательности «Найти рейтинговый матч».

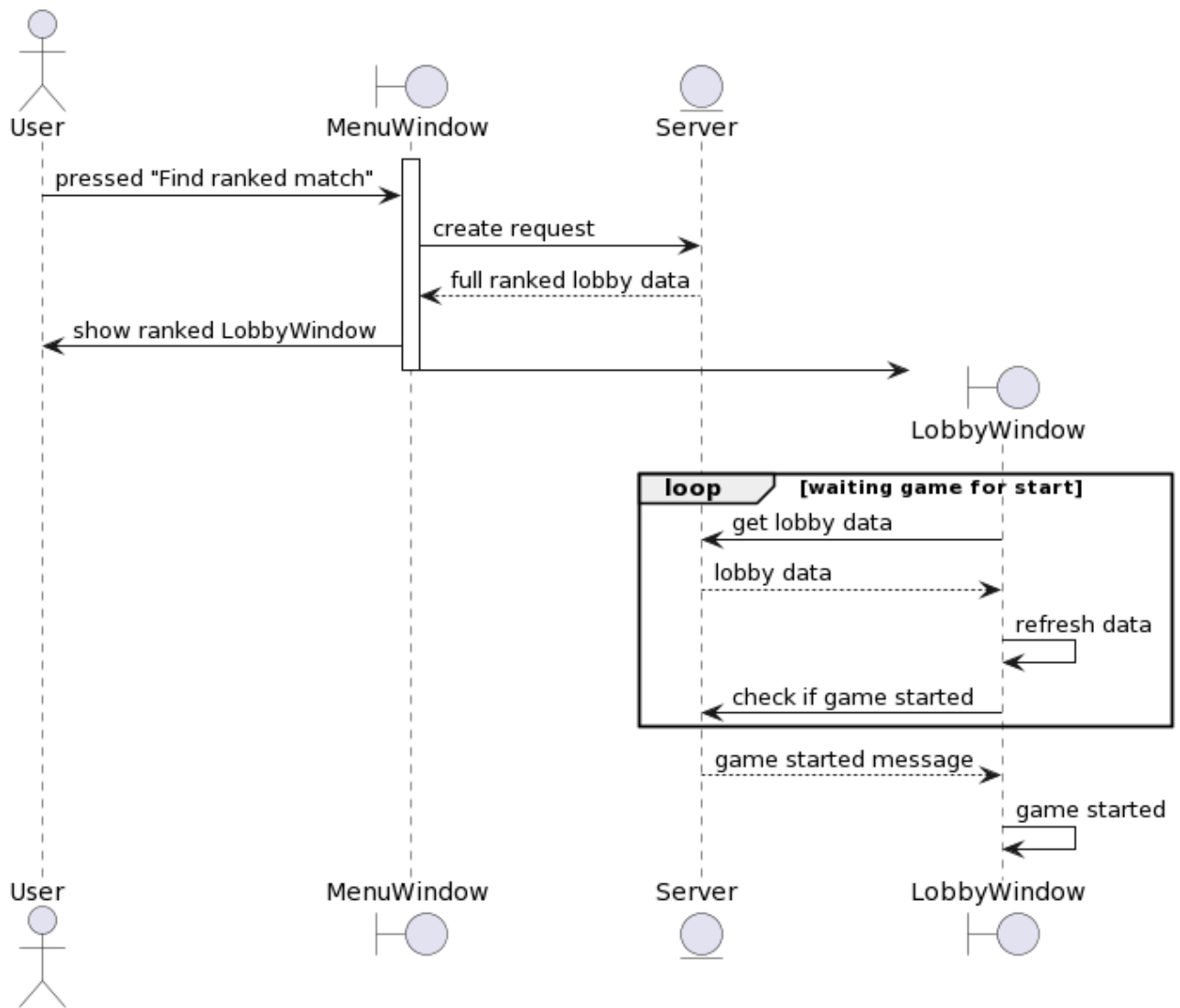


Рисунок 2.8 — Диаграмма последовательности «Найти рейтинговый матч»

Пользователь нажимает на кнопку «Найти рейтинговую игру» в окне главного меню «MenuWindow». К серверу отправляется запрос на поиск и подключение к рейтинговой комнате. От сервера приходит ответ с контекстом комнаты, который наполняет окно лобби «LobbyWindow», отображаемое для пользователя. Пока не начата игра, в этом окне пользователи ждут подключения необходимого количества игроков. Окно «LobbyWindow» обновляет информацию посредством отправки соответствующего запроса к серверу с определённой периодичностью. Пользователи узнают о начале игры, когда на запрос обновления информации комнаты приходят данные с установленными флагами для отображения старта игры.

На рисунке 2.9 изображена диаграмма последовательности «Участвовать в матче».

