

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
Кафедра Интеллектуальных информационных технологий

«К защите допускаю»
Заведующий кафедрой ИИТ
_____ В. А. Головкин
«__» _____ 2022 г.

РАЗРАБОТКА ЭЛЕКТРОННОЙ ККИ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
ПО ДИСЦИПЛИНЕ «СОВРЕМЕННЫЕ ПЛАТФОРМЫ ПРОГРАММИРОВАНИЯ»
ЗА II СЕМЕСТР

КР.ПО4.190333-05 81 00

Листов 19

Руководитель

А. А. Крощенко

Выполнил

П. И. Галанин

Консультанты:

по ЕСПД

А. А. Крощенко

Рецензент

А. А. Крощенко

лист с заданием

Содержание

| | |
|---|----|
| ВВЕДЕНИЕ | 4 |
| 1 АНАЛИЗ ПОСТАВКИ ЗАДАЧИ | 5 |
| 1.1 Перечень функций | 5 |
| 1.2 Требования пользователей | 5 |
| 1.3 Описание предметной области | 6 |
| 1.4 Варианты использования программы в виде диаграмм прецедентов (use case diagram) | 7 |
| 1.5 Первичное описание объектов и классов | 8 |
| 1.6 Диаграмма состояний (statechart diagram) для прецедентов | 9 |
| 2 ПРОЕКТИРОВАНИЕ СТРУКТУРЫ ПРИЛОЖЕНИЯ | 10 |
| 2.1 Графический интерфейс приложения (UI/UX) | 10 |
| 2.2 Общая диаграмма с учётом каркаса (layer diagram) | 12 |
| 2.3 Диаграмма последовательностей (sequence diagram) | 12 |
| 2.4 Диаграмма видов деятельности (activity diagram) | 13 |
| 3 РАЗРАБОТКА АЛГОРИТМОВ ФУНКЦИОНИРОВАНИЯ И СТРУКТУР ДАН- НЫХ | 14 |
| 4 РЕАЛИЗАЦИЯ ПРИЛОЖЕНИЯ И РЕЗУЛЬТАТЫ ИСПЫТАНИЙ | 16 |
| 4.1 Диаграмма компонентов (component diagram) | 16 |
| 4.2 Диаграмма развёртывания (deployment diagram) | 16 |
| 4.3 Тестирование приложения | 17 |
| СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ | 19 |

ПРИЛОЖЕНИЕ А. СХЕМА ПРОГРАММЫ

ПРИЛОЖЕНИЕ Б. ТЕКСТ ПРОГРАММЫ

| | | | | | | | | | |
|-----------|----------|----------|-------|------|--|--|--|--|--|
| | | | | | КР.ПО4.190333-05 81 00 | | | | |
| | | | | | | | | | |
| Изм | Лист | № докум. | Подп. | Дата | | | | | |
| Разраб. | Галанин | | | | <div>РАЗРАБОТКА ЭЛЕКТРОННОЙ ККИ</div> <div>ПОЯСНИТЕЛЬНАЯ ЗАПИСКА</div> <div>УО «БрГТУ»</div> | | | | |
| Пров. | Крощенко | | | | | | | | |
| | | | | | | | | | |
| Н. контр. | Крощенко | | | | | | | | |
| УТВ. | | | | | | | | | |

ВВЕДЕНИЕ

Цель курсовой работы по дисциплине «Современные платформы программирования» - детальное проектирование и программная реализация сетевой игры.

Клиенская часть написана на языке программирования Java Script, с использованием фреймворка ReactJS, с использованием объектно-ориентированного программирования. Серверная часть написана на языке программирования Node Java Script.

Используя объектно-ориентированное проектирование позволяет обходить ряд сложных проблем в программировании, сводя необходимую модификацию программы к её расширению и дополнению. Систематическое применение объектно-ориентированного подхода позволяет разрабатывать достаточно хорошо структурированные, надежные в эксплуатации, просто модифицируемые программные системы. Элементы объектно-ориентированного программирования получили своё развитие, и в настоящее время ООП принадлежит к числу ведущих технологий программирования.

| | | | | | | |
|-----|------|----------|-------|------|------------------------|------|
| | | | | | КР.ПО4.190333-05 81 00 | Лист |
| Изм | Лист | № докум. | Подп. | Дата | | 4 |

1 АНАЛИЗ ПОСТАВКИ ЗАДАЧИ

1.1 Перечень функций

Достижение цели курсовой работы предполагает необходимость создания веб- или мобильного приложения (ОС Android/iOS) со следующим функционалом:

1. Предусмотреть возможность игры с ИИ.
2. Разработать фиксированный набор карт для игры (от 30 и более).
3. Сохранение статистики матчей (опционально - сохранение на удаленном сервере).
4. Поддержка сетевого матча один-на-один.
5. Минимальный пользовательский интерфейс и графическое оформление карт.

1.2 Требования пользователей

Пользовательские требования - описание на естественном языке (плюс поясняющие диаграммы) функций, выполняемых системой, и ограничений, накладываемых на неё.

Ответственный: Системный аналитик

Эти требования должны определять только внешнее поведение системы, избегая по возможности определения структурных характеристик системы. Пользовательские требования должны быть написаны естественным языком с использованием простых таблиц, а также наглядных и понятных диаграмм.

Обязательные требования игрока к игре:

1. Разработать фиксированный набор карт для игры (от 30 и более).
2. Минимальный пользовательский интерфейс и графическое оформление карт.

