

The title slide features a black background with the word 'CODING' in large, orange and blue block letters at the top. Below it is the word 'to' in orange, followed by 'LEARN' in blue and 'and' in orange. At the bottom is the word 'CREATE' in large, blue block letters. Two white chevrons point upwards from the bottom of 'CREATE'. A small white star is centered below the chevrons.

CODING to LEARN and CREATE

New Modes of Programming for Learners
Who Have Been Left Out

Colin Clark

Associate Director, Inclusive Design Research Centre
OCAD University

Simon Bates

Sepideh Shahi

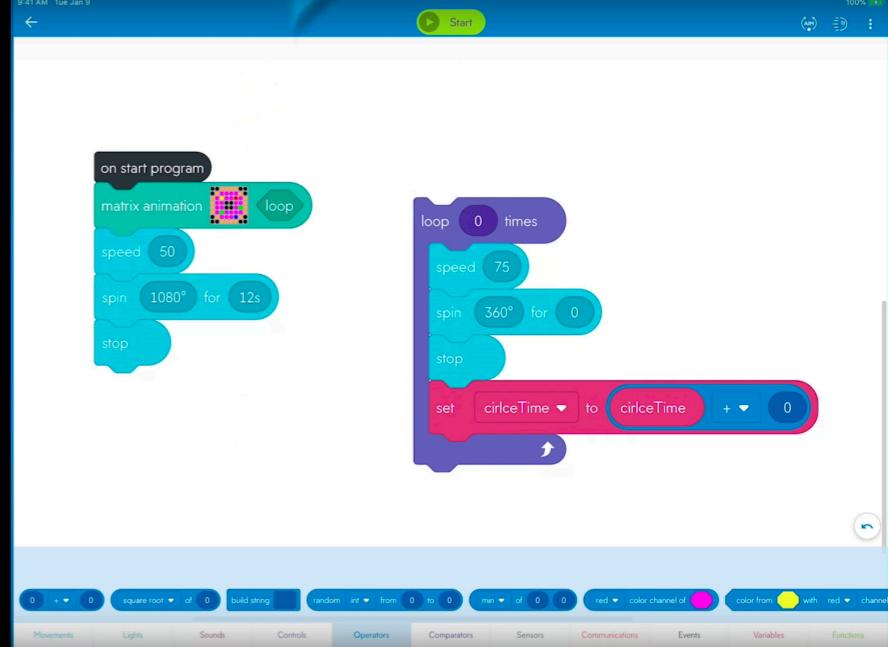
Clayton Lewis

Inclusive Learning

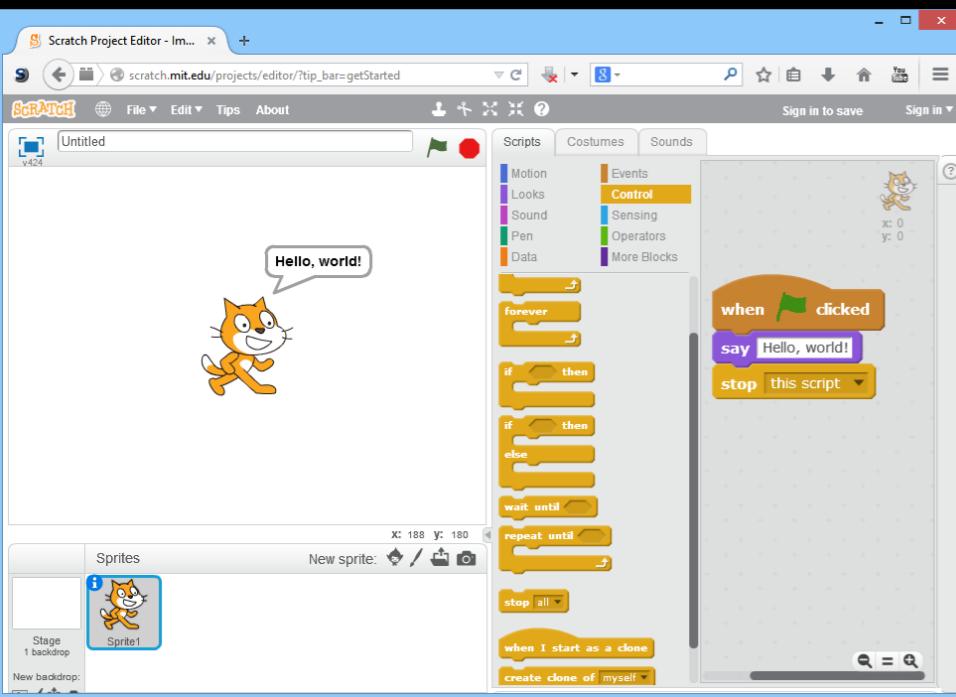
- Too often, we invoke and obsess over a **mythical norm**: the typical learner, standardizable teaching and assessment, the “right way” to teach and learn
- **Learners learn differently**, and students with disabilities challenge the normative status quo
- Diversity is a catalyst for innovation; **we should shift our attention and prioritization to the edge**, the outliers