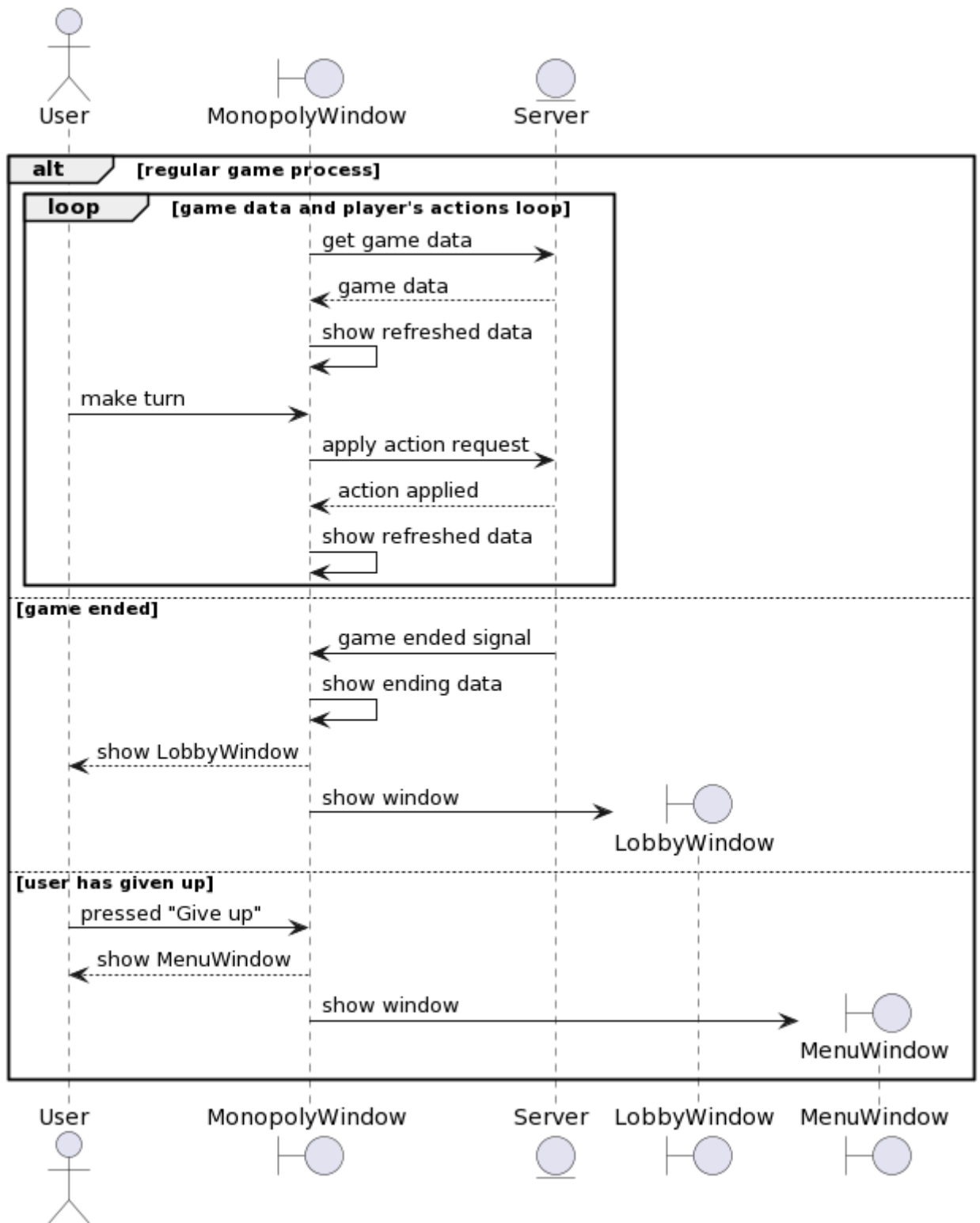


Рисунок 2.9 — Диаграмма последовательности «Участвовать в матче»

В течение игрового процесса окно игры «MonopolyWindow» получает данные игрового контекста от сервера. Игрок может сделать ход (либо пропустить его), что соответствует отправке запроса, после чего окно игры снова получает обновлённые данные для отображения. Если игра окончена, сервер отправляет сигнал клиентскому приложению, которое, в свою очередь, показывает игрокам статистику на конец игры и занятые места. После этого пользователю отображается окно лобби «LobbyWindow». Если игрок во время игровой сессии воспользовался кнопкой сдаться, его перенаправляет на окно главного меню «MenuWindow».

2.4 Формат сообщений к серверу и от сервера

2.4.1 HTTP-запросы

HTTP-запросы используются для получения от серверного приложения такой информации, как: список доступных лобби, основная информация об игроках в лобби (псевдоним, число баллов рейтинга, статус готовности) и информация о самом лобби, в котором пользователь находится — такие запросы необходимо отправлять неоднократно с определённой периодичностью, чтобы корректно отображать любые изменения в настройках комнаты, настройках игры и в списке игроков. Кроме запросов с целью получения информации, используются HTTP-запросы, с помощью которых пользователь может создать лобби, настроить параметры комнаты и игры, начать матч, изменить флаг готовности, подключиться к лобби и т.д. Описание запросов, характерное для этапа проектирования, от клиентского приложения к серверному сведено в таблицу 2.1 (примечание: «ид.» — сокращённо «идентификатор»).

С целью корректной идентификации пользователя и проверки подлинности его запросов используется JWT-токен, который предоставляет сервер при успешной авторизации пользователя.

Таблица 2.1 — Описание HTTP-запросов к серверу на этапе проектирования

Описание	Тип	Передаваемые параметры	Получаемые данные
Авторизация через сервис ВКонтакте	POST	Код, получаемый при успешной авторизации на стороне сервиса	Пара токенов: обновляющий токен и JWT-токен (доступа)
Авторизация через сервис Google	POST	Код, получаемый при успешной авторизации на стороне сервиса	Пара токенов: обновляющий токен и JWT-токен (доступа)
Гостевая авторизация	POST	Код, получаемый при успешной авторизации на стороне сервиса	Пара токенов: обновляющий токен и JWT-токен (доступа)
Обновление JWT-токена (доступа)	POST	Заголовок: обновляющий токен.	JWT-токен (доступа)
Изменение псевдонима	POST	Заголовок: JWT-токен (доступа). Тело запроса: новый псевдоним.	—
Получение информации о пользователе	GET	Заголовок: JWT-токен (доступа). Параметр: уникальный ид. пользователя	Данные о пользователе: уникальный ид., псевдоним, баллы рейтинга, маркер гостя
Получение списка комнат	GET	Заголовок: JWT-токен (доступа).	Список доступных для подключения комнат.
Проверка на участие в игре	GET	Заголовок: JWT-токен (доступа).	Контекст комнаты. Информация о подключенных игроках. Параметры игровой сессии.
Создание комнаты	POST	Заголовок: JWT-токен (доступа). Тело запроса: основной контекст комнаты.	Контекст комнаты с изменёнными данными. Информация о подключенных игроках.
Подключение к комнате	POST	Заголовок: JWT-токен (доступа). Параметр: уникальный ид. комнаты. Тело запроса: пароль (опционально).	Контекст комнаты. Информация о подключенных игроках.

