Prathyush Pramod

Software designer, developer and writer working in Prabros

Software design, learning, and a shared studio

Prathyush Pramod is a software designer and developer who works with his brother Pranav Pramod in Prabros, a two-person studio. Their public portfolio includes product interfaces, web development, icons, and animation for clients, alongside experimental tools and research publications. Prathyush also writes and curates material about programming, mathematics, and ways of representing knowledge under the public handle @prathyvsh. PatternAtlas is one home for those explanations; Prabros is the shared professional practice.[1, 2, 3]

He studied at Newcastle University, according to an identity-linked public self-profile, although his degree and attendance years remain unverified. By 2014, the brothers were publishing collaborative design work while he was undertaking sustained programming study. Their chat-interface essay was receiving overseas coverage in 2016. In 2019–20, his public record broadens into computational history, logic, educational projects, and a Malayalam technology-publishing initiative. PatternAtlas' s sustained mathematical explanations follow in 2021–22; later studio work includes interface experiments and a spreadsheet research collaboration with Nilenso.[4, 5, 6, 7, 8, 9]

His working life is therefore more concrete than its sometimes unfamiliar project titles suggest. It includes making screens and prototypes, studying books, recording unresolved questions, preparing demonstrations, responding to a client' s problem, and exchanging references with his brother and other practitioners. His own description of moving from interface design through development toward mathematics helps explain the growing research archive. The studio work continued alongside that learning.[10, 11, 12]

The public sources are unusually detailed about this work and much less complete about childhood, university experience, residence, and private finances. Many Prabros pages also have a collective voice without an individual byline. The account follows the available life and work record while preserving those distinctions: Prathyush' s individually credited explanations, Pranav' s design presentations, the brothers' shared publications, and other people' s systems collected in their research.

Education, early design work, and a visible apprenticeship, 2014–2015

Newcastle belongs in the account of his formation. A public Quora entry for Prathyush Pramod identifies him as a software designer at Prabros and states that he studied at Newcastle University. The studio affiliation and his distinctive public location phrase connect the entry to this practitioner rather than to unrelated people with the same name. The full profile is inaccessible to this research, so neither a subject of study, a qualification, nor a graduation date can responsibly be added. This establishes university attendance on a self-report basis. It also changes how his later exercise records should be read: self-directed learning is a documented continuing practice, not proof that he had no formal education.[13]

There is no adequate source here for an account of his childhood or the precise sequence between university and employment. The public record becomes much clearer at the level of actual work. Pranav’s Dribbble account dates from January 2011 and describes a freelance interface-design practice. A June 2014 entry presents the final visual design for a Mac application called Footprints and directs readers to Prathyush’s Quartz Composer prototype. In August 2015, Pranav published variations for the Prabros website and again pointed toward an animated version on Prathyush’s page. These are small credits, but they recover an early material arrangement: Pranav was presenting visual designs while Prathyush was associated with making them behave over time.[14, 15, 16]

A visual layout and an interactive prototype answer different questions. The first can specify hierarchy, spacing, typography, and the appearance of controls. The second has to specify what changes when a person acts: movement, timing, state, and feedback. The early credits suggest a collaboration across this boundary, although two examples cannot establish a permanent division of all the studio’s labor. They nevertheless prevent the research archive from swallowing the design career. Pramod was working on interfaces during the same period in which his programming study became publicly legible.

From June 4 to October 17, 2014, he worked through How to Design Programs, the Scheme-based textbook associated with Matthias Felleisen and colleagues. His completion account records 540 pomodoros and estimates 200–250 hours over approximately twenty weeks. He reports learning to sketch solutions on paper before typing; a difficult recursion exercise changed his understanding. He also invites correction of the published answers. These are his measurements and retrospective lessons, rather than an externally audited diary.[17]

The importance of the episode lies in the kind of effort it made necessary. Recursion asks a programmer to solve a problem by defining how smaller instances relate to the whole. As an elementary illustration, counting the items in a list can be expressed as: an empty list has zero items; a nonempty list has one plus the count of its remainder. The difficult step is trusting and specifying the smaller operation rather than mentally carrying every subsequent step. A paper sketch can make that structure available before syntax and execution distract the learner. This example explains the pedagogical stakes; it is not represented as the particular exercise that changed Pramod’s mind.

The exercise record places his later diagrams in a more demanding light. They are not simply attractive illustrations attached to concepts he already knew. Representation was

part of learning how to solve a problem at all. It also reveals the labor concealed by an elegant explanatory page: repeated attempts, uncertainty, and sustained time. The records suggest one reason elementary exposition matters to him: he had himself needed intermediate steps. This is an inference from the learning accounts, not a verified causal history of his later projects.[18, 19]

In September 2015 he began a recorded study of Daniel P. Friedman and Matthias Felleisen’s The Little Schemer. He would return to it in late 2019 and mark completion on December 31. The surviving log therefore documents an interruption and a return, not four years of continuous study. His chapter notes identify helper functions, currying, continuations, and the construction of a Lisp interpreter in Lisp; they also record concepts for which he still wanted more examples.[20]

The January 2020 Prabros review of The Little Schemer turns this study toward criticism of its teaching. It admires helper functions and the way a program can become understandable in pieces, while finding the book’s guidance uneven for a newcomer. The review is unbylined; his separately named repository establishes his own study. Together they show both doing the exercises and considering the reader’s route through them.[21]

An interpreter is especially consequential in this learning trajectory. Instead of merely writing a program in a language, the learner builds a program that reads and evaluates expressions in that language. The machinery previously taken for granted becomes an object of work. The project does not make him the inventor of Scheme or of its semantics. It supplies a route from competent use toward understanding how a computational environment produces its results. That movement, from the surface into its enabling structure and back toward a better surface, becomes recurrent in his practice.

[External image / 外部圖片] Dark mobile interface with a map, colored route markers and a horizontal timeline.
Prabros’ interface presentation for Dash, an always-on personal journal. This is a design/prototype image showing a map and timeline. Prabros; collective studio project credit · Rights retained by the source and respective rights holders; no open license stated. · Image source

The brother collaboration and the practical studio

Prabros identifies itself as a two-person software design studio. Its portfolio describes work across product design, research, prototyping, and engineering, for startups and larger organizations around the world. Named projects include Dash, an always-on personal journal; Portage Path, a farm-management dashboard; Cody, an animated identity for an AI assistant; Roads, web and iOS work involving front- and back-end development; and icons for Intel’s internal tools. The portfolio also names clients including Kayako and Photojojo. These are the studio’s own credits, not independently audited accounts of every contract.[22]

This range explains why a public mathematical researcher would have practical knowledge of typography, animation, software tools, and product workflows. The studio sells ways of making software usable and visible. Its deliverable can be a functioning site, a product interface, an icon system, a prototype, or a visual identity. Those activities demand different skills and different amounts of engineering. A logo animation does not establish responsibility for the underlying AI system; a dashboard credit does not establish ownership of the agricultural business it serves. Keeping the scopes precise makes the work more concrete.

Pranav is central to this arrangement. His public professional profile names interface design, visual design, and user research; his individually credited design presentations reach from Footprints to the later Adjacent Possible experiments. The 2016 German coverage of their chat study explicitly identifies Prathyush and Pranav as brothers. Their shared practice has a sustained working relationship and individually visible contributions; the sources do not disclose private family dynamics.[23, 24, 25, 26]

The evidence also resists a simple assignment of "Prathyush thinks; Pranav makes it look good." In a public study thread, Prathyush credits Pranav's investigations of George Lakoff with directing him toward Saunders Mac Lane's account of mathematics. In the later Dharma reading list, a conversation with Pranav prompts a question about Krishna's advice to Arjuna, and Pranav points him toward Amartya Sen. The brother supplies intellectual encounters as well as visual artifacts. These documented exchanges do not tell us how every day's work is divided, but they establish that the collaboration can alter the research agenda itself.[27, 28]

The economic structure is visible in outline. Prabros solicits design commissions; its public research pages also invite prospective clients to contact the studio. This connects exploratory publication to a service business. An unusual interface proposal can demonstrate judgment and skill before a client supplies a brief. A long research account can show how the studio frames a problem, rather than merely how polished its screens look. That is a plausible commercial function of the publication, inferred from its placement and hiring invitations. The sources do not disclose revenues, rates, the brothers' private incomes, or what fraction of research is paid for by clients.[29, 30]

Two different forms of accountability consequently coexist. A client needs a deliverable that fits a product, deadline, and budget. A public research artifact can remain provisional and invite a reader to continue it. A studio capable of both can carry insights from exploratory work into a commission, but it can also accumulate unfinished projects whose maintenance earns no disclosed fee. The archive makes this tension unusually visible. Version numbers and partial implementations are not incidental blemishes; they are signs of how the practice divides effort among learning, publication, and delivery.

