

Making Microworlds: A Framework for Making Sense by Making Things

...

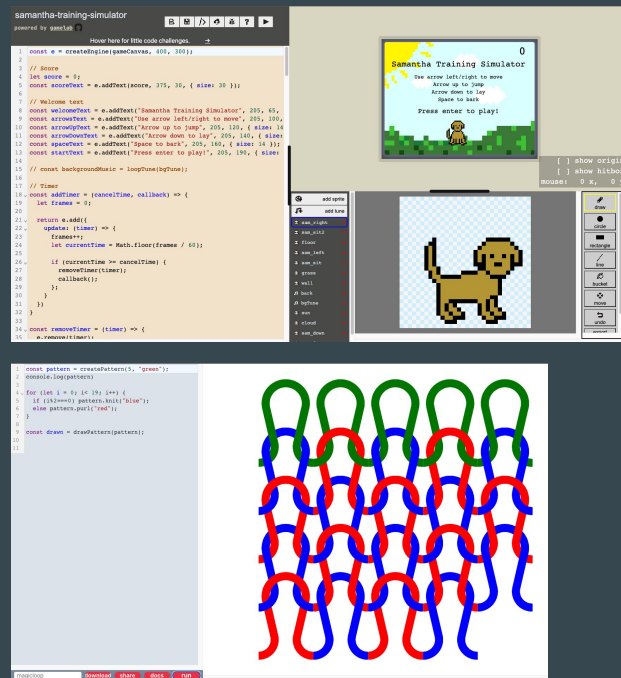
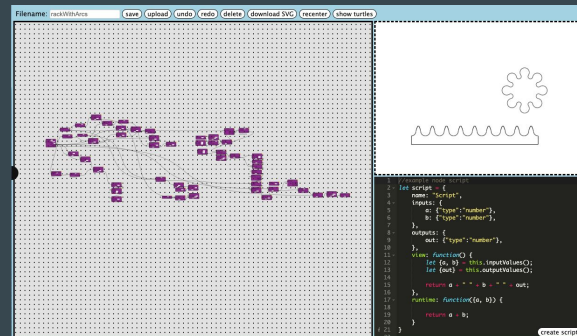
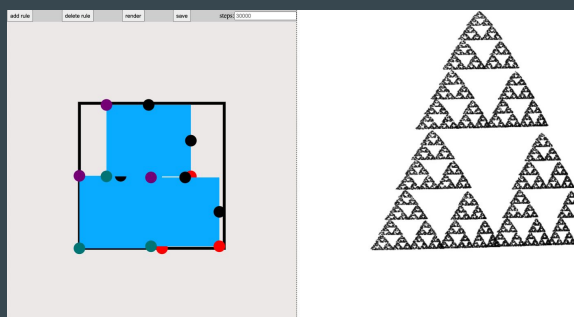
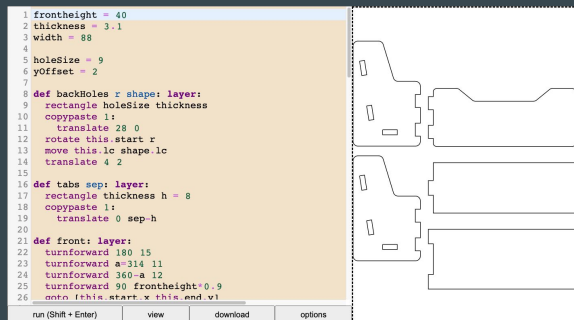
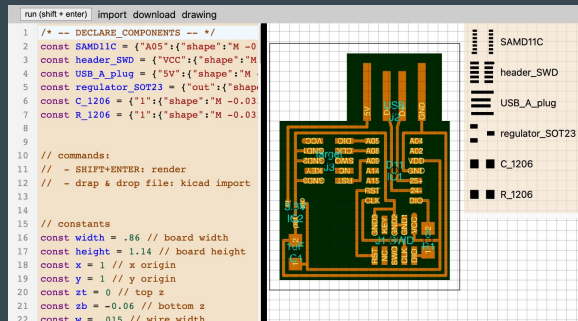
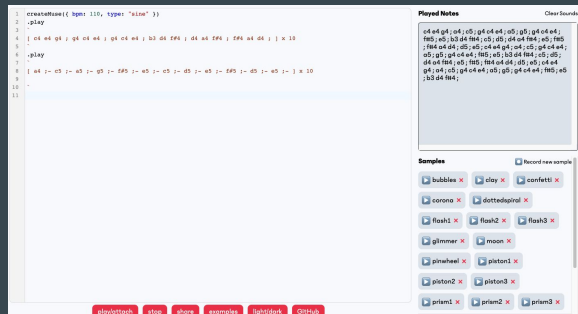
Leo McElroy
April 14, 2022

People learn best by making things they care about, which they share with others.

Physical Tools



Digital Tools



Some Guideposts

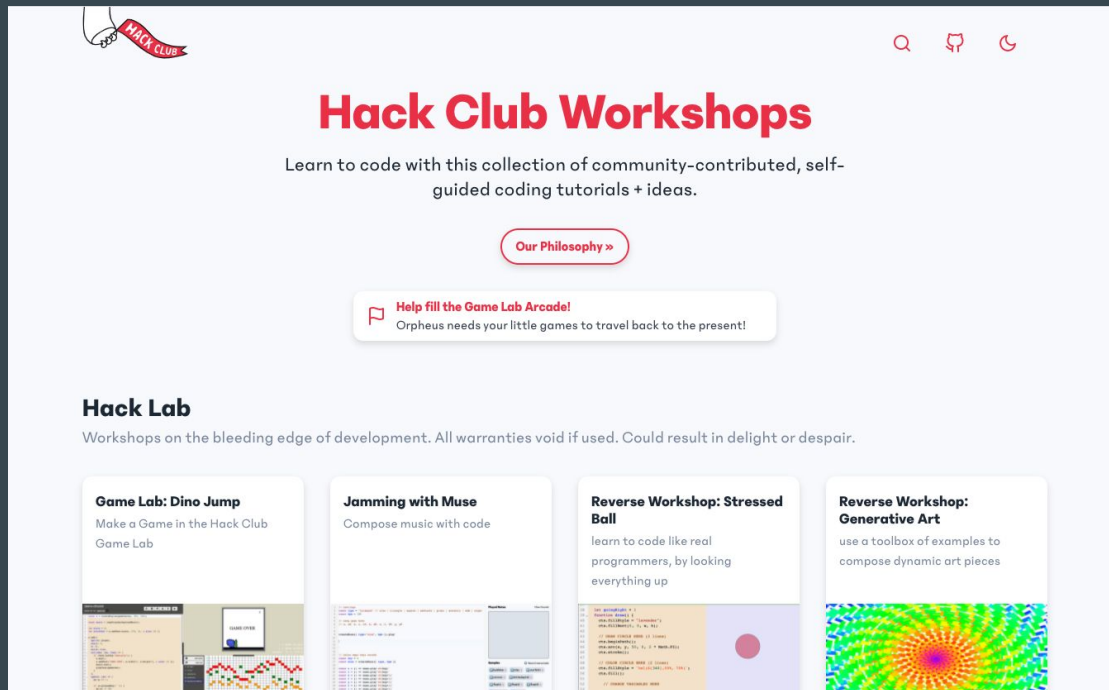
- The status quo: Hack Club Workshops
- Some history on Piaget, Papert, and microworlds
- A tour of microworlds
- Distill qualities of microworlds to help us make more