Окончание таблицы 2.1

Описание	Тип	Передаваемые параметры	Получаемые данные
Подключение к комнате рейтинговой игры	POST	Заголовок: JWT-токен (доступа).	Контекст комнаты. Информация о подключенных игроках.
Обновление информации о комнате	GET	Заголовок: JWT-токен (доступа)	Контекст комнаты. Информация о подключенных игроках. Если игра была начата, присылаются параметры игровой сессии (адрес и порт).
Применение новых настроек комнаты	POST	Заголовок: JWT-токен (доступа). Тело запроса: изменённый контекст комнаты.	—
Переключение готовности	POST	Заголовок: JWT-токен (доступа).	—
Передача другому пользователю права на владение и управление лобби	POST	Заголовок: JWT-токен (доступа). Параметр: уникальный ид. повышаемого пользователя.	—
Исключение пользователя из лобби	POST	Заголовок: JWT-токен (доступа). Параметр: уникальный ид. повышаемого пользователя.	—
Старт игры	POST	Заголовок: JWT-токен (доступа).	—
Отключение от комнаты	POST	Заголовок: JWT-токен (доступа)	—
Удаление комнаты	DELETE	Заголовок: JWT-токен (доступа)	—

2.4.2 Запросы во время игровой сессии

Клиентское приложение получает последовательность сообщений, из которых можно выделить следующие данные:

- идентификаторы карт;
- идентификаторы игровых полей;
- информация об игровых полях;
- информация об игроках;
- информация об изменениях на карте;
- управляющие сигналы.

2.5 Диаграмма классов

На рисунке 2.10 изображена диаграмма классов уровня проектирования. Выделены пять основных классов: класс окна авторизации «LoginWindow», класс окна главного меню «MenuWindow», класс окна лобби «LobbyWindow», класс окна игры «GameManagerWindow» и класс, осуществляющий взаимодействие с сервером, «ServerCommunicator».

Классы окон управляют внутренними процессами при взаимодействии пользователя со соответствующими управляющими элементами, а также обращаются к объекту класса «ServerCommunicator» с целью формирования запроса к серверу и получения информации.

Окно игры «GameManagerWindow» регулирует процессы игровой сессии и, подобно другим окнам, обращается к объекту класса «ServerCommunicator», чтобы получать и отправлять сообщения, взаимодействуя с сервером.

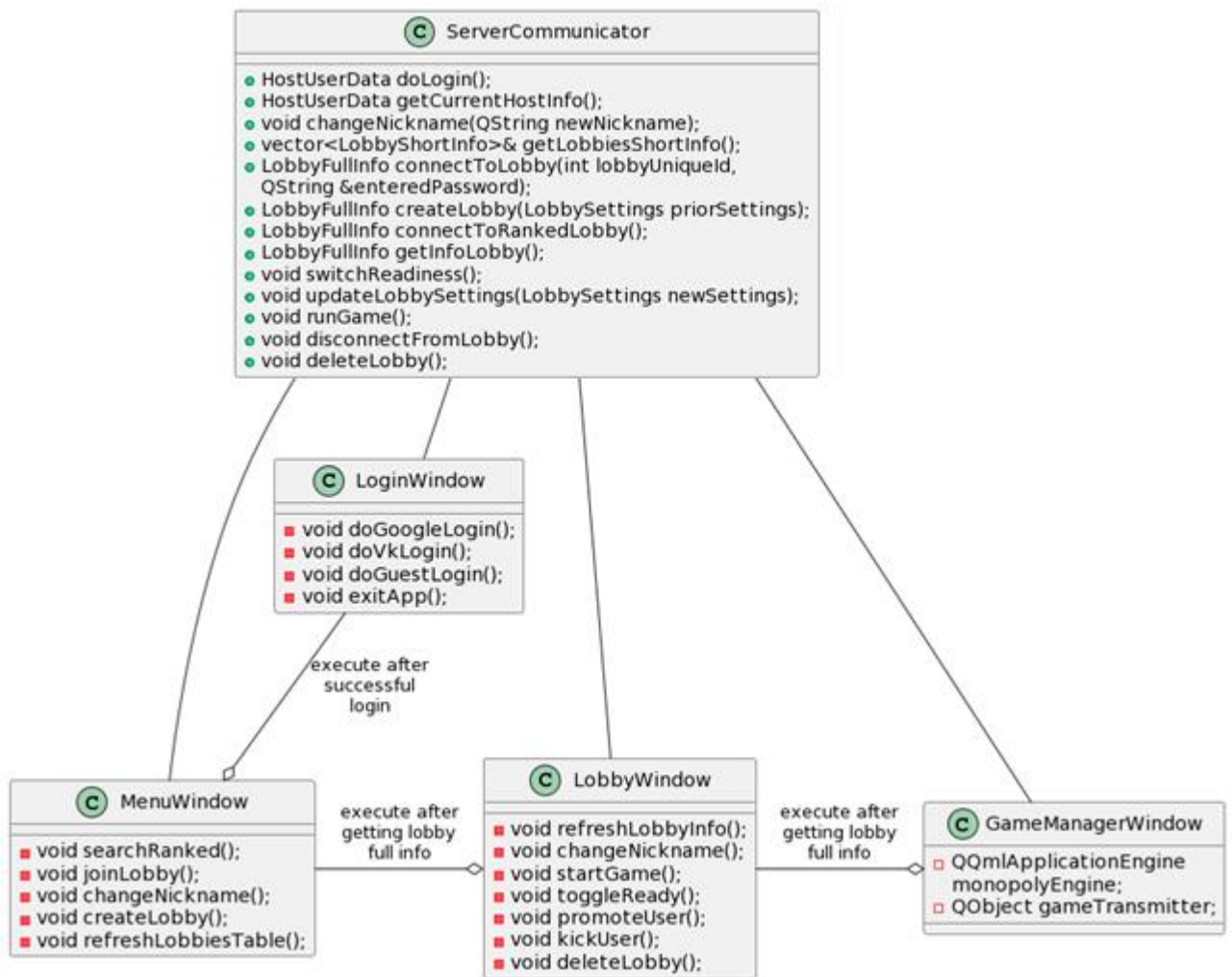


Рисунок 2.10 — Диаграмма классов уровня проектирования

2.6 Выводы по главе

1. Выполнено проектирование, включающее в себя: определение макета предметной области, создание диаграмм пригодности для всех прецедентов, создание диаграмм последовательности, создание диаграммы классов.