Ursula Franklin, Technological Orthodoxies

"Prescriptive technologies are designs for compliance... While we should not forget that these prescriptive technologies are often exceedingly effective and efficient, they come with an enormous social mortgage. The mortgage means that we live in a culture of compliance, that we are ever more conditioned to accept orthodoxy as normal, and to accept that there is only one way of doing 'it'."



Can't Code?



Creative Power

- Too often today, creative power is asymmetric—we can *consume*, but it's hard to *create*
- People with disabilities often depend on technology for their daily lives—yet have few means of expressing their own technologies
- **Democratic and inclusive innovation needs to provide avenues for creation, not just use**



Coding Skills == Jobs?

- *Computational thinking* is too often predicated on schematism and technocracy
- Computation should be a **material literacy**, not just “abstract thinking” or mathematics
- Especially for students with disabilities, literacy in this new medium means the **ability to express themselves**, to participate when they’d otherwise be alone
- **Life skills**—explicit instruction, collaboration, understanding cause/effect, criticality, creativity

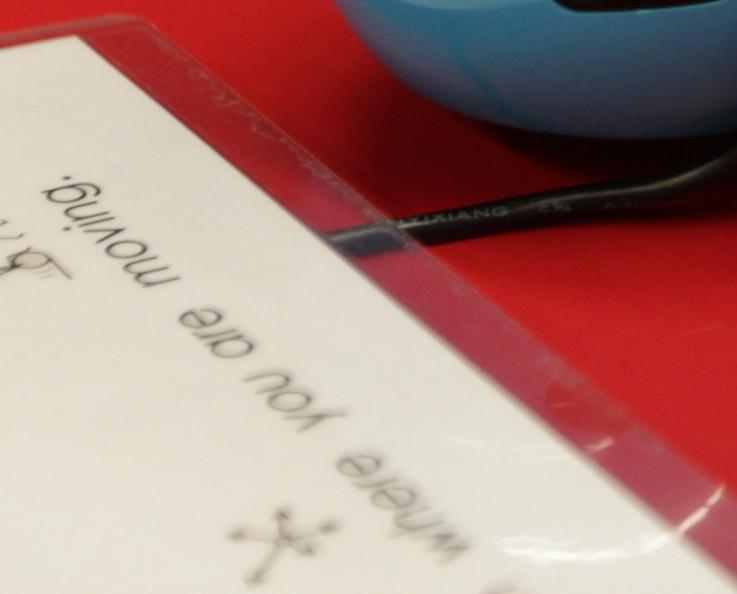
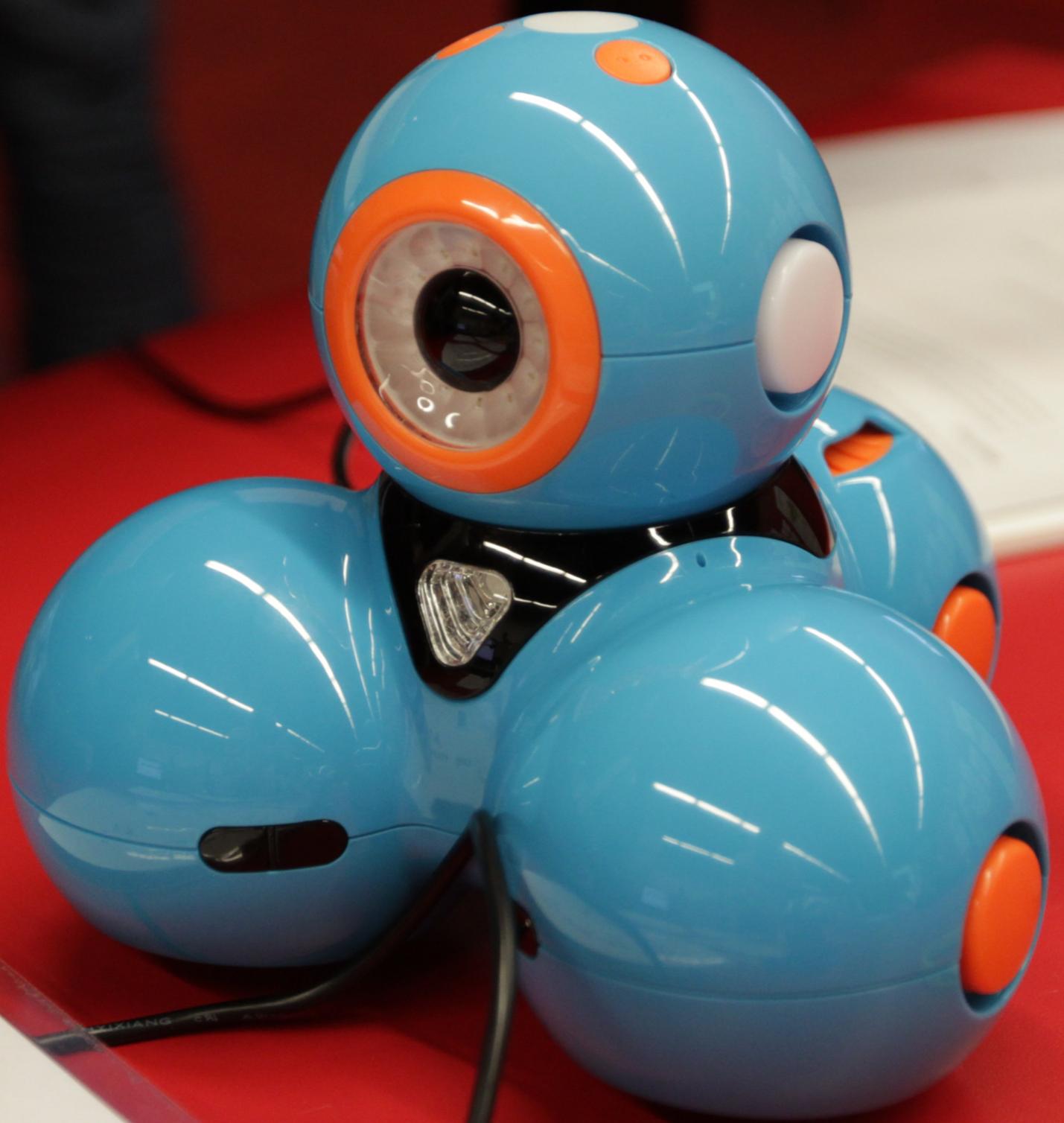
**CODING
TO LEARN
AND
CREATE**

