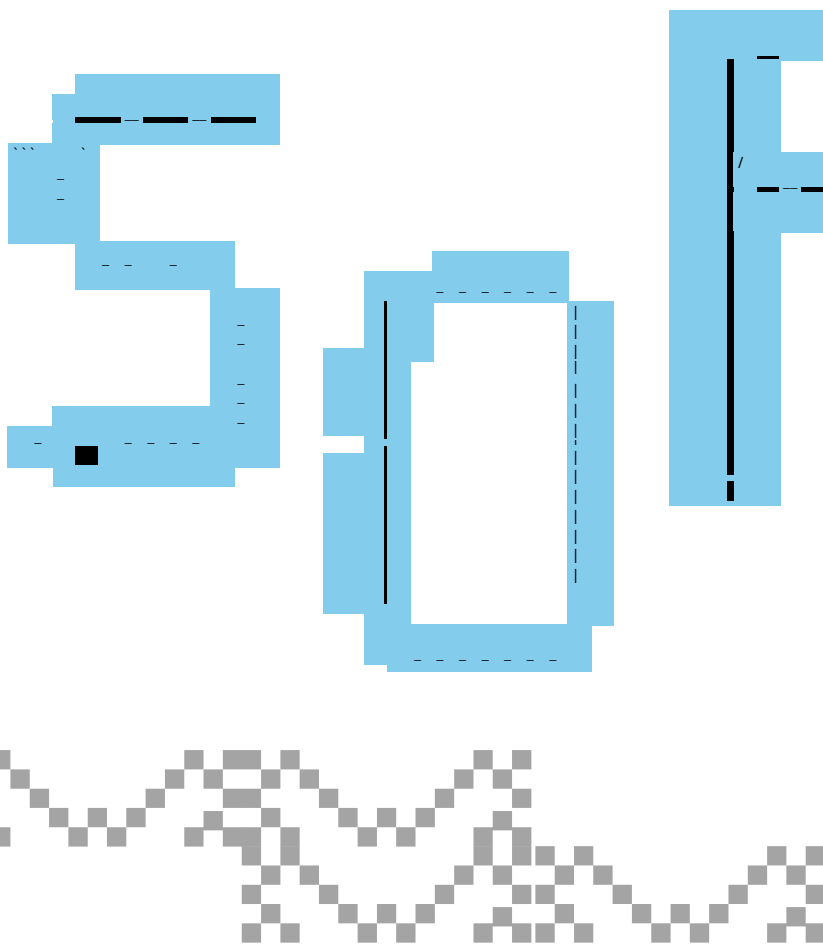


Chapter 2

Softer computing is a design attitude and practice that values friction, care, and participatory agency over seamlessness, automation, and invisible control. Unlike open source or DIY, which often focus on technical transparency or independence, softer computing is fundamentally about relational engagement between maker, tool, and context.

INTRODUCING SOFTER COMPUTING



Softer computing is not a system, nor is it a fixed set of techniques. Rather, it is an attitude - a critical, relational approach to design and technology. At its core, softer computing resists the dominant logics of efficiency, flattening, and automation that underpin today's digital landscape. It proposes an alternative value system: one that prioritizes friction, transparency, constraint, modularity, and the concept of "*slow time*" over seamlessness, opacity, and relentless acceleration.

COMPUTING

softer

၈၈

chapter 2



WHY NOW? SOFTER COMPUTING AS A RESPONSE TO DIGITAL MAXIMALISM

William Powers, *Hamlet's Black-Berry*, 2011

The need for a softer approach has never felt more urgent. The rapid proliferation of generative AI, the dominance of algorithmic curation, and the “*digital maximalism*” that Powers (2011) critiques have together created a climate of design overload. The culture of tech, shaped by capitalist imperatives of constant growth and efficiency, increasingly demands that designers and users alike submit to automated workflows, default templates, and invisible infrastructures. In response, a growing movement for “*degrowth computing*” has emerged, seeking to decouple digital technology from the “growth-focused imperatives of capitalist societies” (Neil, 2022).

Neil, *What might degrowth computing look like?*, 2022

For me, the very idea of “softer” computing comes from reflecting on the language of technology itself.

HARDWARE

“Hardware” describes the physical machinery—

SOFTWARE

“Software,” by contrast, promises malleability:

it is “soft” because it

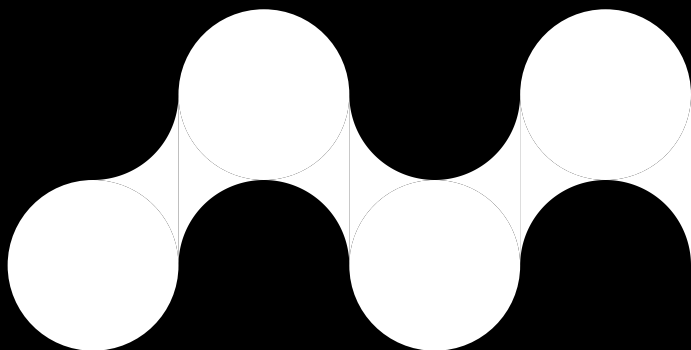
Yet this is a misleading comfort. In practice, contemporary software has become so complex, so deeply layered with proprietary code and abstracted interfaces, that even developers find it difficult to intervene. Far from soft, today's software is often rigid, inaccessible, and resistant to meaningful modification.

