Variables: Coordinating Research with K-5 Classroom Realities

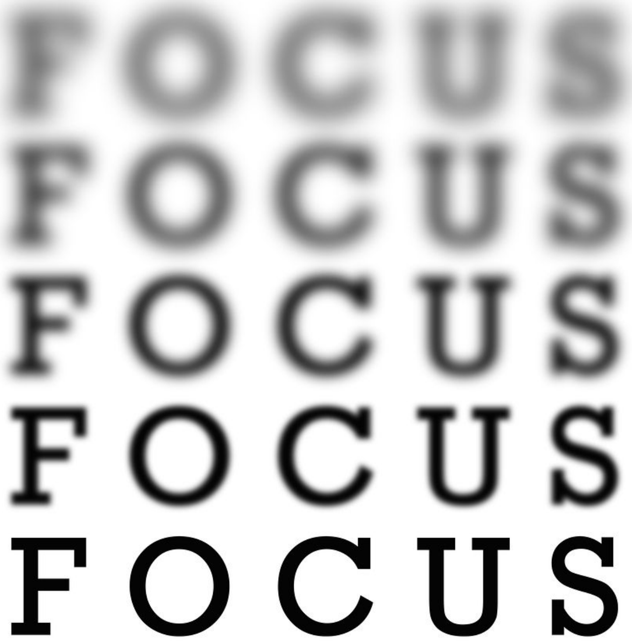
Raspberry Pi Foundation Computing Education Research Seminar

Dr. Katie Rich, Senior Researcher, American Institutes for Research
Carla Strickland, Digital Development Manager, UChicago STEM Education

January 2023

Context: The LTEC Project

- LTEC = Learning Trajectories for Everyday Computing
- NSF STEM+C grant project among University of Chicago, University of Illinois at Urbana-Champaign, and University of Illinois at Chicago
- What big ideas in computer science should be addressed in K-5 curricula?
- How can we express them in ways that make sense to K-5 students and teachers?
- How should they grow across time?



UCHICAGO
STEM EDUCATION

AIR®

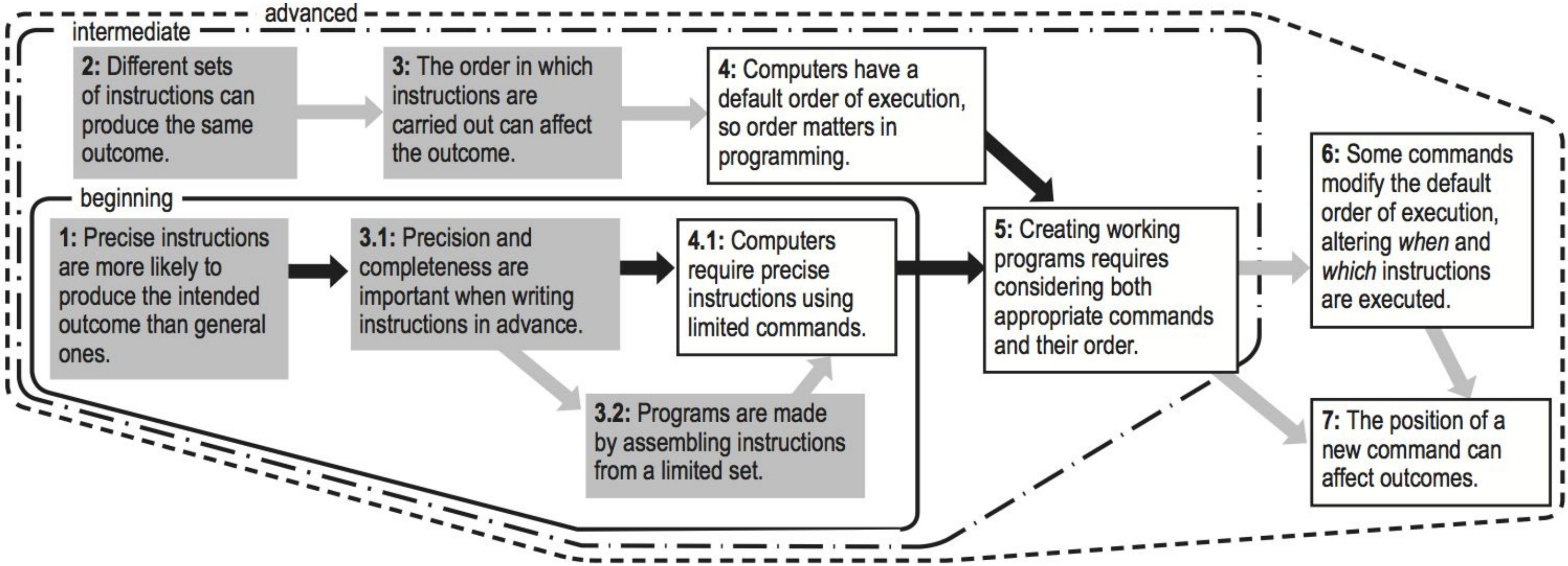
You might know of our learning trajectories (LTs)

Sequence, Repetition, Conditionals (ICER 2017)

Decomposition (ICER 2018)

Debugging (SIGCSE 2019)

Variables (CSE 2020)



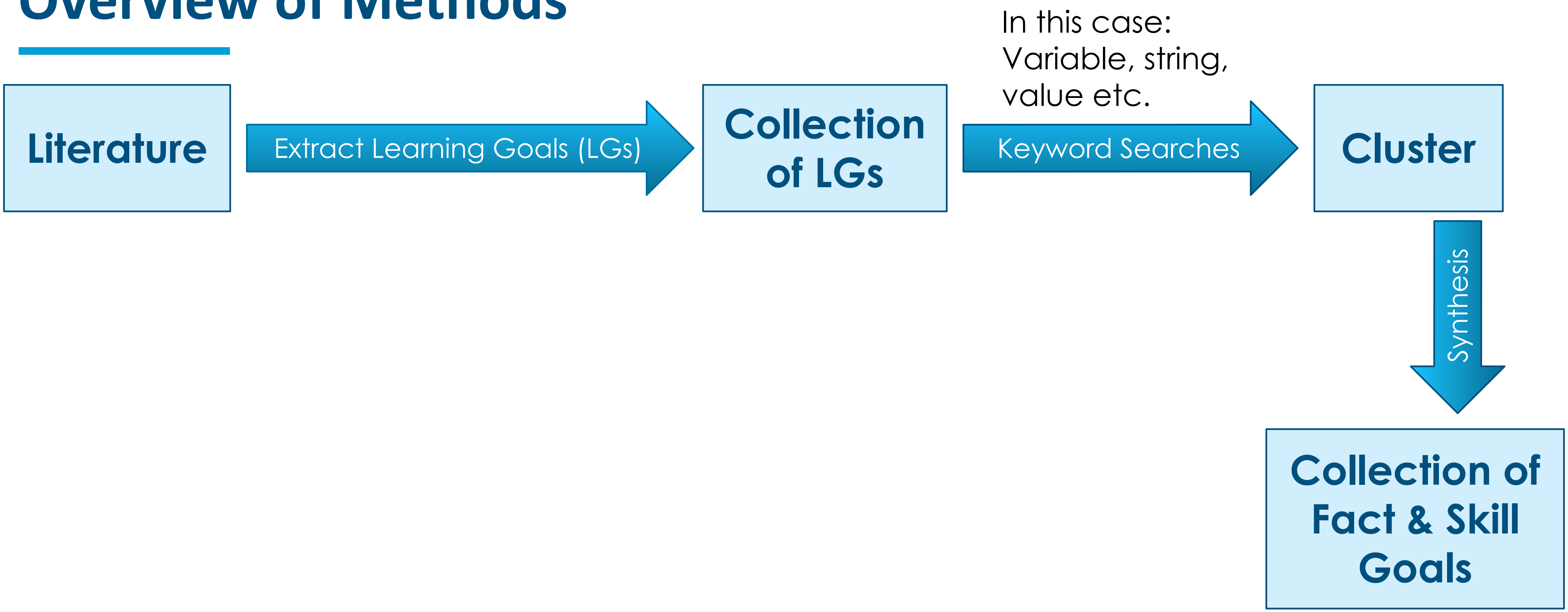
Interactive versions are on our project website:

http://everydaycomputing.org/public/visualization/

UCHICAGO STEM EDUCATION

AIR

Overview of Methods



UCHICAGO
STEM EDUCATION

AIR

Synthesis into Fact and Skill Goals

Cluster

Synthesis

Fact and Skill Goal

Learning Goals

Learn “the importance of initial conditions and parameters.”

“Abstract from the problem the need to initialize and synchronize costume with motion.”

Attend to “initialization” with respect to attributes including “size,” orientation, background, “visibility,” “costumes,” and “position.”

Initialize the properties of a sprite.

Understand what “initialization is and why it is important.”

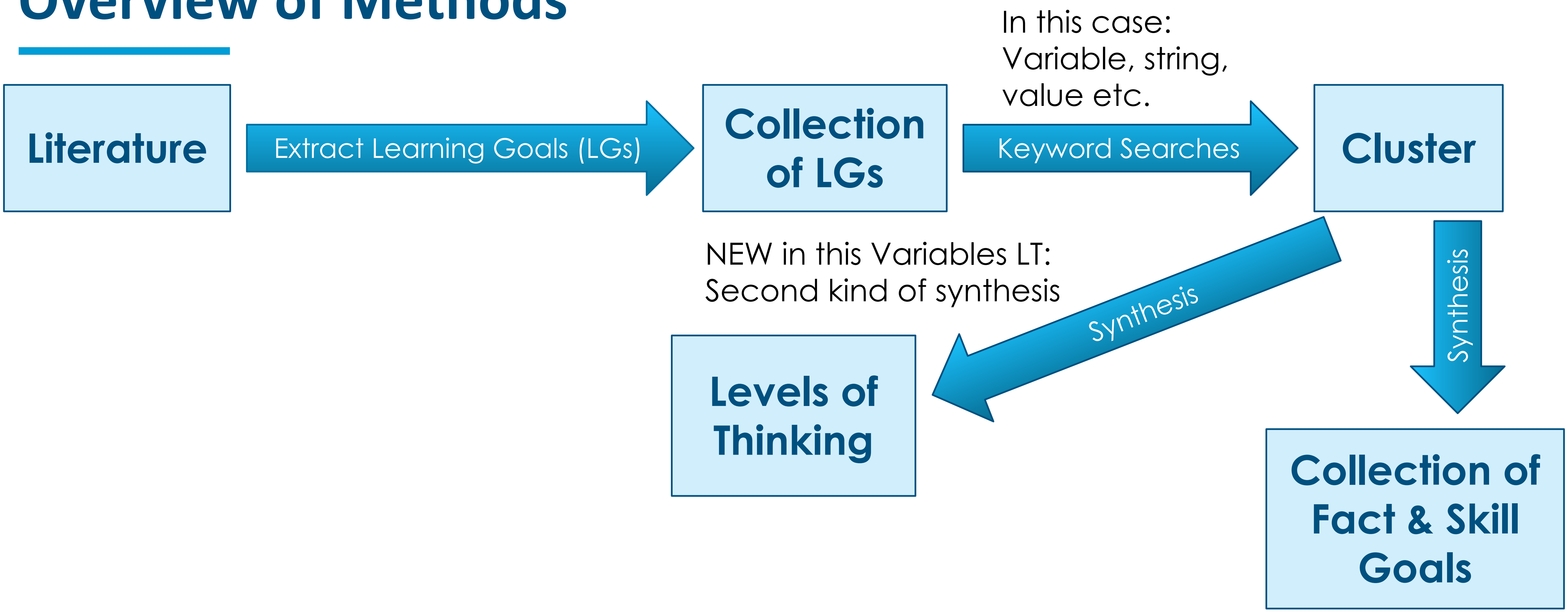
“Consensus” Goal

Know when and why it is important to specify initial conditions and initial values for variables.

UCHICAGO STEM EDUCATION

AIR

Overview of Methods



UCHICAGO
STEM EDUCATION

AIR

New element in Variables LT: Levels of Thinking

Cluster

Synthesis

Level of Thinking

– Review associated learning context for helping students achieve the learning goal

» *How were learners asked to think about the variables?*

» *What mental objects did they consider?*

» *What operations did they carry out on the object?*

Example Annotation

This quiz question relates to variables: "How can you keep track of how many times the bat has been hit by a banana?"

Mental Objects & Operations

Students consider **data** and **storage locations** and **place data into the storage.**

UCHICAGO STEM EDUCATION

AIR

Results: Levels of Thinking

Data User
Students at this level have mental objects representing general pieces of information, and can operate on them in ways that mimic how a program might use the data.

→

Data Storer
Students at this level have mental objects representing variables. They can mentally place objects representing data into the variables, and view the data stored in their variables. These mental actions mimic the actions completed by a computer when it stores data in variables and reads data values.

→

Variable User
Students at this level continue to have mental objects for variables and data, but begin operating on them from the perspective of a programmer rather than a computer, e.g., updating a variable in a general sense rather than in a specific instance.

