ТЗ по онлайн-игре шашки

Azriel Gridfen [ZN]

2016

Table of Contents

[Ключевые задачи и технологический выбор для реализации: 2](#_Toc468701072)

[Front-end 3](#_Toc468701073)

[Начальные требования от заказчика (мои комментарии в скобках): 3](#_Toc468701074)

[Что уже выполнено: 3](#_Toc468701075)

[Что нужно выполнить обязательно: 3](#_Toc468701076)

[Back-end 4](#_Toc468701077)

[Что можно было бы выполнить: 4](#_Toc468701078)

# Ключевые задачи и технологический выбор для реализации:

Интерфейс компоненты (Турниры, Игра, Идущие игры, Чат) – React.js+Redux, вариант компоновки можно подсматривать у [Chess.com](https://www.chess.com/live).

Интерфейс стили – Sass(sass syntax)\CSS (опционально, на время разработки можно добавить отдельный .css стиль, добавлю его в sass в конце работы)

Сообщения чата – Rx.js (опционально, но удобно)

Сообщения других компонентов – на выбор, сейчас есть поток мета-информации, в него по желанию можно отправить турниры\идущие игры, но можно и создать отельный обработчик на обычном js.

Связь с сервером – Socket.io

# Front-end

## Начальные требования от заказчика (мои комментарии в скобках):

Нужна игра в шашки, любое количество одновременных партий (не ограниченное выбранным типом сетевой логики). В начале хотелось бы русские, но потом еще добавить международные. Нужен чат и возможность адмиистратором (возможно и пользователем, но это перещелк одного разрешения так что не суть) как-то создавать турниры.

## Что уже выполнено:

Сами шашки, логика, связь с сервером через WebSocket, комнаты на сервере. Большая часть интерфеса, но написана она на бутстрапе (и кажется foundation, не уверен уже).

## Что нужно выполнить обязательно:

Верстка: интерфейс, доделать для недостающих частей программы (турниры, чат, идущие игры, т.д.) и стилизовать его.

Запланированные модули интерфейса: Меню начала игры, Турниры, Чат, Идущие игры, Друзья.

Программирование: тот же интерфейс на React.js, логика интерфейса (состояние компонентов и проч)

## Компоненты

Чат: отображает сообщения текущего матча (Room на сервере). Вид примерно такой

[<b>Отправитель</b>: сообщение]

[<b>Отправитель</b>: сообщение]

[<b>Отправитель</b>: сообщение]

Сами сообщения будут формироваться как сейчас ходы (Room.moves) - в начале приходит массив старых сообщений с сервера потом к ним могут добавляться новые, обрабатывается это Rx-ом, чтобы выровнять массивы для одного обработчика.

Турниры: (предполагаю массив похожий на Rooms) будут содержать информацию о типе турнира и прочее (meta-данные) и при выборе (открытие во всплывающем окне\лайтбоксе?)  показывать полную информацию + массив участников.  
Идущие игры: будут отображать открытые (Room.closed === false) комнаты массив которых будет получен с сервера.  
Начать игру: простой контейнер с двумя выпадающими списками в которых можно выбрать тип и время проведения игры и кнопкой, последняя готова, нужно написать обработчик выбора в списках.

# Back-end

## Что можно было бы выполнить:

Работа сервер с нереализованными элементами интерфейса. Серверную часть покрывать тестами достаточно необходимо, если не умеете – объясню, примеры есть. (Клиент тоже хотелось бы покрыть тестами, но опомнился слишком поздно и вводить тесты в текущий массив код не очень горит). Общение с клиентом в данный момент ведеться в стиле Redux отправляем действия и сервер\клиент их выполняет создавая новый экземпляр хранилища.

# Приложения

Текущее состояние: <https://askadar.github.io/checkers/>

Репо: <https://github.com/Askadar/checkers>