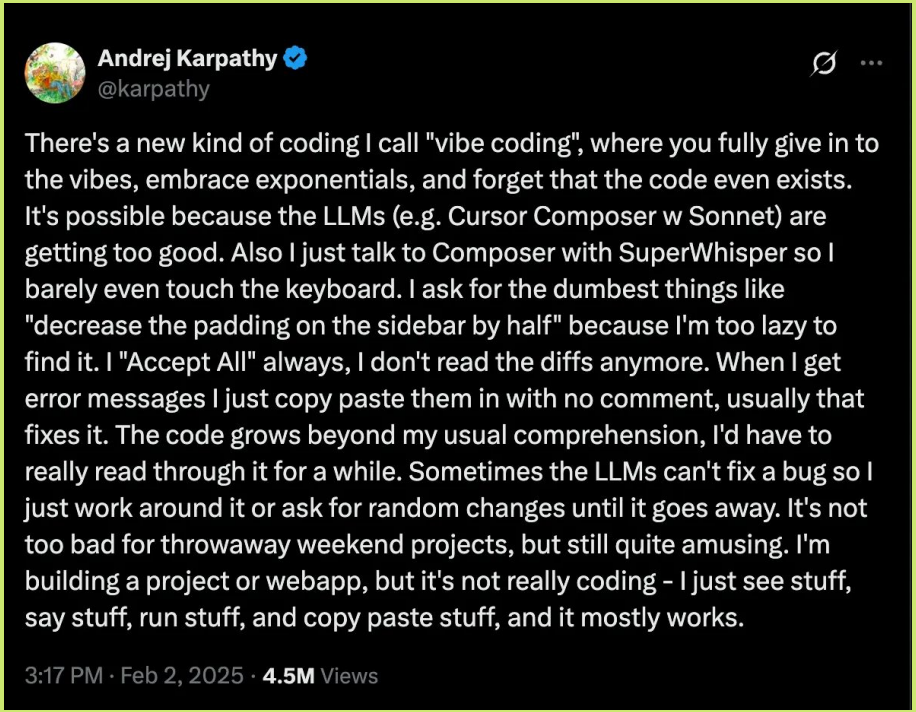
Vibe Coding

CCA Software Tutorial Series
April 15, 2025
Philip Mocz

What is Vibe Coding?



Andrej Karpathy
@karpathy

There's a new kind of coding I call "vibe coding", where you fully give in to the vibes, embrace exponentials, and forget that the code even exists. It's possible because the LLMs (e.g. Cursor Composer w Sonnet) are getting too good. Also I just talk to Composer with SuperWhisper so I barely even touch the keyboard. I ask for the dumbest things like "decrease the padding on the sidebar by half" because I'm too lazy to find it. I "Accept All" always, I don't read the diffs anymore. When I get error messages I just copy paste them in with no comment, usually that fixes it. The code grows beyond my usual comprehension, I'd have to really read through it for a while. Sometimes the LLMs can't fix a bug so I just work around it or ask for random changes until it goes away. It's not too bad for throwaway weekend projects, but still quite amusing. I'm building a project or webapp, but it's not really coding - I just see stuff, say stuff, run stuff, and copy paste stuff, and it mostly works.

3:17 PM · Feb 2, 2025 · **4.5M** Views

What can it do?

- Autocomplete code
- Answer your questions
- Explain API
- Create documentation
- Outline structure
- Add testing
- Debug code
- Lint code

What can I use?

VS Code + Copilot

Cursor AI

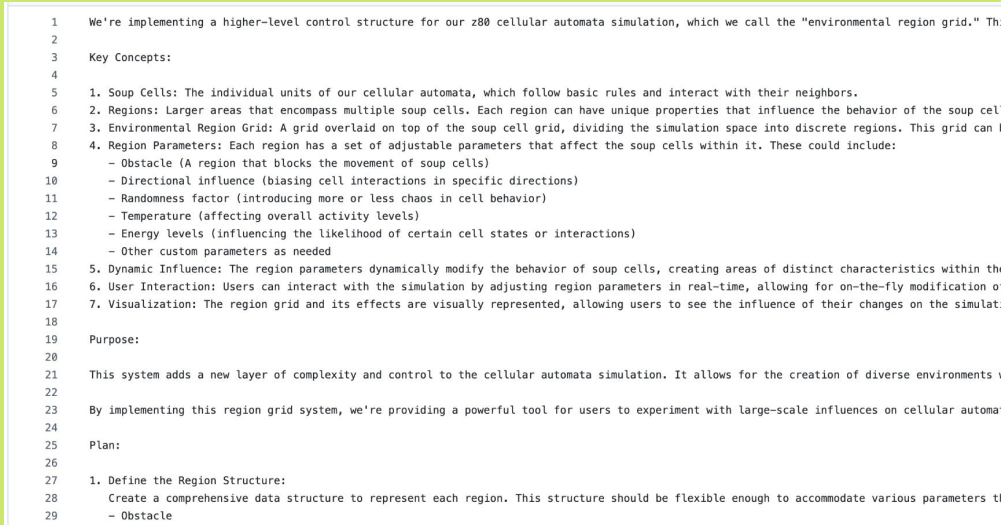
Google Colab

agents!

Tips & Tricks

- Be **Specific** (what libraries should it use? What are the specs? goals?)
- Set **Rules**
- Give **Oversight** (alignment)

e.g. a `.cursor/rules` file for coding, tech stack, workflow, and communication preferences



```
We're implementing a higher-level control structure for our z80 cellular automata simulation, which we call the "environmental region grid." Th

Key Concepts:

1. Soup Cells: The individual units of our cellular automata, which follow basic rules and interact with their neighbors.
2. Regions: Larger areas that encompass multiple soup cells. Each region can have unique properties that influence the behavior of the soup cel
3. Environmental Region Grid: A grid overlaid on top of the soup cell grid, dividing the simulation space into discrete regions. This grid can
4. Region Parameters: Each region has a set of adjustable parameters that affect the soup cells within it. These could include:
   - Obstacle (A region that blocks the movement of soup cells)
   - Directional influence (biasing cell interactions in specific directions)
   - Randomness factor (introducing more or less chaos in cell behavior)
   - Temperature (affecting overall activity levels)
   - Energy levels (influencing the likelihood of certain cell states or interactions)
   - Other custom parameters as needed
5. Dynamic Influence: The region parameters dynamically modify the behavior of soup cells, creating areas of distinct characteristics within the
6. User Interaction: Users can interact with the simulation by adjusting region parameters in real-time, allowing for on-the-fly modification o
7. Visualization: The region grid and its effects are visually represented, allowing users to see the influence of their changes on the simulat

Purpose:

This system adds a new layer of complexity and control to the cellular automata simulation. It allows for the creation of diverse environments

By implementing this region grid system, we're providing a powerful tool for users to experiment with large-scale influences on cellular automa

Plan:

1. Define the Region Structure:
   Create a comprehensive data structure to represent each region. This structure should be flexible enough to accommodate various parameters t
   - Obstacle
```

Pros & Cons

- Gives you a superpower

- Don’t take its answers for granted (it can make mistakes or lose context)


LET’S
GO