Winter is Coding: On Programming & The Freeze Response

Michael Nagle
Researcher
Coda
nagle@coda.io

1. Abstract

Much of the pioneering work in the first UI designs for a programmable computer were grounded in cognitive science. This paper takes the jumping off point that nervous system science can offer new insights into programmable UI design now in the way that cognitive science did 40 years ago. Specifically, we can look at "mathphobia" as a conditioned freeze response to symbolic reasoning. Polyvagal theory suggests orientation to the environment and social connection as two ways to help "thaw" the freeze response. We explore two examples of what that would mean from a UI and UX perspective here.

Figure 1 - Dany knows it deep in her bones: no one can fuck with her now that she's gotten timezone support working in her React app. Her fight response carried the day through her freeze.
(c) HBO's Game of Thrones.

2. An introduction

Figure 2 - This cartoon dog demonstrates the Freeze Response, esp. in the left hand panel. (c) KC Green.

In this essay I look at what recent understanding in neurobiology can tell us about our individual and collective experiences with programming.

What interests me is how programming can become a civic, moral, and creative force in this time of incredible change.

My specific interest is programmable UIs: UIs which facilitate the creation of programs, not just their use.

The basic question is: what keeps programming behind a moat of inaccessibility?

It interests me that despite the immense forces —

- geopolitical: Facebook reports privacy breach after breach and seems to have been pivotal in swinging the last US election
- financial: Uber becomes worth $40 billion in a decade of work

— that we don't see *masses* of people writing their own software.

Here I explore the freeze response — conditioned collectively through schooling and math education— that keeps us from creatively engaging with and iterating on programming, despite programming hugely impacting all of our modern lives.

3. From cognitive science to biology and physiology.

EXPERIMENTAL PERSONAL COMPUTER was built at the Xerox Palo Alto Research Center in part to develop a high-level programming language that would enable nonexperts to write sophisticated programs. The author and his colleagues were also interested in using the experimental computer to study the effects of personal computing on learning. The machine is completely self-contained, consisting of a keyboard, a pointing device, a high-resolution picture display and a sound system, all connected to a small processing unit and a removable disk-file memory. Display can present thousands of characters approaching the quality of those in printed material.

Figure 3 - One source of inspiration for the work on the Alto was Jerome Bruner's cognitive science work on the three human mentalities: enactive, iconic, and symbolic.

I was recently compelled to go back to some of Alan Kay's writings on developing the UIs that inspired modern windowed operating systems after a design review. I knew deep features about human cognition informed the UI design of the original personal computer, the Alto, and not the in vogue but shown to be false claim that "everyone has their own learning styles." Thinking about this, I wanted to read it from the horse's keyboard.

In "User Interface: A Personal View," Kay (1989) writes:

"Therefore, let me argue that the actual dawn of user interface design first happened when computer designers finally noticed, not just that end users had functioning minds, but that a better understanding of how those minds worked would completely shift the paradigm of interaction." (p. 123)

He goes on to expound how looking at three mentalities of humans —

- doing (enactive)
- seeing (iconic)
- and symbolic (...symbolic)

— needed to be leveraged simultaneously to make a powerful and natural UI. Hence: "**doing with images makes symbols.**"

After re-reading this, a new question hit me:

As cognitive science informed designs of the 70s, how could nervous system science inform designs now?

4. Co-regulating with computers

Figure 4 - As human's nervous systems co-regulate each other, how do computers regulate us? (c) Bjork, All Is Full of Love

In What Technology Wants, Kevin Kelly (2010) imagines technology to be a sixth kingdom of life. It has its own wants and needs, its own imperatives. Hence the title: examining what technology as a life form and force in its own right wants.

We can imagining compute as something which we have a living relationship with, just like being in relationship with other people.

Young children learn self-regulation through **co-regulation**: close proximity, attunement, and caregiving with and from a loved one (Calkins 2010.) Not yet able to manage their own nervous systems, touch, eye contact, and call and response language are ways for young children to learn to ride the waves of their own inner world. To get excited, calm back down, and feel loved and safe all the while.