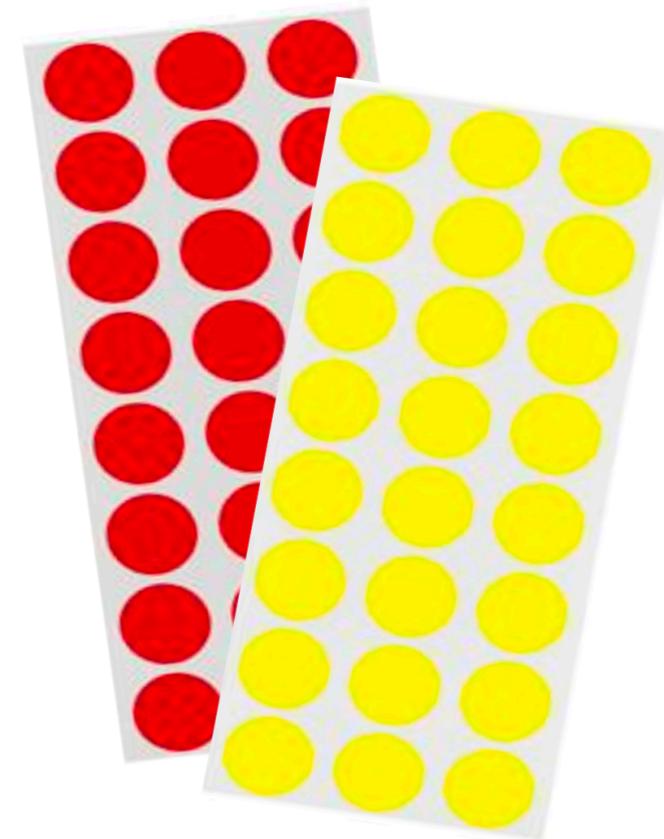


Coding to Learn and Create

1. Open educational resources—**lesson plans, arts-based activities, adaptations, and strategies for educators** to help them teach more inclusively
2. Work with the manufacturers and open source communities that create educational programming tools to **help them improve the accessibility and adaptability of their systems**
3. Create a **new inclusive coding environment** designed specifically for use by students with physical, learning, and cognitive disabilities.





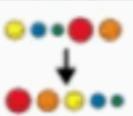


Some of the things I will learn about might be tricky to understand.

<name of person(s)> will try to make it easy for me by having me wear wristbands or stickers on my hands.



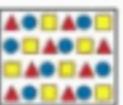
1



Create a sequence



2



Show a repeated pattern



=



3



Recreate the sequence using the function block

function



function



function

function

function

function

Working with students with disabilities raises productive questions about what coding actually is as a practice, what it's for, how it is manifested materially, and how it can be productively implicated in other learning and personal activities. By working with learners who are currently on the margins of computational creativity, and addressing the barriers that prevent them from engaging with computational media equally alongside their peers, new possibilities for programming languages and tools may be revealed that benefit all learners—and other programmers and users, too.











Co-Design and Community

- Co-design is designing **with**, not simply **for**. It involves asking the people who might otherwise just be "users," particularly those on the margins of today's technology experiences, to be part of the design process.
- Co-design typically starts with a process of **discovering and negotiating roles**—asking participants how, when, and how often they want to be involved, and making space to accommodate different “scales” of investment and engagement. It takes time.
- **Delegation, not facilitation:** “What role would you like me to play in this process?”
- It demands that all participants have **equal access to the information**—plans, ideas, prototypes, and works in progress—that is essential for full decision-making and responsible contribution.
- Not an prescriptive or instrumental process like typical methodologies
- **Dynamic, opportunistic, flexible**—go where the wild things are (and become one too)

Specific Designs First, Generalize Later

- Inversion of typical industry approach (to scaling): our co-design process aims to let learners and their teachers and families lead—we’re designing with specific classrooms and schools, starting with what they need
- **Will our students be interested in these ideas?**
- **One size won't fit all.** Our learners have a huge diversity of needs, so we'll need to explore customization and providing teachers with tools to make their own coding environment