Hack Club



<https://map.hackclub.com/>

Status Quo: Hack Club Workshops




The screenshot shows the Hack Club Workshops website. At the top left is the Hack Club logo, a hand holding a red banner that says 'HACK CLUB'. In the top right corner are three icons: a magnifying glass, a GitHub Octocat, and a clock. The main heading is 'Hack Club Workshops' in large red text. Below it is a subtitle: 'Learn to code with this collection of community-contributed, self-guided coding tutorials + ideas.' There is a red button with white text that says 'Our Philosophy >'. Below that is a white box with a red flag icon and the text 'Help fill the Game Lab Arcade! Orpheus needs your little games to travel back to the present!'. The section 'Hack Lab' is followed by a paragraph: 'Workshops on the bleeding edge of development. All warranties void if used. Could result in delight or despair.' Below this are four workshop cards. The first card is 'Game Lab: Dino Jump' with a description 'Make a Game in the Hack Club Game Lab' and a thumbnail showing a game interface. The second card is 'Jamming with Muse' with a description 'Compose music with code' and a thumbnail showing a code editor. The third card is 'Reverse Workshop: Stressed Ball' with a description 'learn to code like real programmers, by looking everything up' and a thumbnail showing a pink ball on a purple background. The fourth card is 'Reverse Workshop: Generative Art' with a description 'use a toolbox of examples to compose dynamic art pieces' and a thumbnail showing a colorful fractal pattern.

Hack Club Workshops

Learn to code with this collection of community-contributed, self-guided coding tutorials + ideas.

[Our Philosophy >](#)

 **Help fill the Game Lab Arcade!**
Orpheus needs your little games to travel back to the present!


Hack Lab

Workshops on the bleeding edge of development. All warranties void if used. Could result in delight or despair.


Game Lab: Dino Jump
Make a Game in the Hack Club Game Lab



Jamming with Muse
Compose music with code



Reverse Workshop: Stressed Ball
learn to code like real programmers, by looking everything up



Reverse Workshop: Generative Art
use a toolbox of examples to compose dynamic art pieces