Computers and smartphones wire in very deeply to us — shifting our habits unconsciously.

How are we co-regulated by computers?

What kind of attunement do we need from them?

5. The Nervous System as a store of biological history

Figure 5 - Unlike Bruner's mentalities, which have clear boundaries from one another, our nervous system stores our history and conditioning, building up in layers over time from birth to present.

In The Body Keeps The Score, Bessel Van der Kolk (2014) outlines the recent work done to understand how trauma, big and small, is remembered by the body. While cognition goes through distinct stages of development, our nervous system accumulates history like a painting accumulates layers of paint.

The layers are formed by experiences that we couldn't integrate at the time, both overwhelming (traumatic) and ordinary, and the nervous system stores emotion and motion (movements) still needing completion.

Van der Kolk cites Stephen Porges' polyvagal theory to understand the nervous system. Unlike Bruner's cognitive mentalities — the doing, seeing, symbolic thinking triad — which can co-exist, Porges (2001) argues that major nervous system states are exclusive. His argument is that when different branches of the vagus nerve are innervated, readying one of three different motor strategies for the body. We start, most recently in terms of evolution, ready for emotion, motion, and communication — social attunement. We then fall back under threat to a system ready to run, jump, and be in conflict — the well known "fight or flight." And then we fall back to an even older system, evolutionarily speaking, ready for immobilization and conserving metabolic resources — the freeze response (seen in animals playing dead.) (p. 8)

These strategies have an exclusive quality to them. We're in one state, one motor strategy, or another.

6. Papert's "Mathphobia" as Freeze

Figure 6 - Freeze is most known in animals playing dead, but also evokes qualities of the cold — a numbing or checking out from present reality.

So: how does symbolic reasoning elicit the freeze response?

Seymour Papert (1980) one of the foundational thinkers on how computers can be sandboxes for children to learn and create with, documents the phenomenon of mathphobia. From Mindstorms:

"Children begin their lives as eager and competent learners. They have to *learn* to have trouble with learning in general and mathematics in particular." (p. 40)

Or from John Holt (1982), arguably the father of American homeschooling, on learning and fear in the 1970s American classroom:

"For many years I have been asking myself why intelligent children act unintelligently at school. The simple answer is, "Because they're scared." I used to suspect that children's defeatism had something to do with their bad work in school, but I thought I could clear it away with hearty cries of "Onward! You can do it!" What I now see for the first time is the mechanism by which fear destroys intelligence, the way it affects a child's whole way of looking at, thinking about, and dealing with life. So we have two problems, not one: to stop children from being afraid, and then to break them of the bad thinking habits into which their fears have driven them." (p. 92)

How does this fear affect us over the course of our lifetime? How do we move out of it?

7. The first way out of freeze: orienting.

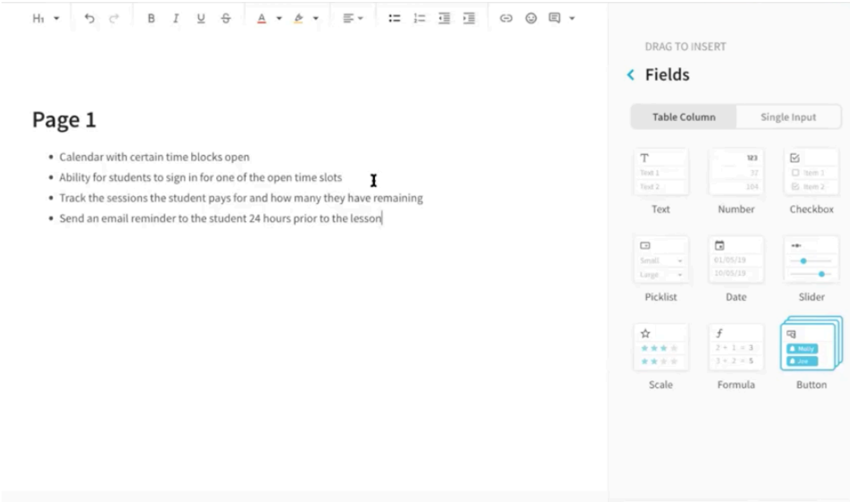

Figure 7 - Coda's Parts Bin (right hand panel) allows user to orient to the computational environment, giving them a way to see and feel the building blocks available tot hem.

