Lightbot and Functions

CSE 120 Winter 2020

Instructor: Sam Wolfson

Teaching Assistants: Yae Kubota Eunia Lee Erika Wolfe

The 10 Neatest Things We've Seen at CES So Far

"The big spectacle of CES hasn't fully spread its wings just yet—Tuesday will be bananas—but we're feeling the early rumblings of what's to come. On Sunday night here in Vegas, the assembled tech press got to take a peek at some of the gadgets and apps launching this week."

- https://www.wired.com/story/ces-2020-photo-gallery-day-1/

Administrivia

- Checkoffs
  - **Done in pairs**, multiple attempts to complete
  - Generally due by the end of the **following** section; later checkoffs are longer (calendar was incorrect – now fixed)
  - Can get checked off by any staff member in section or OH

- Homework submission
  - **Individual work**
  - Submit via Canvas Assignments page
  - Make sure you look at Canvas rubrics
  - Lightbot Functions assignment due Monday (1/13)

Aside: Reading Presentations

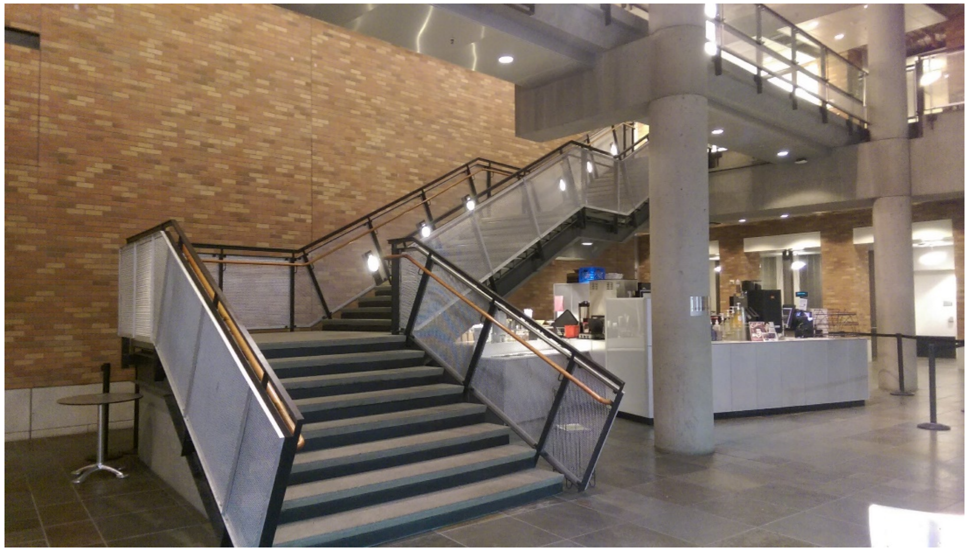
- In Thursday sections this quarter, we'll ask you to form small groups (2-4) and create presentations based on the readings that you did for this week.
- You will be given about 15 minutes to prepare, and 3 minutes to present.
- A small amount of outside research during the preparation period is encouraged but not required.
- Presentation mediums will vary from week to week.

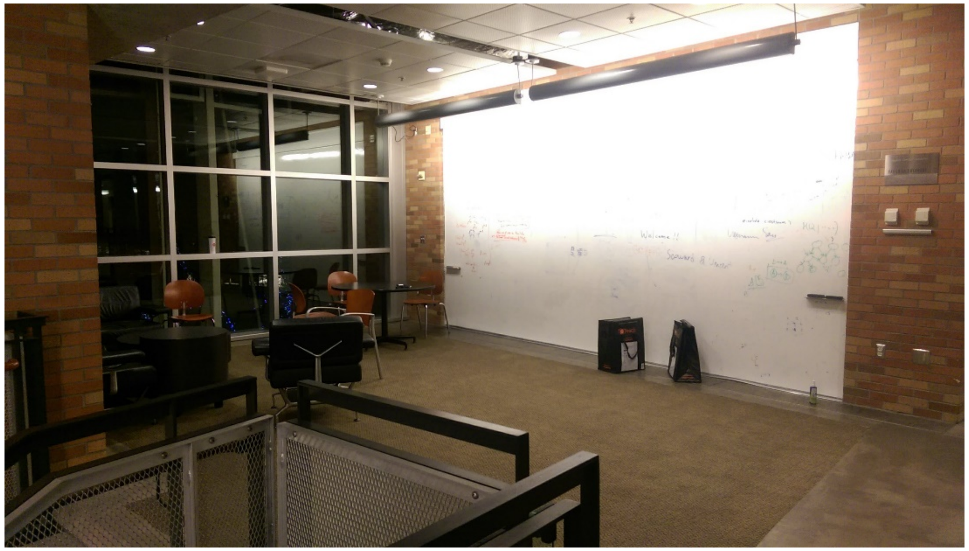
Why?

