Циферки!

Оглавление:

[Процесс: 1](#_Toc397616914)

[Время: 1](#_Toc397616915)

[Количество очков: 1](#_Toc397616916)

[Комбинации: 2](#_Toc397616917)

[Сетевое взаимодействие: 3](#_Toc397616918)

# Процесс:

1. Бросок кубика на очередность;
2. Процесс игры (13 раундов):
   1. Бросок 6 кубиков;
   2. Переброс выбранных кубиков (можно отказаться);
   3. Выбор комбинации;
   4. По желанию можно «Сдаться»;
3. Подсчет очков:
   1. Сумма всех;
   2. Если сумма первых шести комбинаций (Ner) больше либо равна 63, то +35 очков;
4. Определение победителя.

# Время:

Все шахматные варианты: время на одного игрока на все ходы, общая продолжительность и т.д.

# Количество очков:

Минимальное: 6, максимальное: 414.

# Комбинации:

|  |  |  |  |
| --- | --- | --- | --- |
| id | Название | Условие | Очки |
| 0 | Единицы (Ones, 1er) | — | Сумма всех 1 |
| 1 | Двойки (Twos, 2er) | — | Сумма всех 2 |
| 2 | Тройки (Threes, 3er) | — | Сумма всех 3 |
| 3 | Четверки (Fours, 4er) | — | Сумма всех 4 |
| 4 | Пятерки (Fives, 5er) | — | Сумма всех 5 |
| 5 | Шестерки (Sixes, 6er) | — | Сумма всех 6 |
| 6 | Сет (3 of a kind, Dreier Pasch) | 3 одинаковых | Сумма всех |
| 7 | Каре (4 of a kind, Vierer Pasch) | 4 одинаковых | Сумма всех |
| 8 | Фул-хаус (Full house, Full House) | 2 одинаковых и  3 одинаковых | 25 |
| 9 | Маленькая улица (Small straight, Kleine Straße) | 4 в ряд (1..4, 2..5, 3..6) | 30 |
| 10 | Большая улица (Large straight, Große Straße) | 5 в ряд (1..5, 2..6) | 40 |
| 11 | Язь (Yatzy, Yazzee) | 5 одинаковых | 50 |
| 12 | Шанс (Chance, Chance) | — | Сумма всех |

# Сетевое взаимодействие:

|  |  |  |
| --- | --- | --- |
| № | Название запроса | Ответ |
| 0 | Подключение (/api/connectPlayer) | {}: string sessionID; int playersOnline, playersSearching, activeGames; (но передаются все значения string) |
| 1 | Запрос игры (/api/findGame/#players#), “/api/findGame” оставлен для обратной совместимости (players равен 2) | 1. Если sessionID пуст: null 2. Если игра не готова: int playersOnline, playersSearching, activeGames; 3. Если игра готова:  string \_id;  int playerIndex;  [] players;  [] names;  [] rounds; int status; |
| 2 | Выход из очереди поиска игры (/api/stopFindGame) | “1” (просто символ 1 =) ) |
| 3 | Сгенерируй кубики (/api/dices) | {}: [] dices; (объект содержащий массив по имени dices) |
| 4 | Данные игры (/api/rounds/#gid#) | {}: game (смотри п.1.3) |
| 5 | Замена кубиков (/api/dices/#di#) | {}: [] dices; |
| 6 | Выбор комбинации (/api/rounds/#gid#/#ci#) | “1” (просто символ 1 =) ) |
| 7 | Сдайся (/api/giveup) | {}: game (смотри п.1.3) |
| 8 | Отключить свою сессию (/api/disconnectPlayer) | Кол-во активных сессий |
| 9 | Изменить имя (/api/changeName/#name#) | {}: string login |

Дополнительные сведения:

1) все данные в формате JSON, [] — массив, {} — объект;

2) направление всех запросов «от клиента к серверу»;

3) http://188.187.61.24:33322;

4) все запросы GET, в заголовке содержатся cookie, в которых есть sessionID, идентификация по sessionID;

5) #di# заменить на индексы кубиков, которые нужно менять (индекс с 0);

6) #ci# выбрать комбинацию с данным индексом (индекс с 0);

7) #gid# заменить на идентификатор игры (“g1”);

8) #name# заменить на имя игрока (проверяется regexp(“^[A-z0-9\_-]{3,16}$”), т. е. проверяет есть ли что-то кроме букв латинского алфавита, цифр, дефисов и нижних подчеркиваний, а также имеет длину от 3 до 16 символов);

9) #players# заменить на количество игроков в ожидаемой игре, т.е. искать игру для 1, 2, 3, 4 игроков.

Предполагаемый порядок действий:

1. Подключение;
2. Периодическая операция:
   1. Запрос игры;
   2. Если игра не найдена, то ожидание (1 секунда) и переход на пункт 2, иначе пункт 3;
3. Сгенерируй кубики;
4. Периодическая операция:
   1. Данные игры;
      1. Проверка есть ли в ответе от сервера пункт results, если есть, то конец игры;
   2. Произвольное действие:
      1. Сгенерируй кубики;
      2. Замена кубиков;
      3. Выбор комбинации;
      4. Сдайся;
   3. Ожидание (1 секунда), переход к пункту 4.