As a result, the notion that digital environments are open to intervention by ordinary users has become largely illusory. The smooth, seamless structures of modern interfaces conceal the underlying code, hiding opportunities for agency and making it easy to mistake constraint for choice. The GUI (a "graphical user interface") may feel approachable, but it often functions as a form of abstraction that distances us from the real workings of technology (Emerson, 2014).

Emerson, Lori, *Reading Writing Interfaces: From the Digital to the Bookbound*, 2014



HARDWARE



Wikipedia

Computer hardware includes the physical parts of a computer, such as the central processing unit (CPU), random-access memory (RAM), motherboard, computer data storage, graphics card, sound card, and computer case. It includes external devices such as a monitor, mouse, keyboard, and speakers.

SOFTWARE

Wikipedia

Software consists of computer programs that instruct the execution of a computer. Software also includes design documents and specifications.

Software can generally be categorized into two main types:

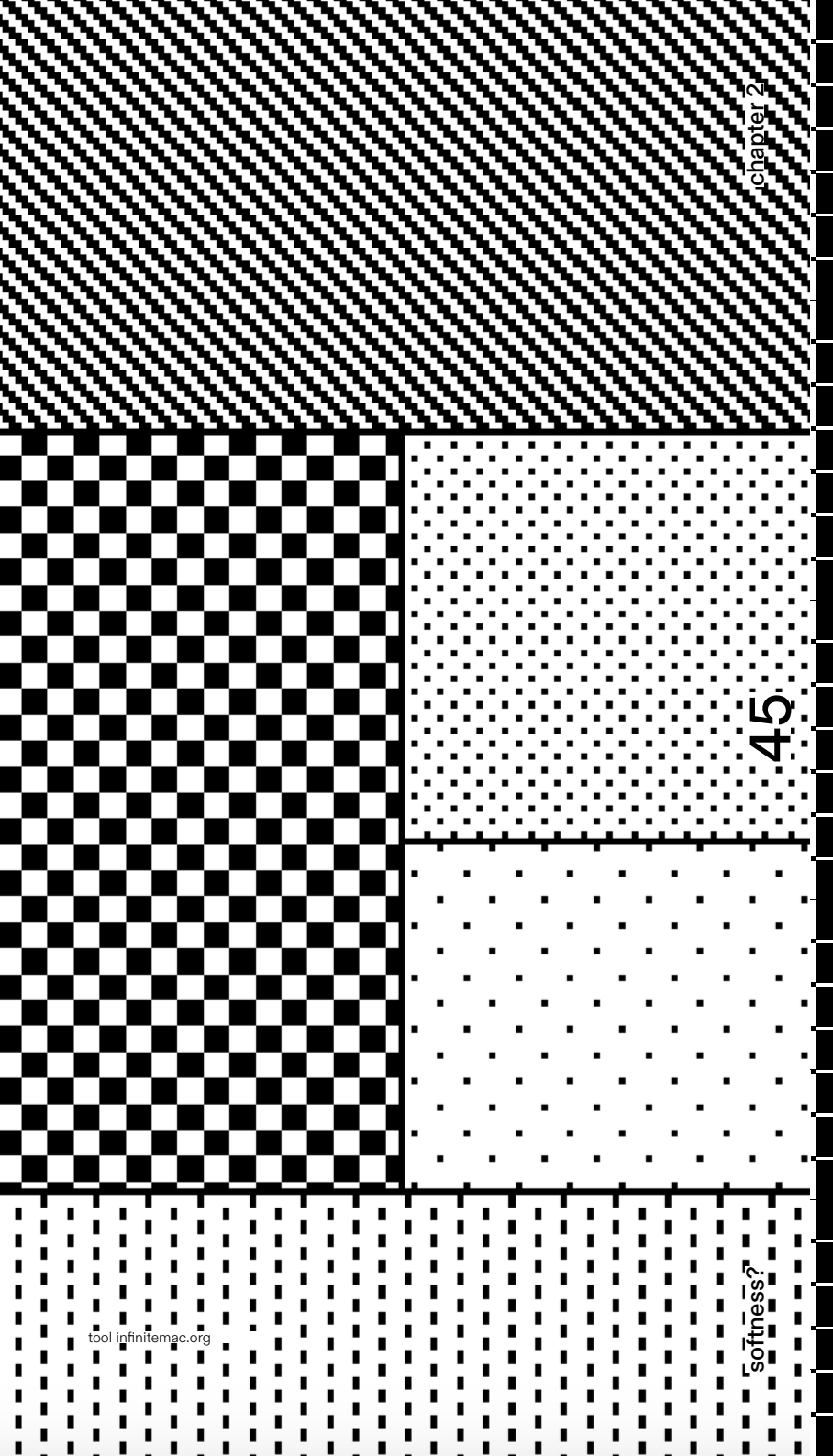
operating systems, which manage hardware resources and provide
1. services for applications 2. application software, which performs specific tasks for users