Желательные требования к игре:

1. Предусмотреть возможность игры с ИИ.
2. Сохранение статистики матчей (опционально - сохранение на удаленном сервере).
3. Поддержка сетевого матча один-на-один.

| | | | | | | |
|-----|------|----------|-------|------|------------------------|------|
| | | | | | КР.ПО4.190333-05 81 00 | Лист |
| Изм | Лист | № докум. | Подп. | Дата | | 5 |

1.3 Описание предметной области

Игрок - человек или машина, который/которая играет в данную игру. В данной игре будут два игрока: мы и игрок противник.

Карты - графический элемент игры, в виде прямоугольника с характеристиками. Карт в игре в количестве 30 штук. 30 штук как и у нас, так и у противника. Карты имеют фракции: мечник, лучник, тяжелая артиллерия; который сдаются в определенное поле боя.

Мечник - категория карты, которая имеет малый урон и сражается с мечниками противника.

Лучник - категория карты, которая имеет малый урон и сражается с лучниками противника.

Тяжелая артиллерия - категория карты, которая имеет высокий урон и сражается с тяжелой артиллерией противника.

Поле боя - игровое поле, которое разбито на 6 частей, из которых три части противника и три части наши.

Поле боя содержит:

1. Тяжелую артиллерию противника.
2. Лучников противника.
3. Мечников противника.
4. Своих мечников.
5. Своих лучников.
6. Свою тяжелую артиллерию.

ПАС - действие, после объявления которого мы объявляем о прекращении бойни и пропускаем ход, в тоже время противнику дается последний выбор карты, после чего подсчитывается игровой счёт. В течении игры можна два раза нажать на кнопку «ПАС». После подсчёта первого «ПАС» карты с поля боя идут на кладбище. Набираются карты до десяти. Происходит игра. После подсчёта второго «ПАС» сравниваются победы и объявляется победитель.

| | | | | | | |
|-----|------|----------|-------|------|------------------------|------|
| | | | | | КР.ПО4.190333-05 81 00 | Лист |
| | | | | | | 6 |
| Изм | Лист | № докум. | Подп. | Дата | | |

1.4 Варианты использования программы в виде диаграмм прецедентов (use case diagram)

Варианты использования программы изображены на рис. 1.1.

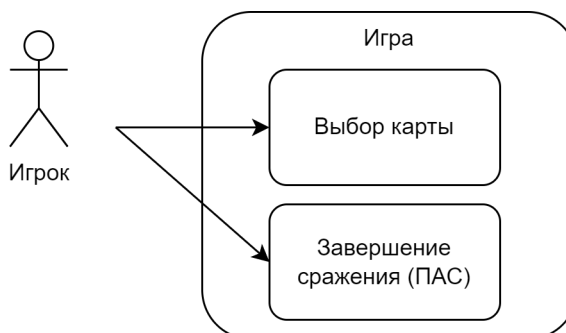


Рисунок 1.1 – Первичное описание прецедентов

Прецедент №1 «Выбор карты»

Назначение: из своей колоды карт выбираем карту, которую ходим выставить на поле.

Исполнители: игрок.

Предусловие: есть карта/карты, которую/которые можно выбрать.

Основной поток событий: игрок выберет карту и карты попадет на поле (мечника, лучника, тяжелой артиллерии), иначе АПС.

Альтернативный поток событий: если у игрока не будет карт, то игрок пропускает ход; если игрок не выбирает карту, то по истечению времени карта выбирается автоматически.

Прецедент №2 «Завершения сражения (ПАС)»

Назначение: завершить бой и узнать результат бойни.

Исполнители: игрок.

Предусловие: есть карты на поле.

Основной поток событий: мы пропускаем ход, а противник выбирает последнюю карту, после чего происходит подсчет сил, иначе АПС.

Альтернативный поток событий: игрок может продолжать выставлять карты на поле.

1.5 Первичное описание объектов и классов

Первичное описание класса игры изображено на рис. 1.2.

| game |
|---------------------------------------|
| # Колода карт противника |
| # Моя колода карт |
| # Карты на руках противника |
| # Мои карты на руках |
| # Кладбище карт противника |
| # Моё кладбище карт |
| # Карты артиллерии на поле противника |
| # Карты артиллерии на моем поле |
| # Карты лучников на поле противника |
| # Карты лучников на моём поле |
| # Карты мечников на поле противника |
| # Карты мечников на моём поле |
| + получить JSON данных о игре () |
| + переместить карту с рук на поле () |
| + прекратить бой () |

Рисунок 1.2 – Первичное описание класса игры

Игра - класс содержащий логику игры.

- Колода карт противника и моя колода карт - начальный стак карточек, из которых в начале игры берется 10 случайных карт.
- Карты на руках противника и мои карты на руках - карты, которые держим в руках, не могут превышать 10 штук. При количестве 0 штук - игрок пропускает ход.
- Кладбище карт противника и моё кладбище карт - карты которые ушли в отбой после первого сражения, так как игрок сделал ПАС и были подсчитаны результаты боя.
- Карты артиллерии на поле противника и карты артиллерии на моём поле - карты, которые выставлены на поле артиллерии. Могут отсутствовать (0 штук).
- Карты лучников на поле противника и карты лучников на моём поле - карты, которые выставлены на поле лучников. Могут отсутствовать (0 штук).
- Карты мечников на поле противника и карты мечников на моём поле - карты, которые выставлены на поле мечника. Могут отсутствовать (0 штук).