2016: a messaging interface begins with someone who

cannot read the indicator

Their chat-interface study was public by July 2016, when the German technology site iphone-ticker recommended it to readers trying new messaging interfaces. The later Prabros index lists the project in December 2022; that date records its place in the present archive, not the original beginning. The contemporary coverage both identifies the brothers and establishes an earlier audience for their design work.[31, 32]

The primary essay starts from observed friction. A mother needs her glasses to interpret the small blue checks used for message status. Prabros proposes making the whole message bubble change appearance. Other proposals use remembered images and approximate time to navigate conversations. The production account names Sketch for assets, Quartz Composer and Facebook’s Origami tools for behavior, Quartz Crystal for rendering, and FFmpeg for GIF output. These are proposed interactions assembled into a public demonstration, not evidence that WhatsApp or iMessage implemented them.[33]

The first example is revealing because it turns a convention into a question. A tiny status mark may seem sufficient to its maker because the maker already knows where to look and what it means. The person trying to use it has a bodily task: notice, distinguish, interpret. Enlarging the meaningful change can reduce that task without changing the message’s informational content. The improvement being investigated is therefore not merely decoration. It concerns where the interface places the burden of perception.

The remembered-image example introduces another kind of mismatch. Applications commonly organize messages as a linear chronological stream. A person may remember “the conversation around that photograph” rather than a date or exact phrase. A navigational representation can use that remembered landmark. Here the designer is not simplifying all of the data; he is giving a different kind of recollection somewhere to attach. This comparison helps explain the later interest in semantic zoom and spatial annotations without claiming that the subsequent projects were already planned in 2016.

The demonstrations also expose the ambiguity of motion. Readers of the German article disagreed: some found the animation distracting or saw only variations on existing features; another emphasized that user testing would have to decide. That is a small contemporary reception record, not a representative survey. It nevertheless identifies a lasting pressure on this practice: an animation must communicate a useful change, while its very fluency can make an untested proposal appear more convincing than the available evidence warrants.[34]

2019–2020: the working programmer turns historical dissatisfaction into research

When Pramod returned to The Little Schemer in 2019, the question had widened beyond a particular language. His October 2020 lambda-calculus introduction looks back on roughly

a decade of working with computer systems. He criticizes the habit of treating passing tests as sufficient understanding and describes the present software stack as an accumulation of decisions, compromises, and accidents. He wants to understand enough of the stack to build a more parsimonious environment. These are his retrospective judgments about practice, not evidence that every inherited technical convention is arbitrary.[35]

The distinction matters. A passing test establishes some specified behavior under some conditions. It does not automatically explain why a language, operating system, or interface has its particular form. Pramod's dissatisfaction concerns the gap between operational success and intelligibility. It is a professional problem before it is an aesthetic one: when an environment repeatedly resists what a maker wants to do, understanding its history and assumptions can become part of making.

Lambda calculus offered a compact place to investigate. In its basic form it uses variables, function abstraction, and function application. Philip Wadler's account of the subject explains how such a small system became foundational to the theory of computation and to relationships between proofs and programs. This technical context helps clarify Pramod's attraction: the system offers a comparatively small conceptual basis from which a wide range of computation can be described. It does not mean that learning it removes hardware, performance, or institutional constraints from software production.[36]

His site frames the work as an illustrated guide and a record for fellow travelers. Its visible version is 0.0.01, with further instructional sections still marked as forthcoming. That status is part of the career evidence. He had made a public entrance into the project, but the introduction cannot be taken as proof that a complete interactive curriculum followed. The aspiration and the artifact must be assessed separately.[37]

The accompanying How to Learn notes describe an earlier reliance on procedural JavaScript and Ruby, then a move through Scheme and recursion toward more fundamental ideas. They criticize symbolic exposition that is immediately legible to trained readers but appears like inscrutable signs to an outsider. He recommends approaching a difficult concept from multiple perspectives and points to interactive and visual mathematical explanation as a model. The notes treat learning as work on one's own capacities, not merely the acquisition of a vocabulary.[38]

This is also the moment when Twitter threads become substantial working surfaces. An accumulating thread begun in October 2019 moves through computational history, notation, and alternative representations. An October 2020 thread proposes a timeline linking thinkers' lifetimes, ideas, and interactions. A November 2020 study thread investigates logic and computation, names MIT's Logitext, and credits Alessandra Carbone's Pathways of Deduction as an important aid. A thread's initial timestamp does not date every later addition; the form is cumulative.[39, 40, 41]

The medium alters the labor of authorship. A diagram can become public before an essay is consolidated. A reader can contribute a reference while the question remains open. The same arrangement makes it easy for suggestive adjacency to outrun a demonstrated

connection. His work needs both the generative freedom of the public notebook and the slower task of checking what its lines actually mean. The later formal explanations can be read as attempts to consolidate selected connections from this broader research environment.

2020: Kerala, Malayalam, and a local technical public

Prabros' s public work has a specific South Indian setting in Makers of Kerala. Its portfolio credits the site' s front- and back-end development; the project describes a platform connecting makers across Kerala. Dated articles in 2020 cover JavaScript and Malayalam language technology. The local connection is evidenced by a publishing and software practice addressed to Malayali makers. It does not justify inferring a birth town, caste, religious community, or a generalized regional disposition from his name.[42, 43]

The February 2020 Vaaku2Vec explainer makes this setting unusually concrete. It offers Malayalam, Romanized Malayalam or Manglish, and English entrances into a word-embedding library developed by Kamal K. Raj and Adam Shamsudeen. It explains why Malayalam' s morphology matters: a language in which components combine into inflected forms cannot simply be treated as an interchangeable container for an English-oriented method. The presentation uses a conversational sequence of questions, diagrams, and links to code and demonstrations. Prabros' s contribution is the documented platform and explanatory context, not invention of Vaaku2Vec.[44]

The choice of three language modes is a design decision about who can participate. English reaches the international technical public, Malayalam connects the explanation to its linguistic object and readers, and Manglish accommodates another familiar way of encountering that language online. These modes need not imply three distinct classes of person. They offer alternative entrances for readers whose reading habits, devices, and technical vocabularies may differ. The mechanism is more specific than a claim that the studio "bridges East and West." It builds a small publishing environment in which a locally consequential technology can be understood and tried.

This work also complicates a universal account of interface accessibility. An algorithm may have a neat formal description while its treatment of words fails to match how a particular language builds them. An explanation may be clear in English while remaining distant from the people whose language supplies the data. The local task is not to reject general computation; it is to make its assumptions and adaptations visible. That fits Pramod' s broader representational questions while retaining the distinct practical stakes of Malayalam technology.

Makers of Kerala later includes a curated cafés project, indexed in August 2023. Its modest scale is useful evidence of a continuing local public, rather than proof of a regional institution or a large membership organization. The portfolio also credits web design for Talks by Women in Logic. Women in Logic is a collective whose own site describes an effort

to support and make visible women working across mathematical, philosophical, and computational logic. The studio credit establishes a contribution to that community's presentation; it does not make Prabros the organizer of the workshops or imply authorship of the current site's subsequent technical changes.[45, 46, 47]

Reading forms without reducing them: literature, mythology, and Christian references

The Prabros book reviews published in early 2020 add a dimension that disappears if the career is treated solely as a catalogue of software ideas. A January review of Paul Cobley and Litza Jansz's Introducing Semiotics appreciates its illustrated introduction while finding some transitions inadequately explained. A March review of Anne Lamott's Bird by Bird attends to voice, humor, persistence, the psychological difficulty of writing, and the religious framework informing the book. Both are unbylined studio publications. They establish a reading world around the practice, without establishing that every sentence is Prathyush's individual judgment.[48, 49]

Semiotics concerns how signs mean; a book about it therefore makes its own arrangement of signs available to criticism. An illustrated introduction can remove one obstacle and introduce another if it omits the relationship between its pieces. The Lamott review concerns a different obstacle: sustaining work when the draft is awkward and recognition uncertain. These are not interchangeable versions of mathematical explanation. Together they enlarge the studio's account of making. The maker has to organize a form that another person can enter and tolerate the incompleteness through which the form comes into existence.