- Think critically about complex technical issues
- Contribute to a discussion in a way that you find interesting & relevant
- Practice your presentation skills

TA Office Hours

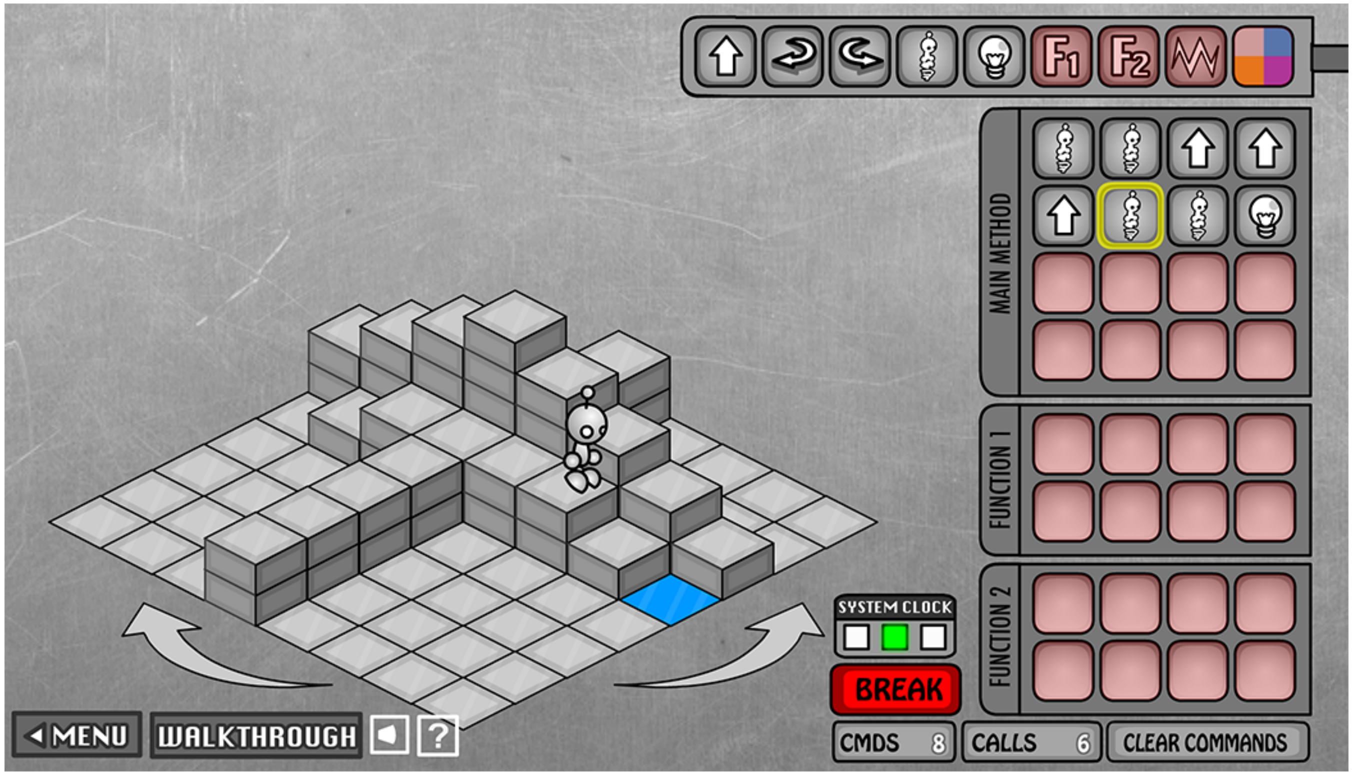
- CSE breakouts
  - Up the stairs in the CSE Atrium (next to the café)
  - At the top of that first flight, the open area with the whiteboard wall is the 2nd floor breakout!
  - Other breakout spaces are directly above on the following floors (just up the stairs)