1.6 Диаграмма состояний (statechart diagram) для прецендентов

Диаграмма состояний для игры изображена на рис. 1.3.

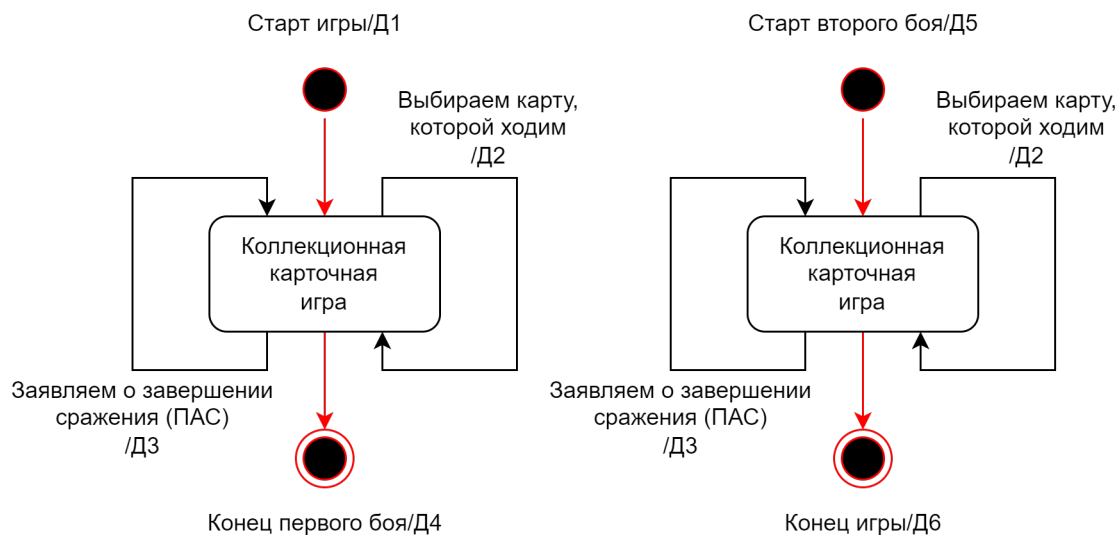


Рисунок 1.3 – Диаграмма состояний игры

- Д1 - Начало игры (выбрали режим: играть с ИИ или играть с другим игроком по сети)
- Д2 - Выбираем карту из колоды, которой будет ходить (мечника, лучника, артиллерию).
- Д3 - Заявляем о завершении боя: мы пропускаем ход, противник кладет последнюю карту, боя заканчивается.
- Д4 - Завершение первого боя и подсчёт сил. Старт второго боя
- Д5 - Старт второго боя.
- Д6 - Завершение второго боя и подсчёт сил. Сравниваются два боя. Объявляется победитель. Завершается игра.

2 ПРОЕКТИРОВАНИЕ СТРУКТУРЫ ПРИЛОЖЕНИЯ

2.1 Графический интерфейс приложения (UI/UX)

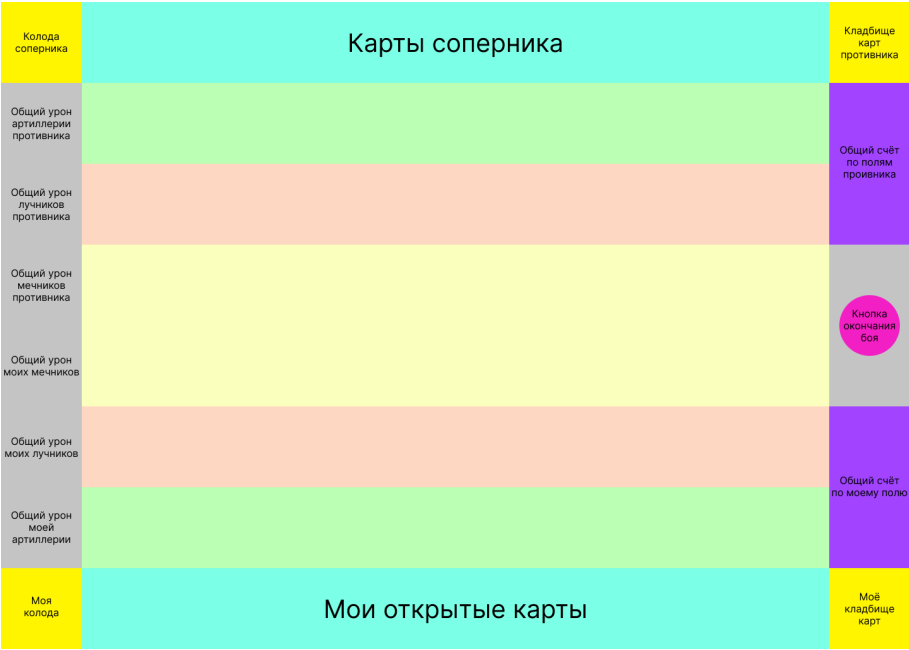


Рисунок 2.4 – Макет интерфейса (UX)