The April 30, 2020 review of Joseph Campbell's Myths to Live By objects to his treatment of Indian autonomy, caste, and sati. It also resists reducing religion to psychological archetypes: Campbell risks "throwing away the lines and words for the alphabets." The criticism concerns what disappears when an interpreter discards the particular form while retaining supposedly universal elements.[50]

Its Christian references include Matthew 5:43–45 on loving enemies and Viktor Vasnetsov's Judgement Day, placed beside a discussion of light, darkness, and eschatology. Neither is identified as specifically Catholic. The further-reading list names Thomas Merton's Symbolism: Communication or Communion? Merton was a Catholic Trappist monk; this is a prospective reading, not proof the reviewer had completed it.[51, 52]

The distinction between communication and communion is suggestive in this context, but the title alone cannot establish the essay's argument or Pramod's adoption of it. What the actual review permits us to say is narrower and stronger: it takes religious forms seriously enough to object when their meaning is stripped down to an inventory of symbols. The interest is in how a form is lived or encountered, rather than only which abstract structure an analyst can extract from it. No inference about Pramod's private faith

follows.

The public religious and symbolic interests also extend beyond Catholicism. The Art of Memory repository gathers material associated with Frances Yates, Hermetic images, and mnemonic traditions. It is a resource collection, with its own historical objects, rather than a disclosed religious identity.[53]

What can be connected is a practice of reading for the relation between form, experience, and understanding. What remains open is whether these encounters amount to a settled worldview. The sources show selections, questions, objections, and passages that struck the authors. They do not supply a confession of faith or authorize a portrait of the brothers' household religion.

Working collections: a repertoire of alternatives, with other people's names retained

Around the historical threads and essays, Pramod built repositories that a reader could revisit from more than one direction. These collections are a substantial part of his work, but their scale should not be mistaken for the scale of his invention. Lambda Calculus Visualizations gathers representations and tools associated with makers including David Keenan, John Tromp, Marius Buliga, Bret Victor, and Chelsea Voss. His editorial contribution is to make alternatives discoverable and comparable, while their authors retain authorship of the collected systems.[54]

A repertoire changes the learner's situation. If one symbolic presentation fails, that failure need not mean the subject is inaccessible. Another representation may expose a dependency, an operation, or a spatial arrangement that the first concealed. The catalogue supplies possible next encounters; it does not prove that each representation teaches every concept equally well. Its intellectual service consists in showing that a notation is a choice among forms and in locating actual makers who have explored those choices.

PL Catalog makes the entrance conditions practical. Its contribution guidance asks for consistent entries and good documentation and favors an accessible playground before local installation. Models of Interaction collects environments such as Baku Hashimoto's GLisp, Matthew Keeter's libfive, and Szymon Kaliski's Protoboard, in which design, editing, and computation are arranged differently. The first collection lowers the cost of trying a language; the second widens the idea of what a programming environment might be.[55, 56]

These decisions concern labor that is often invisible in accounts of intellectual influence. Before a reader can benefit from an unfamiliar tool, somebody must identify it, describe why it belongs, preserve its credit, and provide a workable route into it. Installation can be a substantial barrier for a curious nonspecialist. A browser example can turn a promising name into an experiment. The editor is not merely storing links; he is designing the

sequence through which a reader’s curiosity can become an action.

Networked Notebooks places contemporary note tools beside earlier precedents including HyperCard and Xanadu and explicitly credits Raghuveer as a contributor. It is a record of possible arrangements for connected thought, rather than proof that Pramod uses every application listed. The historical reach matters because it prevents a contemporary interface fashion from appearing to originate without predecessors. The contribution credit matters because the collection’s breadth depends on what other people notice and bring into the conversation.[57]

His Readings on Time, published in March 2020, follows a question associated with Alan Kay through work by John McCarthy, David Reed, Leslie Lamport, and others. The Formal Systems in Biology collection reaches into McCulloch–Pitts models, cellular automata, and Lindenmayer systems; the notation repository ranges across mathematics, music, and mnemonic traditions. These establish a broad reading program. They do not establish biological experiments, a complete scholarly history of notation, or a single theory covering all the collected domains.[58, 59, 60]

The journal accompanying Morphisms of Computational Structures preserves precisely this problem. A proposed relation between closures and actors is accompanied by a reference to Carl Hewitt’s rejection of the equivalence. Other entries remain questions requiring further study. A closure retains access to an enclosing environment; an actor model organizes computation through entities exchanging messages. They may be compared, but the existence of an analogy does not settle whether their semantics coincide. The journal’s record of disagreement is therefore a better indicator of serious mapmaking than a seamless diagram with every line left unchallenged.[61, 62]

The source names in these archives form an adjacent world of the practice: programming-language researchers, visual toolmakers, historical computing figures, online explainers, and readers contributing pointers. They are not all personal friends or mentors. Pramod can encounter a dead inventor through a paper, a contemporary maker through a public demo, or a researcher through disagreement. The network is made operational by artifacts and conversation; a list of eminent influences would conceal those differences.

There is also independent evidence that this editorial and conversational work reaches another person’s production. The author publishing Category Theory Illustrated under the name Jencel Panic thanks Prathyush Pramod for encouraging completion and helping with the book. Andy Matuschak’s note on the detail-in-context problem links Pramod’s semantic-zoom collection. These are bounded acknowledgments of usefulness. They do not make Pramod coauthor of the book, establish a formal collaboration with Matuschak, or measure the entire archive’s reach.[63, 64]

2021–2022: PatternAtlas slows the map down into an explanation

PatternAtlas names its subject as maps of form and content. Its index dates Matrices as Relations to September 2021, an explanation of two interpretations of division to November, the Galois-connection essay to May 2022, and the Curry–Howard–de Bruijn introduction to August 2022. These are dated, versioned publications, not a verified employment sequence. They show selected questions receiving more sustained explanatory treatment after the expansive threads and collections.[65]

The move matters because a diagram that looks comprehensive can remain inaccessible. It may display a relation between named entities without teaching a reader what any of those entities do. Pramod’s stronger explanatory strategy is to unfold a large image into smaller operations, keep the reader’s position visible, and then return to the larger structure. The route through the image becomes part of its meaning. A mapmaker has to arrange an itinerary, not merely place everything on a page.[66, 67]

The Galois essay, version 0.08, treats a relation between two sets as an entrance into concept lattices. It offers a large “Eye of Galois” diagram, smaller navigation motifs indicating the current topic, and a storyboard of constituent objects and operations. The author explicitly assumes familiarity with sets, relations, partial orders, and lattices and leaves further proof-oriented and interactive treatments for later. Its intended accessibility therefore has a threshold: it helps someone connect knowledge already partly acquired, rather than supplying every prerequisite.[68]

To make the subject less forbidding, consider this elementary illustration, constructed here rather than taken from his diagrams. A table relates pieces of fruit to attributes. Apple and pear share “sweet”; if banana is also sweet, asking for all objects possessing that shared attribute expands the selected group. Conversely, asking for all attributes shared by a larger group can leave fewer attributes. The two directions—objects to common attributes and attributes to objects—constrain one another. Formal concept analysis studies such relations and the hierarchies they generate. Ganter and Wille’s foundational account locates it within applied lattice theory and conceptual data analysis.[69]

The example indicates why a relation can become more than a row of labels. A diagram can show how selections generate consequences, which remain stable, and which change when the selection changes. The educational work lies in making those dependencies inspectable. It is distinct from inventing the underlying mathematics. Pramod’s contribution is a proposed route into an established structure, composed from his experience of finding that structure hard to see.