As Experienced Lightbot Players…

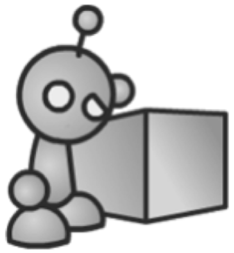
- What are you doing in Lightbot?
  - Commanding a robot through a world of blocks and switches



As Experienced Lightbot Players…

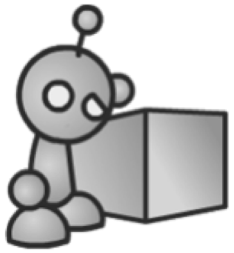
- What are you doing in Lightbot?
  - Commanding a robot through a world of blocks and switches

- Programming is *commanding* an *agent*
  - In this case, the agent is a robot
  - The agent is usually a computer, but could be a person or other device



As Experienced Lightbot Players…

- What are you doing in Lightbot?
  - Commanding a robot through a world of blocks and switches
- Programming is *commanding* an *agent*
  - In this case, the agent is a robot
  - The agent is usually a computer, but could be a person or other device

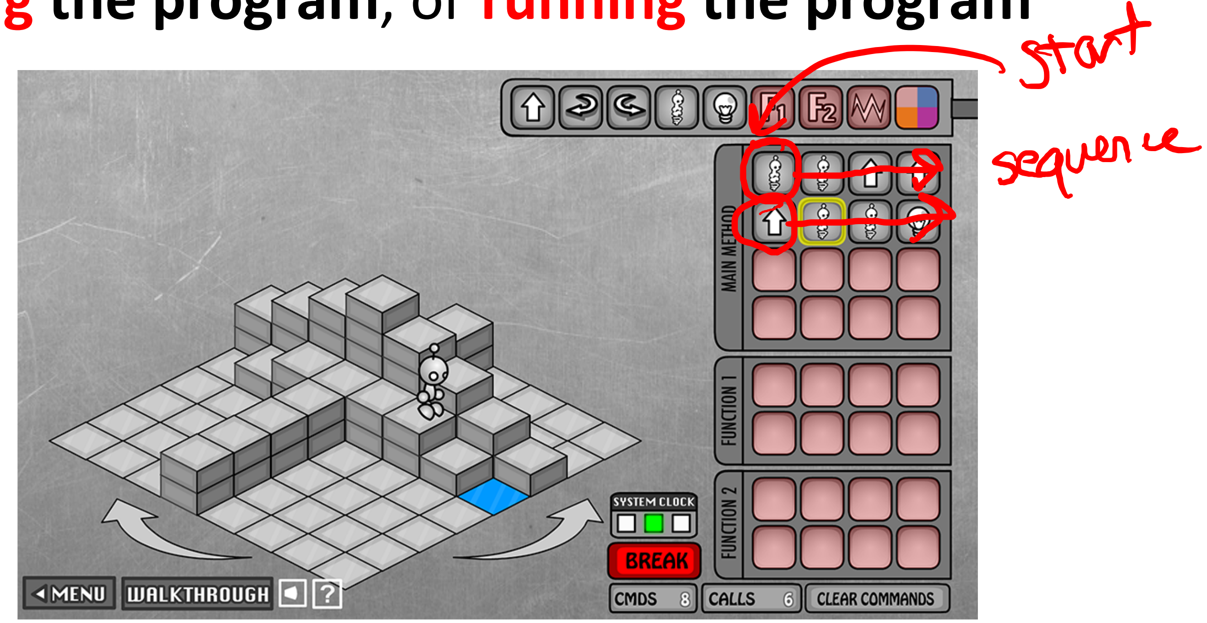


- Direct an agent to a *goal* by giving it *instructions*
  - The agent follows the instructions flawlessly and mindlessly ☆
  - The trick is to find the right instructions to match your *intent*