<https://workshops.hackclub.com/>

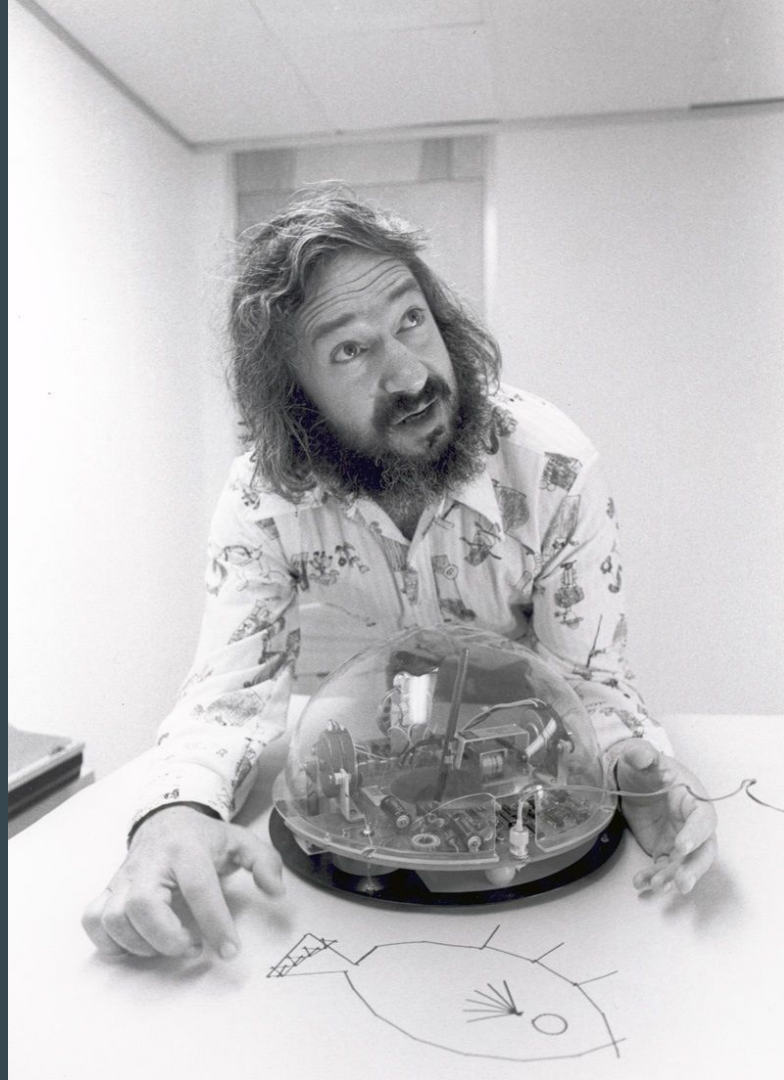
The Expository Model



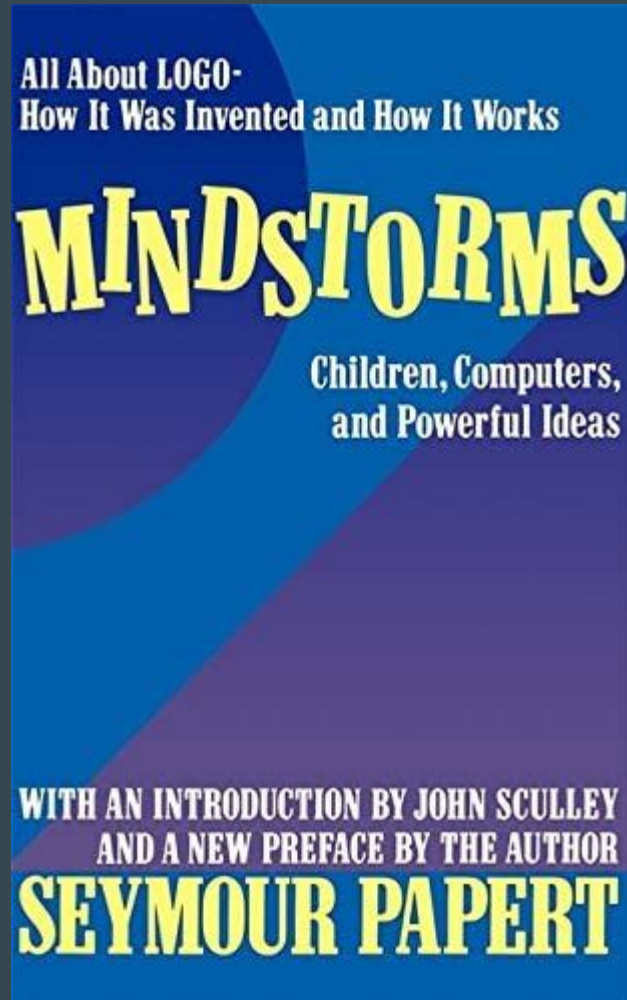
Piaget and the Bourbaki



Seymour Papert



Constructionism



Microworlds

“A growing place for a specific species of powerful idea or intellectual structure.”

Papert's Microworld Criteria

- Relate ideas to something you know
- Give the tools to make those ideas your own
- Offer “the possibility of activities that make activity matter (games, music, etc.)”
- All needed concepts should be available within the experience of the microworld

The First Microworld: Turtle Geometry



<https://microworlds.hackclub.dev/?file=turtle>

Newton's Turtles

- Newton's Laws (2/3)
 - The Law of Inertia: a body remains at rest, or in motion at a constant speed in a straight line, unless acted upon by a force.
 - $F = ma$: when acted upon by a force the time rate of change of its momentum equals the force.
- Laws for our microworld
 - Every Turtle remains in its state of rest until compelled by a command to change that state.
 - The Turtle's speed can be modified only by commands to change speed by some quantity.

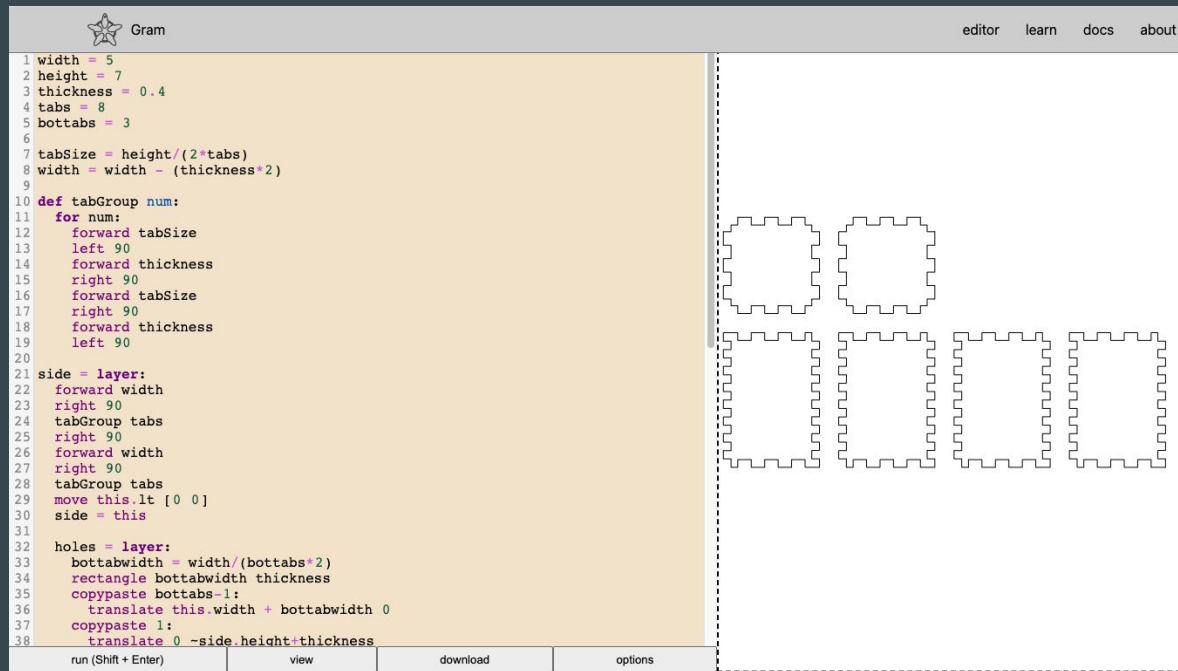
Newton's Turtles



<https://microworlds.hackclub.dev/?file=newts>

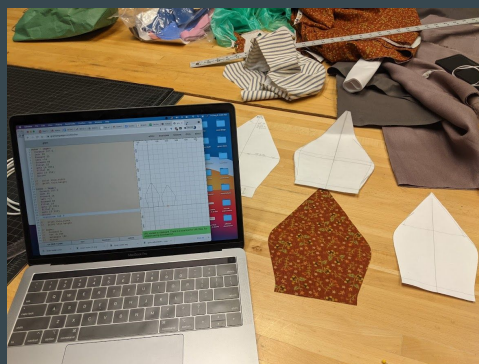
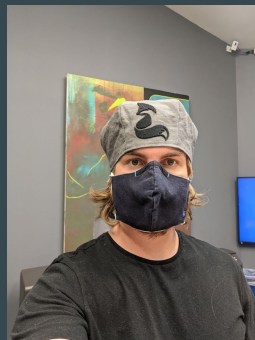
Using code to make things people want and understand.

