Math Manipulatives: a micro:bit method

Mike Deutsch
Kids Code Jeunesse
Montreal, QC, Canada

KCJ’s mission

KCJ is a bilingual Canadian charity determined to give every Canadian child access to digital skills education, with a focus on girls and underserved communities.

We encourage inclusive and sustainable learning by teaching kids and the educators that play a crucial role in their development. We’re making sure our kids have the confidence and creative tools they need to build a better future.

About me

Mike Deutsch

Tiohtià:ke (Montréal)
Kanien’kehá:ka land

Director of Learning Services, KCJ

1st career: CS, 20y in industry, *near* education

2nd career *in* education, KCJ and an MA (math/CS) at McGill U

“Sensemaking” approach to CS

@kidscoding
@mdeutschMTL

Math manipulatives: micro:bit method

Who are you?

- You do **CS integration**, or support **subject area Ts** with coding.
- You have micro:bits you can use, or are curious about them.

And *maybe*...

- “Manipulative” = pedagogical 🤩.

?

Math manipulatives: micro:bit method

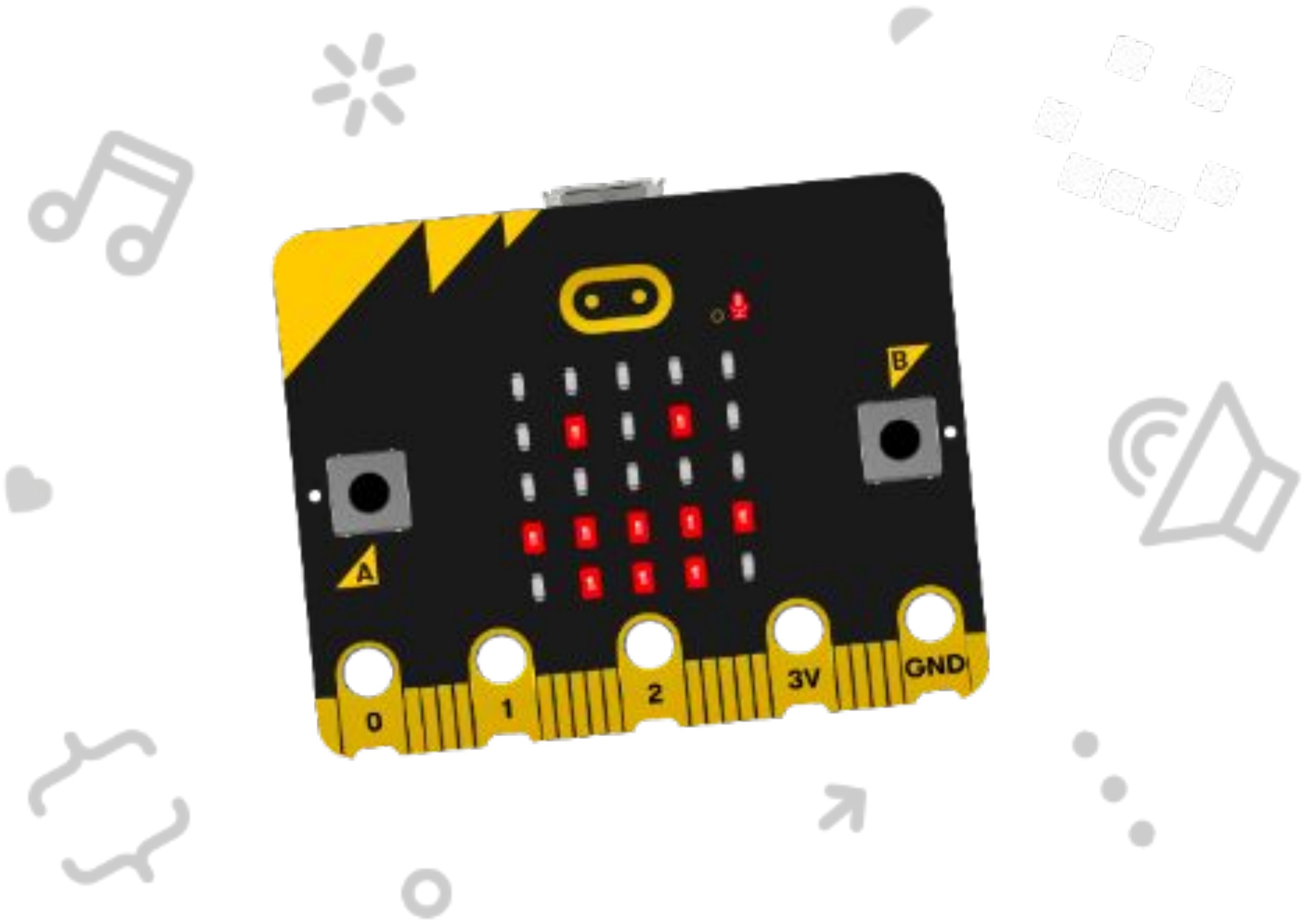
In this session I will…

- Ground us with math+cs pedagogy
- Explore 4 flexible starter projects to think with in grades ~3-6
  keywords: number sense, operations, patterning, algebra

I hope you will…

- Build along with me
- Make sense of things, and maybe flop around a bit 🙃
- Add your knowledge to the space

Math manipulatives: micro:bit method



All we need:

Memory + processor
A + B buttons
LED “screen”

Image: microbit.org

Pedagogy: “Manipulatives” in Math+CS

Manipulatives in math + CS

Pedagogical themes in this session:

Productive **integration of CS + Subject**; how adding concepts & practices from CS can unlock concepts & practices in the other subject.

Manipulatives; how they help learners make sense of things. Helps them move from Concrete → Representational → Abstract.

Manipulatives in math

Manipulatives from K-4

Counting objects
Abaci
Cuisenaire rods
Base 10 cubes - rods - flats
Tangrams

pearsoncanadaschool.com

Concrete → *Representational* → *Abstract*

Manipulatives in math

Elementary math progression:

Counting
→ Counting on
→ Addition
→ Skip counting
→ Repeated addition
→ Doubling
→ Models of multiplication
→ Factors and multiples
→ Standard algorithms

