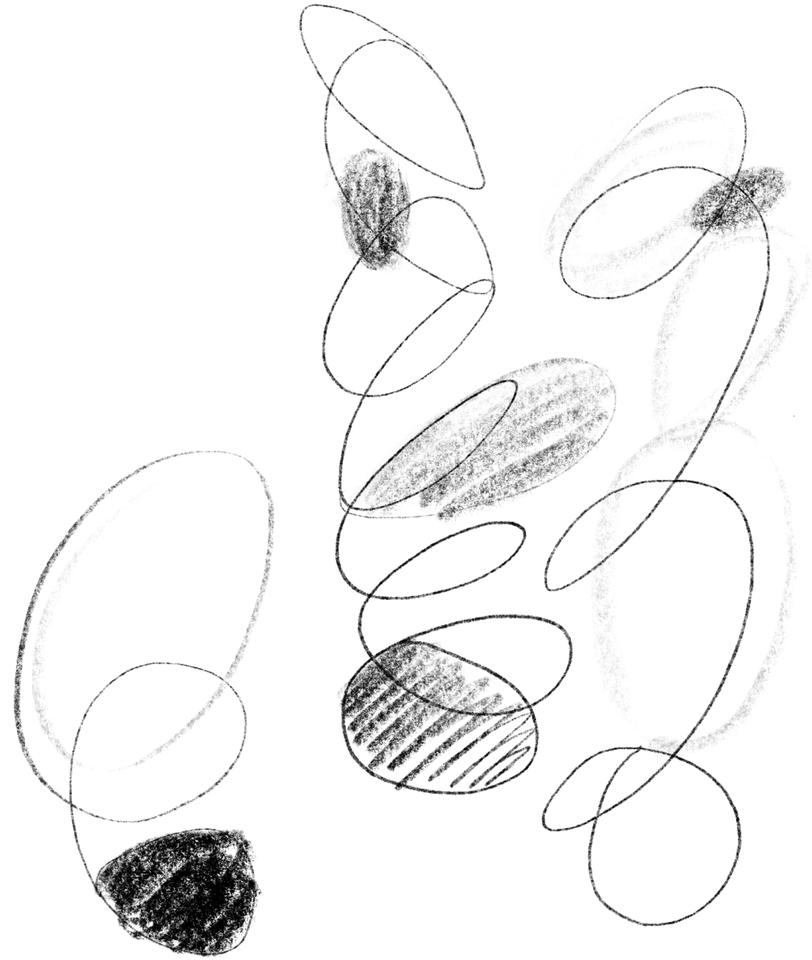
Cairn:
AI
interactive
experience
dev
kit



Cairn is a versatile web development framework that empowers developers to create AI-driven interactive websites with minimal setup. It is designed for creating experiences that focus on gesture communication with mouse clicking — a fundamental gesture ingrained in our digital behavior. Cairn invites developers to join in exploring the possibilities of more intuitive, playful, and potentially profound ways of engaging with artificial intelligence.

This project examines our fundamental relationship with AI by asking the following questions:

When we step away from viewing AI as merely a problem-solving tool, what meaningful insights might emerge?
Can our relationship with AI evolve beyond the traditional request-response paradigm?

The development process was assisted by Claude.

