How I'm fighting the AI Brain Fry

NOVEMBER 8, 2026

If you work with Claude Code or Codex from the terminal, you know this one.

You start a task. The AI takes time to reply, so you start another one. Then the first one replies, but the reply needs one clarification, one follow up and one side task — and each of them wants its own tab. Then the second tab's reply comes in. You juggle. And all of a sudden you find yourself hopping between 20 threads, trying to keep all those contexts in your brain RAM.

Until you can't. Your brain gets exhausted, your focus smashes, and you don't know anymore what you were doing, where you started, and where you were supposed to finish.

A lot of people are suggesting taking breaks, thinking before typing, limiting scope. I have my own experiment.

I decided this way of working has two major problems: sequencing and context. So I organised my work a bit like a priority queue.

Here is how it actually works.

The queue lives in git. Every ticket is a markdown file in the repo, right next to the code it talks about, and the state of the board is a committed snapshot. Nothing important lives in a chat window and nothing lives in my head. If a session dies, the queue is still there.

An agent works that board in cycles. It researches new tickets, builds the ones a human has tagged as safe to build, runs the tests and a browser check, and verifies what is already deployed. For anything touching sensitive code it builds on a branch and asks first. It never deploys to production — that part is mine.

What it cannot do is decide. So everything that needs me lands in one place I call my owner inbox. A script reads the whole board and hands me a prioritised list of exactly the items waiting on my judgement — review this change, answer this question, rule on this tradeoff.

The reading surface is a Claude artifact. One page, rebuilt from git on every cycle, mirroring the whole board plus a review page per ticket. So I don't scroll back through a terminal trying to reconstruct what a change does. I open one link, read what was built and why, and give my verdict. The verdict goes back into git, onto the card. The page is only a viewer — git is the record.

Then comes the part that matters most.

I take one item. I load its context. I make a decision or request an action. And I close it, by putting it back into the queue.

That closing is the whole trick. The ticket doesn't keep hanging in my memory as a thing to remember about. The context closes with it. So when I pick the next item from the queue, I can actually focus. One thing at a time.

It is still exhausting. But I don't feel distracted or lost.

The challenge was, and still is, the tooling around it. Getting the context served up in a digestible manner, so loading a ticket takes a minute and not twenty. And keeping tickets from going into endless loops because they are blocked waiting for a ruling only I can give.

It's a living organism and I keep improving it. But it felt like a huge step forward in my productivity with AI tools, so I thought it was worth sharing.