THE DECEPTIVE PROMISE OF “SOFTNESS”

The characterization of software as “soft” obscures the reality of contemporary computing. Most people today interact with technology through platforms designed for immediate clarity and efficiency.

We are discouraged from “wasting time” on things that do not offer instant, frictionless communication. This dynamic is especially pronounced in the design of websites, which have become so streamlined and standardized that the development process itself mirrors software production: modular, automated, and largely uncustomizable for the average user. The web is no longer “soft”; it has become, as Zhang (2019) writes, a built environment and a social space that is monopolized, commodified, and colonized by corporate interests.

Gary Zhexi Zhang, *Reenvisioning the Internet: Create Tools that Reveal its Ideological Infrastructures*, 2019



VERNACULAR WEBS AND LOST FREEDOM

Olia Lialina, Turing Complete
User, 2015

Looking back, the amateur web of the mid-1990s, as described by Olia Lialina (2015), offers a powerful contrast to the present. This was a time when the web was unregulated, personal, and always “under construction.” The web, in Lialina’s terms, was communal: every page was an act of care, each contribution a unique addition to a shared space. Lialina calls this the “vernacular web,” a term that may seem strange now but precisely captures the ethos of the era—a web made by its users, not just for them.

GeoCities, founded in 1994, stands as a testament to this spirit of digital amateurism. At its peak, it hosted 38 million personal websites - spaces for self-representation and connection, unpolished but fiercely individual. While GeoCities is now gone, this ethos survives in platforms like Neocities, which archives and extends the legacy of amateur web-building. In just the last few years, Neocities has grown to host nearly a million sites, providing free space for people to make the web their own once again.

Simple Net Art



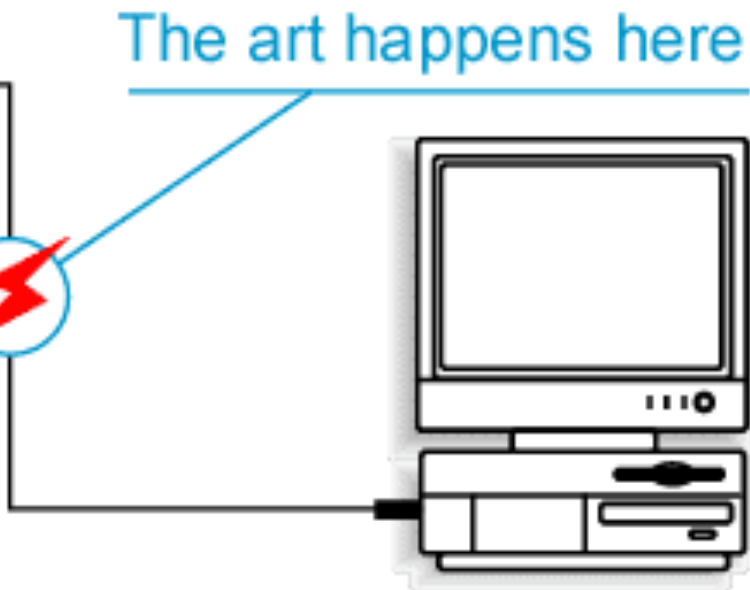
MTAA ca. 1997

MTAA, Simple Net Art Diagram

“Art as an active site, open to the
Information as a stance, opposing
Participation as the norm, including
Time as a fluid state, now to be

Art Diagram

chapter 2



gram, ca. 1997. Animated GIF.

49

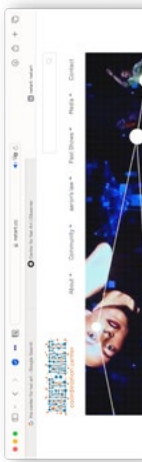
those who care
tion in the air.
sion across the board.
ignored”

—MTAA

vernacular web

The broader culture of net art, community coding, and decentralization continues to thrive in projects like Rhizome, the Center for Net Art, and the School for Poetic Computation, as well as decentralization initiatives such as DWeb and the Internet Archive. These efforts work to actualize a more distributed, collectively-owned internet, one that foregrounds communal stewardship, creative experimentation, and the possibility of self-determination.

the Center for Net Art



Rhizome





the School for Poetic Computation



SO WHAT IS SOFTER COMPUTING?