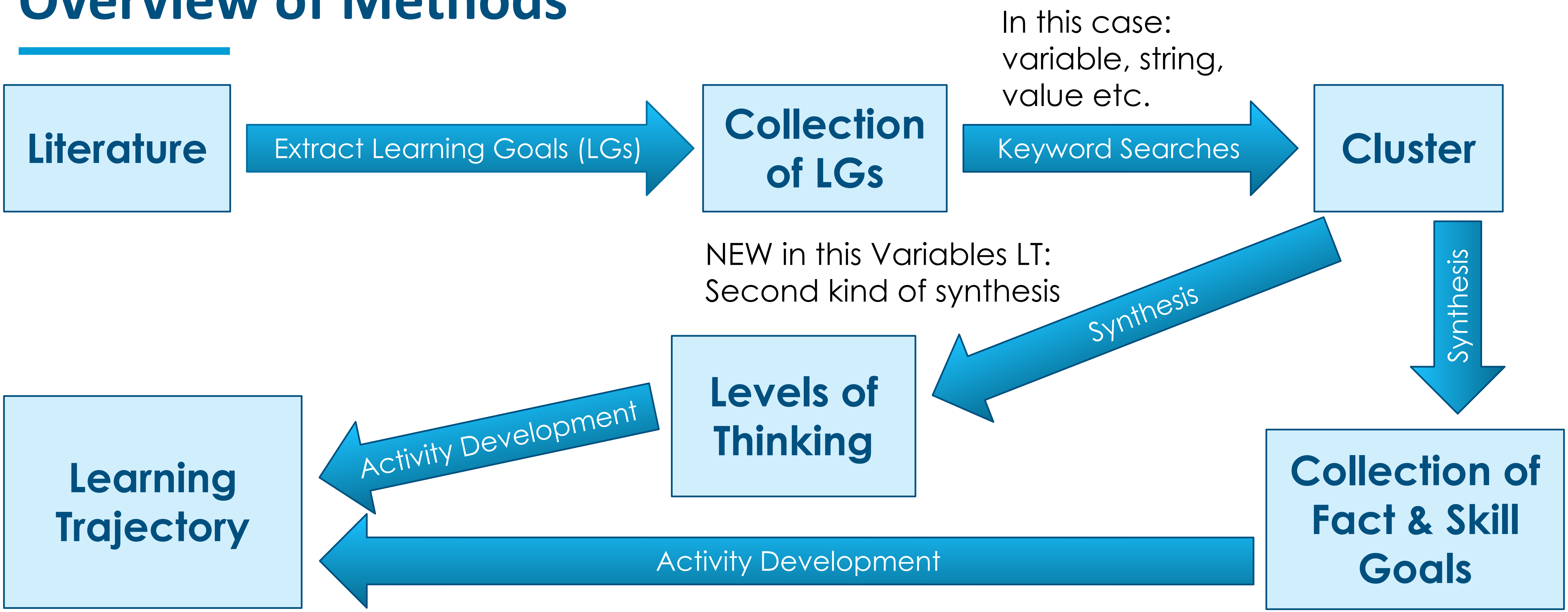
→

Variable Creator
Students at this level can take more highly specified cognitive actions on variables, such as distinguishing between initialization and updating, and associate those cognitive actions to programming commands. They can coordinate multiple operations.

UCHICAGO
STEM EDUCATION

AIR®

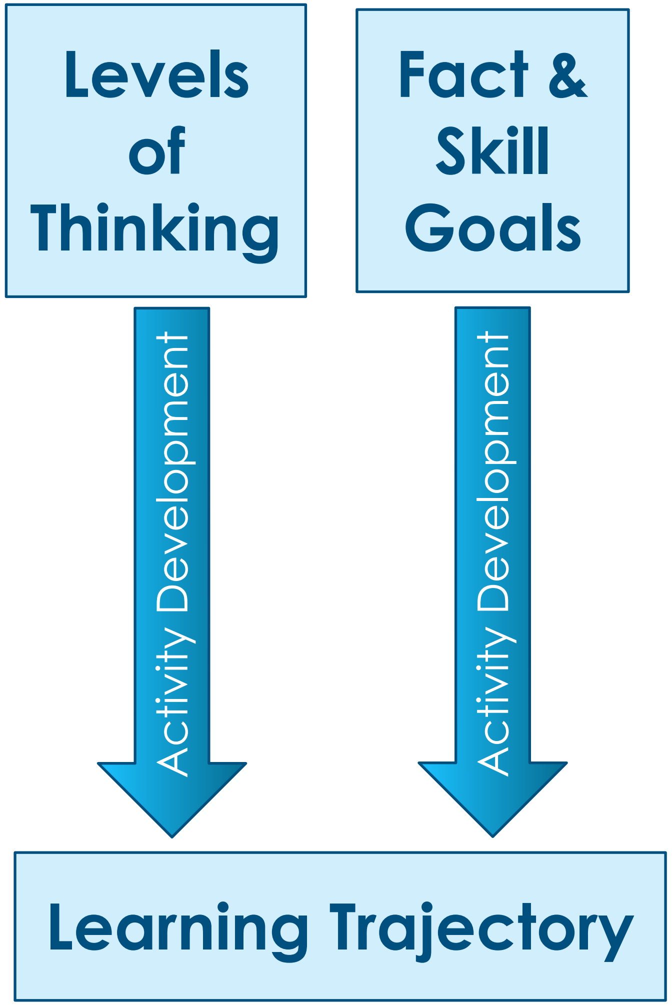
Overview of Methods



UCHICAGO STEM EDUCATION

AIR

Role of Activity Development

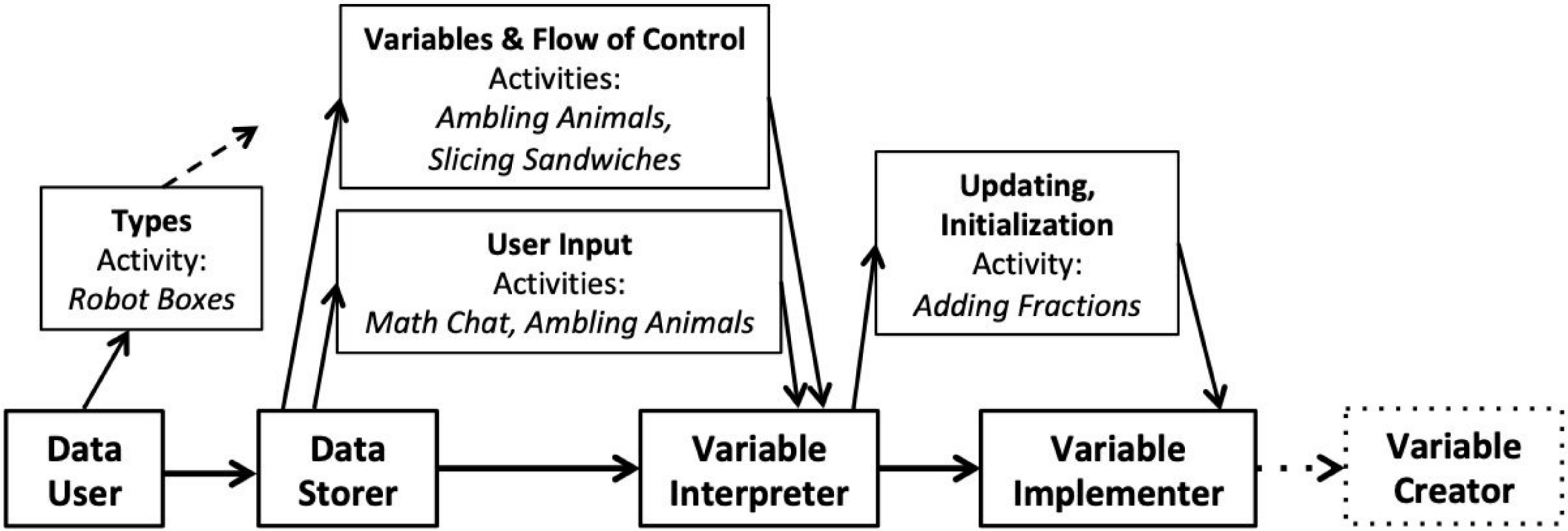


- We were attentive to fostering progression through the levels of thinking, but that left a lot of room to move.
- Critical step for tailoring a Learning Trajectory to constraints of the context
  - *Limited instructional time*
  - *Age level and instructional conditions*
  - *Programming environment*
- We used these considerations to select which fact and skill goals to address and how to shape activities.