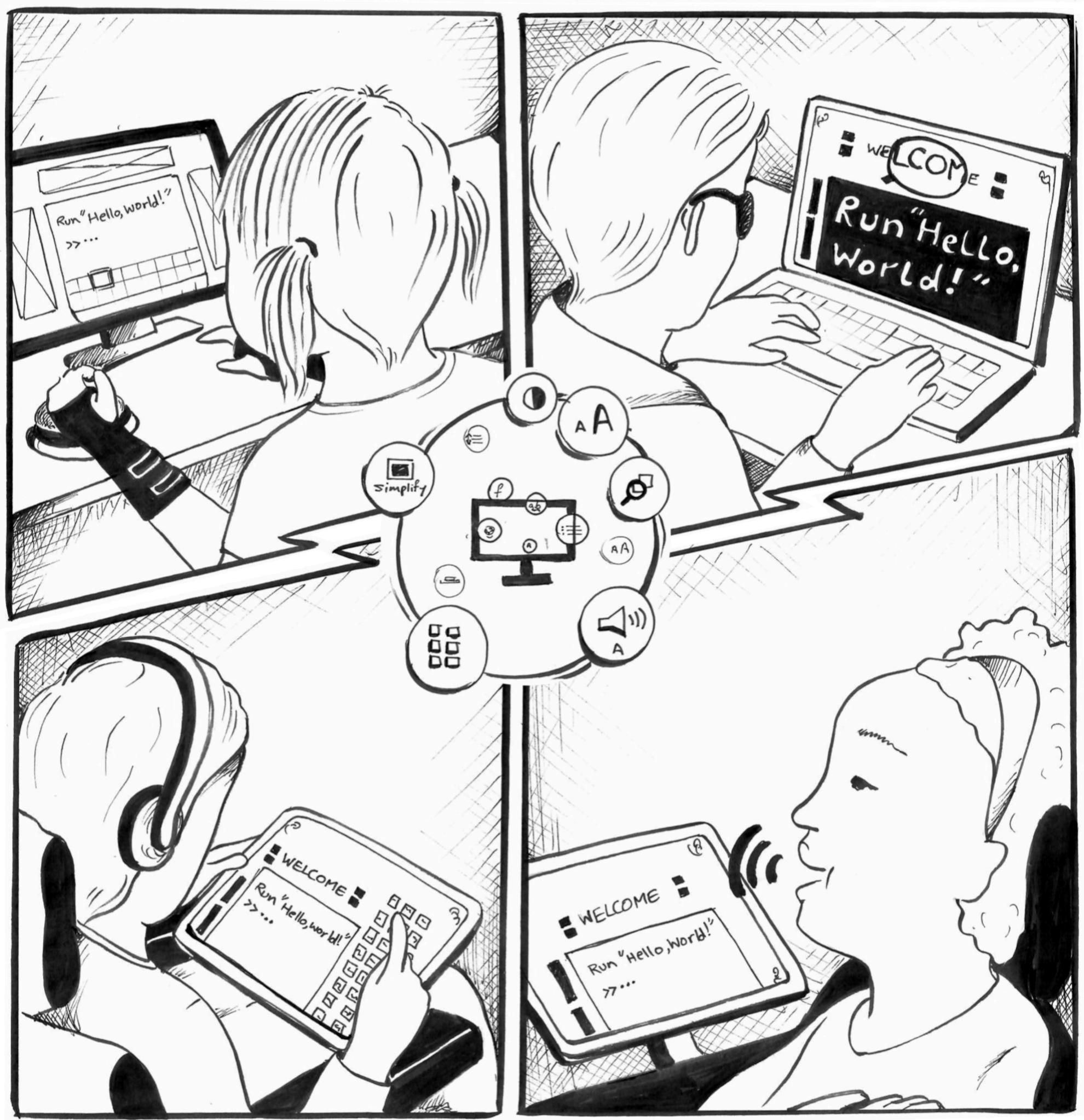
Continuous Prototyping

- Always have some working examples of what you're thinking of
- An invitation to participate: unpolished enough to prompt feedback and alternatives
- Functional enough to try out, but not so set that it limits creativity
- Start exploring the under-explored (e.g. user interfaces optimized for eye gaze or switch access instead of mouse/keyboard)



Inclusive Coding

- **Multiple simultaneous representations** of program code—visual, textual, and auditory
- **Flexible user interfaces** that support simplification, focus modes, and the ability to access additional guidance or customize the timing and layout of the system
- Teacher-driven **learning scaffolding** and unobtrusive examples, guides, video tutorials, and explicit instruction prompts, without requiring students to leave the programming environment
- Support for **assistive technologies** such as eye gaze input, reading supports, magnifiers and screen readers, etc.