Gram Language

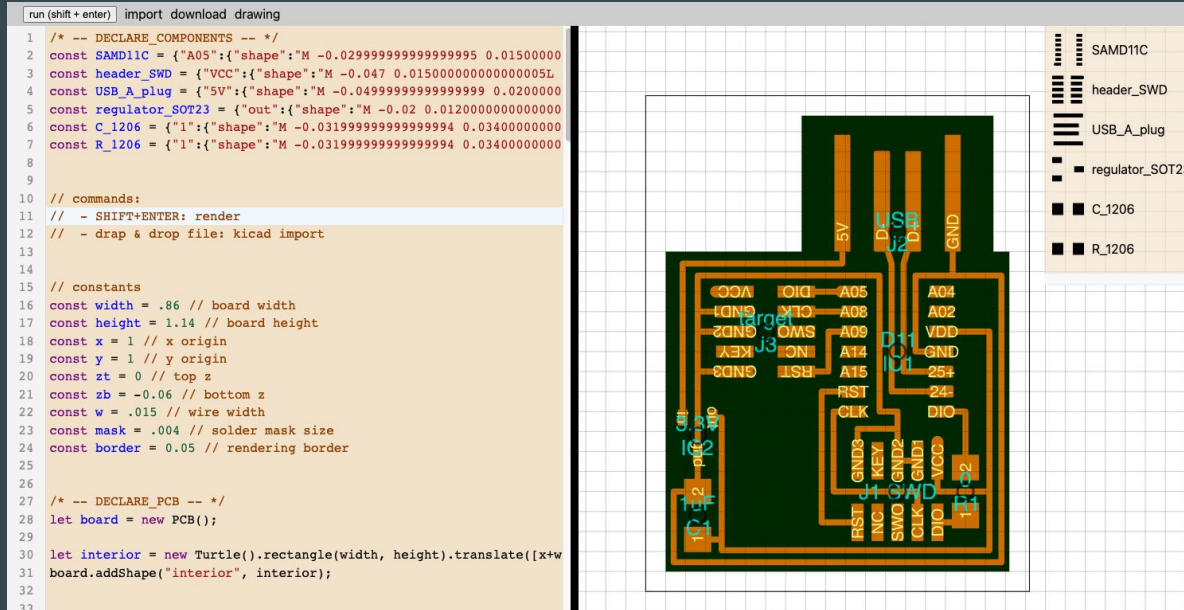


<https://gramlanguage.com/#/home>

Gram Language: Projects

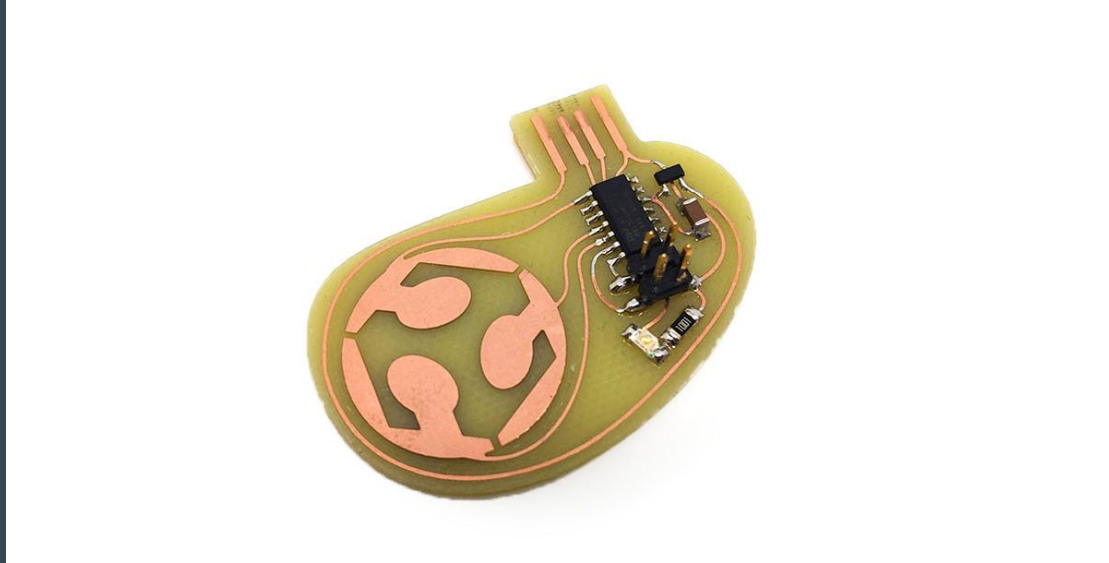


SVG PCB



<https://leomcelroy.com/svg-pcb-website/#/home>

SVG PCB: Projects



Muse Language

```
1 createMuse({ bpm: 110, type: "square" }).play`
2
3 [
4   [a3 c4 e4 a4]++++ ;;;
5   [f3 a3 f4 a4]++++ ;;;
6   [d3 a3 c4 a4]++++ ;;;
7   [c4 e4 g4 c5]++++ ;;;
8   [b3 d4 g4 b4]++++ ;;;
9 ] x 4
10
11 `
12 createMuse({ bpm: 110, type: "sawtooth" }).play`
13
14 [;;;] x 5
15
16
17 ;
18 [
19   [a1 a1;- a1 a1;- a1 a1;- a1 a1;- ;;] x 2
20
21   [d1 d1;- d1 d1;- d1 d1;- d1 d1;- ;;]
22
23   [d1 d1;- d1 d1;- d1 d1;- d1 d1;- ;;] ^2
24
25   [d1 d1;- d1 d1;- d1 d1;- d1 d1;- ;;] ^2
26 ] x 3
27
28 `
29
30 createMuse({ bpm: 110, type: "triangle" }).play`
```

Played Notes Clear Sounds

ff#4 a4 d4; d5; e5; c4 e4 g4; a4;
c5; g4 c4 e4; a5; g5; g4 c4 e4;
ff#5; e5; b3 d4 ff#4; c5; d5; d4 a4
ff#4; e5; ff#5; ff#4 a4 d4; d5; e5;
c4 e4 g4; a4; c5; g4 c4 e4; a5; g5
; g4 c4 e4; ff#5; e5; b3 d4 ff#4; c5;
d5; d4 a4 ff#4; e5; ff#5; ff#4 a4 d4;
d5; e5; a3 c4 e4 a4; ; ; ; ; ; ; ; ; ;
; f3 a3 f4 a4; ; ; ; ; ; ; ; ; ; d3 a3
c4 a4; ; ; ; ; ; ; ; ; ; c4 e4 g4 c5;
; ; ; ; ; ; ; ; ; ; b3 d4 g4 b4; ; ; ; ;
; ; ; ; ;