2. Совместно и согласованно с разработчиком серверной части были определены необходимые HTTP-запросы для последующей реализации взаимодействия клиентского и серверного приложений.

3 Реализация и тестирование

3.1 Инструменты разработки

В качестве инструментов разработки были использованы библиотеки фреймворка Qt [9]. Они позволяют работать с графическим интерфейсом, выполнять обработку регулярных выражений, а также осуществлять сетевое взаимодействие. В течение игровой сессии клиентское и серверное приложения обмениваются большим количеством запросов. Для того, чтобы оптимизировать данные запросы, было решено использовать протокол сериализации «Protocol Buffers» [10].

Интерфейс пользователя в игре представлен окном, содержимое которого определяется модулями декларативного языка программирования QML, использующего TypeScript. Для написания QML-файлов все необходимые типы предоставляются модулем Qt Quick [11].

Все используемые в игре рисунки имеют формат масштабируемой векторной графики SVG — это позволяет иллюстрациям не терять качество при любом размере отображения. Созданные изображения показаны на рисунке 3.1.



Рисунок 3.1 — Изображения формата SVG

3.2 Тестирование приложения

Тестирование выполнялось с использованием макросов и функций библиотеки QTest [12]. В таблице 3.1 приведены результаты тестирования.

Таблица 3.1 — Результаты тестирования

Модуль	Метод	Тестовый случай	Результат
FileManager	Создание директории с необходимыми файлами.	Проверка на существование созданных файлов.	Создана директория со всеми необходимыми файлами.
	Сохранение принятых настроек комнаты и игры.	Чтение настроек из файла. Сравнение полученных настроек.	Прочитанный объект соответствует исходному.
	Сохранение предпочтения пользователя о виде игровых костей.	Чтение и сравнение значений.	Прочитанное значение соответствует исходному.
	Обработка JSON значений, содержащих токены.	Запись токенов в файл. Чтение и сравнение значений.	JSON значения (токены) корректно обработаны и записаны в файл. Прочитанные значения соответствуют полученным в начале.
SourceStructs	Определение типа комнаты для сервера	Пароль отсутствует.	PUBLIC (публичный).
		Пароль есть.	PRIVATE (приватный).
	Определение вида окончания игры для сервера	Очки ограничены, ходы не ограничены.	SCORE (по очкам).
		Очки не ограничены, ходы ограничены.	TURN (по ходам).
		Очки и ходы ограничены.	BOTH (и по очкам, и по ходам).

3.3 Инструкция по сборке

Исходный код приложения представлен в GitHub репозитории [13].

При реализации приложения использовалась утилита qmake, и все файлы и библиотеки подключаются с помощью редактирования файла проекта «Client-Monopoly.pro». Собрать и скомпилировать проект можно с использованием среды QtCreator. В качестве компилятора можно использовать MinGW (Desktop Qt 6.4.2 MinGW).

При успешной сборке утилита qmake создаст все необходимые файлы Makefile, из которых затем можно будет автоматически собрать проект посредством утилиты make.

Порядок ввода команд для сборки:

```
$ cd ~/monopoly_project
```

```
$ qmake Client-Monopoly.pro
```

```
$ make
```

3.4 Документация

С помощью системы генерации документации Doxygen [14] была автоматически создана документация в формате HTML на основе специальных комментариев в исходном коде.

Директория с названием «doxygen_doc» размещена вместе с проектом клиентского приложения игры «Монополия» в репозитории GitHub и содержит все необходимые файлы для просмотра документации. Доступ к основной странице по html/index.html.

Порядок действий для генерации документации:

1. Ввод команды для создания конфигурационного файла:

```
$ doxygen -g "doxy_monopoly.cfg"
```

2. Настройка созданного конфигурационного файла, в котором как минимум указываются: расположение директории с исходными файлами

проекта и директория, в которую будет помещена сгенерированная документация.

3. Ввод команды для генерации документации:

```
$ doxygen "doxy_monopoly.cfg"
```

4. Для обновления текущей версии документации необходимо вводить следующую команду:

```
$ doxygen -u "doxy_monopoly.cfg"
```

3.5 Выводы по главе

1. Описаны инструменты разработки, с помощью которых выполнялась реализация приложения.

2. Выполнено тестирование приложения, состоящее из модульных тестов.

3. Приведена инструкция по сборке приложения.

4. Описана документация, созданная автоматически посредством использования функционала кроссплатформенной системы документирования Doxygen на установленных метках-комментариях в исходном коде.

ЗАКЛЮЧЕНИЕ

В результате проделанной работы были выполнены следующие задачи:

- проанализированы аналоги разрабатываемого приложения;
- сформулирована спецификация требований на основе рассмотренных аналогов;
- выполнен и описан процесс проектирования;
- создано клиентское приложение игры «Монополия», функционирующее согласно спецификации требований.

На фоне рассмотренных аналогов приложение имеет ряд отличительных преимуществ: реализация удобного интерфейса, системы лобби и рейтинговых матчей. Сам процесс игры уникален: особые поля, изменяющаяся по стадиям игровая карта, возможность напрямую взаимодействовать с игроками.

Тестирование реализованного приложения выполнено с помощью автоматических модульных тестов.