UCHICAGO STEM EDUCATION

AIR

Results: Variables Learning Trajectory



UCHICAGO STEM EDUCATION

AIR®

Action Fractions: A coherent collection of Math+CT lessons

Lesson Name	Everyday Math Lesson; G4 Math Topic	CS/CT Focus	*Action Fractions* lesson # (of 25)
Robot Boxes (unplugged)	2-2A; Area and Perimeter (of rectangles)	Introduce and define Variables.	15
Math Chat	2-2B; Area and Perimeter (of rectangles)	See Variables used to store values (user input) to perform later calculations in a Scratch program.	16
Ambling Animals	3-6A; Comparing Fractions (on a number line)	Track the use of a Variable in a Scratch programs. Modifying the code to use the Variable in a desired way (conditional on user input).	18
Slicing Sandwiches	3-6A; Comparing Fractions (same denominator)	Use a Variable in a Conditional statement and Repeat block.	19
Adding Fractions	5-3A; Adding Fractions (same denominator)	Add a Variable to code to efficiently represent the same value in multiple lines.	21

UCHICAGO STEM EDUCATION

AIR

Defining “Variables” for Math+CT

Framing:

- What is the extent of the overlap between mathematics and computer science, and what are the key differences to be addressed in integrated materials?
- How may similarities and differences in discipline-specific definitions affect student’s introductory and developing understanding of each term and impact students’ later/cumulative understandings in the partner discipline?
- What should be done to anticipate these factors and help elementary students (both novice and young learners) to develop this academic language, with nuance, in an integrated setting?

UCHICAGO STEM EDUCATION

AIR®

“Variable” – Shared Language

- In common language, “variable” is used as both a noun and an adjective.

- In Math, variables follow introductory work with “symbols.” Number sentences, equations, tables of values, graphs. The most popular ones are x and y. Independent vs. Dependent is important. One popular misconception is that values of related variables seem like they automatically update. We call this the “Excel problem.”

- In CS, variables are important for calculations, parameters, and determining object locations, etc. The initial assignment statement may look like a number sentence but it is not. Values of even seemingly “related” variables (per initial assignment) **will not automatically update.**

UCHICAGO STEM EDUCATION

“Variable”

G3 exposure in math, with the most work in G6 and upper grades. CS exposure is highly dependent on language choice. Some languages deliberately avoid/mask work with variables (e.g. does not appear in CS Unplugged).

Resulting Synergies: Common usage as a symbol, placeholder, something that can change.

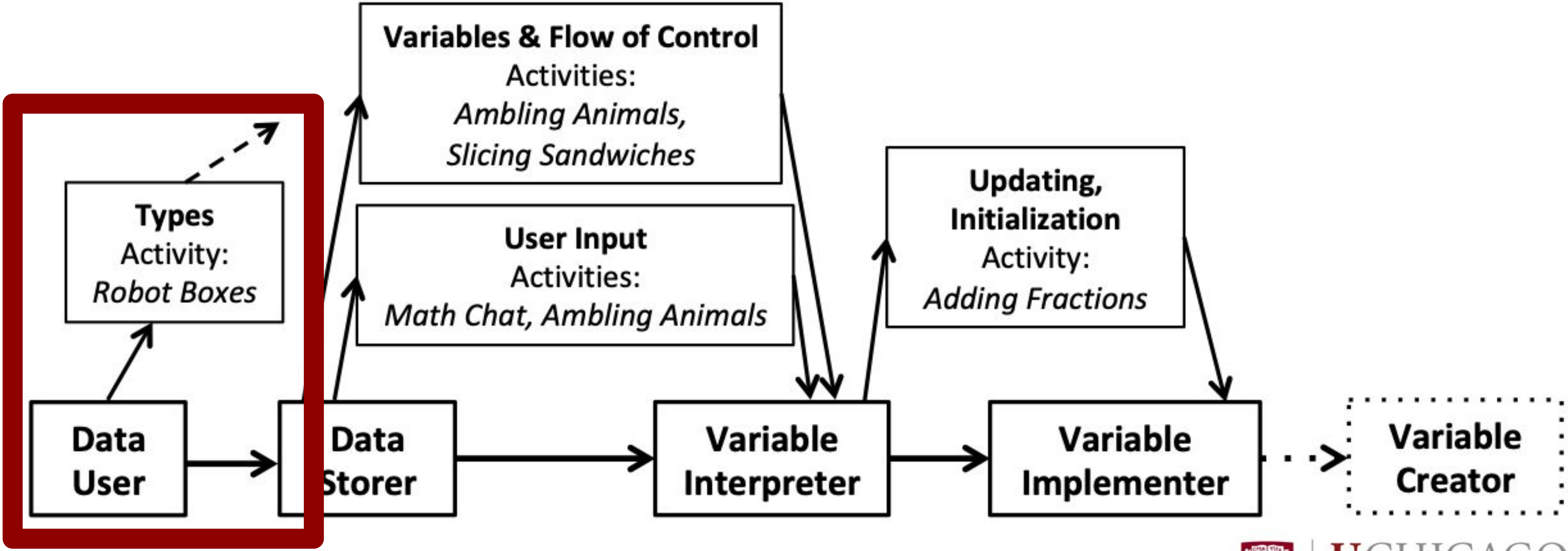
Resulting Tensions & Popular misconceptions: Very problematic in CS with expectations about automatically updating values (“Excel Problem” in prev. slide).

Initial recommendations: As math dominates usage, CS must do more to stress the difference between assignment statements and number sentences/equations (variable as unknown) in order to avoid these kinds of mistakes. CS should avoid using tables of values like math to track values – this implies dependencies.

UCHICAGO STEM EDUCATION

AIR

Results: Variables Learning Trajectory



UCHICAGO
STEM EDUCATION

AIR®

Activity Example: *Robot Boxes*

`message` = "I am a robot and I love rectangles."