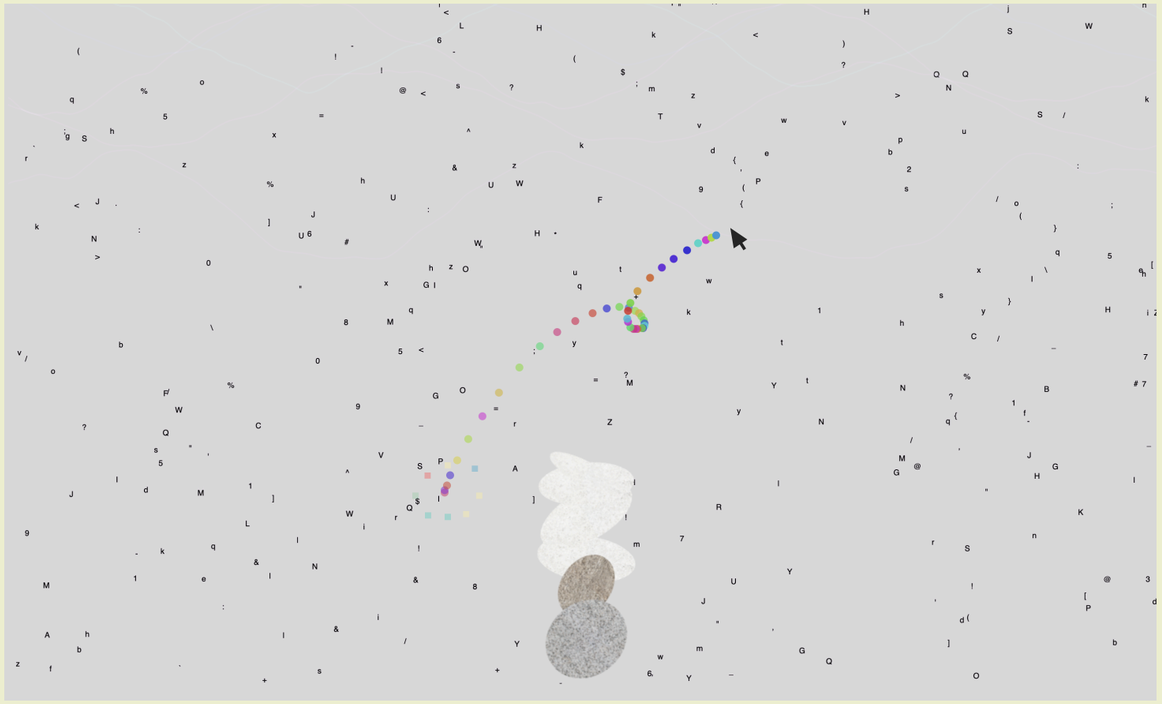
https://github.com/urverkmi/cairn

Independent project
December 2024

Demo Experiences

Experience Cairn through two interactive demos that showcase its potential for creating unique interactions with artificial intelligence. While both demos interpret the same fundamental mouse clicking patterns, each reveals a distinct approach to translating these gestures into meaningful AI responses.

Let's explore a common interaction scenario across both demos: imagine a user creating a series of rapid, clustered clicks on the screen. Each demo transforms this input into a remarkably different experience:

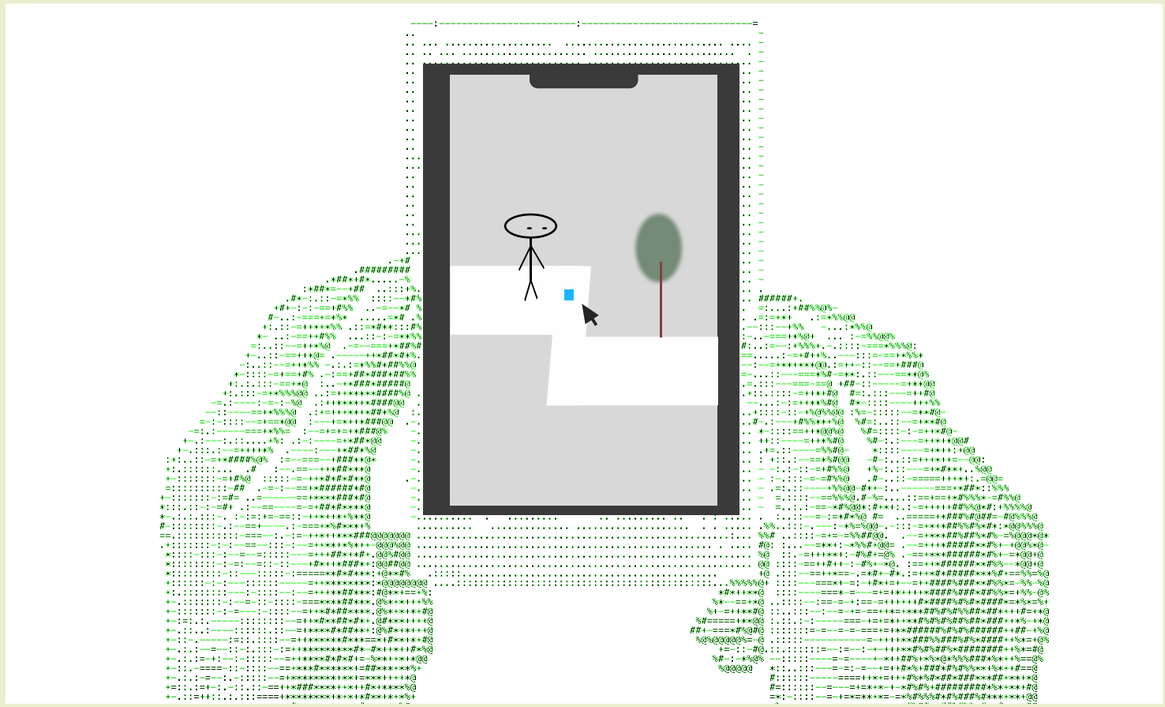


https://urverkmi.github.io/cairn/

Demo experience 1: Sensei Cairn

"Sensei Cairn" seeks to determine if the interactions that we impose are reflective of our consciousness. Based on the click pattern, the sentinent Cairn character would wake and comment with a philosophical dialogue, connecting the interactions to themes of chaos, existentialism, and intentionality.