Гибкая архитектура разработанного клиентского приложения позволит выполнить дальнейшее расширение игровых возможностей:

- добавление в игру событий;
- увеличение количества особых игровых полей;
- создание редактора пользовательских карт с возможностью импорта и экспорта;
- внедрение возможностей для настраивания внешнего вида игровых фишек за определённые достижения игроков.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Результаты запросов по ключевому слову «monopoly» / GitHub : сайт. – URL: <https://github.com/search?q=monopoly> (дата обращения: 07.01.2023).
2. Результаты запросов по ключевым словам «monopoly client server» / GitHub : сайт. – URL: <https://github.com/search?q=monopoly+client+server> (дата обращения: 07.01.2023).
3. Репозиторий «gmichaeljaison/monopoly» / GitHub : сайт. – URL: <https://github.com/gmichaeljaison/monopoly> (дата обращения: 07.01.2023).
4. Репозиторий «AlexandreChamard/CA314_monopoly» / GitHub : сайт. – URL: https://github.com/AlexandreChamard/CA314_monopoly (дата обращения: 07.01.2023).
5. Репозиторий «mbpolan/tyranny» / GitHub : сайт. – URL: <https://github.com/mbpolan/tyranny> (дата обращения: 07.01.2023).
6. Репозиторий «ddunig2/MonopolyProject» / GitHub : сайт. – URL: <https://github.com/ddunig2/MonopolyProject> (дата обращения: 07.01.2023).
7. Ларман, К. Применение UML и шаблонов проектирования / К. Ларман. – Москва: Вильямс, 2014. – 624 с.
8. Процесс разработки программного обеспечения ICONIX / Блог программиста — программирование и алгоритмы : сайт. – URL: <https://pro-prof.com/archives/4126> (дата обращения: 12.02.2023).
9. Фреймворк Qt версии 6.4 / Документация Qt : сайт. – URL: <https://doc.qt.io/qt-6/index.html> (дата обращения: 17.03.2023).
10. Протокол сериализации «Protocol Buffers» / Документация Protocol Buffers : сайт. – URL: <https://protobuf.dev> (дата обращения: 18.03.2023).
11. Модуль Qt Quick версии 6.4.2 / Документация Qt : сайт. – URL: <https://doc.qt.io/qt-6/qtquick-index.html> (дата обращения: 18.03.2023).
12. Пространство имён QTest / Документация Qt : сайт. – URL: <https://doc.qt.io/qt-6/qtest.html> (дата обращения: 21.03.2023).
13. Репозиторий «AlekseyLapunov/Client-Monopoly» / GitHub : сайт. –

URL: <https://github.com/AlekseyLapunov/Client-Monopoly> (дата обращения: 21.03.2023).

14. Система документирования Doxygen / Doxygen : сайт. – URL: <https://www.doxygen.nl/> (дата обращения: 21.03.2023).

ПРИЛОЖЕНИЕ А

Правила и особенности игры

Начало игры

Игра начинается с определения сервером порядка ходов участников случайным образом — это соответствует назначению игрокам цветов фишек, которые изображены на рисунке А.1 — они отображают расположение игроков на карте. В начале игры фишки всех игроков располагаются на первом поле. Сервер размещает особые поля карты посредством случайной генерации.

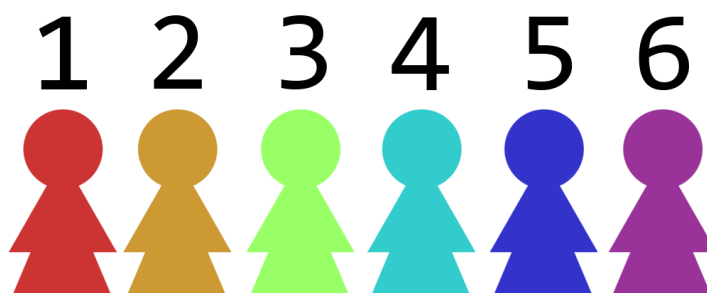


Рисунок А.1 — Цвета фишек игроков

Ход игрока

В начале хода игрока бросаются две игральные кости, затем их значения складываются, и фишка играющего перемещается на соответствующее поле, номер которого определяется суммой номера предыдущей клетки и суммой выпавших значений кубиков (данный процесс показан на рисунке А.2). Если поле, на котором находится фишка игрока, является активируемым, участнику даётся фиксированное время на ход — за это время игрок имеет право применить действие активной клетки, либо досрочно завершить свой ход.



Рисунок А.2 — Перемещение фишки игрока

Типы полей

В игре предусмотрены поля разного типа (все поля представлены на рисунке А.10):

– начальное поле (изображено на рисунке А.3): поле, с которого начинают игру все участники. При попадании фишки игрока на данное поле повторно в течение игры, ему полагается денежный бонус. Количество полей на карте: 1;



Рисунок А.3 — Вид начального поля

– обычное поле (изображено на рисунке А.4): не имеет влияния на игровой процесс;

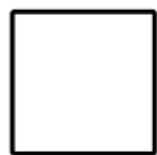


Рисунок А.4 — Вид обычного поля

– поле с активом (изображено на рисунке А.5): имеет активное действие, позволяющее игроку приобрести бизнес за определённую цену. Количество полей на карте: 8 с лесопилками, 4 с ТЭС, 3 с АЭС;



Лесопилка



ТЭС



АЭС

Рисунок А.5 — Вид полей с активами трёх типов

– поле с ресурсом (изображено на рисунке А.6): имеет активное действие, позволяющее игроку развернуть на данном поле свою добывающую станцию. Количество полей на карте: по 3 поля на каждый тип ресурса;



**Лесной
массив**



**Залежи
угля**



**Залежи
урана**

Рисунок А.6 — Вид полей с ресурсами трёх типов

– поле с отпуском (изображено на рисунке А.7): игрок, фишка которого попала на данное поле, вынужден пропустить свой ход в следующей очереди ходов, при этом ему полагается денежный бонус. Количество полей на карте: 2;



Рисунок А.7 — Вид поля с отпуском

– поле с диверсией (изображено на рисунке А.8): имеет активное действие, позволяющее игроку выбрать клетку с занятой другим участником добывающей станцией или бизнесом, которая будет освобождена от владения, при этом данную клетку можно будет приобрести бесплатно любому участнику. Кроме этого, цель диверсии (игрок, потерявший актив или станцию) лишается 5% своего текущего денежного запаса. После использования поле с диверсией перемещается на место случайного обычного поля. Количество полей на карте: 3.