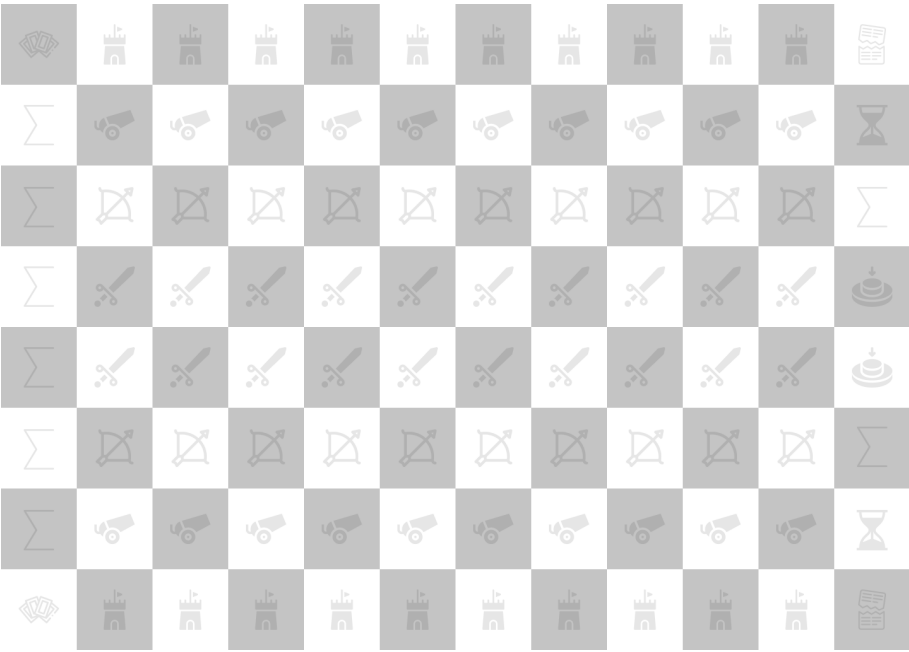


Рисунок 2.5 – Дизайн интерфейса (UI)



Рисунок 2.6 – Урон 1

мечником



Рисунок 2.7 – Урон 2

мечником



Рисунок 2.8 – Урон 4

мечником



Рисунок 2.9 – Урон 5

мечником



Рисунок 2.10 – Урон 1

лучником



Рисунок 2.11 – Урон 5

лучником



Рисунок 2.12 – Урон 6

лучником



Рисунок 2.13 – Урон 1

артиллерией



Рисунок 2.14 – Урон 5

артиллерией



Рисунок 2.15 – Урон 6

артиллерией

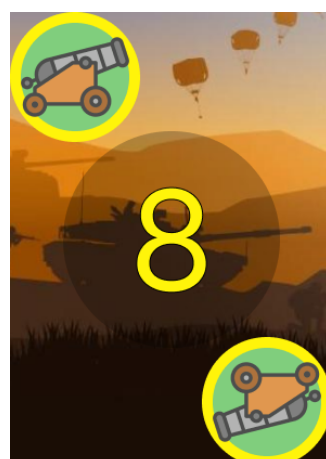


Рисунок 2.16 – Урон 8

артиллерией

2.2 Общая диаграмма с учётом каркаса (layer diagram)

Общая диаграмма с учётом каркаса изображена на рис. 2.17.

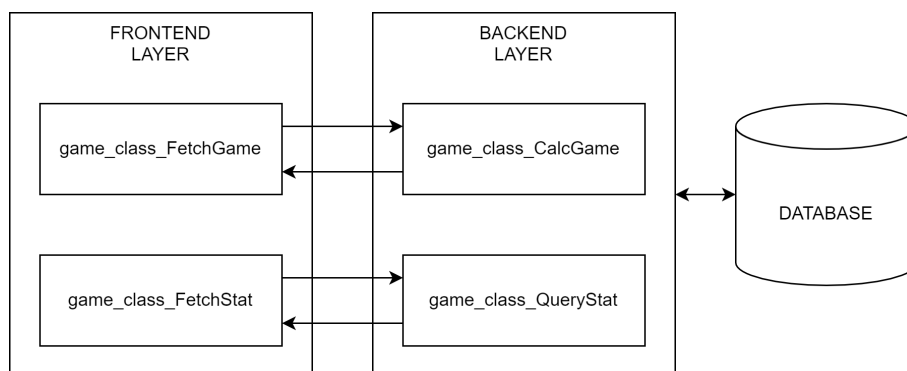


Рисунок 2.17 – Общая диаграмма с учётом каркаса

2.3 Диаграмма последовательностей (sequence diagram)

Диаграмма последовательностей изображена на рис. 2.17.