`length` = 3

`width` = 4

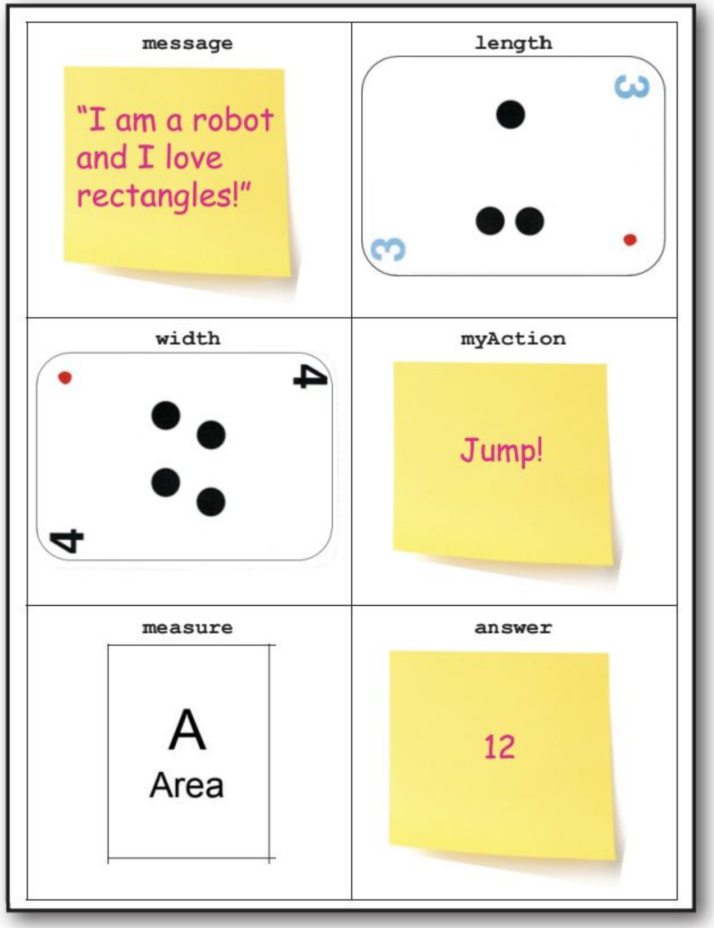
`myAction` = Jump!

`measure` = Area

`answer` = 12

ROBOT Program

1. Say"I am a robot and I love rectangles.";
2. Move 3 steps;
3. Say "I am 3 units long!"
4. Turn right;
5. Move 4 steps;
6. Say "I am 4 units wide!"
7. Turn right;
8. Move 3 steps;
9. Turn right;
10. Move 4 steps;
11. Do Jump!;
12. Wait for Calculator to fill `answer` box.
13. Say "Area is 12"



▶ Wrap Up

WHOLE CLASS	SMALL GROUP	PARTNER	INDEPENDENT

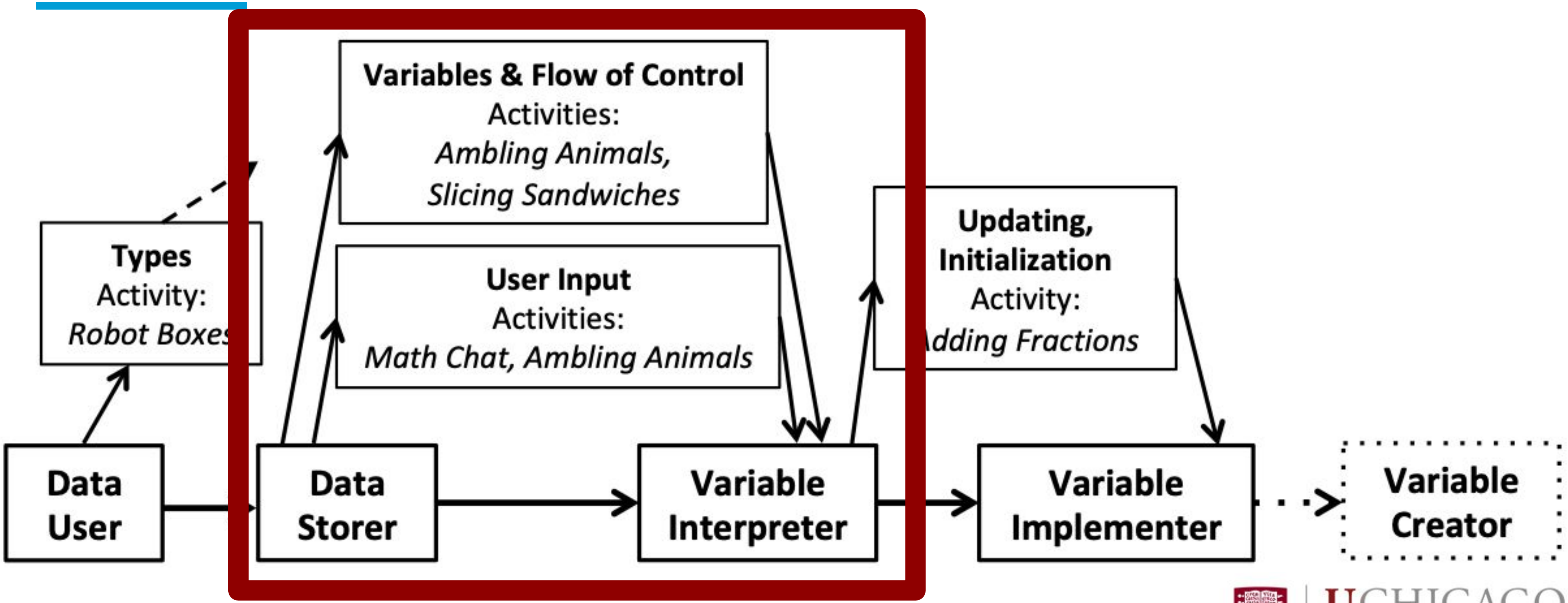
Summarize Explain to the class that **`message`**, **`length`**, **`width`**, **`myAction`**, **`measure`**, and **`answer`** are called **variables**.

- *A* **variable** *holds a value that can change. Variables are used in the program as placeholders, and they hold the values that are assigned to them by the computer running the program.*
- *A variable can hold a number, some text, or other types of values. For example, we have a variable holding an action-command:* **`myAction`**.
- *Variables can be used to make programs more general, so that they can do more than one thing, or do things in different ways.*
- *Variables can also be helpful when we want to use the same value multiple times in a program.*

Choose one of the program variables. Draw a box on the board labeled with the given variable. List the different values that the variable held while students were playing their games. For example, **`message`**: "Rectangles rock," "I love rectangles," etc. Do not write these down as a list in the box, instead--for each new value, erase or cross out the old value to reinforce the idea that the variable holds only one value.