Рисунок А.8 — Вид поля с диверсией

– поле со стрелкой (изображено на рисунке А.9): определяет, в какую сторону пойдёт фишка игрока, если путь лежит через неё. После прохождения через данное поле игрок за определённую плату может выбрать, в какую именно сторону будет повернута стрелка. В случае, если выбор не будет сделан, указатель повернётся по часовой стрелке.



Рисунок А.9 — Вид поля со стрелкой

Начальное поле



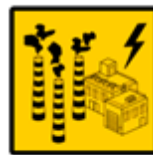
Обычное поле



Поле с активом



Лесопилка



ТЭС



АЭС

Поле с ресурсом



*Лесной
массив*



*Залежи
угля*



*Залежи
урана*

Поле отпуска



Поле с диверсией



Поле со стрелкой



Рисунок А.10 — Все виды полей

Активы

В игре можно купить активы трёх видов: лесопилка, тепловая электрическая станция (ТЭС) и атомная электрическая станция (АЭС); среди

перечисленных активов, лесопилки приносят наименьший доход, а АЭС — наибольший.

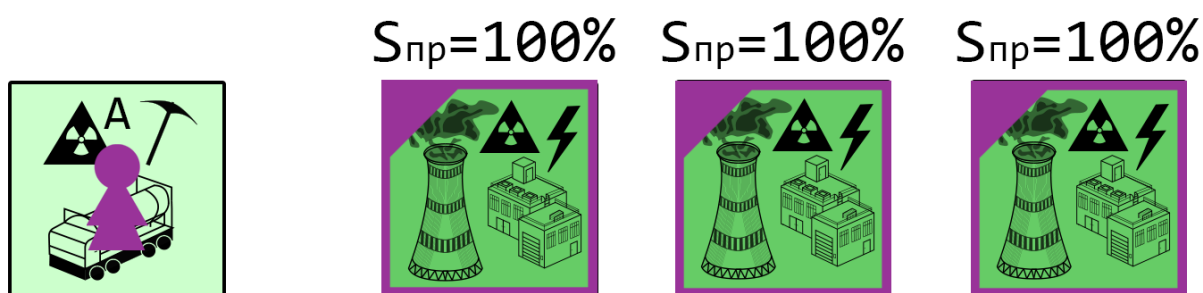
Каждый ход актив под владением игрока приносит ему прибыль. При попадании фишки игрока на уже купленный им актив, он получает дополнительную выплату в размере 0,5 от прибыли, которую производит данный актив. Если фишка игрока переместилась на актив, занятый другим игроком, он выплачивает последнему налог в размере 3% от своего текущего денежного запаса.

Ресурсы

Существует зависимость производств от следующих ресурсов: лесных массивов, залежей каменного угля и залежей урана соответственно списку активов. Зависимость выражается в следующем:

– пример №1: игрок А имеет в своём распоряжении три атомных электрических станции. Установив добывающую станцию урана, он повышает прибыль каждого из своих трёх активов на 25%, в сумме увеличивая свой доход от АЭС на 75%. Приведённый пример представлен на рисунке А.11;

До постройки добывающей станции:



После постройки добывающей станции:

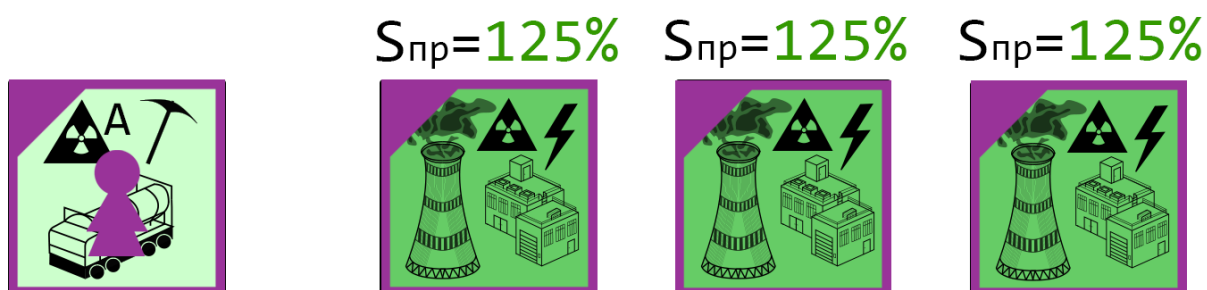


Рисунок А.11 — Зависимость активов от ресурсов (пример №1)

– пример №2: игроки А и Б в сумме имеют в своём распоряжении три тепловые электрические станции. Игрок В, установив добывающую станцию каменного угля, будет получать часть прибыли (по 25% за ТЭС) активов игроков А и Б. Это, в свою очередь, значит, что игроки А и Б будут получать от своих ТЭС на 25% меньше прибыли. Пример проиллюстрирован на рисунке А.12.