Samples ☒ Record new sample

bubbles

clay

confetti

corona

dottedspiral

flash1

flash2

flash3

glimmer

moon

stacked

states

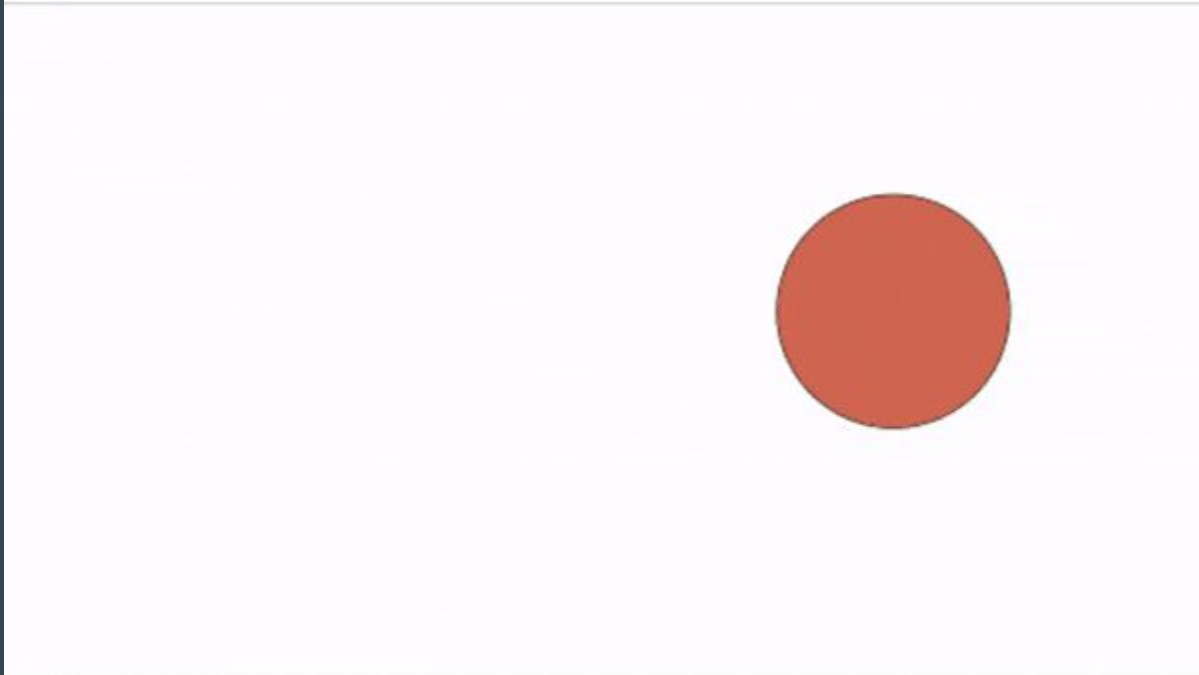
[play/attach](#) [stop](#) [share](#) [examples](#) [light/dark](#) [GitHub](#)

<https://muse.hackclub.com/>

Muse Tunes!

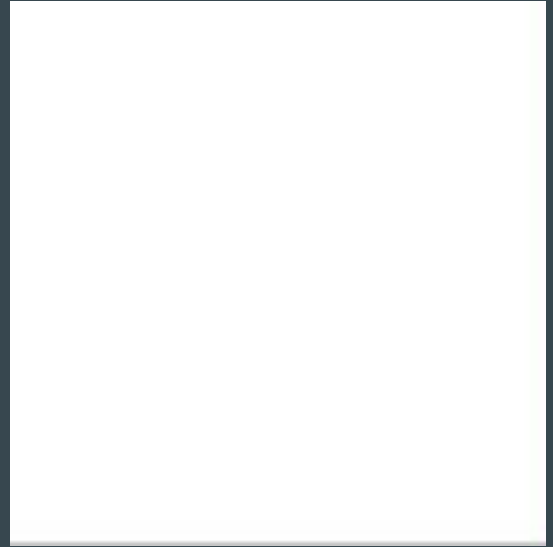
- Fill-Up Glassy
- Circus
- Twinkle-Twinkle
- Samples Sample
- Starter

Reverse Workshop: The Stressed Ball

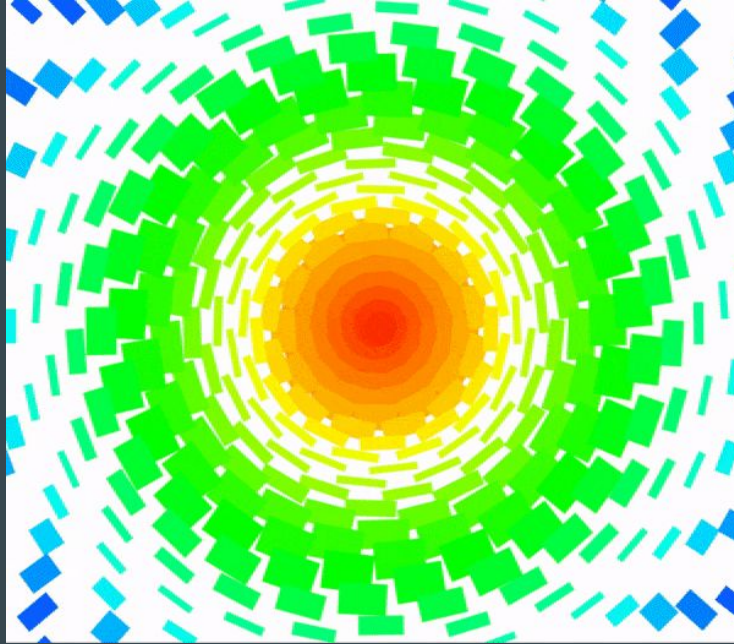


https://workshops.hackclub.com/stressed_ball/

Stressed Ball: Projects



Reverse Workshop: Generative Art



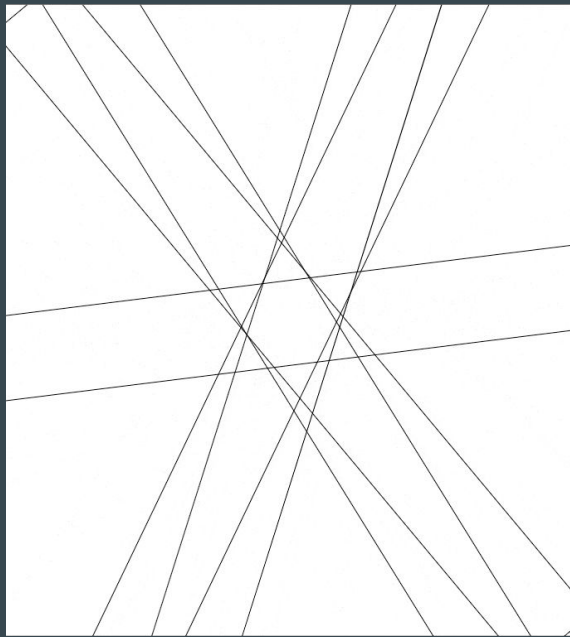
https://workshops.hackclub.com/reverse_workshop_generative_art/