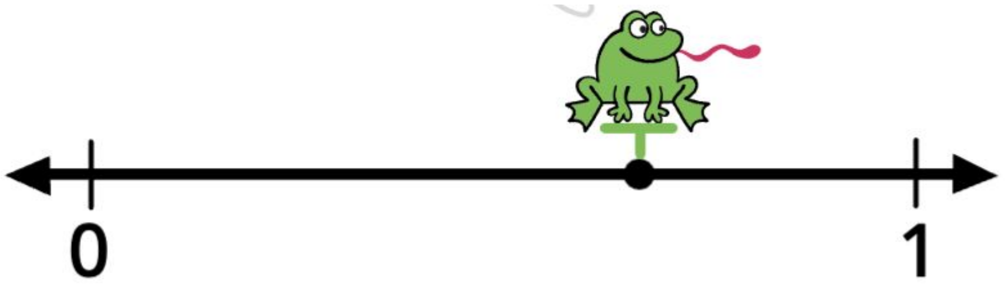
Ask students to think about how variables could store information about things in the real world. For example, information about a car could be stored in variables describing color, make, or speed. A person could be described using variables for age, height, weight, or hair color.

Results: Variables Learning Trajectory



Note. Placeholder for notes, sources, and permissions (if needed). "*Note.*" (including a period) is italicized.

Activity Example: *Ambling Animals*



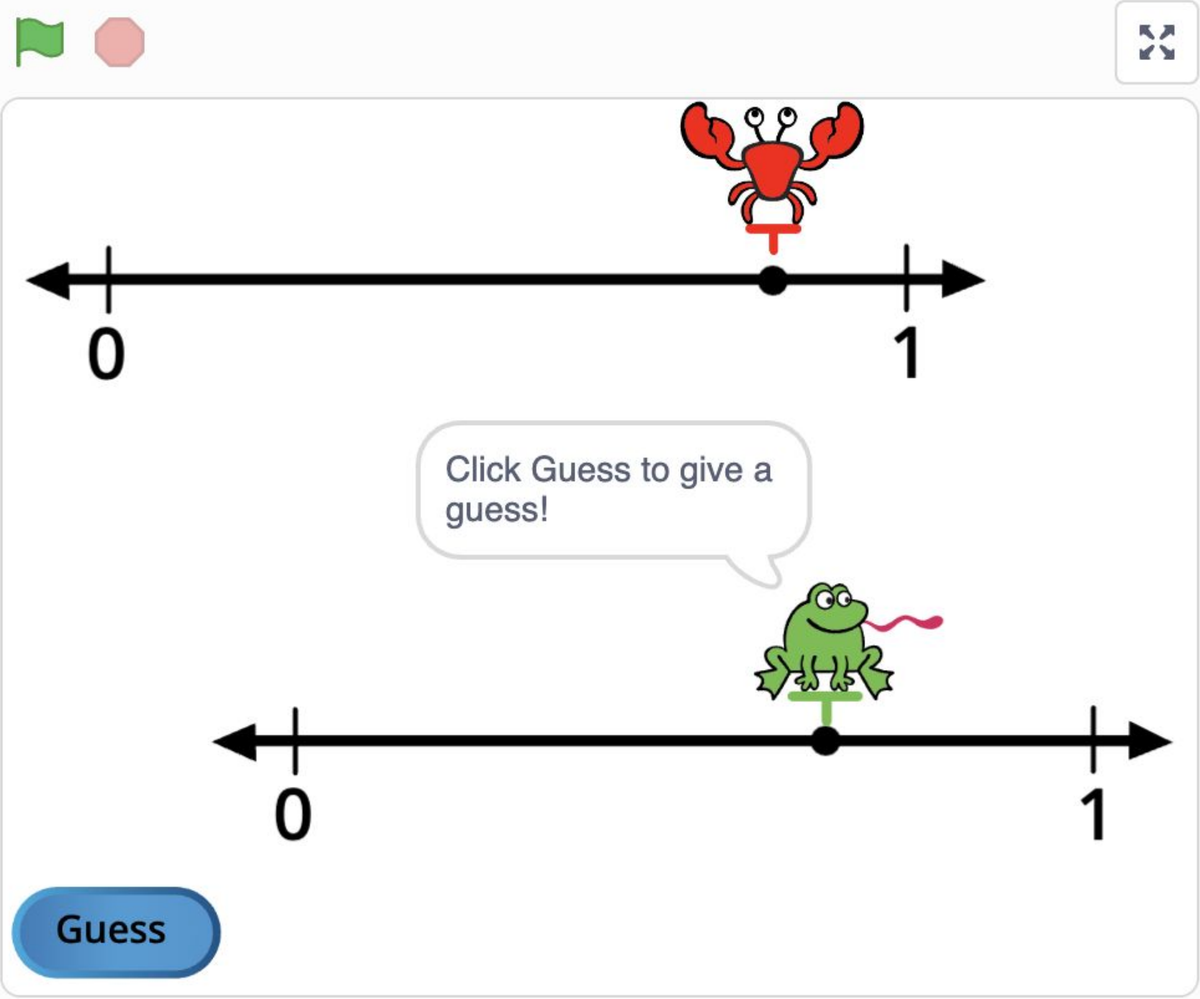
Locating and comparing **fractions** on **number lines**.

UCHICAGO STEM EDUCATION

AIR

UCHICAGO STEM EDUCATION

Activity Example: *Ambling Animals*



Thanks to **ze_ne** for the original project **Ambling Animals**.

Thanks to **nbanholzer** for the original project **Fraction Circles 1a**.

Instructions

Click the green flag to start. Which animal stops on the larger fraction?

Notes and Credits

Click See inside. Use the blocks provided to complete the script on the Guess button sprite.

Activity Example: *Ambling Animals* (actual student work)

Let's look at some actual student work!

Our original 'Ambling Animals' Scratch Project:

https://scratch.mit.edu/projects/259190866

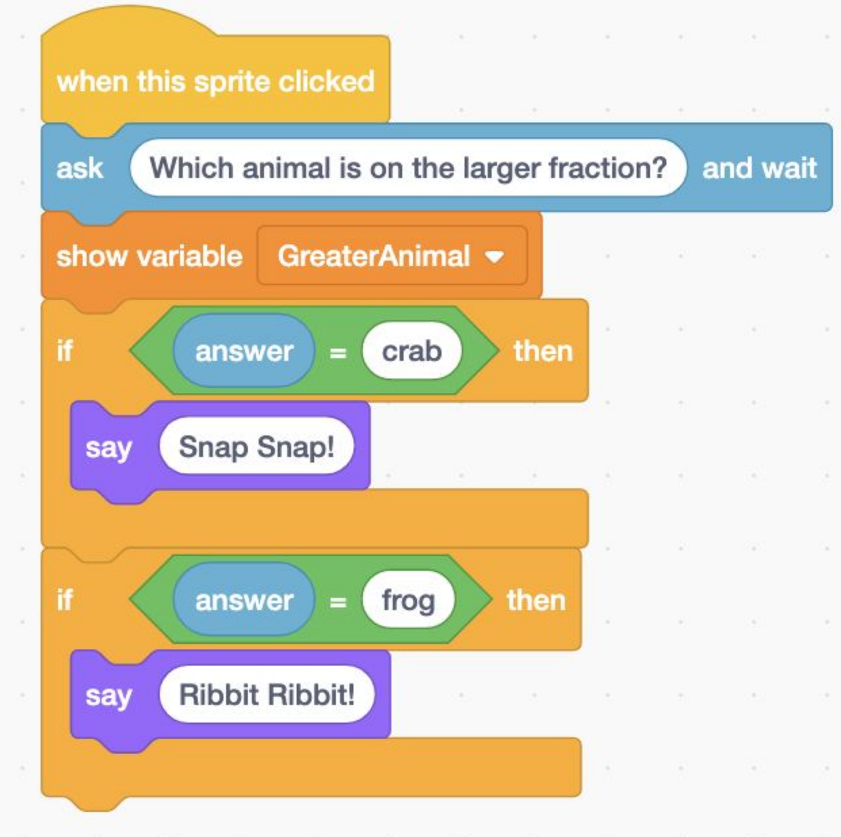
Code from a student's remixed project that uses variables.