> Time flows like water between my stones.
>
> You humans scatter your meanings across centuries, yet always return to create your patterns - never quite random, never fully ordered.
>
> Makes me wonder - do you build these cairns to mark your path, or do we mark something deeper in you? There's a wild defiance in stacking stone upon stone, telling gravity and entropy to go to hell, even if just for a moment.
>
> Your fingerprints fade from my surface, but your restlessness remains.



https://urverkmi.github.io/promenade/

Demo experience 2: Take a Virtual Promenade!

"Take a Virtual Promenade!" is designed as a delightful, fieldnotes-style web app. Through simple mouse clicks, you guide a stick figure along a winding path, where each step potentially reveals a small moment of discovery. The experience transforms your clicking patterns into two forms of AI response:

1. The narrative description of the gift

   > I found this magical butterfly wing for you! It seems to match your energetic spirit today - full of movement and vibrant colors. In our game world, butterfly wings are said to grant swift movement and bring moments of joy to their holders. Would you like to add it to your inventory?

2. A graphical representation of the gift

How does it work?

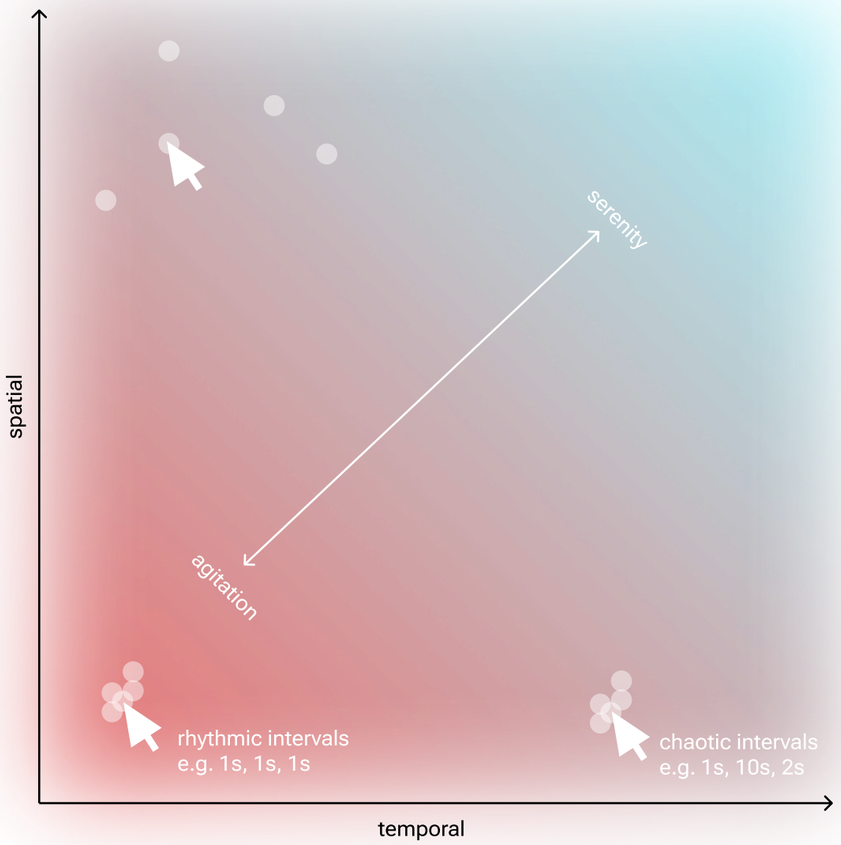
The framework is created with Anthropic intelligence and p5.js visuals.

The flow diagram shows how interactions are encoded, aggregated, and analyzed to be sent to the backend API. The parameters defining a pattern are its spatial and temporal complexities — basically, how rhythmic and how scattered.

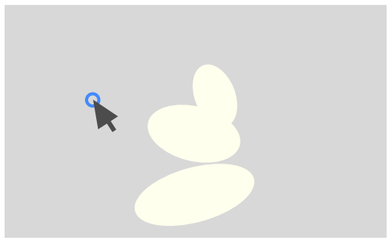
The processed interaction data is then channeled to Claude, which interprets the emotional undertones embedded in the interaction patterns.