The August 2022 Curry–Howard–de Bruijn essay makes its personal starting point explicit. As a designer and developer, he had found the formal presentations opaque. He recounts working with Hans Halvorson’s introductory logic, Chris Taylor’s exposition of algebraic data types, and Sergei Winitzki’s explanation of computational content. He develops the equivalence between passing two arguments successively and passing them as a pair, with recurring visual motifs identifying type, logic, computation, and data. The source itself misnames Halvorson in one sentence; the linked book identifies Hans.[70]

The broader correspondence needs qualification. In appropriate formal systems, a proposition corresponds to a type, a proof to a program inhabiting that type, and proof simplification to program evaluation. Wadler explains both its depth and its dependence on a particular logic and programming language. It is not the assertion that any ordinary program is proof of whatever claim one wants to make about it. The qualification is essential if a memorable slogan is to become useful understanding.[71]

An everyday programming illustration helps make the local issue legible. A pure function could take a tax rate and a price together and return an amount. An equivalent arrangement takes the rate first, producing a function that awaits the price. Both can compute the same result while arranging the user's encounter differently. The second arrangement can retain the rate while being reused for several prices. This is an illustrative explanation of currying, not a claim that Pramod's essay uses taxation or that all programs with side effects preserve every equivalence.

2023–2025: the interface becomes a place for thought to remain in context

From March 2023, Prabros's Adjacent Possible series examines ordinary interface operations: copying, stashing material, previewing changes, searching, reordering messages, and swapping objects. Entries have their own dates; Space Pins arrives in May 2025. Prathyush's public post calls the series Pranav's, and Pranav's individual design account presents its artifacts. This makes the later exploratory interface work part of the brother collaboration while preserving Pranav's specific authorship where it is stated.[72,73]

The series' title suggests proximity to something buildable rather than a complete replacement of computing. A proposal often begins with a small friction that a person already recognizes. The work asks whether the environment contains the right operation for the intention. That can be a more radical question than redesigning the same toolbar: an interface determines which actions count as single actions and which require improvised sequences.

The Swap Objects account follows one such intention across spreadsheets, design layouts, code, and lists. Two objects need to exchange positions or roles, but many interfaces require temporary storage, repeated moves, or editing through an intermediate state. The discussion draws on Donald Norman and Jef Raskin and considers broader permutations. This is a proposed operation with examples, not a released universal feature operating across all applications.[74]

The conceptual stakes are easier to see through an ordinary case. If two priorities in a task list are reversed, a user may want exactly one exchange. An interface that requires several moves exposes accidental intermediate arrangements and makes the user manage them. A built-in swap recognizes the intended relation directly. This illustration is explanatory; it is

not evidence that the studio measured an error reduction. The design’s appeal must still be tested against discoverability, ambiguous selection, and the expectations of the particular application.

JSON Reader supplies a smaller application project within the same world. Its public description offers a visual view of nested JSON data, selective exploration of objects, and a persistent path bar. JSON is a common way of structuring data with named fields and nested objects or arrays. A reader investigating a deeply nested item needs both local detail and a way to know how that item belongs to the whole. The path bar preserves that relationship while the view changes.[75]

This is a different use of “map” from a historical catalogue. Here navigation has to behave. The interface must support a change of focus while retaining enough context for the reader to return or compare. One can appreciate the mechanism without claiming a measured usability outcome: the project description establishes what the application attempts to offer, not how every user performs with it. The distinction between a functioning object and a validated improvement remains important.

Space Pins makes placement itself an operation. The proposal attaches annotations, reminders, and other information to meaningful positions in an interface. It acknowledges that an encompassing version would require operating-system or web-standard support. An individual studio cannot supply those dependencies through a convincing demonstration alone. Compatibility, event handling, application boundaries, and other producers’ cooperation are part of the prospective design problem.[76, 77]

That institutional limit connects the speculative work back to Pramod’s earlier dissatisfaction with inherited systems. Some difficulties can be addressed by changing a local screen; others require changing what the environment allows its components to know about one another. The prototype is useful partly because it exposes that difference. Its operation can be demonstrated locally while its broader realization depends on decisions elsewhere. A vocation in interface design therefore includes knowing which problem is within one’s deliverable and which remains a request to a platform.

The semantic-zoom collection makes the relation between local detail and overall context an explicit research subject. It includes historical approaches such as StretchText and Pad++ alongside contemporary work by Amelia Wattenberger, Orion Reed, and others. Semantic zoom changes the kind or amount of information shown as scale or focus changes; it does more than enlarge the same marks. The collection also considers changes in function and representation. Its examples retain their makers’ names.[78]

As a general illustration, a distant view of a document might show sections; approaching one section might reveal paragraphs, then editing controls. The design problem is not simply how much text to hide. The transition must help the person understand that the local object is the same object they located in the larger view. Abrupt replacement can destroy that continuity even if each individual screen is clean. Matuschak’s detail-in-context note treats this as an open problem in his own work and includes Pramod’s

catalogue among its resources.[79]

Production labor is part of the series' evidence. One account records thirty takes to capture a demonstration. The final smooth sequence is a constructed explanation requiring timing, selection, and repetition. It should be judged as such: it can reveal a possible interaction with considerable precision while concealing the messy contingencies of ordinary use. The motion-design skill associated with the early brother collaboration remains active in the later research, now in the service of making a design argument publicly observable.[80]

Bean, 2023–2026: a client supplies a different working situation

The Bean project brings Pramod and Pranav into a documented collaboration with another engineering organization. Nilenso approached Prabros for open-ended research into spreadsheet design. The studio's account names close work with Prabhanshu and Neena, describes a free grid in which structure could develop incrementally, and admits that the full vision was not achieved. The public site combines historical research, literature reviews, design prototypes, and a link to a playground. Its May 2026 index date is a publication anchor, not the beginning of the entire undertaking.[81]

The software repository supplies an earlier and distinct chronology. By October 31, 2023, Nilenso's account described a grid, a parser, a small formula-language interpreter, and reactive dynamic arrays. A June 23, 2025 update said the project was no longer actively maintained. The code credits Prabhanshu Gupta and Ravi Chandra Padmala; the repository identifies ClojureScript and an AGPL license. These facts should remain separate from Prabros' design-research credit and its 2026 publication. There is no basis for presenting Pramod as sole author of the software or treating the 2026 articles as evidence of a newly launched, maintained spreadsheet business.[82]

This chronological distinction makes the project's professional world clearer. The client was not an anonymous patron paying for an artwork. Nilenso describes itself as an employee-owned product and engineering consultancy, with work in functional programming and software delivery. Prabros entered a setting in which another team already had engineering capabilities and a computational object to develop. The public credit establishes collaboration; it does not disclose the original fee, the contract's dates, or an exhaustive assignment of every person's tasks.[83, 84]

For the brothers, the commission supplied a problem with users, inherited conventions, a research literature, and implementation constraints. Their previous public experiments could begin from a compelling gesture. Spreadsheet design requires asking what happens to formulas, dependencies, informal annotations, and existing habits when a gesture changes. It is a good case through which to see how their practice works under another institution's support, rather than an opportunity to assign all subsequent inquiry to

private genius.

The user of a spreadsheet is often an expert in something other than programming: accounts, operations, research, scheduling, or a particular organization. A grid allows that person to arrange values, calculations, headings, and notes in one place. The freedom can also create ambiguity. A machine may need to know which cells form a table, which are labels, and which are observations. The user may recognize those distinctions from position or context without ever explicitly declaring them. This is the kind of gap the team’s research had to investigate.

The research also reconstructed alternatives in spreadsheet history. Bean’s history page places VisiCalc beside later approaches such as Lotus Improv and Scheme in a Grid. This is the team’s curated account, whose individual priority claims are not all independently verified here. Its practical role is clear: before deciding which conventions to retain, the designers sought examples of other ways to arrange calculation, data, and interaction.[85]

The Prabros review of Bonnie Nardi and James R. Miller’s The Spreadsheet Interface treats the spreadsheet user as a domain expert and emphasizes visible data, a relatively small formula vocabulary, and immediate feedback. It also recognizes limits of modularity and global structure. The July review of Advait Sarkar and Andrew Gordon turns toward how people actually learn spreadsheet techniques: motivation differs, colleagues and inherited sheets matter, and a feature must repay the attention required to acquire it. These are the studio’s readings of prior studies, not newly conducted Bean user research.[86, 87]

The difference from Pramod’s own study logs is instructive. His 2014 exercise project committed hundreds of measured hours to learning a computational method. A person preparing a work report may have no desire or capacity to make that investment. The research asks the designer to respect this circumstance rather than universalize the researcher’s temperament. It is a concrete professional adjustment: explaining the inner structure of a system is useful only when the intended user can afford and benefit from that encounter.