Order of Instructions

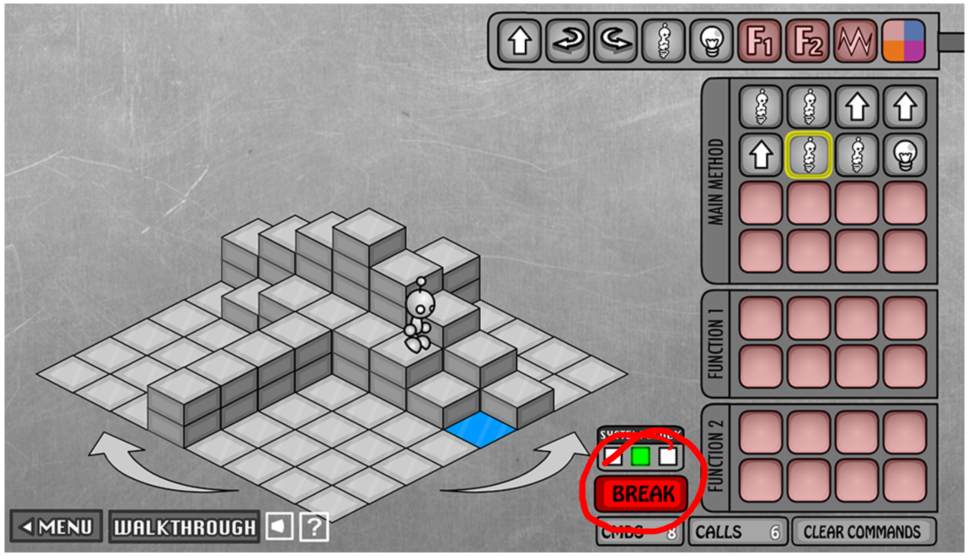
- Instructions are given in order (*i.e.* in a sequence)
  - The 1st instruction is completed, then the 2nd, then the 3rd, …

- *You* issue the instructions and the agent follows them
  - When the agent is following your instructions, this is called **executing the program**, or **running the program**