- Rapid, clustered clicks suggest heightened states like anxiety or agitation
- Measured, dispersed clicks are interpreted as calm and serene

Then, AI generates contextually appropriate responses based on perceived user state. The system's interpretation of these patterns allows for more nuanced and organic interactions.



1. detect interactions



2. encode & aggregate

```
Interactions [
  {
    loc: [50, 50],
    type: 'click',
  },
  {
    loc: [100, 80],
    type: 'click',
  }
]
```

3. analyze & send to worker

based on customized function

e.g. a probabilistic model determining user engagement

4. get API response

optional: look in cache for hits

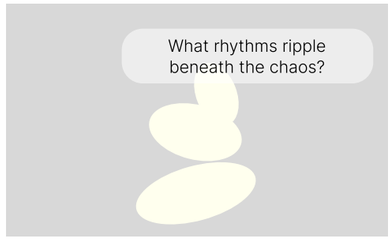
key	xxxxx
API response	"What rhythms ripple beneath the chaos? ..."

If no hit, make API call and store to cache

5. process API response

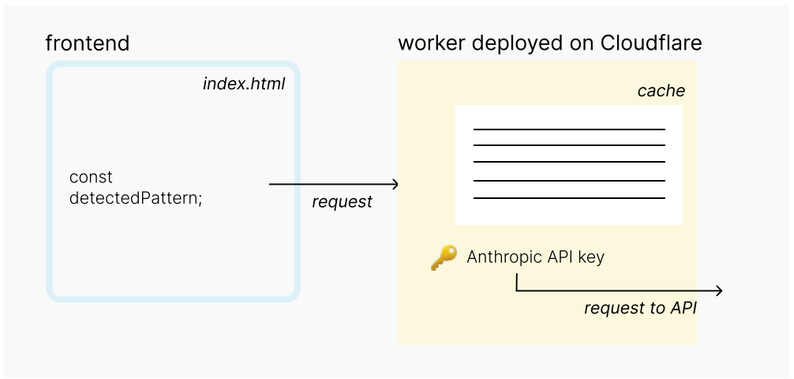
based on customized function

e.g. display on screen:



The framework comes with the groundwork for connecting to Cloudflare, a content delivery network providing features such as secure credential management and external caches. This ensures the secrets are not exposed through client side page, and it also makes the frontend code more lightweight.

A cache can be stored with normalized request parameters and the corresponding API response. This effectively optimizes API usage as the responses can be re-used across different sessions — given a high traffic, it can potentially save up to 80% compared to direct API calls.



How to start using it?

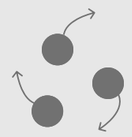
The framework provides various pipelines of interpreting and presenting AI outputs, moving beyond simple text display.

```
/* Start guide */
/* Choose a processing pipeline from the following list that suits your design intent */

/*
```

pipeline:	prompt	AI response	Result

graphic	Describe the color and shape information.	`[{shape: "cylinder", color: "blue"},`
		`{shape: "ellipse", color: "blue"},`
		`{shape: "ellipse", color: "purple"},`
		`{shape: "line", color: "black"},`
		`{shape: "circle", color: "yellow"}]`
			

choreo	Describe the movement qualities, spatial changes, and durations.	`[{movement: "swaying", duration: "3000"},`
		`{movement: "spiral", duration: "1500"},`
		`{movement: "scatter", duration: "300"}]`
			

musical	Describe the notes and durations.	`[{note: "A4", duration: "200"},`
		`{note: "C5", duration: "200"},`
		`{note: "E5", duration: "200"},`
		`{note: "C6", color: "1000"},`
		`{note: "E6", color: "1000"}]`
			

```
*/

/* You are also welcome to create your own pipelines by defining the following functions */

    F(Interaction[]) = send_to_worker::bool // When to send the packed data to create the prompt?

    F(Interaction[]) = prompt::string // How is the prompt written?

    F(api_response)  = page_action() // What to do with the API responses?
```

Conclusion

This project represent a broader vision: moving toward AI as a source of serendipitous discovery. Through these unscripted interactions, we explore how artificial intelligence can contribute to experiences that surprise, delight, and perhaps reveal unexpected insights. We open possibilities for unexpected insights and meaningful exchanges that aren't driven by immediate needs or explicit queries.

The project is named after the carefully balanced stone stacks that guide hikers along trails. Cairn provides developers with foundational building blocks while encouraging creative exploration.