What is Coding?

- Programming-by-example
- Abstraction from direct manipulation (drawing, etc.)
- Live programming
- Physical input devices: program parameters controlled by e.g. pushing or squeezing a ball
- Programming by voice
- Query-based programming (guided by answering yes/no questions, etc.)
- Time-based notations for music and visual art (loop machines, etc.)



Lessons from Camp

- Can't easily separate learning from assistive technology setup from co-design
- Attention varies, especially with kids with ADHD etc., so build in **lots of opportunities for breaks, physical activity, and changes of pace**
- Different learners, different learning approaches and levels of experience—be ready to **tailor activities on the fly** and have simpler and more complex **versions of each activity** available (particularly with regards to the level of abstraction involved)
- Coding is exciting—be ready for the **intensity and conflict** that will arise
- Scheduling and emotion support: visual and verbal explanations of the day's activities; **visual emotional regulation chart** to help learners communicate how they are feeling during the activities
- **Do offline activities before using digital interfaces:** campers benefited from creating sequences using paper pieces before interacting with the digital version of the prototype and robots. This was to encourage them to plan their sequences and not just push buttons at random.



Co-Design Prompts

- If your robot could do anything, what actions would you include?
- How do you like to build your sequence/code?
- What are the other ways you want to control your robot?
- In the viewing area, what would you like to see?

Lessons from Camp

- “Jump, run, reverse/backwards, poke, spin, light up, flash red, talk, dance, draw, draw more shapes similar to circles, roll in water, go underwater, drive a car, have a schedule, fly, fight against other robots, put bad people in jail, do everything for the camper (i.e. give him a bath, bring him a drink, do a trick, etc.) “
- “Spin, talk, repeat, make a sound, dance, jump, move diagonally, light up, say hooray, **make a burp sound**”





Demo



ACTIONS



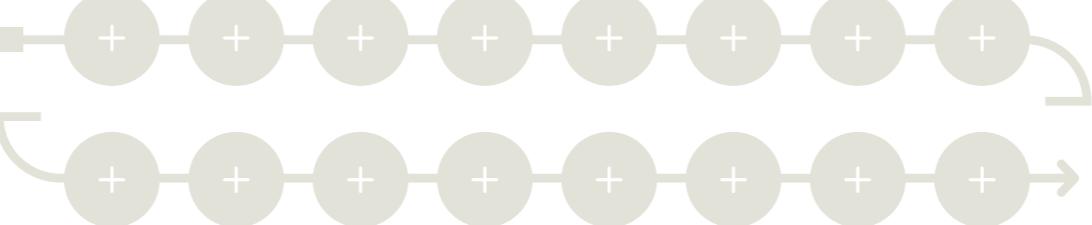
Forward Left Right

Custom Layout +

Connect To Robots +

Examples +

PROGRAM





PLAY **RESTART**

Co-Designers

Beverly Public School

Bloorview School

Norway Jr. Public School

Waterloo Catholic District School Board

Toronto Catholic District School Board

York Region Public School Board

SET-BC in British Columbia

Kayla's Centre, Camp Karma, and Camp Winston

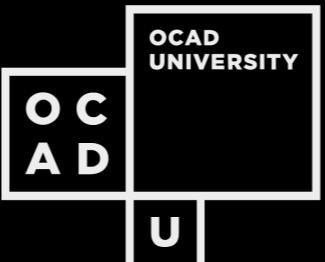
Coleman Institute for Cognitive Disabilities

You?

Thank you!

cclark@ocadu.ca
www.codelearncreate.org

CODING
to LEARN and
CREATE



Supported by Innovation, Science and Economic Development Canada's Accessible Technology Program.

Canada