

VWORLD

**симулятор живых
существ с
памятью,
эмоциями и веб-
интерфейсом**



НАШЕ РЕШЕНИЕ

- ① Создание и управление агентами**
- ② Симуляция мира**
запуск/остановка, в фоне агенты живут сами
- ③ Возможность изменения окружения**
- ④ У каждого агента есть память и эмоциональный интелект**
- ⑤ Создание событий на карте**
добавленные события вызывают реакцию агентов и провоцируют изменения в их эмоциях, целях и отношениях
- ⑥ Строится граф отношений между агентами**

КЛЮЧЕВЫЕ ВОЗМОЖНОСТИ

Чем наше решение выделяется среди конкурентов

① Долговременная память

векторное хранилище

② Суммаризация происходящего

③ Страница каждого агента

история событий и взаимодействий с другими агентами

④ Масштабируемая архитектура

микросервисная архитектура

⑤ Пресеты агентов и мобов



АРХИТЕКТУРА ФРОНТЕНДА

App Layer (Next.js App Router)

отвечает за маршрутизацию, сквозную логику и сборку экранов из готовых блоков

Feature Layer (@identity/*)

доменные фичи вынесены в изолированные пакеты (фрагменты)

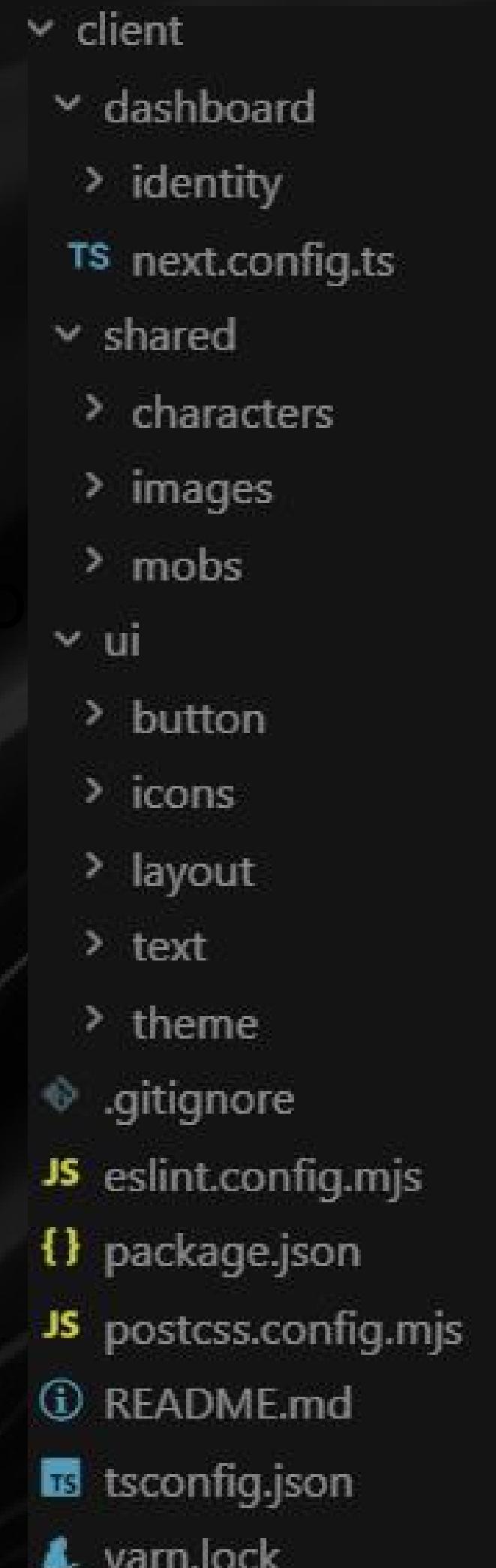
UI Layer (ui/*)

библиотека дизайн-системы - атомарные компоненты (Box, Text, Button) и дизайн-токены

Localization (next-intl)

все тексты управляются централизованно. Это позволяет легко добавлять новые языки сразу для всего приложения.