The July 2026 review of Hadley Wickham’s Tidy Data adds another issue. Wickham’s paper proposes consistent organization—variables in columns, observations in rows, and separate tables for types of observational units—to facilitate analytical tools. Prabros’s reviewer asks how the underlying choice of what counts as an observation is made and criticizes explanations that leave that choice implicit. The review is sometimes technically tentative and says so; it should not be treated as a definitive refutation of data tidying.[88, 89]

As a simple illustration, a person can organize a household budget for reading by month, with notes beside unusual expenses, while an analytical program may prefer records with explicitly named fields. Neither arrangement explains its purpose solely through the visible rectangle. The transformation can preserve numerical values while losing contextual marks. This is an example supplied here to clarify the design problem, not a reported participant’s budget or a new Bean test.

The September review of Lyn Bartram, Michael Correll, and Melanie Tory’s Untidy Data takes such informal organization seriously. It attends to annotations, multiple grains of detail, working copies that preserve a master sheet, and layouts through which people explore and understand data. It also objects to generalizing too far from a limited study. The July Lish review, meanwhile, examines a proposed structure based on templates and coherent ranges and recognizes further development and validation needs. Reading several approaches gives the studio alternatives without requiring it to proclaim one universal replacement for the grid.[90, 91]

The published designs translate selected concerns into operations. Bean’s documentation explores selecting an object or range before choosing an action, making references through direct manipulation, showing function arguments in place, and previewing table joins. It also investigates how headings and data could be distinguished in varied layouts, including structures that do not fit an ideal rectangle. Unresolved cases remain visible. These are design explorations; the documentation does not establish adoption or population-wide gains in accuracy.[92]

One of these proposals is straightforward to describe. In the usual formula-first encounter, a person must know or find the appropriate operation, then specify its inputs. An object-first encounter begins with the data the person already recognizes and makes possible operations available around that selection. A preview offers a result to inspect before committing. This explanatory contrast identifies the proposal’s intended change in sequence; it is not a claim that the alternative removes the need to understand what a join or formula does.

Prabros’s pattern catalogue extracts a vocabulary from the work, including attention investment, data grain, marginalia, and protected master sheets. A separate review of human–spreadsheet interaction examines the relation between visible symbols, context, and meaning, while criticizing terms it finds insufficiently explained. The point of such notes is professional as well as intellectual: collaborators need usable names for recurring situations if a prototype is to address them deliberately.[93, 94]

Publishing this material after the software’s active maintenance had ended gives it a different status from a product launch. The public research can preserve what the team learned, invite discussion, and demonstrate the studio’s methods to a future client. The site visibly solicits design work. The available record does not tell us whether publication was an original contractual deliverable or a later independent decision. Its value should therefore be described in the terms supported: a substantial public account of a partial collaborative project, with design work, research synthesis, and software authorship credited separately.[95, 96]

2026: new tools, a new logic prototype, and another return to learning

The March 2026 Cavalry page records learning motion design through credited tutorials and experiments. Its nodes, timeline, and scripting interest the author as parts of a system to understand. An experiment combines duplicated circles, linked parameters, noise, and visual effects. The closing Cloud of Unknowing passage places that learning beside contemplation.[97]

The quotation offers a direct Catholic contemplative reference. Attributed to the anonymous late-medieval Cloud of Unknowing, it compares contemplation with carpentry: the reader becomes "the material it works upon." The Christian Classics Ethereal Library identifies the work as a medieval monastic guide. Its pre-Reformation setting places it within the Catholic contemplative inheritance. Its appearance here establishes a chosen text, not religious membership.[98, 99]

The placement deserves interpretation because it follows an account of learning software. The surrounding page investigates nodes, dependencies, and motion; the closing passage reverses the familiar maker's relation to material. One is not only mastering a tool but also being changed through practice. That juxtaposition does not make software training a sacrament, nor prove that the quotation caused a design method. It lets the publication hold two ways of understanding work together: gaining control over a system and allowing an activity to alter the person undertaking it. In a career full of public returns to learning, that is a substantive tension.

The episode records Pramod and the studio beginning another learning project after years of professional work. The early Quartz Composer credits had already connected him to motion through a node-based tool. Cavalry is another environment with its own conventions; previous skill does not automatically supply its conceptual model. The account does not prove a direct migration from one tool to the other, but it records continued acquisition of the skills through which interface ideas can be made visible.[100, 101]

The 2026 Dharma list records another conversation with Pranav. Prathyush asks tentatively about Krishna's advice to Arjuna, follows his brother's recommendation toward Amartya Sen, and collects scholarship on Hindu law, ritual, and liberation. The list documents the route of an inquiry; it does not claim that he has read every title or settled the philosophical question. This is a further concrete instance of the brothers' intellectual collaboration alongside their software work.[102]

Eris, documented in August 2026, brings Pramod's logic research toward a directly manipulable environment. Early notes investigate semantic zoom and graphical containers for logical operators, including conjunction, disjunction, and implication. His public account presents progress toward graphical theorem proving. The available record establishes an emerging interface and experiments; it does not certify a complete proof checker or the soundness of a finished formal system.[103]

Conjunction provides an elementary way to understand the task. If a premise states both P and Q, a valid rule permits extracting P. A graphical environment might make the premise, its parts, and the extraction visible and manipulable. The hard work is not drawing the

container alone. An implemented proof environment has to know which operations are legal, preserve dependencies, and distinguish a valid derivation from a merely persuasive arrangement. This example is explanatory, not a claim that every required mechanism is present in Eris.

The move toward such an environment changes the responsibility of his visual work. A diagram can invite someone to reason about a correspondence; a tool purporting to manipulate proofs must additionally enforce or faithfully represent the rules on which that reasoning depends. His earlier study of logic, programming, and representations makes the undertaking understandable, but does not make its success inevitable. The public record ends with work in progress.

The person, the practice, and what remains unsettled

Pramod’s career, as it can be reconstructed publicly, includes university attendance, a long-running brother collaboration, commercial design and development, measured self-directed study, mathematical writing, online editorial work, local language publishing, and a substantial research commission. The parts have different dates and conditions. Early design credits and the 2016 messaging study precede the mature PatternAtlas essays. Bean’s code predates its 2026 research presentation. The religious texts belong to specific reviews and learning pages, not to a disclosed personal religious identity.

His public self-conception as designer, developer, and mathematical learner fits a working life in which understanding must repeatedly be acquired rather than merely announced. One sees him returning to books, comparing representations, crediting a brother’s suggestion, asking for more examples, and publishing intermediate states. Those habits are documented. They do not supply a complete personality portrait away from work, and the sources give too little information about residence, domestic routines, university years, or private finances to fabricate one.

The studio makes the breadth materially possible in a way that an account of interests alone cannot show. Pranav’s visual and interface work, Prathyush’s programming and research, clients’ commissions, other engineers’ implementation, and public interlocutors’ references all enter the artifacts. A polished demonstration depends on assets, behavior, rendering, and capture. A mathematical page depends on prior studies and the slower labor of finding an explanatory route. A Malayalam explainer depends on linguistic specificity and the developers whose work it introduces. These are particular relationships and tasks, rather than a general claim that creative people collaborate.

The main difficulty of the practice is also visible. There can be more promising relationships than one person or a small studio can explain, implement, test, and maintain. A catalogue can remain useful while its links age. A prototype can communicate a good idea without surviving ordinary use. A visually generous lesson can still leave a beginner below its entrance threshold. Pramod’s own partial versions and the discontinued Bean code

make this a concrete condition of the work, not a hypothetical objection invented to balance praise.

Chronological cross-check

The dates below distinguish a life fact, a work period, and a publication. They supplement the account above.