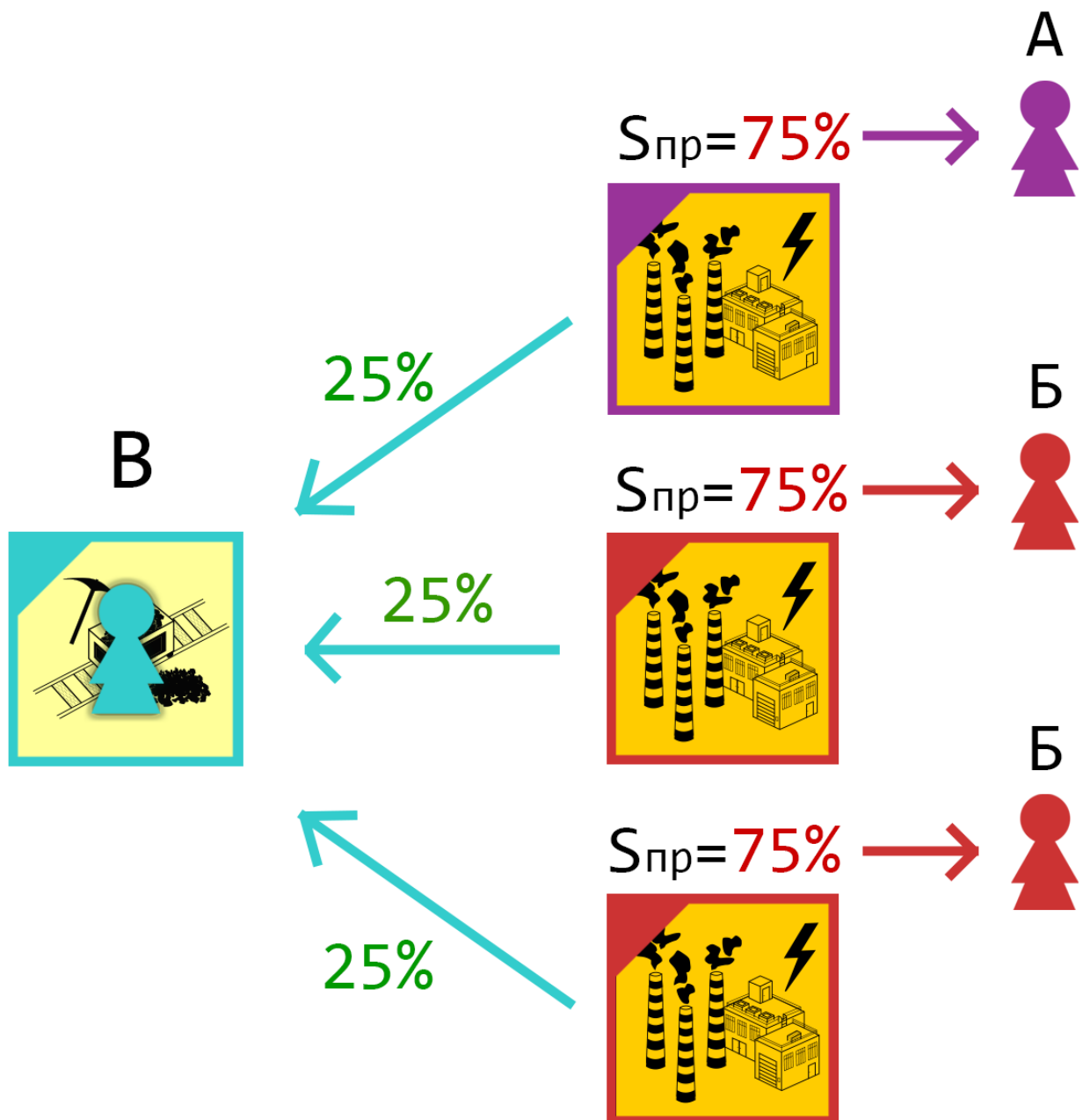


Рисунок А.12 — Зависимость активов от ресурсов (пример №2)

Игрок, фишка которого попала на поле с активом другого участника, может перекупить это поле в своё владение, если у него есть сумма, равная значению, вычисляемому по формуле (А.1).

$$S = 1,5 \cdot S_0 + k \cdot \left[\left| \frac{S_{\text{Д.и.п.}} - S_{\text{Д.и.в.}}}{k} \right|^{1,2} \right], \quad (\text{А.1})$$

где S — сумма, которую платит перекупающий игрок;
 S_0 — базовая стоимость актива;

$S_{\text{д.и.п.}}$ — доход перекупающего игрока;

$S_{\text{д.и.в.}}$ — доход игрока-владельца актива;

k — регулирующий коэффициент, рассчитывающийся по формуле (A.2).

$$k = 10^{\text{digits}(\text{avg}(S_{\text{д.и.п.}}, S_{\text{д.и.в.}})) - 1}, \quad (\text{A.2})$$

где k — регулирующий коэффициент;

$\text{digits}(x)$ — функция, возвращающая число разрядов в целой части числа x (например: $\text{digits}(1000,294) = 4$; $\text{digits}(29492) = 5$);

$\text{avg}(x, y)$ — функция, возвращающая среднее значение между числами x и y ;

$S_{\text{д.и.п.}}$ — доход перекупающего игрока;

$S_{\text{д.и.в.}}$ — доход игрока-владельца актива;

В таком случае 50% этих денег перейдёт бывшему владельцу производства.

Стадии игры

В игре предусмотрено три стадии игры:

– первая стадия. Ходы 1-24. Данная стадия характеризуется тем, что на карте имеются поля со всеми активами, кроме активов атомных электрических станций. Кроме полей с активами присутствуют поля с отпуском;

– вторая стадия. Ходы 25-48. Карта дополняется вертикальным мостом посередине. На клетках-развилках размещаются особые поля со стрелками. На месте некоторых обычных полей появляются поля со всеми ресурсами, кроме залежей урана;

– третья стадия. Ходы 49-72 (либо меньше или больше 72 — зависит от настройки в пользовательском лобби). Карта дополняется горизонтальным мостом посередине, в итоге в центре карты формируется перекрёсток. На месте

некоторых обычных полей появляются поля с диверсиями, ресурсами урана, и активами атомных электрических станций.

Процесс изменения карты продемонстрирован на рисунке А.13.