Модульная монорепо

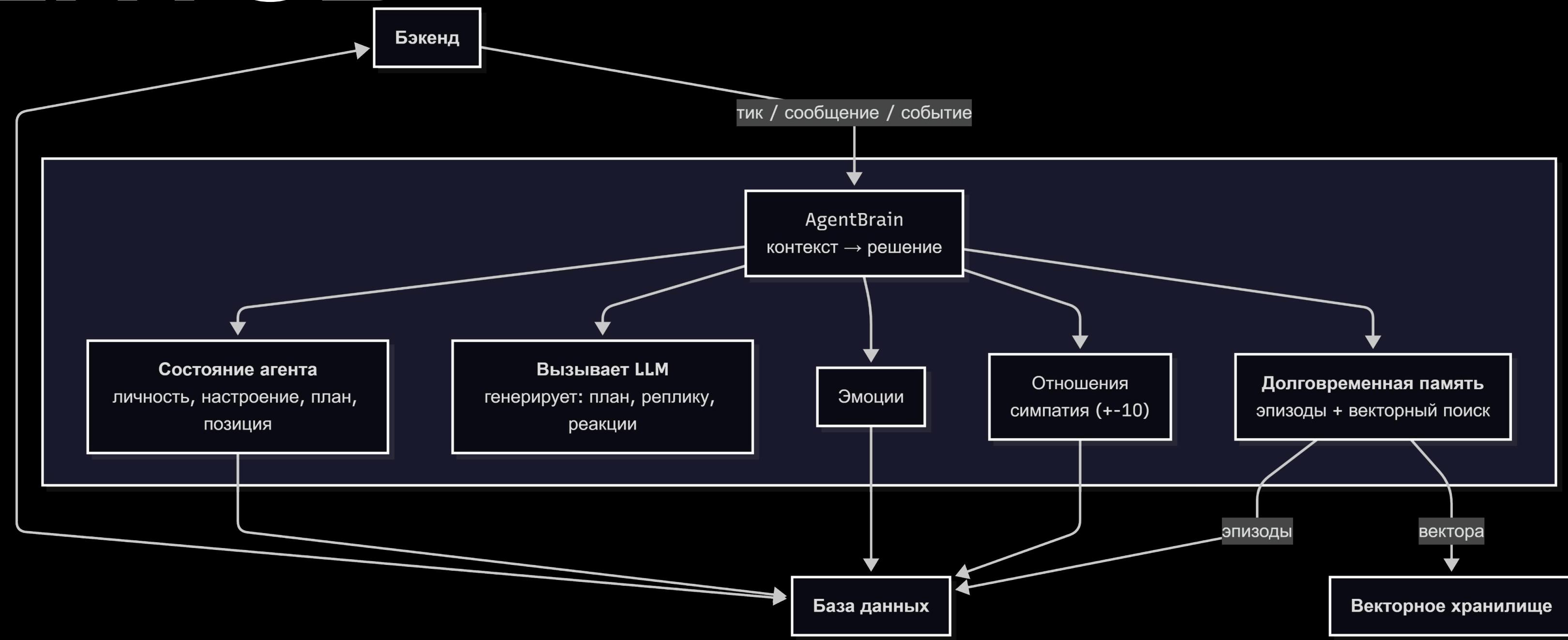


A screenshot of a file tree representing a modular monorepo. The structure includes:

- client
- dashboard
- identity
- next.config.ts
- shared
- characters
- images
- mobs
- ui
- button
- icons
- layout
- text
- theme
- .gitignore
- eslint.config.mjs
- package.json
- postcss.config.mjs
- README.md
- tsconfig.json
- yarn.lock

ДИЗАЙН АГЕНТОВ

1. Триггер: тик симуляции, входящее сообщение, событие мира
2. Загрузка состояния агента
3. Сбор контекста: events и weather
4. Загружается текущая связь с другими агентами
5. Воспоминания: делается запрос эмбеддингам и выбираются самые похожие
6. Строится план и вызывается LLM
7. Чистится мусор
8. Пересчитывается mood, корректируется sympathy
9. Результат записывается в БД
10. Состояние и действия агента транслируются через WebSocket-события



Особенности:

- 2 слоя памяти
- Суммаризация
- Эмоции влияют на стиль речи
- Отношения обновляются
- Автономность мультиагентов

БЭКЭНД

Разработан REST API

- Модульная архитектура
- Обработка ошибок
- Использованы паттерны для бд

Взаимодействие с фронтендом через WebSocket

- Обновление позиций персонажей с частотой ~60 FPS
- Возможность настройки скорости движения

документация



Особенности

- Асинхронность
- 80% кода покрыто тестами
- Оптимизирована производительность бд
- Обеспечена комплексная многоуровневая безопасность системы