Date or period / Evidence and meaning

Date or period: Undated university period
Evidence and meaning: Public self-profile states study at Newcastle University; degree and years unverified.[104]

Date or period: 2014–2015
Evidence and meaning: Pranav’s Footprints and Prabros website presentations point to Prathyush’s prototypes or animation.[105, 106]

Date or period: June–October 2014
Evidence and meaning: Recorded How to Design Programs study and retrospective completion account.[107]

Date or period: September 2015; late 2019
Evidence and meaning: Separate Little Schemer learning episodes; completion entry December 31, 2019.[108]

Date or period: July 2016
Evidence and meaning: Contemporary German coverage establishes the chat study’s public presence and identifies the brothers.[109]

Date or period: 2019–2020
Evidence and meaning: Accumulating computational-history and logic threads; public learning essays.[110, 111, 112]

Date or period: 2020
Evidence and meaning: Makers of Kerala language-technology explanation and Prabros book reviews.[113, 114, 115]

Date or period: 2021–2022
Evidence and meaning: Dated PatternAtlas lessons, Galois and Curry–Howard–de Bruijn essays.[116]

Date or period: March 2023 onward
Evidence and meaning: Adjacent Possible entries; Space Pins dated May 2025.[117]

Date or period: October 2023; June 2025
Evidence and meaning: Bean code’s described development state; subsequent end of active maintenance.[118]

Date or period: 2026
Evidence and meaning: Cavalry learning notes, Bean research publication and paper reviews, Dharma list, and early Eris notes.[119], [120], [121], [122]

Sources and reading notes

Research checked October 3, 2026. The principal evidence is primary: Pramod's named repositories and essays, Prabros's collective publications and portfolio, Pranav's individual design credits, and the Bean software team's own record. Contemporary German coverage supplies an independent date and identification of the brothers. Malayalam and English versions of the local technology explainer were consulted. Technical publications supply limited explanatory context; the profile does not imply that every book in a curated list was read by Pramod or by this researcher. Public examples constructed in the narrative are identified as illustrations, not observations of his users.

Prabros reviews and project pages without an individual byline remain attributed to the studio. Indexed self-profile information is used narrowly for Newcastle attendance; qualification and dates are unresolved. Publication dates do not automatically establish the start of work. Client credits establish a stated scope, while the public sources do not disclose contract values, incomes, representative usability results, or a complete employment history.

---

1. Prathyush Pramod, GitHub profile, especially the named introduction, professional self-description, and links to Prabros and PatternAtlas. Self-report; not a full CV.↩
2. Prabros, design portfolio and service account. Describes a two-person studio, services, clients, and project-specific scope. Its undated twelve-year retrospective does not establish a founding year.↩
3. Prabros project index. Visible "Maker Duo" heading and HTML description naming Prathyush and Pranav; dated archive entries. The present archive reindexes some older work.↩
4. Prathyush Pramod's public Quora entry attached to "Post", indexed description accessed October 3, 2026: identifies software design at Prabros and study at Newcastle University. The linked profile and About page were inaccessible. No degree, subject, graduation, or attendance years were verified.↩
5. Pranav Pramod, "Footprints Final User Interface", indexed June 25, 2014. The text credits Prathyush's Quartz Composer prototype separately from Pranav's visual interface.↩
6. Nicolas, "Lesetipps: Ideen für bessere Messenger – Apple auf einem Auge blind", iphone-ticker, July 11, 2016. German contemporary coverage explicitly identifies the two Indian software designers as brothers. The messenger section and its comments were read; the unrelated Apple accessibility profile is not used as evidence about them.↩

7. Makers of Kerala, mission and dated article index. The studio's development credit is documented separately in its portfolio.↩

8. PatternAtlas index. Dates and versions for the cited lessons; synopsis errors are not automatically adopted as mathematical facts.↩

9. Prabros, Bean overview, indexed May 29, 2026. Describes the Nilenso approach, Prabhanshu and Neena collaboration, intended environment, and partial realization. Index date is not used as the commission date.↩

10. Prathyush Pramod, GitHub profile, especially the named introduction, professional self-description, and links to Prabros and PatternAtlas. Self-report; not a full CV.↩

11. Pramod, "How to Learn". Entire public working note read, including the account of JavaScript/Ruby, Scheme, symbolic presentation, and multiple approaches to learning. Its exact composition date is not established.↩

12. Prabros, "Interface Ideas for Chat Applications". Actual proposal, observed indicator problem, navigation ideas, and production-tool chain. The present archive's December 2022 entry is distinguished from the contemporary 2016 coverage.↩

13. Prathyush Pramod's public Quora entry attached to "Post", indexed description accessed October 3, 2026: identifies software design at Prabros and study at Newcastle University. The linked profile and About page were inaccessible. No degree, subject, graduation, or attendance years were verified.↩

14. Pranav Pramod, Dribbble professional profile and individual work. Self-description of interface, visual, and research practice; account membership January 2011 does not itself date Prabros's founding.↩

15. Pranav Pramod, "Footprints Final User Interface", indexed June 25, 2014. The text credits Prathyush's Quartz Composer prototype separately from Pranav's visual interface.↩

16. Pranav Pramod, "Prabros Website Variations", indexed August 5, 2015. Points readers to Prathyush's anticipated animated version; evidence of a stated collaboration, not proof of the animation's subsequent delivery.↩

17. Pramod, How to Design Programs exercise repository. Retrospective start/end dates, pomodoro accounting, estimated hours, lessons, and invitation to correct answers.↩

18. Pramod, How to Design Programs exercise repository. Retrospective start/end dates, pomodoro accounting, estimated hours, lessons, and invitation to correct answers.↩

19. Pramod, "How to Learn". Entire public working note read, including the account of JavaScript/Ruby, Scheme, symbolic presentation, and multiple approaches to learning. Its exact composition date is not established.↩

20. Pramod, The Little Schemer repository, especially its chapter notes and 2015/2019 logs. December 31, 2019 completion entry; recorded intervals should not be collapsed into uninterrupted study.↩

21. Prabros, “The Little Schemer: A Review”, January 30, 2020. Complete studio review consulted; its pedagogical judgments are distinguished from Pramod’s individually credited study log.↩

22. Prabros, design portfolio and service account. Describes a two-person studio, services, clients, and project-specific scope. Its undated twelve-year retrospective does not establish a founding year.↩

23. Pranav Pramod, Dribbble professional profile and individual work. Self-description of interface, visual, and research practice; account membership January 2011 does not itself date Prabros’s founding.↩

24. Pranav Pramod, “Footprints Final User Interface”, indexed June 25, 2014. The text credits Prathyush’s Quartz Composer prototype separately from Pranav’s visual interface.↩

25. Nicolas, “Lesetipps: Ideen für bessere Messenger – Apple auf einem Auge blind”, iphone-ticker, July 11, 2016. German contemporary coverage explicitly identifies the two Indian software designers as brothers. The messenger section and its comments were read; the unrelated Apple accessibility profile is not used as evidence about them.↩

26. Prathyush, public post identifying Pranav’s continuing Adjacent Possible series, May 28, 2025, indexed text consulted; and Pranav’s individually credited design portfolio. Does not assign every collectively published detail to one brother.↩

27. Pramod, logic/computation study thread begun November 9, 2020; original status. Named resources, reported formative encounters, and credit to Pranav’s Lakoff-related inquiry. Its mathematical use of “trinitarianism” is not evidence of Christian belief.↩

28. Pramod, Dharma reading list, public 2026 research record. Entire short note read, including the conversation with Pranav, Sen reference, and reading suggestions. Its tentative question is not represented as an established conclusion in Hindu philosophy.↩

29. Prabros, design portfolio and service account. Describes a two-person studio, services, clients, and project-specific scope. Its undated twelve-year retrospective does not establish a founding year.↩

30. Prabros, Bean research blog index. Intended readership, literature program, project account, and invitation to seek design services.↩

31. Nicolas, “Lesetipps: Ideen für bessere Messenger – Apple auf einem Auge blind”, iphone-ticker, July 11, 2016. German contemporary coverage explicitly identifies the two Indian software designers as brothers. The messenger section and its comments were read; the unrelated Apple accessibility profile is not used as evidence about them.↩