Generative Art: Example Sets

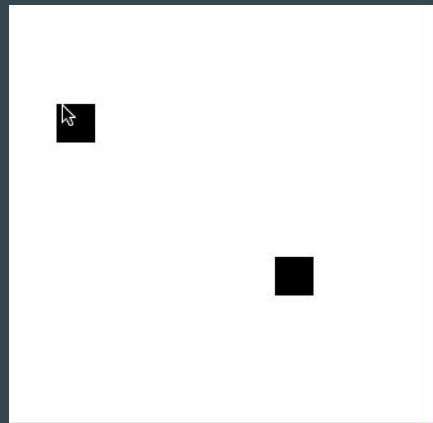
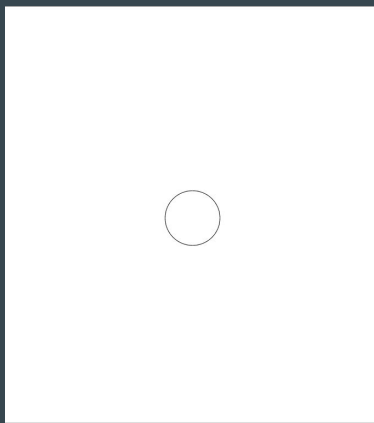
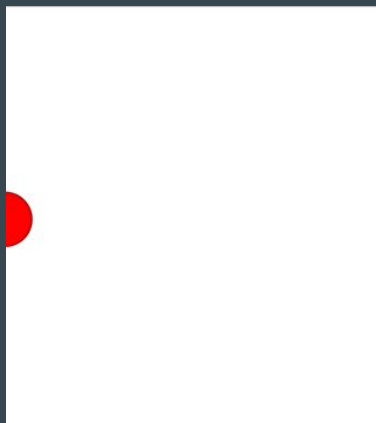


<https://hackclub.github.io/js-canvas-examples/>

Generative Art: Projects



Generative Art Projects → Games



Tiny Game Engine

A minimal template can be found in the [live editor](#) under "Examples" -> "Game Template".
A more full example game can be found under "Examples" -> "Full Game Example" or [here](#).
The source and more documentation is available on [GitHub](#).

DRAW PLAYER

```
const e = new Engine(canvas);
const ctx = e.ctx;

e.add({
  x: 150,
  y: 150,
  draw: (obj) => {
    ctx.fillStyle = "blue";
    ctx.fillRect(obj.x, obj.y, 20, 20)
  },
});

e.start();
```

run

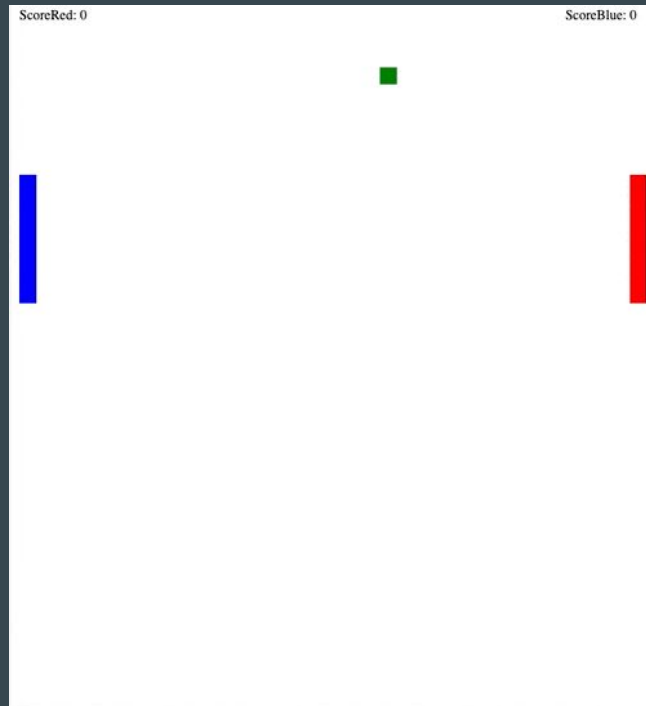
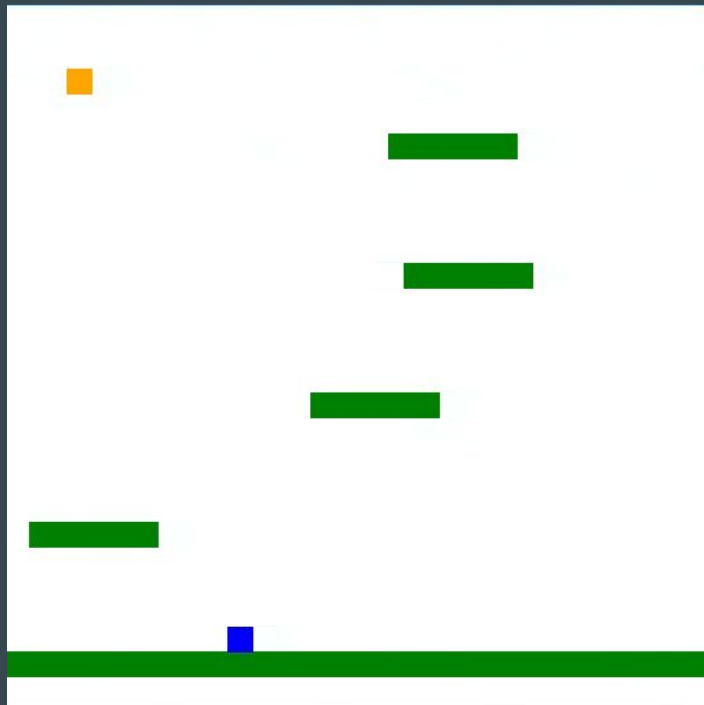
ADD GRAVITY

```
const e = new Engine(canvas);
const ctx = e.ctx;

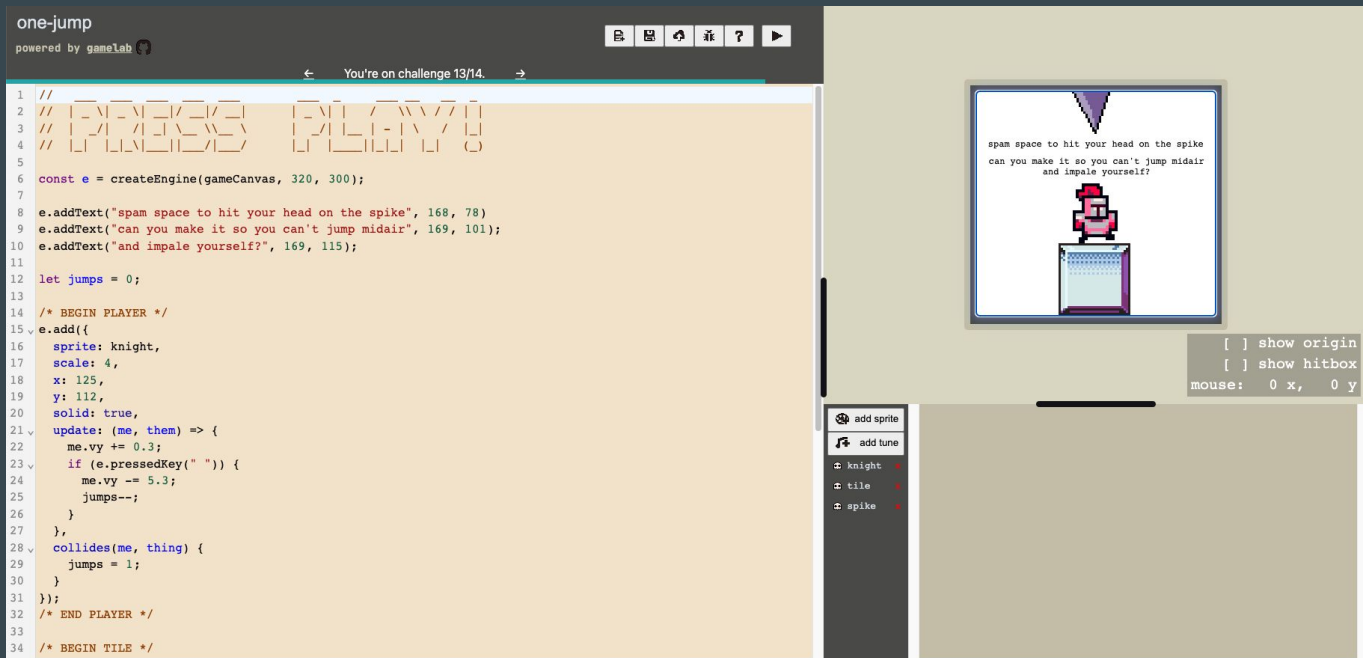
e.add({
  x: 150,
```