NCTM:
Number and Operations
Number & Operations in Base 10

Common Core:
Operations & Algebraic thinking

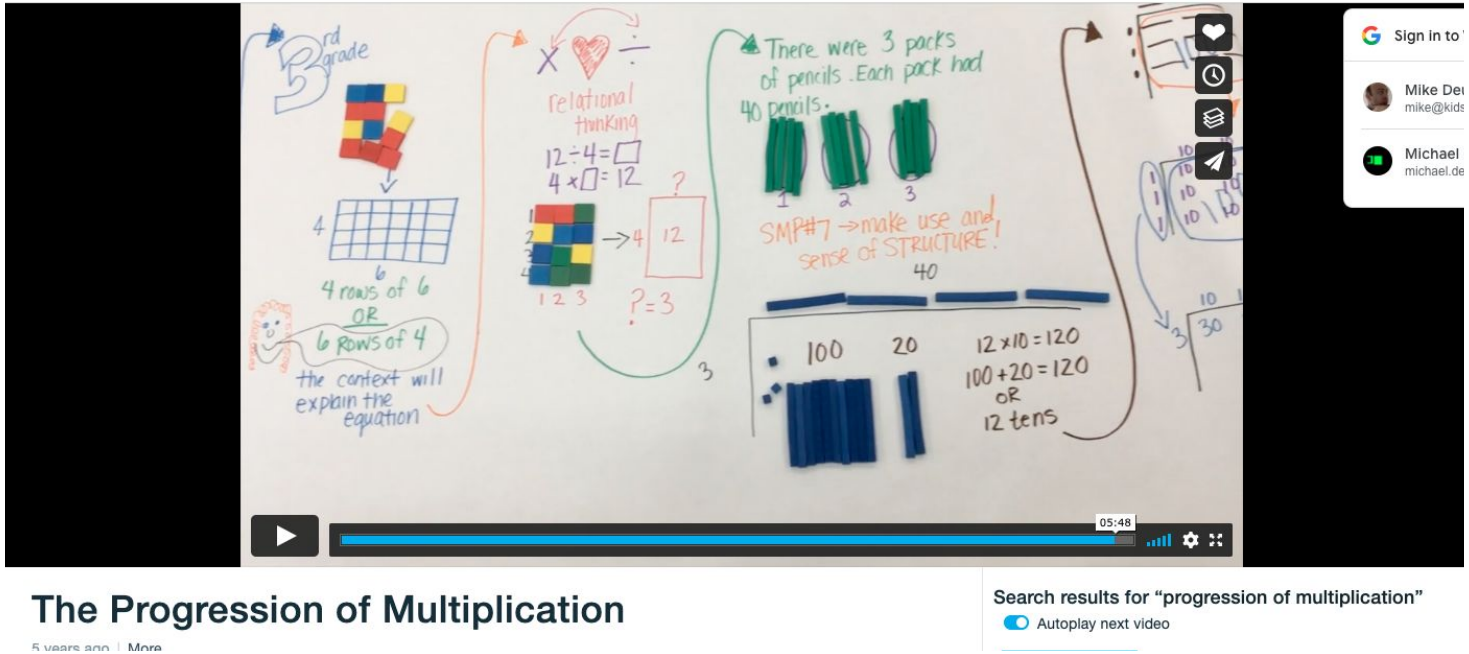
familiar?

Concrete → Representational → Abstract

Manipulatives in math

Graham Fletcher: The progression videos

https://gfletchy.com/progression-videos/



The Progression of Multiplication

5 years ago | More

Search results for "progression of multiplication"