32. Prabros project index. Visible “Maker Duo” heading and HTML description naming Prathyush and Pranav; dated archive entries. The present archive reindexes some older work.↩

33. Prabros, “Interface Ideas for Chat Applications”. Actual proposal, observed indicator problem, navigation ideas, and production-tool chain. The present archive’s December 2022 entry is distinguished from the contemporary 2016 coverage.↩

34. Nicolas, “Lesetipps: Ideen für bessere Messenger – Apple auf einem Auge blind”, iphone-ticker, July 11, 2016. German contemporary coverage explicitly identifies the two Indian software designers as brothers. The messenger section and its comments were read; the unrelated Apple accessibility profile is not used as evidence about them.↩

35. Pramod/Prabros, Understanding Lambda Calculus introduction, version 0.0.01, 2020. First-person motivation and stated educational aim; later sections remain forthcoming on the consulted page.↩

36. Philip Wadler, Propositions as Types, version V10, November 29, 2014. Introduction and opening treatment of lambda calculus consulted for technical context. Supports the qualified correspondence between specified formal systems; not evidence of a personal relationship with Pramod.↩

37. Pramod/Prabros, Understanding Lambda Calculus introduction, version 0.0.01, 2020. First-person motivation and stated educational aim; later sections remain forthcoming on the consulted page.↩

38. Pramod, “How to Learn”. Entire public working note read, including the account of JavaScript/Ruby, Scheme, symbolic presentation, and multiple approaches to learning. Its exact composition date is not established.↩

39. Pramod, computational-history thread begun October 28, 2019; original status. Accumulating thread: subsequent entries are not all dated to its first post.↩

40. Pramod, timeline study thread; original status, October 2020. Documents a proposed historical representation, not a completed comprehensive database.↩

41. Pramod, logic/computation study thread begun November 9, 2020; original status. Named resources, reported formative encounters, and credit to Pranav’s Lakoff-related inquiry. Its mathematical use of “trinitarianism” is not evidence of Christian belief.↩

42. Prabros, design portfolio and service account. Describes a two-person studio, services, clients, and project-specific scope. Its undated twelve-year retrospective does not establish a founding year.↩

43. Makers of Kerala, mission and dated article index. The studio’s development credit is documented separately in its portfolio.↩

44. Makers of Kerala, Malayalam Vaaku2Vec explainer, February 24, 2020, and English version. Language menu includes Manglish. Developers Kamal K. Raj and Adam Shamsudeen are expressly credited. Versions are treated as one underlying article for derivation limits.↩

45. Makers of Kerala, cafés project, indexed August 2023. Only the public description and

index were used; no unseen list of individual cafés is asserted.↩

46. Prabros, design portfolio and service account. Describes a two-person studio, services, clients, and project-specific scope. Its undated twelve-year retrospective does not establish a founding year.↩

47. Women in Logic, collective's stated purpose and institutional structure. Read for context of the studio's portfolio credit; the current site also credits later technical help and is not attributed wholesale to Prabros.↩

48. Prabros, review of Introducing Semiotics, January 27, 2020. Unbylined studio review; discussion is attributed accordingly.↩

49. Prabros, review of Anne Lamott's Bird by Bird, March 27, 2020. Actual response to the book's writing advice and religious frame; not a claim of Catholic authorship or of the brothers' faith.↩

50. Prabros, review of Joseph Campbell's Myths to Live By, April 30, 2020. Complete review consulted; biblical quotation identified without reproducing it; Merton listed prospectively.↩

51. Prabros, review of Joseph Campbell's Myths to Live By, April 30, 2020. Complete review consulted; biblical quotation identified without reproducing it; Merton listed prospectively.↩

52. Thomas Merton Center, Bellarmine University, "Thomas Merton's Life and Work". Institutional account verifies Catholic conversion and Trappist vocation. Used only to identify the tradition of the writer listed in the Prabros review, not to claim Pramod read or adopted Merton's complete work.↩

53. Pramod, Art of Memory. Public overview and references to Yates and mnemonic/Hermetic traditions; a resource collection, not a complete scholarly argument or religious confession.↩

54. Pramod, Lambda Calculus Visualizations. Curated systems retain their respective makers' authorship.↩

55. Pramod, PL Catalog. Contribution and entry conditions consulted alongside the catalogue; a repository date does not establish invention of its languages.↩

56. Pramod, Models of Interaction. Catalogue and documented examples of design/code environments.↩

57. Pramod and contributors, Networked Notebooks. Complete public overview consulted, including prior-art section and Raghuveer credit.↩

58. Pramod, "Readings on Time", March 10, 2020. Curated inquiry into programming and systems history; does not establish endorsement of every listed approach.↩

59. Pramod, Formal Systems in Biology. Used as evidence of reading and editorial scope, not original biological research.↩

60. Pramod, Notation. Public repository spanning representational traditions, with abbreviated or unfinished headings.↩

61. Pramod, research journal. Entire short note read; preserves the disputed closure–actor comparison and unresolved questions.↩

62. Pramod, Morphisms of Computational Structures. Working compilation of correspondences, references, and open questions; sections do not have uniform evidentiary status.↩

63. Jencel Panic, Category Theory Illustrated: about and acknowledgments. Actual acknowledgment of Pramod's encouragement and help; no larger authorship claim is inferred.↩

64. Andy Matuschak, "The detail-in-context problem". Entire working note read; links the semantic-zoom collection among references to an unresolved problem.↩

65. PatternAtlas index. Dates and versions for the cited lessons; synopsis errors are not automatically adopted as mathematical facts.↩

66. Pramod, "Structures and Processes of a Galois Connection", version 0.08, indexed May 31, 2022. Entire visible introductory installment read, including navigation, prerequisites, and forthcoming sections.↩

67. Pramod, "An Elementary Introduction to Curry Howard de Bruijn Correspondence", version 0.13, indexed August 31, 2022. Visible installment read, including the author's reported difficulty, named sources, currying discussion, diagrams, and incompleteness. General illustrative examples in this profile are separate from his figures.↩

68. Pramod, "Structures and Processes of a Galois Connection", version 0.08, indexed May 31, 2022. Entire visible introductory installment read, including navigation, prerequisites, and forthcoming sections.↩

69. Bernhard Ganter and Rudolf Wille, Formal Concept Analysis: Mathematical Foundations, Springer, 1999. Publisher's overview and contents consulted for disciplinary context; the profile does not claim to have read the subscription-only volume in full.↩

70. Pramod, "An Elementary Introduction to Curry Howard de Bruijn Correspondence", version 0.13, indexed August 31, 2022. Visible installment read, including the author's reported difficulty, named sources, currying discussion, diagrams, and incompleteness. General illustrative examples in this profile are separate from his figures.↩

71. Philip Wadler, Propositions as Types, version V10, November 29, 2014. Introduction and opening treatment of lambda calculus consulted for technical context. Supports the qualified correspondence between specified formal systems; not evidence of a personal relationship with Pramod.↩

72. Prabros, Adjacent Possible. Complete public series index and presentation notes; March 2023 start and subsequent individual entries, including the thirty-take capture note.↩

73. Prathyush, public post identifying Pranav's continuing Adjacent Possible series,

May 28, 2025, indexed text consulted; and Pranav's individually credited design portfolio. Does not assign every collectively published detail to one brother.↩

74. Prabros, "Swap Objects". Actual examples, precedents, and proposed extension beyond two objects; not a claim of universal implementation.↩

75. Prabros, JSON Reader. Entire short description consulted; establishes proposed navigation mechanisms and an application link, without measured performance claims.↩

76. Prabros, "Space Pins", May 28, 2025. Actual proposal and acknowledged need for broader platform support.↩

77. Pranav Pramod, "Space Pins" design presentation. Individual credit supplements the studio's collective publication.↩

78. Pramod, Semantic Zoom. Entire public overview consulted, including historical precedents, contemporary makers, proposed uses, and related references.↩

79. Andy Matuschak, "The detail-in-context problem". Entire working note read; links the semantic-zoom collection among references to an unresolved problem.↩

80. Prabros, Adjacent Possible. Complete public series index and presentation notes; March 2023 start and subsequent individual entries, including the thirty-take capture note.↩