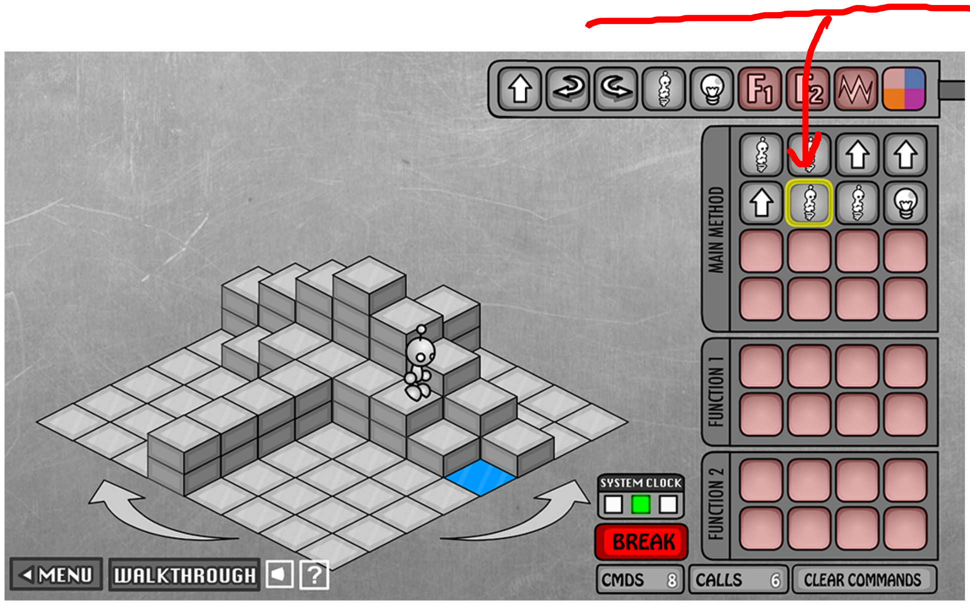
Order of Events

- The instructions are programmed *ahead of time*, and then executed *later*
  - The programmer cannot intervene until the program has finished executing or is terminated prematurely
- The instructions must be correct for the agent to achieve its goal



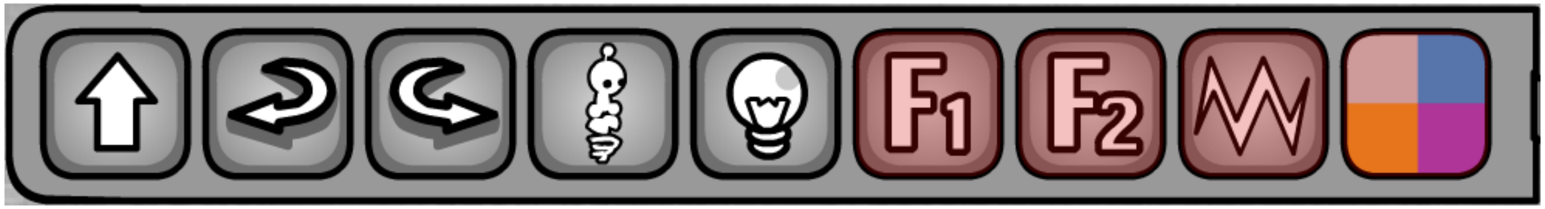
Point of View

- Programming requires you to take the agent's point of view
  - Because it is a sequence of instructions, you must account for everything that happened before (*i.e.* trace the program)
  - There is usually an indication of where you are currently in the program (sometimes called a *program counter*)



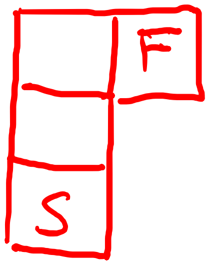
Limited Instructions

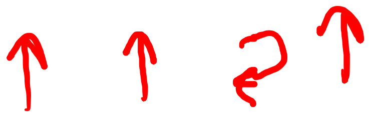
- The number and type of instructions is always limited



  - The agent can only do certain pre-defined actions

- The agent can only execute one instruction at a time
  - Must learn how to specify complex tasks using just these simple actions





Limited Instructions

- Limited instructions is a reality of *all* computing
- A computer's hardware/circuitry can only execute a small number of instructions – usually about 100
  - Many are just different versions of the same idea

Amazing Fact:

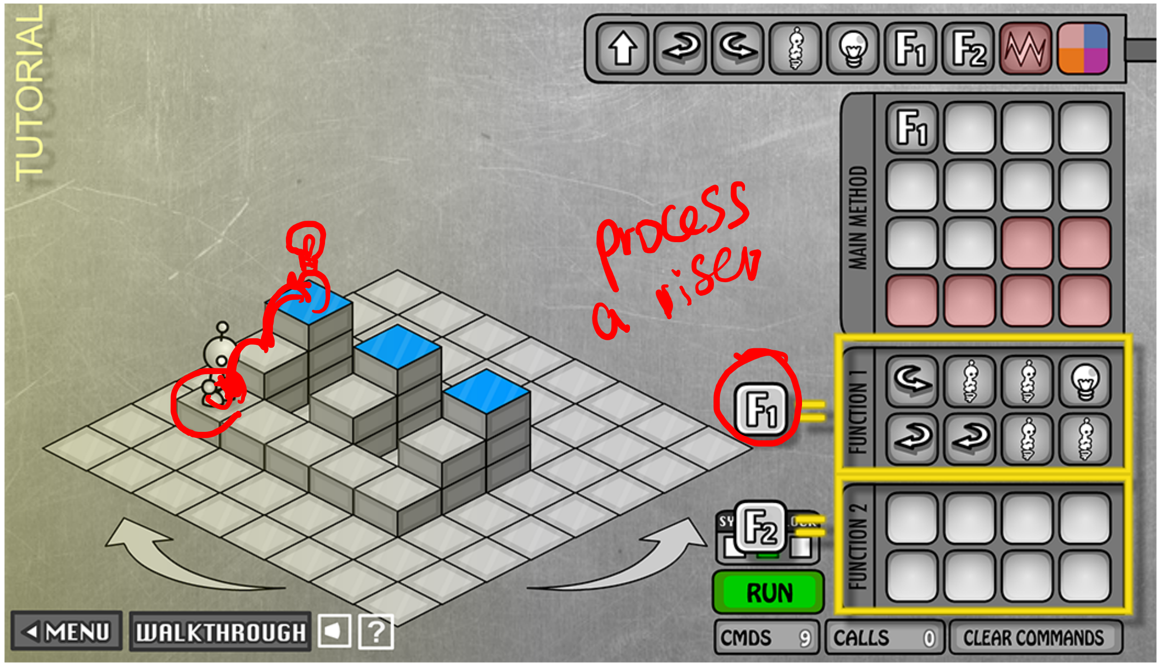
In theory, a computer with just SIX instruction types could compute all known computations!

Limited Instructions in Programming

- Programming would be excruciatingly tedious if you could only use the basic instructions
  - The early days of programming (c. 1950’s) were tedious and error-prone
  - No one would be a programmer no matter how much it paid!
  - The amazing applications we see today would not exist

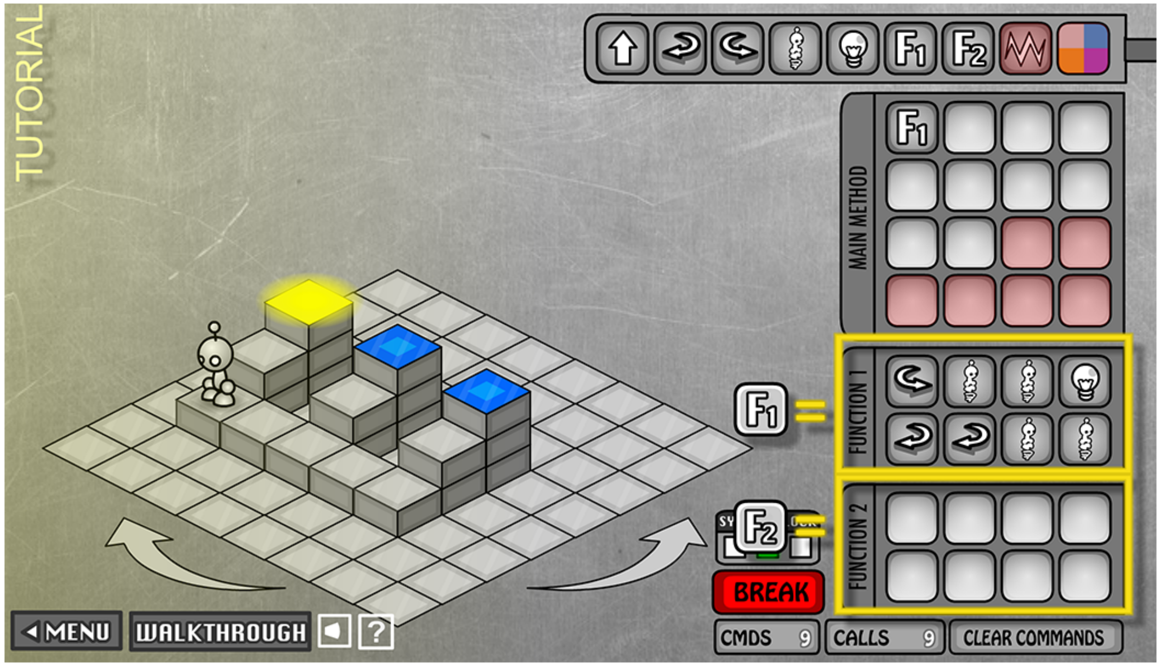
Solution: Functions!