<https://hackclub.github.io/mini-engine-examples/>

Tiny Game Engine: Projects

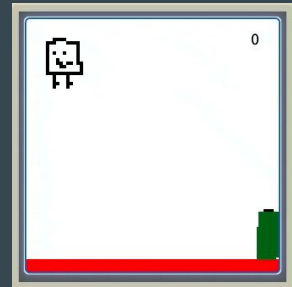
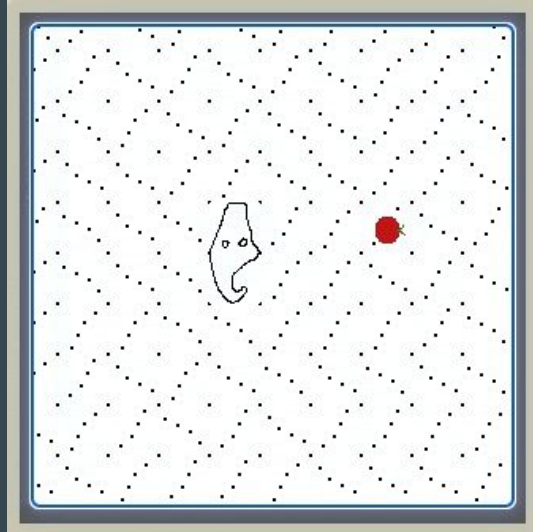
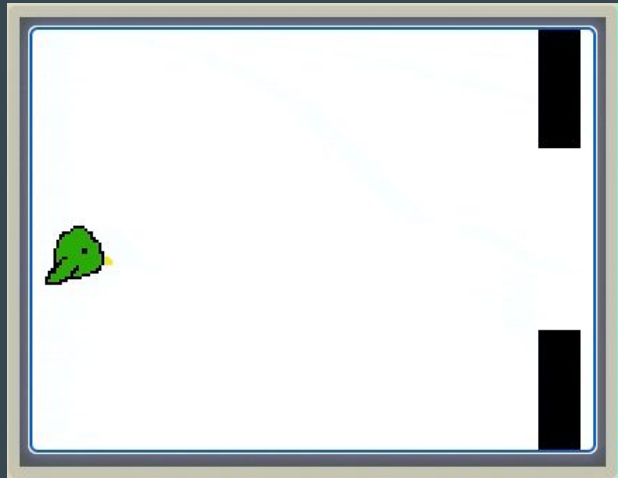


Microworld: Game Lab



<https://gamelab.hackclub.com/>

Game Lab: Projects



The Construction Kit

Can be

- Language
- Library

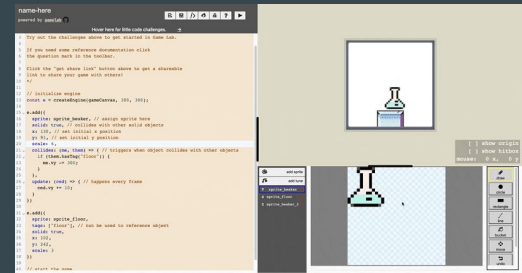
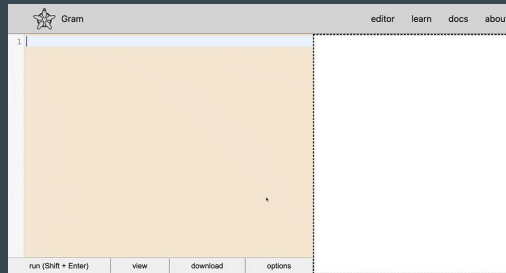
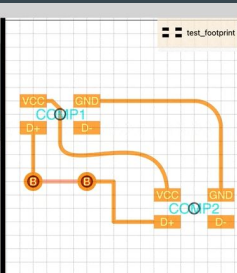
Should be

- Small, “mind-sized bites”
 - You should be able to hold it all in your head
- Composable
- Sufficiently specific to define a “species” yet sufficiently abstract to allow expression

The Environment (Reactivity)