https://scratch.mit.edu/projects/310774172/ (Riley's)

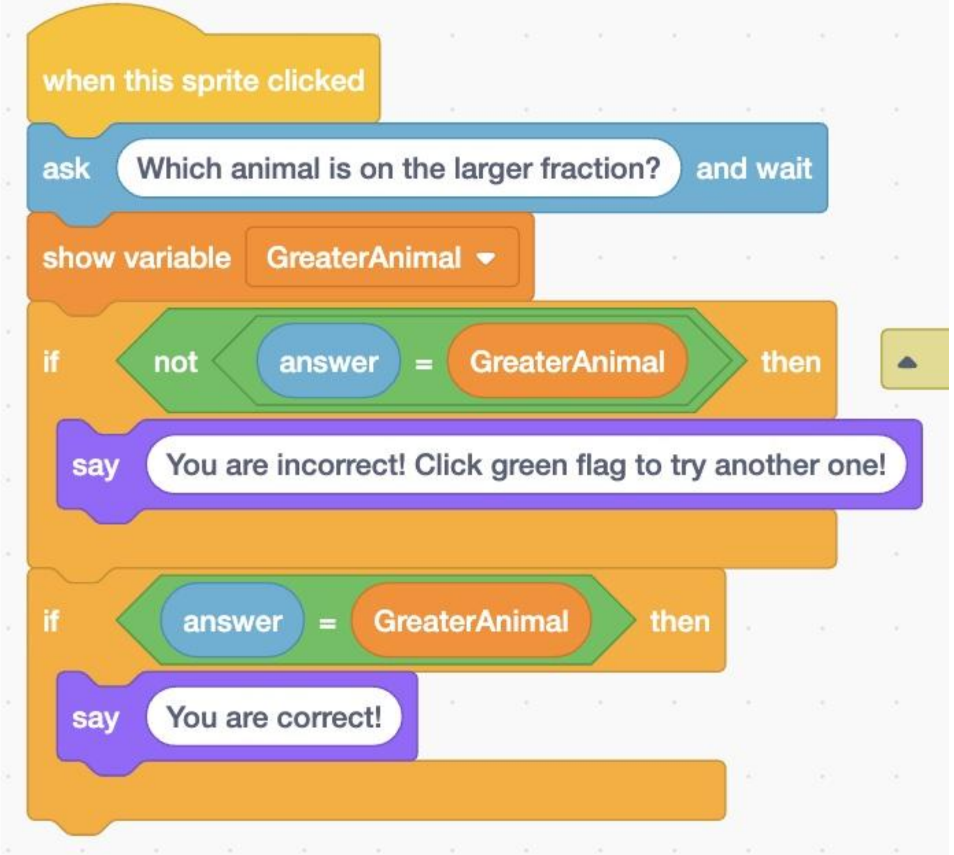
UCHICAGO STEM EDUCATION

AIR

Activity Example: *Ambling Animals* (code comparison)