- Functions allow us to create new, more complex subtasks for our agents
  - Below, F1 is a function to "process a riser"
  - We can call a function by name (F1) to execute its instructions



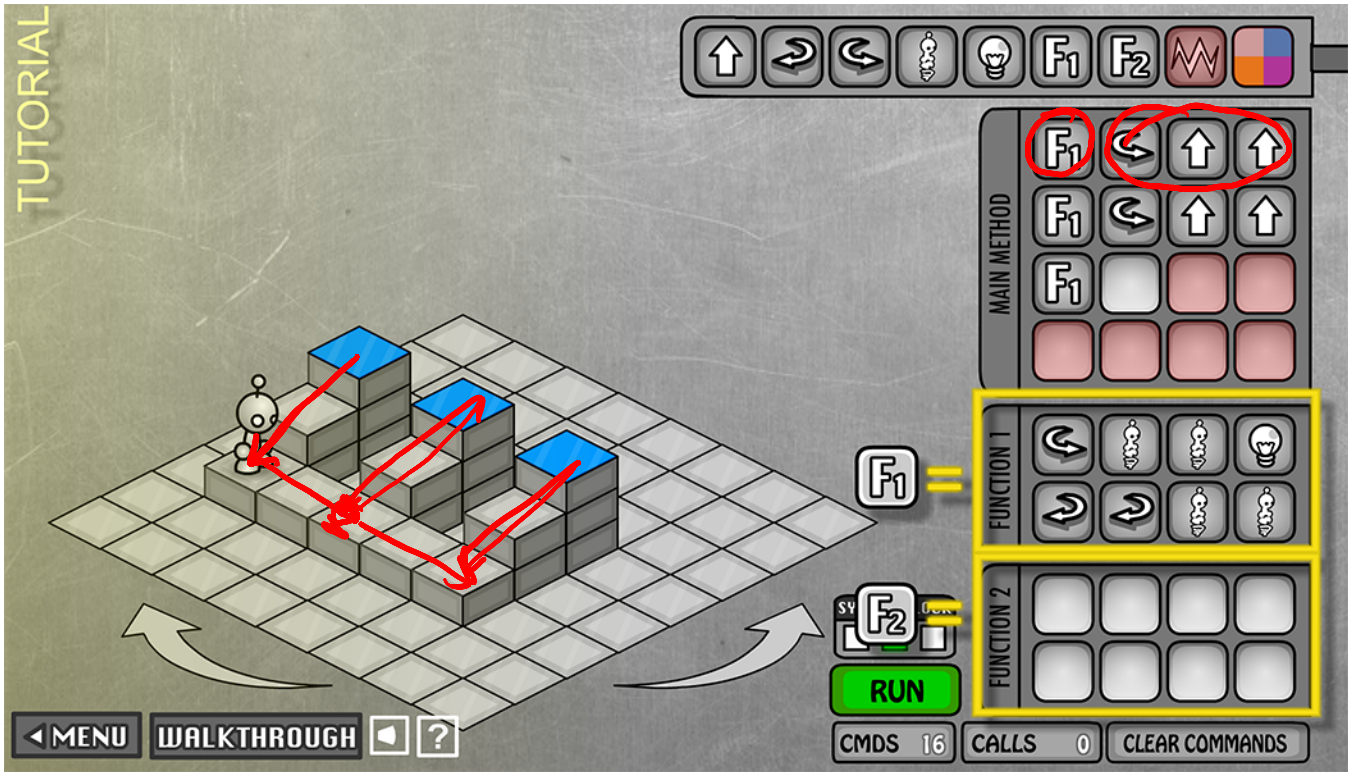
Choosing Functions

- The goal is to break down a complex problem into smaller/simpler ones ← detail removal
- Look for common patterns ← Generalization!
  - "Process a riser" looks like a useful sub-problem because there are three of them