If the nervous system accumulates like layers of paint, and we cope with our fear in school by mild, persistent freeze, then how do we thaw our ability to think symbolically? And how could a computer help us regulate through that freeze?

A first step is orientation. Orientation is a connection to the environment through our five senses. One way computers can help is by orienting us to the environment at hand. In programmable UIs, parts bins are one way to help a new user orient. A parts bin visually lays out a map of concepts, building blocks, and key relationships and hierarchies in an environment. Scratch's color-coded blocks based palette is perhaps the most famous example.

At Coda, we implemented a Parts Bin as a prototype. Coda itself is a document surface with computational building blocks: tabular data with a query language, and ways to trigger actions in those tables as well as externally (for example: automatically send a Slack message to the person whose turn it is to be on-call for customer support each workday.) The goal is to support businesses in building applications that match their unique workflows.

One of our big challenges as a team is to reach an audience not yet used to formal, symbolic, "technical" thinking. The word "technical" was mirrored back to us in initial interviewing — it's a great tool, our best users said, but sometimes it's too technical to share.

The Parts Bin prototype lays out the building blocks available in a drag and drop interface. We saw users reach for familiar patterns — checkboxes to start lists, and as in Figure 7, calendars to start making appointment booking software. The user interviewed above was, using the Parts Bin, able to successfully understand Coda's paradigm of a calendar being a presentation view of a table. Presentation views being driven by tables are normally one of the key conceptual hurdles in Coda — the UI which allows users to start from visually orienting, exploring, and finding the pieces which speak to their job at hand (making an appointment booking system) show in this early study show a lot of promise to extend capability to a "non-technical" audience.

8. The second way out of freeze: social connection

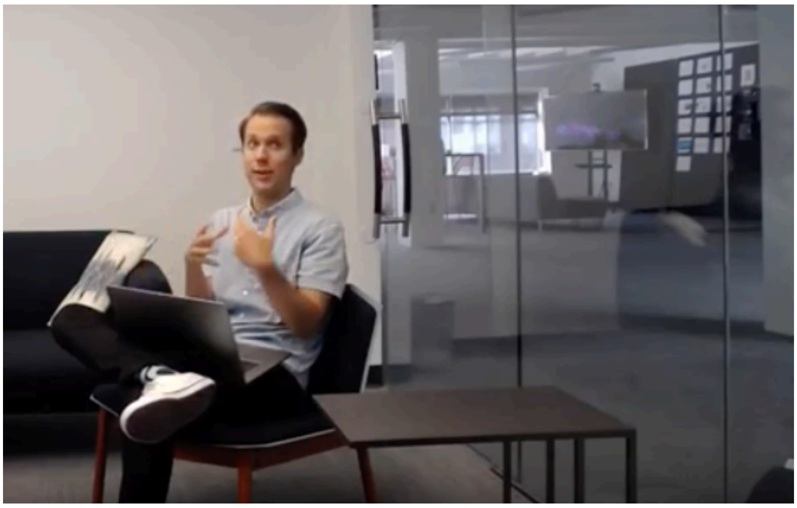

Figure 8 - Justin, a Codan, leading a video chat human on boarding session. We offer 45m sessions to users over video chat where we walk through examples and help a user get started making their business app. It's been one of our most effective onboarding strategies.

Porges' also writes about the social attunement system. That when we are connected to other people, we come out of our alarm physiology and back into the here and now. This is seen as the most recent evolution of our nervous system, the most advanced strategy we have for coping with threat being simple social connection and banding together.

We see this in the wild again with Scratch — the home page being a community portal — and other sites like DIY.org and Glitch. At Coda one of our initiatives around social connection ha s been what we've called "Human Onboarding." It is simply a way for any new user, building out their business applications in Coda, to setup a 45m video session with us. In it one of us — usually Justin shown above — will walk through relevant application templates and help the user get started building their own business application.