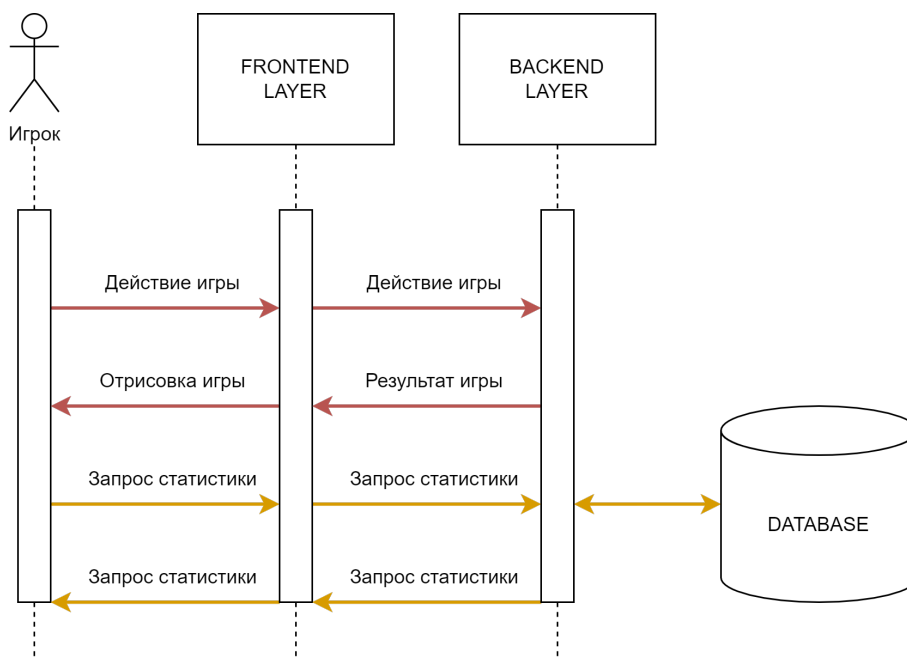


Рисунок 2.18 – Диаграмма последовательностей

2.4 Диаграмма видов деятельности (activity diagram)

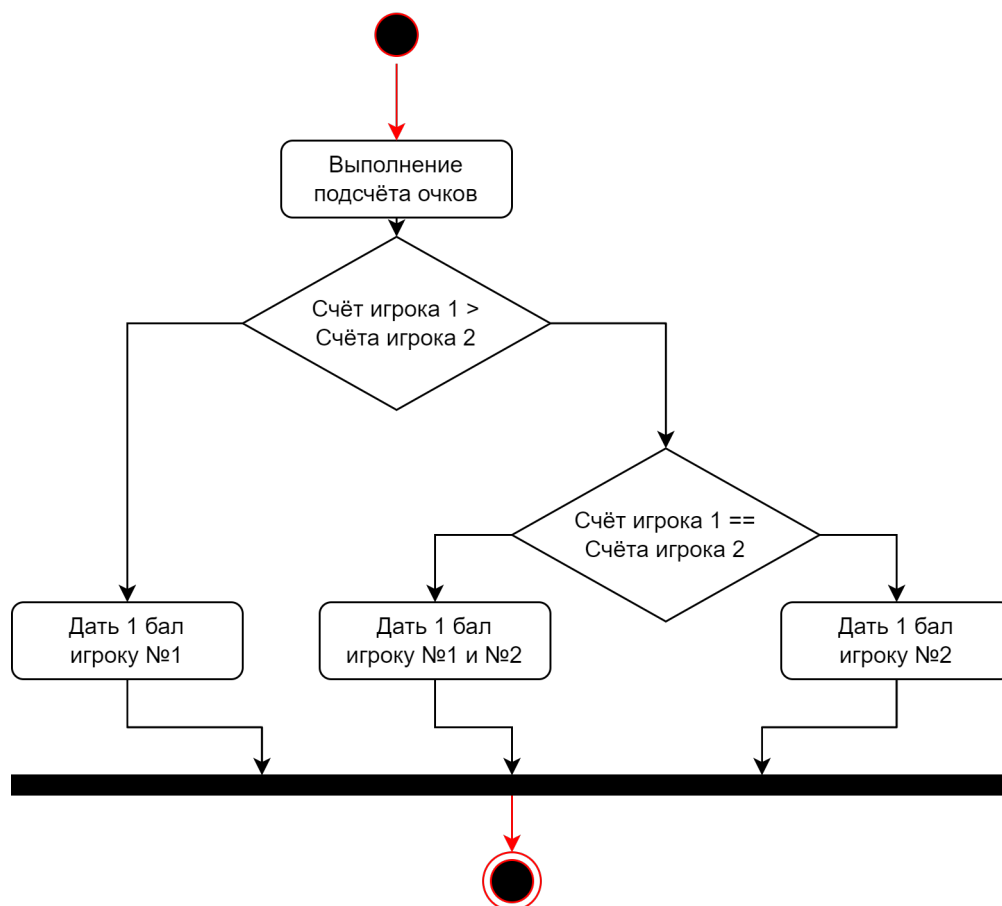


Рисунок 2.19 – Диаграмма вида деятельности счётчика баллов

3 РАЗРАБОТКА АЛГОРИТМОВ ФУНКЦИОНИРОВАНИЯ И СТРУКТУР ДАННЫХ

Алгоритм нахождения сумарного урона игрока

Исходные данные: нет данных

Алгоритм:

1. counter = 0. Переход к пункту 2
2. i = 0. Проходимся по массиву карточек карточек. Переход к пункту 2.1.
 - 2.1. Если «тип карточки» равен «на войне», то переход к пункту 2.2, иначе переход к пункту 2.3.
 - 2.2. counter = counter + «урон карточки». Переход к пункту 2.3.
 - 2.3. i = i + 1. Если i < «размер массива», то переход к пункту 2.1, иначе переход к пункту 3.
3. return counter

Алгоритм перетаскивания карточки на поле боя

Исходные данные: 1) ИД карты

Алгоритм:

1. Если это не наш ход, то переход к пункту 12, иначе переход к пункту 2.
2. Если мы выбрали ПАС, то переход к пункту 12, иначе переход к пункту 3.
3. Если противник не ПАСовал, то переход к пункту 4, иначе переход к пункту 6.
4. Блокируем доступ к нашим картам. Переход к пункту 5.
5. Разблокируем доступ к картам противника. Переход к пункту 6.
6. Перём карту под ИД равным id. Переход к пункту 7.
7. Рендерим наши карты. Переход к пункту 8.
8. Рендерим наш общий урон. Переход к пункту 9.
9. Рендерим наш урон мечников. Переход к пункту 10.
10. Рендерим наш урон лучников. Переход к пункту 11.
11. Рендерим наш урон тяжелой артиллерии. Переход к пункту 12.
12. Конец.