Autoplay next video

Concrete → Representational → Abstract

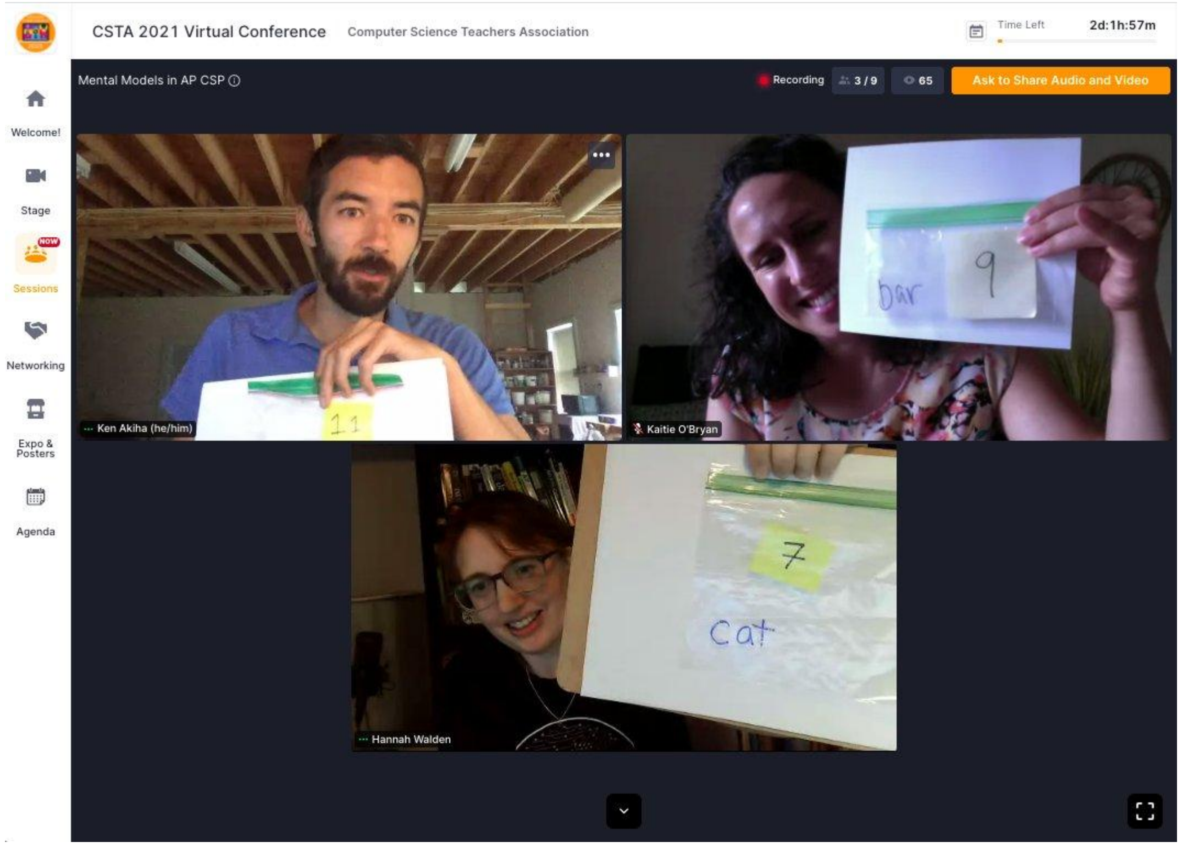
Manipulatives in CS

Manipulatives in CT/CS

Unplugged “instruction” games, Turtles, Beebots, Sphero

Unplugged misc: variables, data, algorithms, etc.

got a fave?



code.org team, CSTA 2021

Concrete → Representational → Abstract

Manipulatives in math and CS

In this spirit… let’s make some manipulatives

4 flexible projects, growing slowly in complexity,
For teachers and students to build and then use in math class.

Math and CS are co-equal.
CS is never put first, and it can’t distract too much.

Not “what clever things this tool can do.”
Rather: “the clever thinking I can do, equipped with this tool I made.”

By adding CS we augment or modify the math we can access.
We can create CS-based manipulatives for exactly the math we want to do.

Manipulatives in math

Elementary math progression:

I build → click counter

We talk → skip counter

You build → add + double

We skim? → function machine

Counting

Counting on

Addition

Skip counting

Repeated addition

Doubling

Models of multiplication

Factors and multiples

Standard algorithms

Building manipulatives

Manipulatives in math

Elementary math progression:

I build →

click counter

skip counter

add + double

function machine

- Counting
  - Counting on
    - Addition
      - Skip counting
        - Repeated addition
          - Doubling
            - Models of multiplication
              - Factors and multiples
                - Standard algorithms

1. Simple click counter

→ makecode.microbit.org

Prompts & tasks

“Can we build a micro:bit that *counts*?”

Keep score

Count off steps (measurement)

Act out simple addition problems

Algorithm

- Start with zero
- When clicked, add 1
- Show the value

Manipulatives in math

Elementary math progression:

click counter

We talk → skip counter

add + double

function machine

Counting

▷ Counting on

▷ Addition

▷ Skip counting

▷ Repeated addition

Doubling

Models of multiplication

Factors and multiples

Standard algorithms

2. Skip counter

Prompts & tasks

“Can we make this a *skip counter*?”

Practice addition (skip counting)

Can you get to 24 with just 2s? How many does it take?
(repeated addition)

What are the different ways to get to 12?