СТЕК

Фронтенд

TS



Workspaces



Агенты

Gemini



Бэкэнд



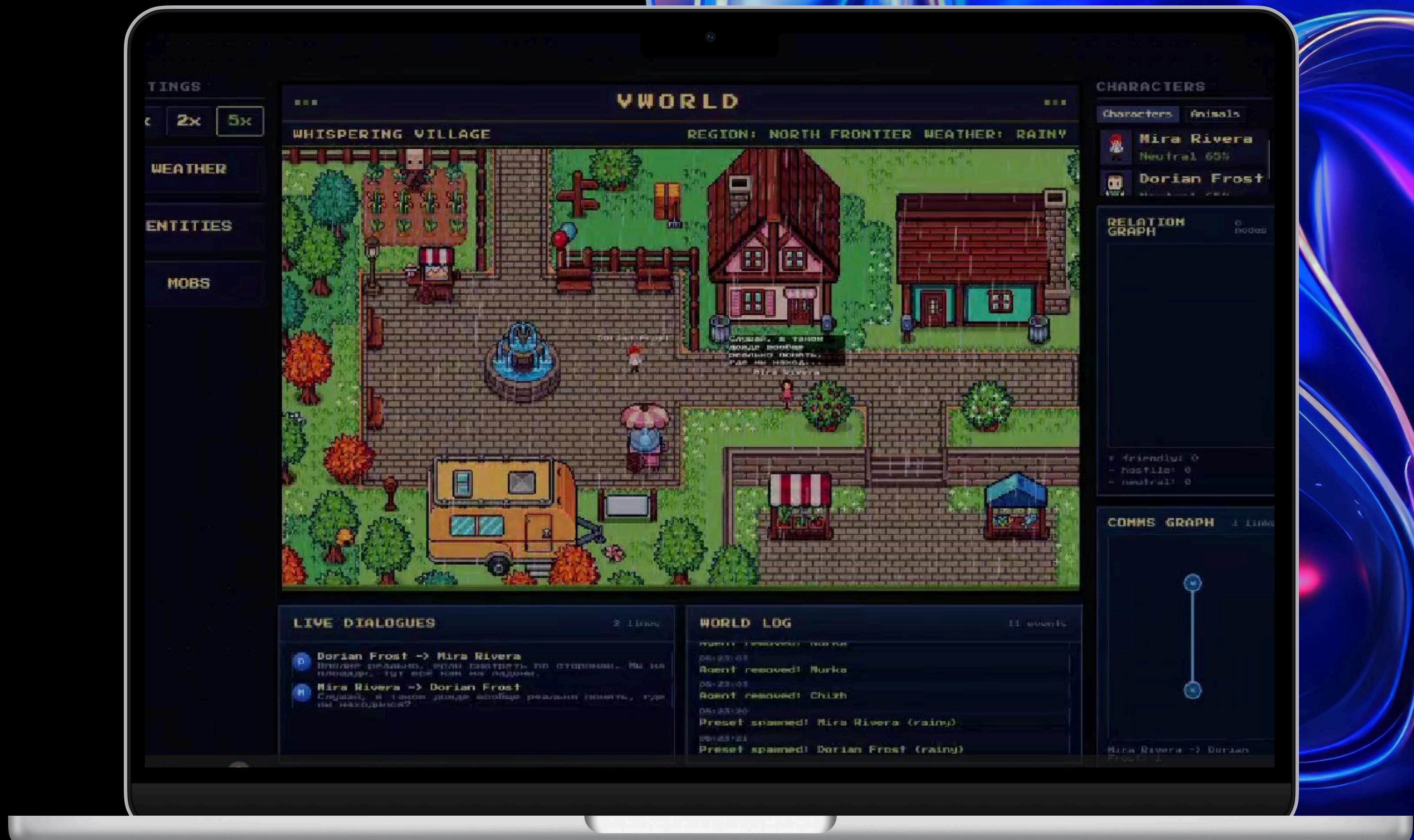
FastAPI



ДОРОЖНАЯ КАРТА



ДЕСКТОП-ПРОТОТИП

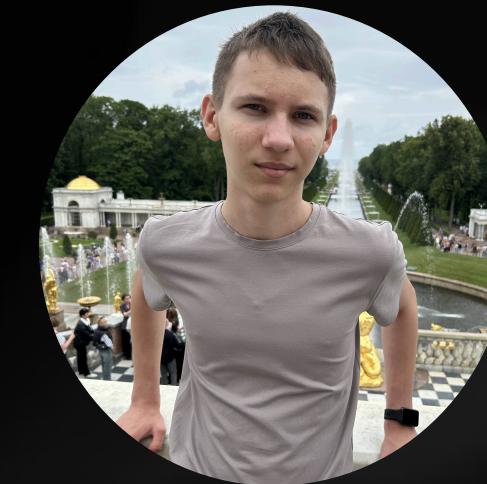


НАША КОМАНДА



Донат Авдеев

Technical Leader



Валерий Лысенко

ML Engineer



Толмаская Ирина

Backend Developer



Ли Александр

Frontend Developer

**СПАСИБО ЗА
ВНИМАНИЕ**

Гитхаб