Алгоритм ПАСа у нас

Исходные данные: нет данных

Алгоритм:

1. Если у нас есть ПАС, то переход к пункту 6, иначе переход к пункту 2.
2. Устанавливаем значение true, что мы ПАСанули. Переход к пункту 3.
3. Отключаем доступ к нашим картам. Переход к пункту 4.
4. Если противник не ПАСовал, то переход к пункту 5, иначе переход к пункту 6.
5. Разблокировываем карты противника. Переход к пункту 6.
6. Конец.

Выходные данные: мы пропустили ход, противник делает сколько угодно ходов, пока сам не ПАСёт.

Алгоритм ПАСа у противника

Исходные данные: нет данных

Алгоритм:

1. Если у противника есть ПАС, то переход к пункту 6, иначе переход к пункту 2.
2. Устанавливаем значение true, что проивник ПАСанул. Переход к пункту 3.
3. Отключаем доступ к картам противника. Переход к пункту 4.
4. Если мы не ПАСовал, то переход к пункту 5, иначе переход к пункту 6.
5. Разблокировываем наши карты. Переход к пункту 6.
6. Конец.

Выходные данные: противник пропустил ход, мы делаем сколько угодно ходов, пока мы не ПАСём.

Структура данных таблицы «Результаты игры»

Листинг: Создание таблицы «Результаты игры»

```
CREATE TABLE 'GameResults' (  
  'code' INT NOT NULL AUTO_INCREMENT ,  
  'date' DATE NOT NULL ,  
  'player1' VARCHAR(120) NOT NULL ,  
  'player2' VARCHAR(120) NOT NULL ,  
  'player1_damage' INT NOT NULL ,  
  'player2_damage' INT NOT NULL ,  
  PRIMARY KEY ('code')  
) ENGINE = InnoDB CHARSET=utf8 COLLATE utf8_general_ci;
```

| | | | | | | |
|-----|------|----------|-------|------|------------------------|------|
| | | | | | КР.ПО4.190333-05 81 00 | Лист |
| | | | | | | 15 |
| Изм | Лист | № докум. | Подп. | Дата | | |

4 РЕАЛИЗАЦИЯ ПРИЛОЖЕНИЯ И РЕЗУЛЬТАТЫ ИСПЫТАНИЙ

4.1 Диаграмма компонентов (component diagram)

Диаграмма компонентов изображена на рис. 4.20.

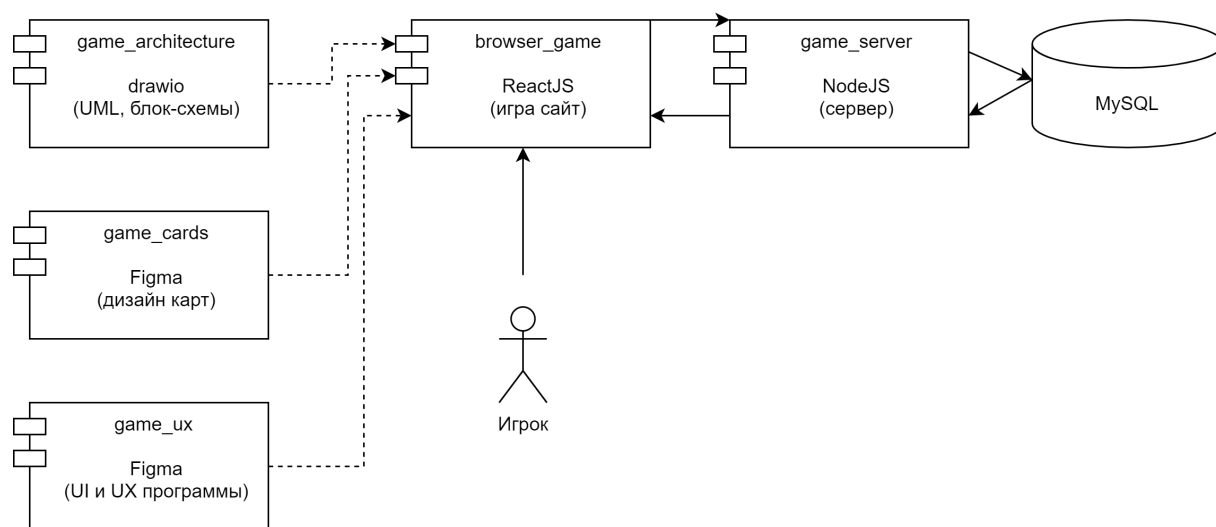


Рисунок 4.20 – Диаграмма компонентов

4.2 Диаграмма развёртывания (deployment diagram)

Диаграмма развёртывания изображена на рис. 4.21.