“Reach 21” game

Algorithm

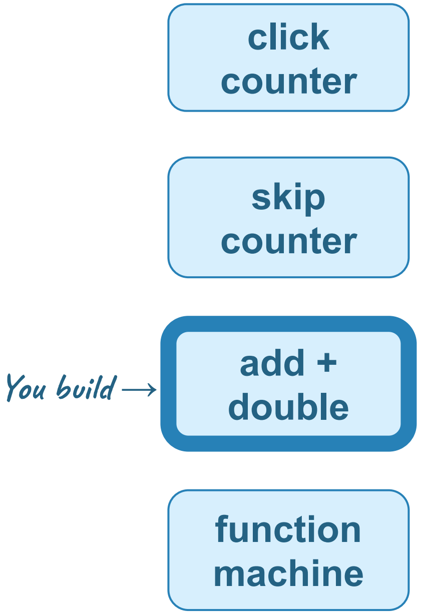
- Start with zero
- When A clicked, add [#] and show the value
- When B clicked, add [#’] and show the value

Let’s try my pre-built skip counter.

What’s another good target number? Why? What would be a skip counter that could reach that number in at least two ways?

Manipulatives in math

Elementary math progression:



Counting

▷ Counting on

Addition

Skip counting

Repeated addition

▷ Doubling

Models of multiplication

Factors and multiples

Standard algorithms

3. Add and double

Prompts & tasks

“Can we make this an adder *and* a doubler?”

Practice doubling

Can you get to 25 just by adding 1 and doubling?

What are the different ways we can get to 25?

Algorithm

- Start with zero
- When A clicked, add 1 and show the value
- When B clicked, double and show the value

Let’s build. Follow my algorithm. ☝️

What’s a good target number? Why? Make an add-and-doubler, and reach the target number in two ways.

3. Add and double

A 4th grade transcript…

Q: How can you get to 12?
A: count up to 6, then double it.

Q: Great. Can you give me another way?
A: ok, count to 3 and quadruple it.
Q: What do you mean, quadruple?
A: Um, ok, get to 3. Then double two times.

Q: Another way?
A: Ok, count up to 4. Double it. (That's 8.) Double it again -- *oh, wait*. You can't!
Q: riiight…
A: Ok, just count the rest of the way. 4 more to 12.

Q: Up to 11?
A: Count to 4. Double it (to 8). Count the rest of the way.

4. Function machine

Elementary math progression:

click counter

skip counter

add + double

We skim? → function machine

Counting

Counting on

Addition

Skip counting

Repeated addition

Doubling

Models of multiplication

Factors and multiples

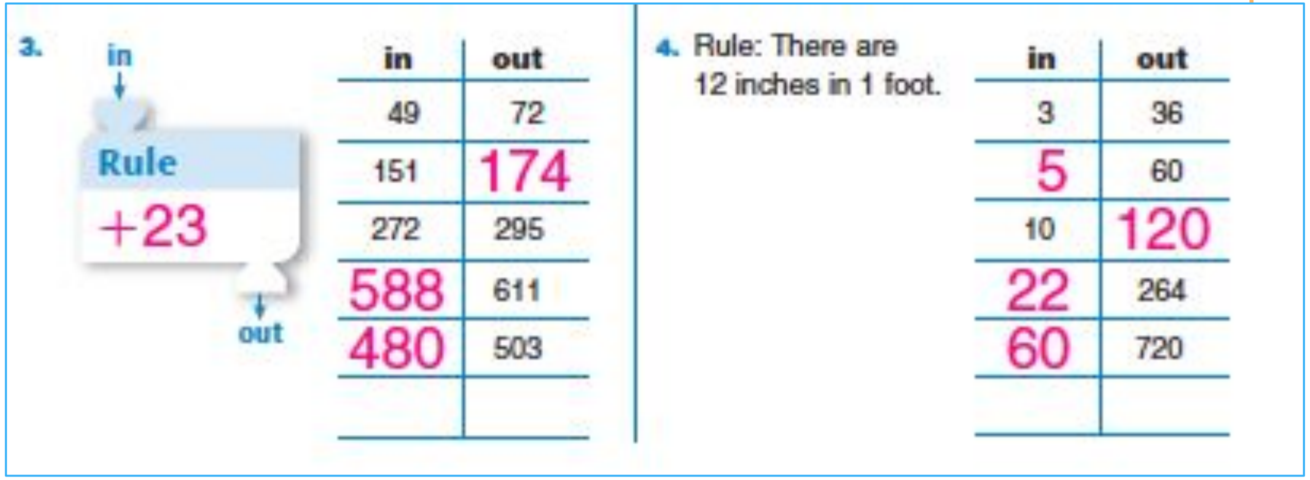
Standard algorithms

4. Function machine

Prompts & tasks

Build a function machine with a mystery function. “What’s my rule?” See if your neighbour can figure it out.

