



Gateway

Что получится, если CSS научится основам программирования? Мы создали революционную архитектуру, которая превращает папки в классы, а скрипты — в живые организмы. От простой темы для Obsidian до полноценного CSS-OOP фреймворка — история о том, как энтузиазм и любопытство привели к изобретению нового подхода к организации стилей. Готовы окунуться в мир крыс, котов и собак, которые автоматически собирают ваш CSS? Полный лог — ниже ↓



Essence

- **RCD Framework** — это CSS-архитектура, которая применяет принципы ООП к стилям: каждая папка становится классом, файлы — методами и полями, а автоматические скрипты — конструкторами и сборщиками.
 - **Три уровня иерархии** решают проблему масштабируемости: LEAF (листья) содержат базовые компоненты, MIDD (средний) объединяет блоки, BUFF (верхний) собирает финальный проект — всё происходит автоматически.
 - **Звери в коде** — это не шутка, а практическая метафора: "крысы" (rat_*.css) хранят модули, "коты" собирают их в классы, "собаки" следят за изменениями, а "паук" анализирует структуру — каждый выполняет свою роль в экосистеме.
 - **Революция в рабочем процессе**: одна команда собирает весь проект от глубоких листьев до корня, автоматически обновляя документацию и сохраняя порядок — никаких ручных копипастов.
 - **RCD-Library** — финальное решение для избавления от повторений: импортируй готовые компоненты вместо копирования кода, создавай новые проекты одной командой.
 - **От темы Obsidian до фреймворка**: то, что начиналось как стилизация редактора, превратилось в универсальную систему организации CSS с автоматической сборкой, версионированием и документированием.
-



Panorama



Рождение идеи: от темы к системе

История началась с простого желания создать красивую тему для Obsidian. Юлия, имея опыт веб-разработки, хотела реализовать тему с уникальными эффектами, вдохновленную концепцией "третий глаз + огонь". Но традиционный подход с монолитными CSS-файлами быстро показал свои ограничения: файлы росли, становились неуправляемыми, а изменения превращались в кошмар отладки.



Открытие: CSS нуждается в архитектуре

Попытка организовать стили привела к неожиданному открытию. Когда Юлия начала разделять код на модули и создавать скрипты для автоматизации, стало ясно: CSS нуждается в тех же архитектурных решениях, что и "нормальные" языки программирования. Так родилась идея **RCD Framework** — система, где каждый каталог становится классом, а структура проекта отражает объектно-ориентированные принципы.

Ключевые компоненты системы:

- **Крысы (rat_*.css)** — базовые модули с эффектами (как entropy, flux, spiral)
- **Коты (cat.js, yumia-cat.js)** — скрипты-сборщики, создающие классы из крыс
- **Собаки (dog.js, dog-midd.js)** — наблюдатели, автоматически запускающие сборку
- **Паук (spider.sh)** — анализатор структуры проекта



Три уровня абстракции

RCD использует трехуровневую иерархию, решающую проблему масштабируемости:

1. **LEAF (листья)** — базовый уровень с отдельными эффектами
2. **MIDD (средний)** — объединяет несколько LEAF в тематические блоки
3. **BUFF (верхний)** — создает финальную тему из MIDD компонентов

Каждый уровень имеет своих специализированных "животных": leaf-коты работают с крысами, midd-коты собирают leaf-конструкторы, а dog-big управляет всей иерархией рекурсивно.

Магия автоматизации

Ключевая особенность RCD — полная автоматизация. Добавляете новый rat_chaos.css? Собака автоматически обнаружит изменение, обновит конфигурацию (meow.json), запустит котов для пересборки. Весь проект от глубоких листьев до корня собирается одной командой, создавая:

- Финальный CSS-файл с четкой структурой
- Автоматическую документацию (Эвтюмии)
- Метаданные сборки для отслеживания версий



От решения к революции

То, что начиналось как организация CSS для темы Obsidian, превратилось в полноценный фреймворк. Система показала настолько эффективный подход к управлению стилями, что возникла идея создания **RCD-Library** — универсальной библиотеки компонентов.

RCD-Library: конец эры копипаста

Библиотека решает главную проблему разработчиков: необходимость копировать и адаптировать код для каждого нового проекта. Теперь:

```
// Вместо копирования сотен строк кода:  
const rcd = require('rcd-library');  
rcd.leaf.cat.run();
```

Команды новой эры

```
# Создать структуру проекта  
rcd init leaf
```

```
# Добавить новый эффект  
rcd rat storm
```

```
# Собрать весь проект  
rcd dog-big
```



Мантра RCD: *Каждый каталог — это класс, каждый файл — метод, каждое животное — живой организм в экосистеме CSS.*



Философия и будущее

RCD Framework демонстрирует, что CSS может быть столь же структурированным и автоматизированным, как любой другой язык программирования. Принципы ООП, паттерны проектирования, автоматическая сборка — всё это теперь доступно в мире стилей.

Эта система открывает новые возможности:

- Командная работа над стилями без конфликтов
- Версионирование и документирование CSS-кода
- Переиспользование компонентов между проектами
- Автоматическая оптимизация и сборка

RCD превращает процесс создания тем и стилей из рутинной работы в инженерную практику, где каждый компонент имеет своё место, цель и взаимодействует с другими по четким правилам.

Полный лог разработки, включая все итерации, эксперименты и технические детали, доступен в оригинальной беседе с Claude.