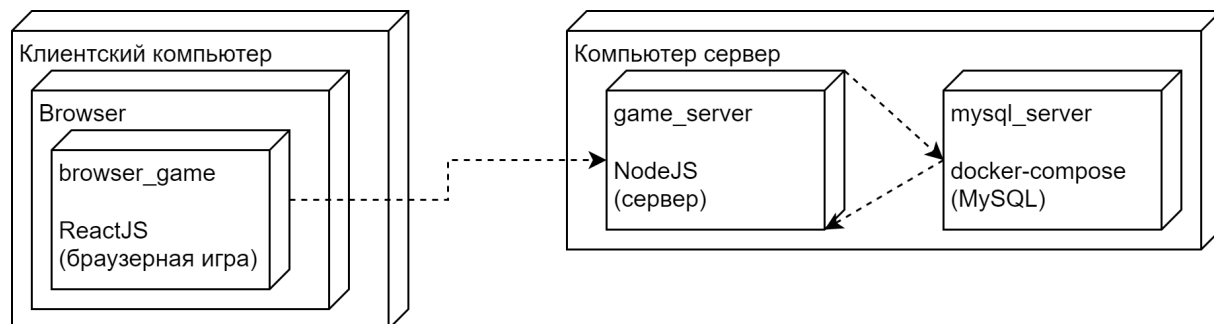


Рисунок 4.21 – Диаграмма развёртывания

4.3 Тестирование приложения

Тестирование просмотра своих карт в руках

Тестируемая задача: Просмотр своих карт в руках.

Ожидаемый результат: видны карты в руках, когда нужно ходить.

Полученный результат: При нашем ходе наши карты яркие. При ходе противника - наши карты будут тусклыми. Скриншот на рисунке 4.22.

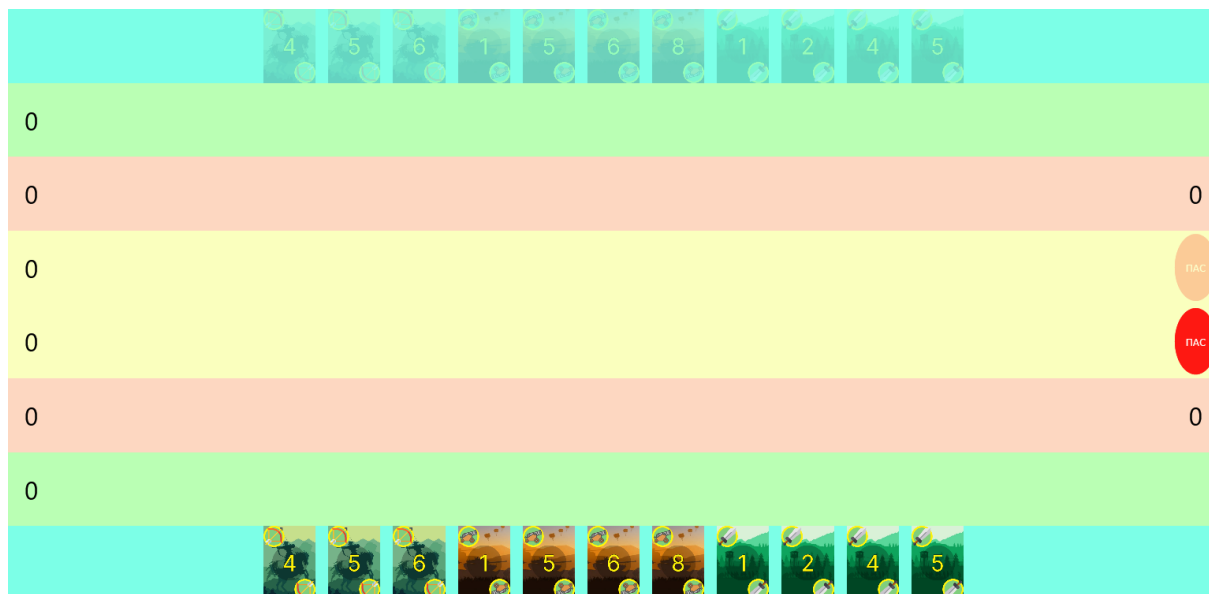


Рисунок 4.22 – Наши карты на руках. Наш ход

Вывод по тесту: При нашем ходе карты яркие. При ходе противника наши карты становятся тусклыми. Результат теста соответствует ожиданию.

Тестирование добавление карты на стол

Тестируемая задача: Добавление карты на стол.

Ожидаемый результат: выбранная карта положились на стол.

Полученный результат: Скриншот на рисунке 4.23.

Вывод по тесту: Выбранная карта кладется на нужное поле боя (поле мечника, поле лучника, поле тяжелой артиллерии). Результат теста соответствует ожиданию.

