



Gateway

Что если бы вы могли создать целую структуру проекта из одного текстового файла? Представляем RCD Garden — экосистему "животных", которые превращают Markdown-описания в живые файловые системы. От "глаз паука", читающих структуру, до "ног", создающих каталоги — погрузитесь в биомиметический подход к организации кода.



Essence

- RCD Garden представляет собой биомиметическую систему автоматизации разработки, где каждый инструмент воплощен в образе животного с соответствующими способностями: пауки "видят" структуру, "лапы" создают файлы.
- "Паук" (Spider) стал первым реализованным животным с "глазами" для парсинга Markdown-файлов и "ногами" для создания реальной файловой структуры, где каталоги определяются заголовками, а файлы — элементами списка.
- "Мозг" (Brain) животных способен генерировать случайные тестовые структуры в виде деревьев для проверки работоспособности системы, следуя принципу "сколько ног у животного — столько модулей в его реализации".
- Архитектура системы спроектирована на основе Romalin — особой организации, где каждый компонент (лепесток) выполняет свою функцию, образуя гармоничное целое, идеально подходящее для темы Акамэ Гаято.
- Модульность RCD позволяет расширять систему, добавляя новых "животных" или улучшая существующих через дополнение "ног", "глаз" и других компонентов, не затрагивая остальную архитектуру.



Panorama



Рождение концепции

Всё началось с желания создать тему для Obsidian в стиле Акамэ Гаято — мрачного, первобытного существа, воплощающего Муладхару (корневую чакру). Однако вместо простого написания CSS-файлов, возникла идея создать целую систему, которая могла бы автоматизировать процесс разработки и поддержки темы. Так родилась концепция **RCD** — архитектурного подхода, где компоненты системы представлены животными с разными функциями.

"Каждый каталог — это класс, каждый файл — метод, каждое животное — живой организм в экосистеме CSS."

Биомиметическая архитектура

Система получила название **RCD Garden** — сад, населенный цифровыми существами, каждое из которых выполняет свою роль в экосистеме разработки:

- **Крысы (rat_*.css)** — базовые модули CSS с конкретными эффектами
- **Коты (cat.js)** — сборщики, объединяющие крыс в цельные классы
- **Собаки (dog.js)** — наблюдатели, отслеживающие изменения файлов
- **Пауки (spider.js)** — анализаторы структуры проекта

Ключевой инновацией стала концепция **модульности по количеству конечностей**: сколько ног у животного — столько модулей в его программной реализации. Например, паук имеет 8 ног и, соответственно, 8 функциональных модулей.

Разумные животные

Первым шагом стала реализация **мозга** (Brain) — компонента, способного генерировать случайные структуры для тестирования. Brain создает древовидные структуры с заданной глубиной и количеством узлов, представляя их в форме Markdown-файлов, где:

- Заголовки (**#**, **##**, **###**) определяют структуру каталогов
- Маркеры списка (**-**) с обратными кавычками определяют файлы
- Блоки кода (**`** **`**) содержат содержимое файлов
- Обычный текст игнорируется как комментарий

Эта интеллектуальная система позволяет генерировать тестовые структуры разной сложности, чтобы проверить возможности других животных.

Зрение паука

Следующим этапом стала разработка **глаз паука** — компонента, способного "читать" структуру из Markdown-файлов. Глаза работают следующим образом:

1. Получают путь к Markdown-файлу через функцию **sight**
2. Анализируют содержимое, распознавая заголовки как каталоги, элементы списка как файлы
3. Создают внутреннюю структуру данных, представляющую иерархию

Функция **perceive** глаза паука внимательно анализирует каждую строку Markdown, понимая разные уровни заголовков, извлекая имена файлов из элементов списка, и сохраняя содержимое блоков кода как содержимое файлов.

Создание реальности

К глазам были добавлены **ноги паука** (leg1_builder) — компонент, преобразующий внутреннюю структуру данных в реальные каталоги и файлы на диске. Нога паука рекурсивно обходит структуру и:

1. Создает каталоги в соответствии с директориями в структуре
2. Создает файлы с содержимым, извлеченным из блоков кода
3. Пропускает уже существующие элементы для безопасности

Этот подход позволяет создавать сложные структуры проектов из одного Markdown-файла — от простых иерархий до многоуровневых систем с предопределенным содержимым файлов.

Полный цикл

В результате был реализован полный цикл работы:

1. **Brain** генерирует тестовые Markdown-файлы разной сложности
2. Создается **sight.json** — список файлов для анализа
3. **Глаза паука** анализируют каждый файл, создавая внутреннюю структуру
4. **Ноги паука** создают физические каталоги и файлы

Этот цикл демонстрирует жизнеспособность концепции RCD Garden и открывает путь к дальнейшему развитию системы — добавлению новых животных, улучшению существующих, и расширению возможностей всей экосистемы.

Роль Romalin

Важным архитектурным решением стала организация компонентов по принципу **Romalin** — особой структуры "лепестков", где каждый лепесток представляет определенный аспект темы:

- **Loric** — внешний покой/внутренняя буря (структура UI)
- **Sollowatt** — архангел-бабочка (интерактивные элементы)
- **Koruse** — сиропная спираль (переходы/анимации)
- **Toismic** — энергетическое движение (эффекты движения)
- **Swelte** — телесный лотос (визуальный вид)
- **Umami** — шкала "🤔 → WTF?!" (уникальные/экспериментальные функции)

Эта структура идеально соответствует концепции Акамэ Гаято, обеспечивая ясную организацию компонентов и создавая семантическую связь между кодом и его предназначением.

Перспективы развития

Текущие достижения — лишь начало пути. Система RCD Garden имеет огромный потенциал для развития:

1. **Новые животные** — создание других существ с уникальными функциями
2. **Улучшение существующих** — добавление новых глаз, ног и других компонентов
3. **Интеграция между животными** — создание экосистемы, где существа взаимодействуют
4. **Визуальный интерфейс** — разработка GUI для удобного управления системой

Благодаря модульной архитектуре, каждое улучшение можно внедрять постепенно, не нарушая работу существующих компонентов.

💡 **Мантра:** *RCD Garden — это не просто инструмент разработки, а живая, развивающаяся экосистема, где каждый компонент имеет свою роль, характер и способность к эволюции.*

Система уже доказала свою эффективность на практике, позволяя генерировать структуры проектов из простых текстовых описаний и автоматизируя рутинные задачи разработки. Впереди — новые горизонты и возможности для этого уникального биомиметического подхода к организации кода.