

Парный обратный сапер

В качестве тестового задания, пожалуйста, реализуйте следующую клиент-серверную мини игру. Старайтесь не тратить на данное задание более трех часов, если время закончилось, но Вы не успеваете реализовать все - присылайте то, что уже готово. Главная цель - иметь законченную, корректно работающую часть задания.

Правила игры:

В игре участвуют два игрока, они ходят по очереди на квадратном поле размером $N \times N$ ($N \leq 6$), на котором в случайных клетках при старте игры генерируются M (M - нечетное) алмазов. Изначально все клетки закрыты для обоих игроков.

Во время хода игрок открывает любую закрытую клетку. При открытии клетки обоим игрокам показывается либо число (количество клеток от 0 до 8, окружающих данную, содержащих алмаз), либо алмаз. Если игрок открыл число, то ход переходит другому игроку, если игрок открыл алмаз, то он делает еще один ход.

Игра заканчивается, когда все алмазы были найдены. Побеждает тот игрок, который собрал больше алмазов.

Техническая реализация:

Для технической реализации данной задачи используйте:

1. Браузерную клиент напишите на **Vue**, мы не будем смотреть на красоту дизайна, основной упор стоит сделать на работоспособности данного интерфейса и читаемость кода.
2. Серверную часть реализуйте на **NestJS**, используя ESLint и Prettier.
 - a. Создание игры реализовать через POST запрос, принимая в качестве параметров N и M (размер поля и количество алмазов соответственно).
 - b. Дальнейшую коммуникацию в течение игры реализуйте с помощью WebSocket соединения.
3. Готовый результат пришлите в виде ссылки на репозиторий Github.

Дополнения:

Любые разумные дополнения к правилам и функционалу приветствуются. Если непонятно, что имелось в том или ином месте - сделайте так, как считаете нужным, но напишите обоснование комментарием.

Если Вы не укладываетесь в три часа, но хотите показать необходимость в реализации какого-либо функционала - оставьте TODO: в нужном месте с описанием необходимого функционала.

Оценка:

Тестовое задание будет оцениваться по следующим критериям:

1. Смысловая завершенность фичи
2. Отсутствие багов
3. Красота и читаемость кода
4. Корректность работы кода в различных ситуациях