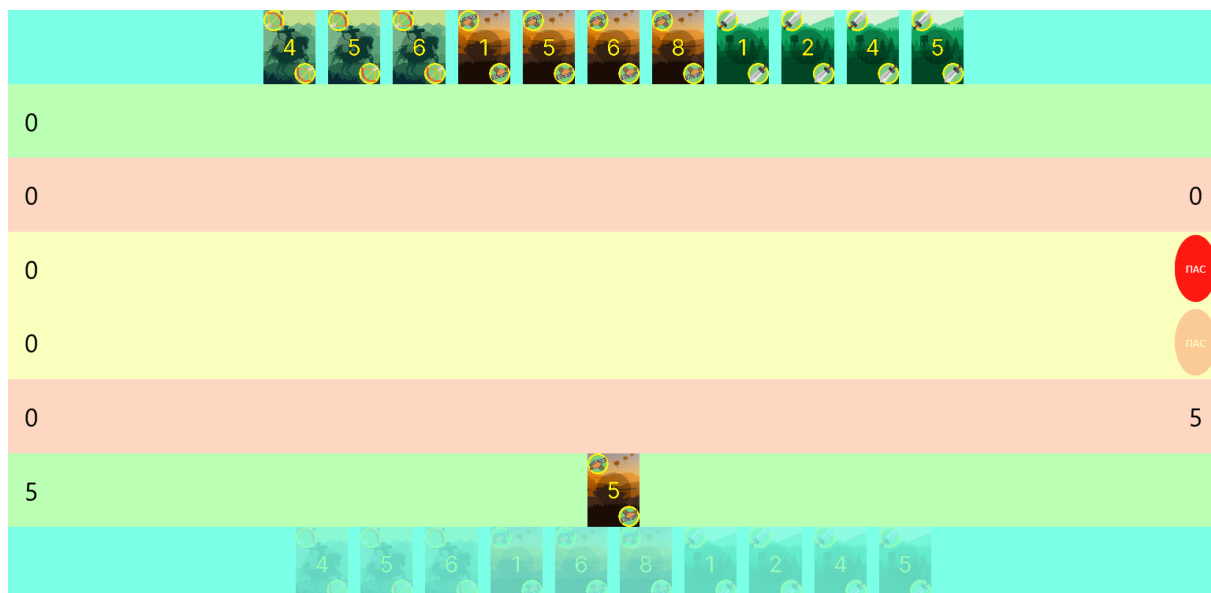


Рисунок 4.23 – Переташили карту на боевое поле

Тестирование окончания боя

Тестируемая задача: Окончание боя.

Ожидаемый результат: мы пропускаем ход, а противник делает последний ход.

Полученный результат:

После того как мы ПАСанули, то противник может делать неограниченное количество ходов, пока сам не ПАСанёт. Как два игрока сделали ПАС, то происходит подсчёт общего урона, который выводится на экран. Скриншот на рисунке 4.24.

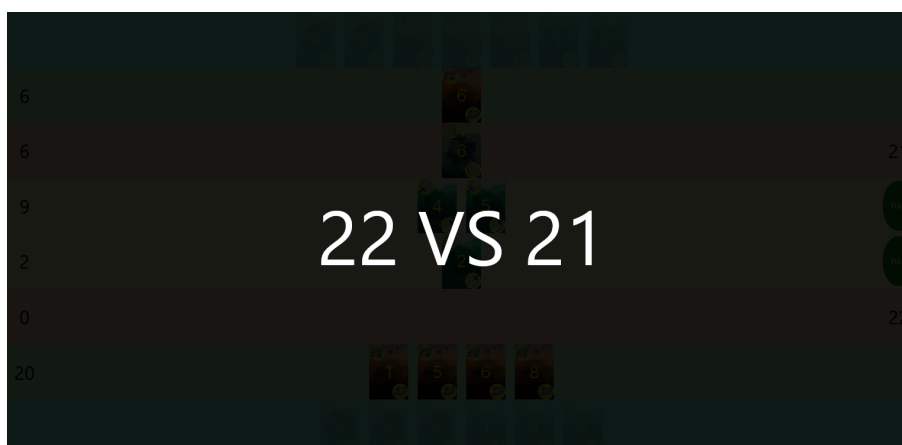


Рисунок 4.24 – Окончание боя

Вывод по тесту: После того как ма ПАСанули, то мы не могли ходить. Противник мог делать неограниченное количество ходов, пока сам не ПАСанул. Как все сделали ПАС, то произошел подсчёт общего урона и вывелся результат на экран.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Получение GET и POST запросов на Node.js - YouTube - [Электронный ресурс]
Режим доступа: <https://www.youtube.com/watch?v=YMJDUNHuccvA> Дата доступа: 19.11.2021.
2. Подключение к базе данных MySQL в Node.js - YouTube - [Электронный ресурс]
Режим доступа: <https://www.youtube.com/watch?v=YhuozY-qplI> Дата доступа: 19.11.2021.
3. Модули Node.js, require - YouTube - [Электронный ресурс] Режим доступа: <https://www.youtube.com/watch?v=1PkarXC-9TQ> Дата доступа: 19.11.2021.
4. Manual installation steps for older versions of WSL | Microsoft Docs - [Электронный ресурс] Режим доступа: <https://aka.ms/wsl2kernel> Дата доступа: 19.11.2021.
5. Леоненков А. В. Самоучитель UML 2. – СПб.: БХВ-Петербург, 2007. – 576с.
6. Почему в папке 'React-router-dom' нет экспорта Switch? — Хабр Q&A - [Электронный ресурс] Режим доступа: <https://qna.habr.com/q/1103400> Дата доступа: 15.05.2022.
7. React Router | Upgrading from v5 - [Электронный ресурс] Режим доступа: <https://reactrouter.com/docs/en/v6/upgrading/v5> Дата доступа: 15.05.2022.
8. reactjs - How to build a 404 page with react-router-dom v6 - Stack Overflow - [Электронный ресурс] Режим доступа: <https://stackoverflow.com/questions/67050966/how-to-build-a-404-page-with-react-router-dom-v6> Дата доступа: 15.05.2022.
9. AugustinSorel/multiplayerTicTacToe: simple multiplayer made with react and socket io - [Электронный ресурс] Режим доступа: <https://github.com/AugustinSorel/multiplayerTicTacToe> Дата доступа: 16.05.2022.