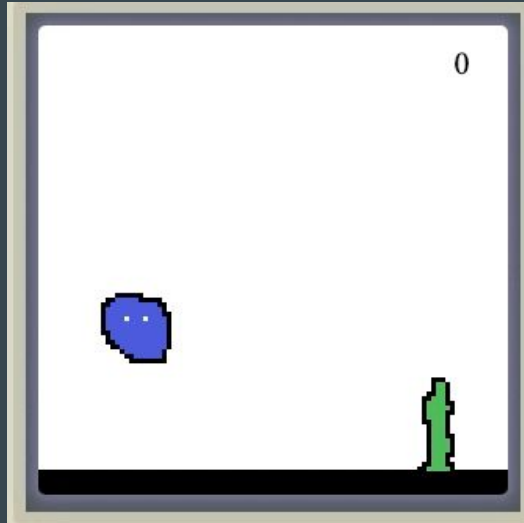
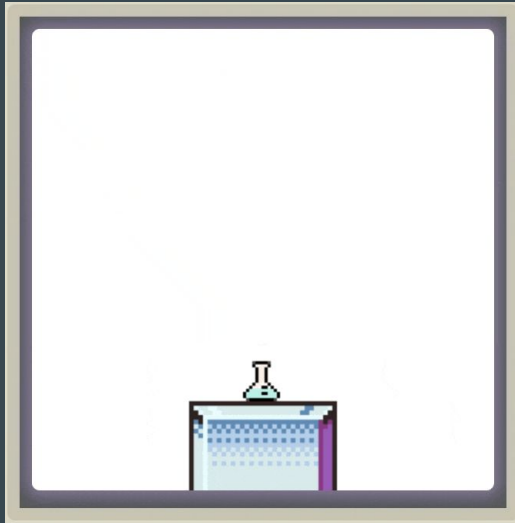
- Make ideas relatable
- Make activity matter
- Help the learner understand what they said
- Help them say what the mean

```
run(shift+enter) import download drawing
19 // mac: Command + Option + J
20 // Windows/Linux: Shift + CTRL + J
21
22 // included: Turtle, PCB, pcb
23
24 /* -- DECLARE PCB -- */
25 let board = new PCB();
26
27 /* -- ADD_COMPONENTS -- */
28 let test_comp1 = board.add(test_footprint, {translat
29 let test_comp2 = board.add(test_footprint, {translat
30 let v1 = board.add(via(0.02, 0.035), {translate: {t
31 let v2 = board.add(via(0.02, 0.035), {translate: {t
32
33
34
35
36
37
38
39
40
41
42 /* -- ADD_WIRES -- */
43 board.wire([test_comp1.pad("D+"),
44 v1.pos], 0.015)
45
46 board.wire([v1.pos,
47 v2.pos], 0.015, "B.Cu")
48
49 board.wire([v2.pos,
50 [v2.posX+0.1, v2.posY],
```



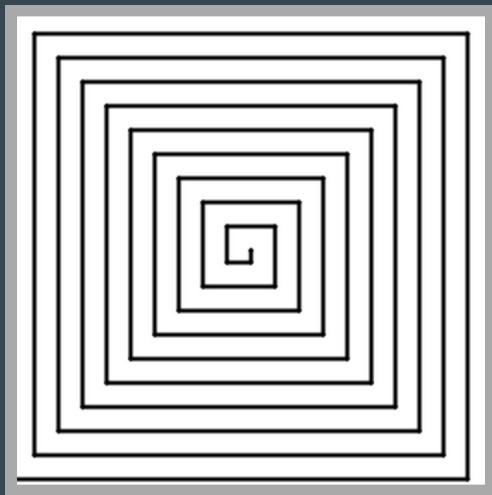
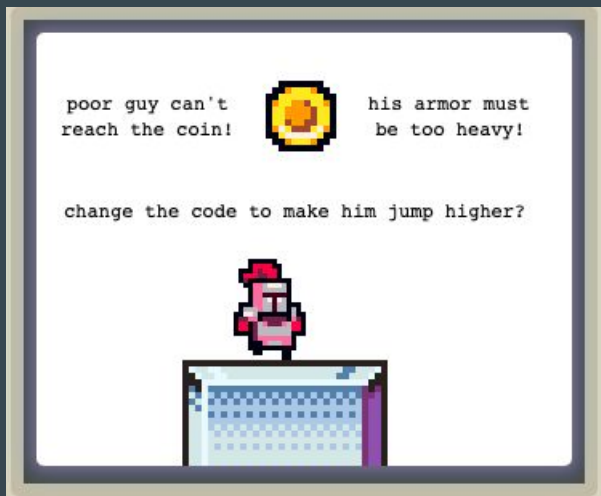
Examples and Documentation

- Show us useful patterns
- Give us starting points
- Inspire us



Prompts/Challenges

- Give a quest
- Don't have to get where you direct them to, just need to get going



Sharing

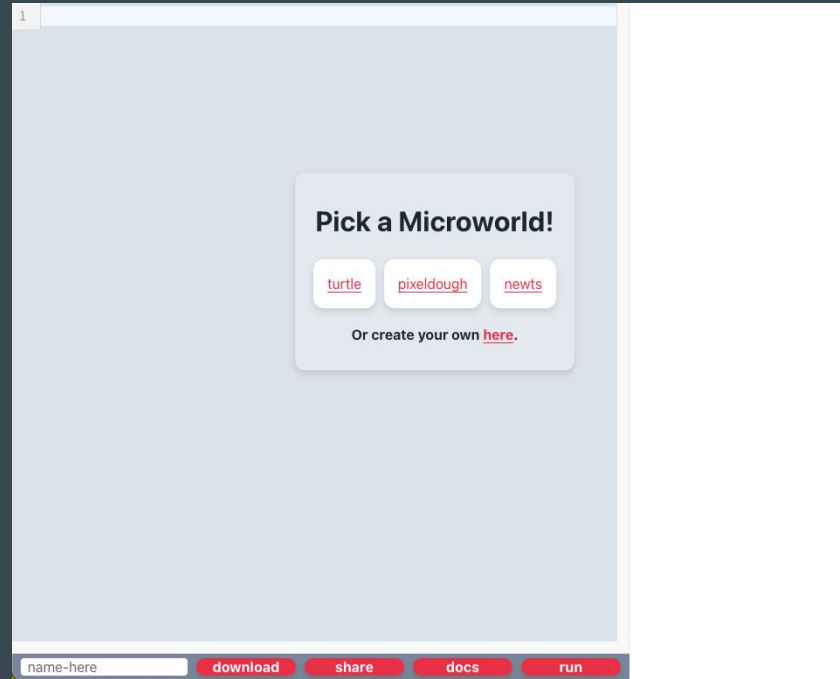
- Make it easy to connect with a community



What's in our microworld?

- Powerful ideas
- A construction kit
- An environment
 - Helps us relate to ideas
 - Reflect on expressions
- Explanation on how to use our kit as examples
 - Shows useful patterns
 - Gives us a place to start
 - Inspires us
- A prompt to give us direction when starting
- A community to provide feedback and draw inspiration from

A Meta-Microworld



<https://microworlds.hackclub.dev/>

Thank you! Questions?

- leomcelroy.com
- Everything we saw today is open-source
 - <https://github.com/hackclub/>
 - <https://github.com/leomcelroy/>