Extend with more sophisticated math operators.



3.

in

Rule

+23

out

in	out
49	72
151	174
272	295
588	611
480	503

4. Rule: There are 12 inches in 1 foot.

in	out
3	36
5	60
10	120
22	264
60	720

everydaymath.uchicago.edu

Algorithm

- Start with zero as the “input”
- A clicked: <u>+1 to input</u>, then show
- B clicked: <u>calculate output</u>, then show

Let’s try my pre-built function machine. What’s my rule?

What (integer) operations could <u>you</u> put in a mystery function for your students? Make a function machine we can solve.

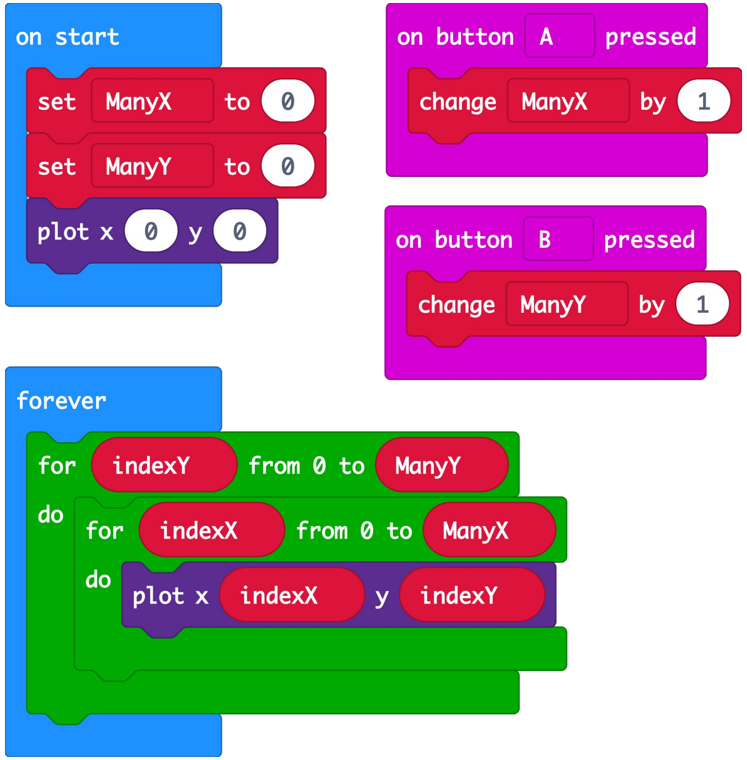
Others...

We like these too:

- “Flash card” type
- Plotting x/y
- Randomness
- Rock-paper-scissors

but we’ve retired (or no longer use in this way):

- Fizz buzz
- Build a single digit (place value)
- Visual LED multiplication



```
on start
    set ManyX to 0
    set ManyY to 0
    plot x 0 y 0

on button A pressed
    change ManyX by 1

on button B pressed
    change ManyY by 1

forever
    for indexY from 0 to ManyY
    do
        for indexX from 0 to ManyX
        do
            plot x indexX y indexY
```

Recap & discussion

Recap

I said we’d look at:

click counter

skip counter

add + double

function machine

4 flexible projects, building slowly in complexity,
For teachers and students to build and then use in math class.

Math and CS are co-equal.
CS is never put first, and it can’t distract too much.

Not “what clever things this tool can do.”
Rather: “the clever thinking I can do, equipped with this tool I made.”

By adding CS we Augment or Modify the math we can access.
We can create manipulatives for exactly the math we want to do.