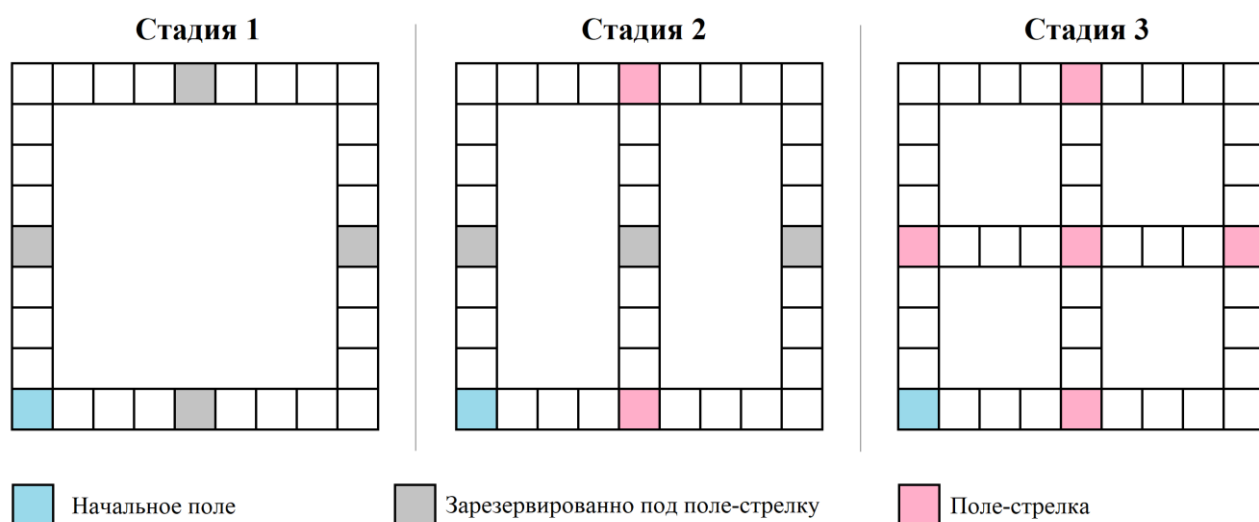


Рисунок А.13 — Расширение карты по стадиям

Конец игры

Предусмотрено несколько сценариев окончания игры и определения победителя:

- по балансу: победителем считается игрок, первым набравший установленную в настройках сумму для победы на своём счёте;
- по балансу игроков на последний ход (если игра ограничена числом ходов): победителем считается игрок, имеющий наибольшую сумму на счету;
- по последнему оставшемуся игроку: если все игроки, кроме одного, сдались или отключились без дальнейшего присоединения к игре, то оставшийся игрок назначается победителем, несмотря на свой денежный баланс.

Возможность сдаться и отключение игрока

В обычном матче предусмотрена возможность сдаться. Если игрок воспользовался данной возможностью, то его денежный баланс будет разделён поровну между оставшимися участниками.

При незапланированном отключении игрока от матча устанавливается пауза на одну минуту, предоставляющая игроку небольшое количество времени для возвращения в игру. По истечению данного времени игра продолжается, а фишкой отключившегося игрока управляет «бот», однако игрок всё ещё сможет вернуться в игру, пока не прошло 10 общих ходов с момента его отключения. На 11 ход игрок будет отключен от матча, его фишка будет удалена из игры, купленные им клетки — освобождены, и его денежный баланс так же будет разделён поровну между оставшимися участниками, как это работает при использовании игроком возможности сдаться.

Для каждого игрока пауза устанавливается единожды, то есть при повторном отключении игрока после его возвращения игра уже не будет приостановлена. Ходы «бота», оставшиеся до отключения, не выполняются при повторном подключении игрока к матчу.

«Бот»

«Бот» служит для управления в течение 10 общих ходов фишками игроков, которые отключились от игры ввиду незапланированной ситуации.

Если его фишка попала на активируемое поле, то «бот» по умолчанию скупает свободные активы, но игнорирует возможные действия любых других полей. «Бот» не способен перекупать чужие активы. Он не является кандидатом на победу, даже если его сумма на счету достигнет значения, необходимого для победы, или если в игре наступил последний ход, баланс «бота» в таблице игроков игнорируется. Однако, стоит учитывать, что если денежный баланс

«бота» достигнет значения, достаточного для завершения игры, это инициализирует конец игры, и его баланс будет игнорироваться.

Проводить диверсию по отношению к «боту» можно, так как он просто является заместителем игрока, который отключился. По аналогии, его активы также можно перекупать.

Рейтинговый матч

Рейтинговая игра начинается только с четырьмя игроками. В рейтинговом матче отсутствует возможность сдать. Отключившийся игрок, не присоединившийся далее по игре в течение 10 общих ходов с момента отключения, на 11-тый ход уже не сможет подключиться обратно и получит штраф в размере «-10» RP («Rating Points» — «Баллы Рейтинга»).

Настройки рейтинговой игры фиксированные:

- конец игры по ходам: 72 хода;
- конец игры по балансу: не ограничено;
- время на ход: 8 секунд;

Количество очков RP, на которое изменяется рейтинг игроков по окончанию игры, определяется занятым местом:

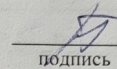
- первое место: +10 RP;
- второе место: +5 RP;
- третье место: -5 RP;
- четвёртое место: -10 RP.

Министерство науки и высшего образования РФ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
Кафедра вычислительной техники

УТВЕРЖДАЮ

Заведующий кафедрой

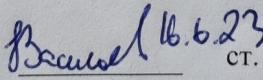

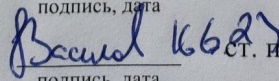
 О.В. Непомнящий
подпись

«22» 06 2023 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 — Информатика и вычислительная техника

Игра «Монополия». Клиентская часть

Руководитель	 16.6.23 подпись, дата	ст. преподаватель	В.С. Васильев
Выпускник	 16.06.23 подпись, дата		А.Ю. Лапунов
Нормоконтролер	 16.6.23 подпись, дата	ст. преподаватель	В.С. Васильев

Красноярск 2023