To me, “*softer computing*” signals a holistic new approach to technology, one grounded in care, criticality, and the ongoing struggle to reclaim agency over the tools that shape our world. It means acknowledging the power structures embedded in our technological systems, and refusing to accept their values as neutral or inevitable. Mindy Seu (2022) poses the dilemma starkly:

‘DO WE SHAPE THE FUTURE BY CREATING THE TOOLS AND TECHNOLOGIES NECESSARY TO ACHIEVE OUR GOALS, OR DO THE TECHNOLOGIES CREATED AND IMPOSED ON US BY TECH GIANTS LIKE META BUILD THE FUTURE ON OUR BEHALF?’

The practice of *softer computing*, as I define it, sides firmly with the former. It is a call to take part in the making of futures, not simply to inherit those imposed by others.

Softer computing is not a rejection of technology but a refusal of its commodification and instrumentalization, it's a resistance to computational maximalism. It is about creating space for friction, ambiguity, and the "slow time" needed for genuine reflection. The mainstream tech industry's endless pursuit of frictionlessness, efficiency, and hyper-automation leaves little room for difference or dissent. Softer computing intentionally interrupts this logic, insisting on the value of discomfort, uncertainty, and the human hand.

Ursula K. Le Guin's *Carrier Bag Theory of Fiction* (1986) inspires my understanding of this approach. She argues that technology, rather than a tool of conquest or domination, can be reimagined as a "*cultural vessel*", a container for stories, relationships, and possibilities. Similarly, Luna Maurer (n.d.) calls for "*friction*" in digital design, arguing that meaningful connection emerges from encountering, rather than erasing, obstacles. Maurer observes that "*transactions with machines are now smoother than interactions with fellow humans*," a fact that reveals how the pursuit of seamlessness has become a form of social alienation.



Ursula K. Le Guin's *Carrier Bag Theory of Fiction*, 1986
Luna Maurer, *Designing Friction*, n. d.

Miriam Rasch, Friction, 2020

Luna Maurer, Designing Friction,
n. d.

Friction, then, is not merely an inconvenience - it is a necessary condition for agency, awareness, and connection. Philosopher Miriam Rasch (2020) contends that the removal of friction produces passivity, leaving users “absorbing” technology rather than questioning or reshaping it. Maurer echoes this: “*Friction perceived as an obstacle might, in fact, be a possibility for connection.*” In this way, designing with and for friction means exploring the boundaries of our tools, inviting creative error and imperfection, and pushing against the uniformity imposed by platforms.

Softer computing is fundamentally about computing with limits - embracing constraint, transparency, and modularity as positive values. It resists the fantasy of endless technological expansion and instead prioritizes care, attention, and situated knowledge. This perspective resonates with the “Computing within Limits” research community, which advocates for ecological awareness and ethical responsibility in digital development.

Spencer Chang, essay We're All
(Folk) Programmers, 2024

Practically, this stance transforms how we approach both design and code. I see programming as a radical act - akin to “*writing a story*” (Chang, 2024) - not because it is about building entirely new systems, but because it makes space for small-scale invention and participation. Softer computing supports the emergence of alternative tools, platforms, and methods that foreground collective creation and reframe what “human-centered” design should mean. For me, a true human-centered practice is not merely anthropocentric; it is a form of care that recognizes the interdependence of people, communities, and environments.

Importantly, embracing softness does not mean designing technology to be intentionally flawed or less usable. Rather, it means rethinking what perfection even is, leaving room for serendipity, error, and user agency. It is about decentralizing technology so that even small actions like tinkering, hacking, remaking matter in the larger discourse. In the hands of practitioners, this philosophy is realized in communities like the School for Poetic Computation, or in projects like 'Computing within Limits', where technical and ecological boundaries are sources of innovation rather than constraints to be eliminated. Franklin (1999) argued,

In a moment when digital infrastructures are increasingly proprietary, predatory, and insular, the softer approach seeks out poetic and defiant forms of making. Softer computing, I believe, has the potential to reshape graphic communication design not only by decentralizing and diversifying existing structures but by radically reimagining human-computer interaction as a field of autonomy, creativity, and alternative futures.

While many creative communities embrace DIY or open-source approaches, softer computing is distinct in its embrace of *imperfection, smallness, and slowness*. It resists the logic that everything must scale, be universally accessible, or optimized for productivity at all costs. However, this approach has limits: too much friction can hinder access, and too much smallness can isolate. Softer computing aims to balance these tensions by remaining open to re-evaluation and critique.

“Technology is not the sum of the artifacts, of the wheels and gears, of the rails and electronic transmitters.

Technology is a system... Technology involves organization, procedures, symbols, new words, equations, and, most of all, a mindset.”