Starting Code

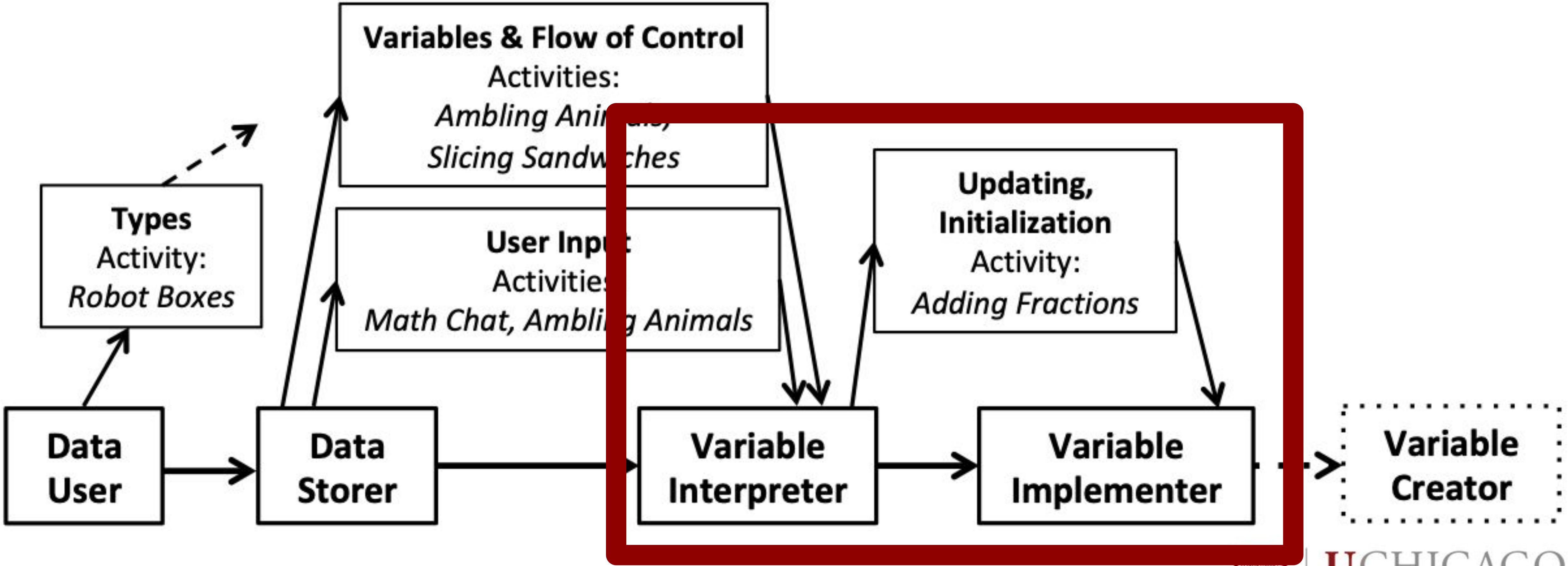


Remixed Code

UCHICAGO STEM EDUCATION

AIR

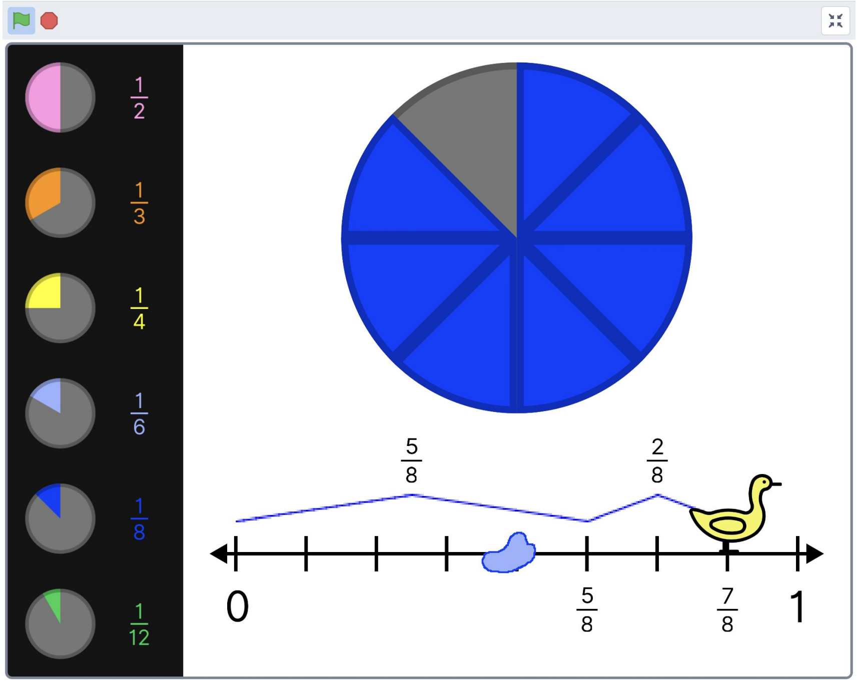
Results: Variables Learning Trajectory



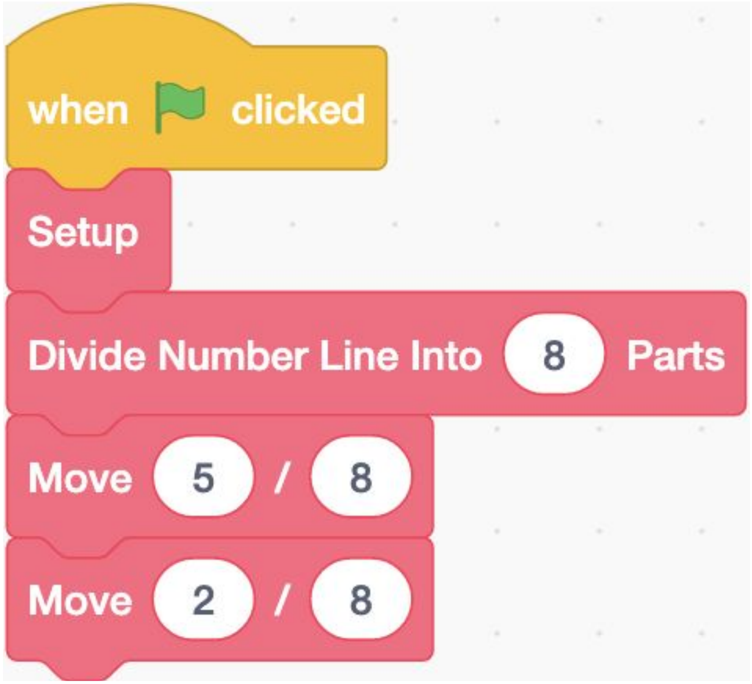
UCHICAGO
STEM EDUCATION

AIR

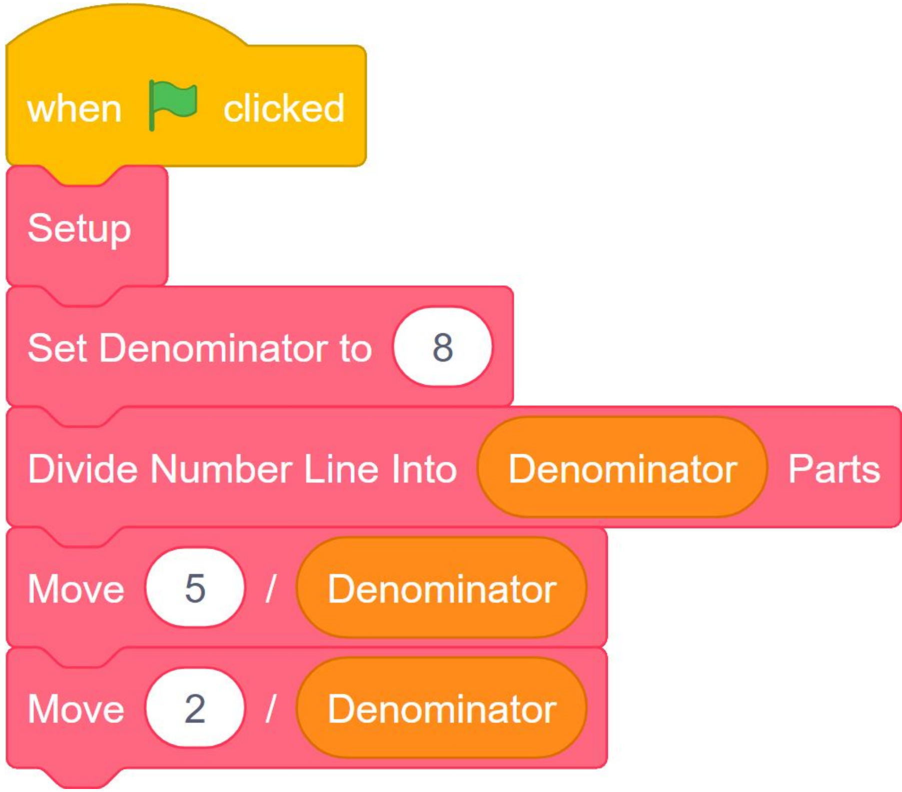
Activity Example: *Adding Fractions*



Scratch Project Stage



‘Inefficient’ Code



‘Efficient’ Code

Limitations

Context

- Dominated by U.S. school system, academic articles
- Lit review concluded in 2016

Language

- English only

UCHICAGO STEM EDUCATION

Takeaways

Conceptual

- Levels of thinking form a flexible base for designing instruction.
- There is a lot to unpack between the levels we originally labeled “Variable User” and “Variable Creator.”
- Our expertise with *young* learners, not just *novice* learners, allowed us to recognize when further splits were necessary.

Practical

- Contriving integrated scenarios for CS can be problematic. e.g. efficiency argument
- Generalist elementary teachers are very well predisposed to integrated instruction so design to meet their needs. e.g. spiraling; tagging focus CGs
- Shared Language has benefits & drawbacks for Students and Teachers that integrated instruction must address.

AIR

Katie Rich

Senior Researcher
krich@air.org

@KatietheCurious

Carla Strickland

Digital Development Manager
castrickland@uchicago.edu

@CisforCarla

AMERICAN INSTITUTES FOR RESEARCH® | AIR.ORG