81. Prabros, Bean overview, indexed May 29, 2026. Describes the Nilenso approach, Prabhanshu and Neena collaboration, intended environment, and partial realization. Index date is not used as the commission date.↩

82. Prabhanshu Gupta and Ravi Chandra Padmala/Nilenso, Bean software repository. Development-state note October 31, 2023; no-active-maintenance update June 23, 2025; authors, stack, license. Read independently from Prabros's design site.↩

83. Nilenso, public account of employee ownership, product/engineering services, and technical practices. No inference about the Bean fee or individual wages is made.↩

84. Prabros, Bean overview, indexed May 29, 2026. Describes the Nilenso approach, Prabhanshu and Neena collaboration, intended environment, and partial realization. Index date is not used as the commission date.↩

85. Prabros, Bean history of spreadsheets. Entire public catalogue consulted as a design-research lineage; not an independent verification of every historical priority or date.↩

86. Prabros, review of Nardi and Miller, "The Spreadsheet Interface: A Basis for End-User Programming", June 25, 2026. Actual review, not a new Bean participant study.↩

87. Prabros, review of Sarkar and Gordon, "How Do People Learn to Use Spreadsheets?", July 3, 2026. Complete review consulted for learning, motivation, and attention-investment questions.↩

88. Hadley Wickham, "Tidy Data", Journal of Statistical Software 59(10), September 12, 2014, DOI 10.18637/jss.v059.i10. Published abstract consulted for the proposal's own

stated scope; not claimed as a full-paper reading.↩

89. Prabros, review of Wickham, “Tidy Data”, July 14, 2026. Complete review read; its ontology questions and tentative interpretations are attributed to the reviewer.↩

90. Prabros, review of Bartram, Correll, and Tory, “Untidy Data: The Unreasonable Effectiveness of Tables”, September 15, 2026. Complete review read, including marginalia, working copies, and objections concerning the study’s generalizability.↩

91. Prabros, review of “Structuring Spreadsheets With ‘Lish’ Data Model”, July 27, 2026. Complete review read. The model belongs to Alan Hall, Michel Wermelinger, Tony Hirst, and Santi Phithakkitnukoon.↩

92. Prabros, Bean design documentation. Complete publicly retrieved page read, including operation sequence, references, previews, tables, varied layouts, and unresolved cases. Not representative user-test results.↩

93. Prabros, Bean patterns. Public research vocabulary and links to its derivation; patterns are treated as aids to inquiry, not universal laws.↩

94. Prabros, review of Andrea Kohlhase, “Human–Spreadsheet Interaction”, July 20, 2026. Complete review read; contextual distinction between symbols, information, and knowledge, with its critique of terminology.↩

95. Prabhanshu Gupta and Ravi Chandra Padmala/Nilenso, Bean software repository. Development-state note October 31, 2023; no-active-maintenance update June 23, 2025; authors, stack, license. Read independently from Prabros’s design site.↩

96. Prabros, Bean research blog index. Intended readership, literature program, project account, and invitation to seek design services.↩

97. Prabros, Cavalry learning notes, March 29–30, 2026 onward. Complete publicly retrieved page read, including tutorial credits, node/timeline learning, metaball experiment, and concluding Cloud of Unknowing passage. Collective publication; no unverified individual byline assigned.↩

98. Prabros, Cavalry learning notes, March 29–30, 2026 onward. Complete publicly retrieved page read, including tutorial credits, node/timeline learning, metaball experiment, and concluding Cloud of Unknowing passage. Collective publication; no unverified individual byline assigned.↩

99. Christian Classics Ethereal Library, Cloud of Unknowing, work information. Consulted for the medieval monastic context of the text quoted in the Cavalry notes. The modern rendering appearing there is not assigned an unverified translator.↩

100. Pranav Pramod, “Footprints Final User Interface”, indexed June 25, 2014. The text credits Prathyush’s Quartz Composer prototype separately from Pranav’s visual interface.↩

101. Prabros, Cavalry learning notes, March 29–30, 2026 onward. Complete publicly retrieved page read, including tutorial credits, node/timeline learning, metaball experiment, and concluding Cloud of Unknowing passage. Collective publication; no unverified individual byline assigned.↩

102. Pramod, Dharma reading list, public 2026 research record. Entire short note read, including the conversation with Pranav, Sen reference, and reading suggestions. Its tentative question is not represented as an established conclusion in Hindu philosophy.↩

103. Prabros, Eris, August 17 and 20, 2026 entries. Entire short project record read; early graphical logic and semantic-zoom work, with formal correctness and completeness unverified.↩

104. Prathyush Pramod's public Quora entry attached to "Post", indexed description accessed October 3, 2026: identifies software design at Prabros and study at Newcastle University. The linked profile and About page were inaccessible. No degree, subject, graduation, or attendance years were verified.↩

105. Pranav Pramod, "Footprints Final User Interface", indexed June 25, 2014. The text credits Prathyush's Quartz Composer prototype separately from Pranav's visual interface.↩

106. Pranav Pramod, "Prabros Website Variations", indexed August 5, 2015. Points readers to Prathyush's anticipated animated version; evidence of a stated collaboration, not proof of the animation's subsequent delivery.↩

107. Pramod, How to Design Programs exercise repository. Retrospective start/end dates, pomodoro accounting, estimated hours, lessons, and invitation to correct answers.↩

108. Pramod, The Little Schemer repository, especially its chapter notes and 2015/2019 logs. December 31, 2019 completion entry; recorded intervals should not be collapsed into uninterrupted study.↩

109. Nicolas, "Lesetipps: Ideen für bessere Messenger – Apple auf einem Auge blind", iphone-ticker, July 11, 2016. German contemporary coverage explicitly identifies the two Indian software designers as brothers. The messenger section and its comments were read; the unrelated Apple accessibility profile is not used as evidence about them.↩

110. Pramod, computational-history thread begun October 28, 2019; original status. Accumulating thread: subsequent entries are not all dated to its first post.↩

111. Pramod, logic/computation study thread begun November 9, 2020; original status. Named resources, reported formative encounters, and credit to Pranav's Lakoff-related inquiry. Its mathematical use of "trinitarianism" is not evidence of Christian belief.↩

112. Pramod/Prabros, Understanding Lambda Calculus introduction, version 0.0.01, 2020. First-person motivation and stated educational aim; later sections remain forthcoming on the consulted page.↩

113. Makers of Kerala, mission and dated article index. The studio's development credit is documented separately in its portfolio.↩

114. Makers of Kerala, Malayalam Vaaku2Vec explainer, February 24, 2020, and English version. Language menu includes Manglish. Developers Kamal K. Raj and Adam

Shamsudeen are expressly credited. Versions are treated as one underlying article for derivation limits.↩

115. Prabros, review of Joseph Campbell's Myths to Live By, April 30, 2020. Complete review consulted; biblical quotation identified without reproducing it; Merton listed prospectively.↩

116. PatternAtlas index. Dates and versions for the cited lessons; synopsis errors are not automatically adopted as mathematical facts.↩

117. Prabros, Adjacent Possible. Complete public series index and presentation notes; March 2023 start and subsequent individual entries, including the thirty-take capture note.↩

118. Prabhanshu Gupta and Ravi Chandra Padmala/Nilenso, Bean software repository. Development-state note October 31, 2023; no-active-maintenance update June 23, 2025; authors, stack, license. Read independently from Prabros's design site.↩

119. Prabros, Cavalry learning notes, March 29–30, 2026 onward. Complete publicly retrieved page read, including tutorial credits, node/timeline learning, metaball experiment, and concluding Cloud of Unknowing passage. Collective publication; no unverified individual byline assigned.↩

120. Prabros, Bean overview, indexed May 29, 2026. Describes the Nilenso approach, Prabhanshu and Neena collaboration, intended environment, and partial realization. Index date is not used as the commission date.↩

121. Pramod, Dharma reading list, public 2026 research record. Entire short note read, including the conversation with Pranav, Sen reference, and reading suggestions. Its tentative question is not represented as an established conclusion in Hindu philosophy.↩

122. Prabros, Eris, August 17 and 20, 2026 entries. Entire short project record read; early graphical logic and semantic-zoom work, with formal correctness and completeness unverified.↩