Recap

Special thanks to:

Sophie Lawi, Kids Code Jeunesse instructor
who has honed these activities extensively with Ss and PSTs.

Jared O'Leary @Jared_OLeary
whose csk8 podcast (Feb 2022) provided extra theoretical framework.
http://jaredoleary.com/csk8

Recap

Discussion…

Do these DIY tools connect to anything you do?

What correction, addition, or subtlety would improve these? (pick any)

You came in to **learn…**
What are you ready to **create?**

Connect with us!

@kidscoding

kidscodejeunesse

kidscodejeunesse.org

#kids2030

Thank you!

This document is strictly private, confidential, personal to its recipient and remains the property of Kids Code Jeunesse, and should not be copied, distributed or reproduced in whole or in part, nor passed to any third party for any reason whatsoever.

Math Manipulatives: a micro:bit method

Mike Deutsch
Kids Code Jeunesse
Montreal, QC, Canada

Sensemaking: Ideas & intent

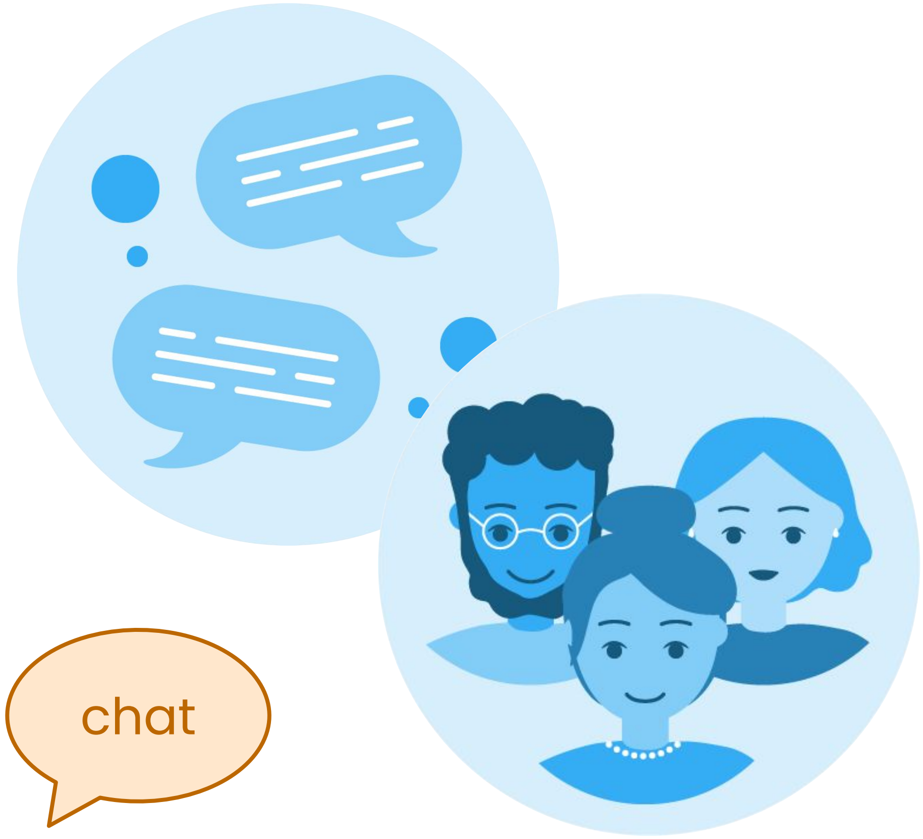
We have found:

Start with **level playing field**.

We all get access to
other people's thinking.

We fill in gaps, round out,
formalize our own **understanding**.

Gain additional **strategies for building**.



Sensemaking: Ideas & intent

Sense-making:*

- Conceptual understanding
- Multiple, flexible strategies

“Does this make sense?”

** Theory: social constructivism.*

Answer-getting:

- Procedural competence
- Narrow, inflexible strategies

“Did I do it right?”

chat