The rates of these users — getting started through a live video session — continuing on with making their custom business apps has one of the highest we've seen. To me it shows that looking at helping people start with programming can be a question of creative UI, but it can also be a question of how to design simple human connection into the user experience.

9. Languages that match the task at hand

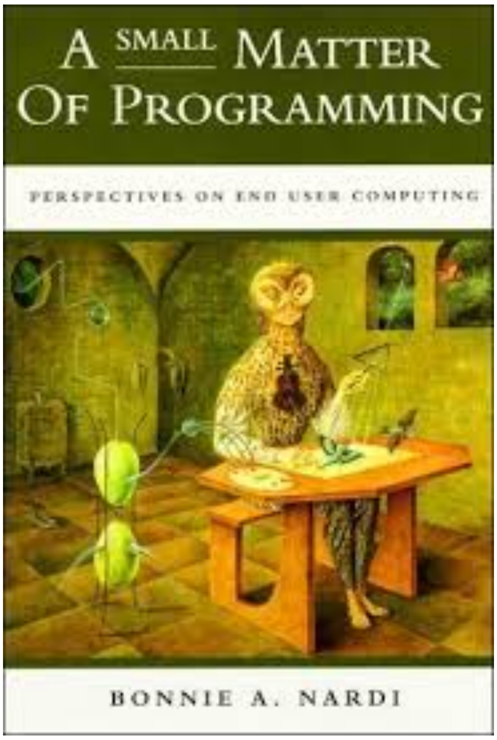

Figure 9 - Cover of Nardi's A Small Matter of Programming.

In A Small Matter of Programming, Bonnie Nardi (1993) documents cases where symbolic languages have evolved in folk and computational contexts. Knitting and baseball scorekeeping are in the wild examples of domain specific languages, and spreadsheets and CAD environments as computational examples which have had widespread end-user adoption.

What Nardi argues for in Small Matter is leaning into symbolic languages. That we see them in our culture appearing in the wild and that humans are very capable of learning domain-specific languages.

To her, an essential point is that the language is designed to the task at hand. She introduces the notion of a language being too "low-level" — forcing the user into the implementation details of the language.

At Coda this has given us a heuristic — while we occasionally consider a purely visual language, that it is fine, and even helpful, to have a text-based symbolic language at the heart of things (even if more visual UI is layered on top of that language.) But the "low-level" test offers us a check in the user experience: did the imagined user have to just learn a few new primitives to translate their workflow into Coda, or did they have to leave that paradigm and think about things specific to Coda? Said more directly, a recruiter does have to think about their flow of managing and scheduling candidates, but they should never have to think about whether a returned object is a single element or an array.

10. Summary

I hypothesize that most people who learn how to use a symbolic system have to contend with a history of freeze, conditioned by their experience in math classes in schooling.

Orientation to the environment and social connection are two ways of moving through this freeze.

This opens a larger question:

What else can biology, physiology, and the new sciences of the nervous system and trauma tell us about what we need from our computers to have access to our full selves, including our symbolic thinking?

How can better UI and UX thaw the freeze around symbolic thinking, and help evolve computing to the dynamic, creative, liberating medium it was meant to be?

11. References

Holt, John. (1982) **How Children Fail.** Da Capo Lifelong Books.

Kay, Alan. (1989) **User Interface: A Personal View**. Multimedia: from Wagner to Virtual Reality. pp. 121 - 131.

Kelly, Kevin. (2011) **What Technology Wants.** Penguin Books.

Nardi, Bonnie. (1993) **A Small Matter of Programing.** MIT Press.

Papert, Seymour. (1980) **Mindstorms.** Basic Books.

Porges, SW. **The polyvagal theory: phylogenetic substrates of a social nervous system.** International Journal of Psychophysiology 42 (2). pp. 123-146

van der Kolk, Bessel. (2014) **The Body Keeps the Score.** Penguin Books.