Choosing Functions

- One possible solution is shown below:
  - 17 commands, 29 calls



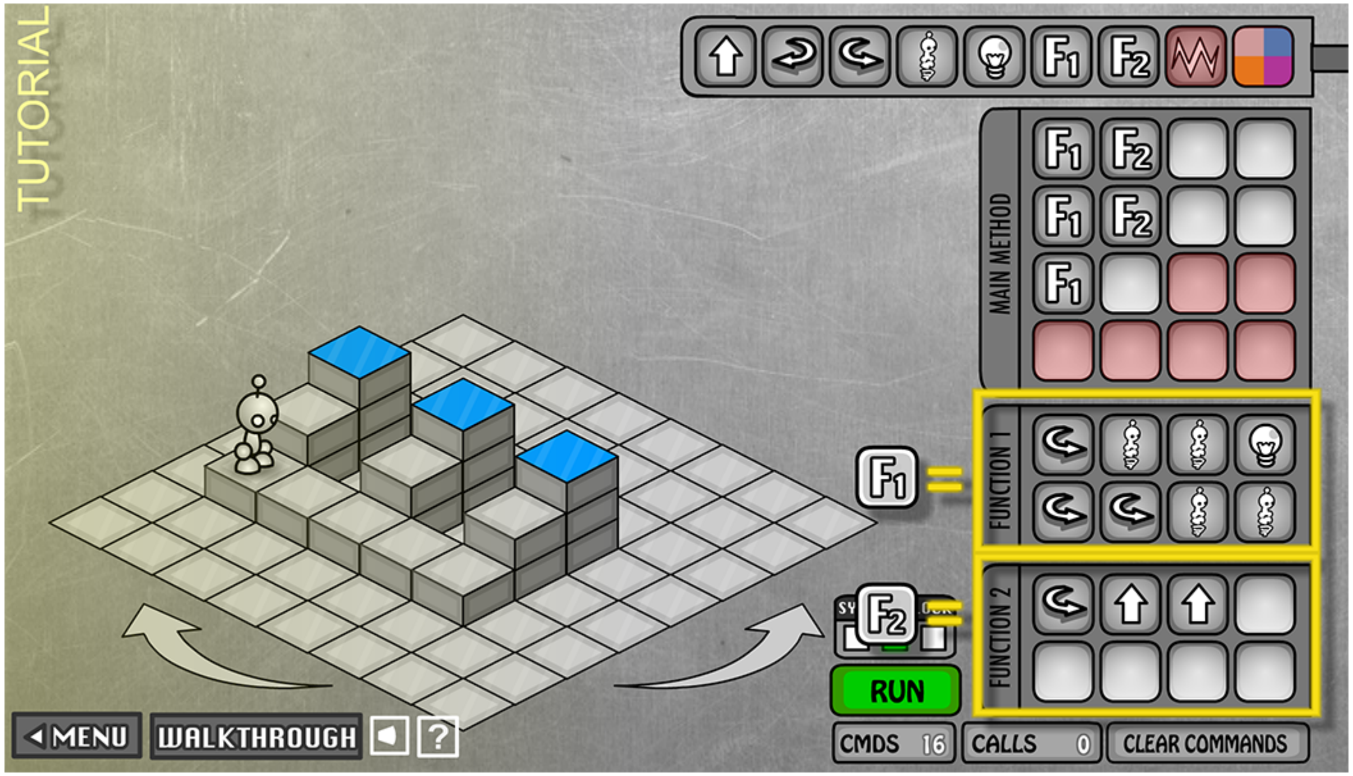
9

8

17

Choosing Functions

- Modified solution is shown below:
  - Now F2 is a function to "move to next riser"
  - 16 commands, 31 calls

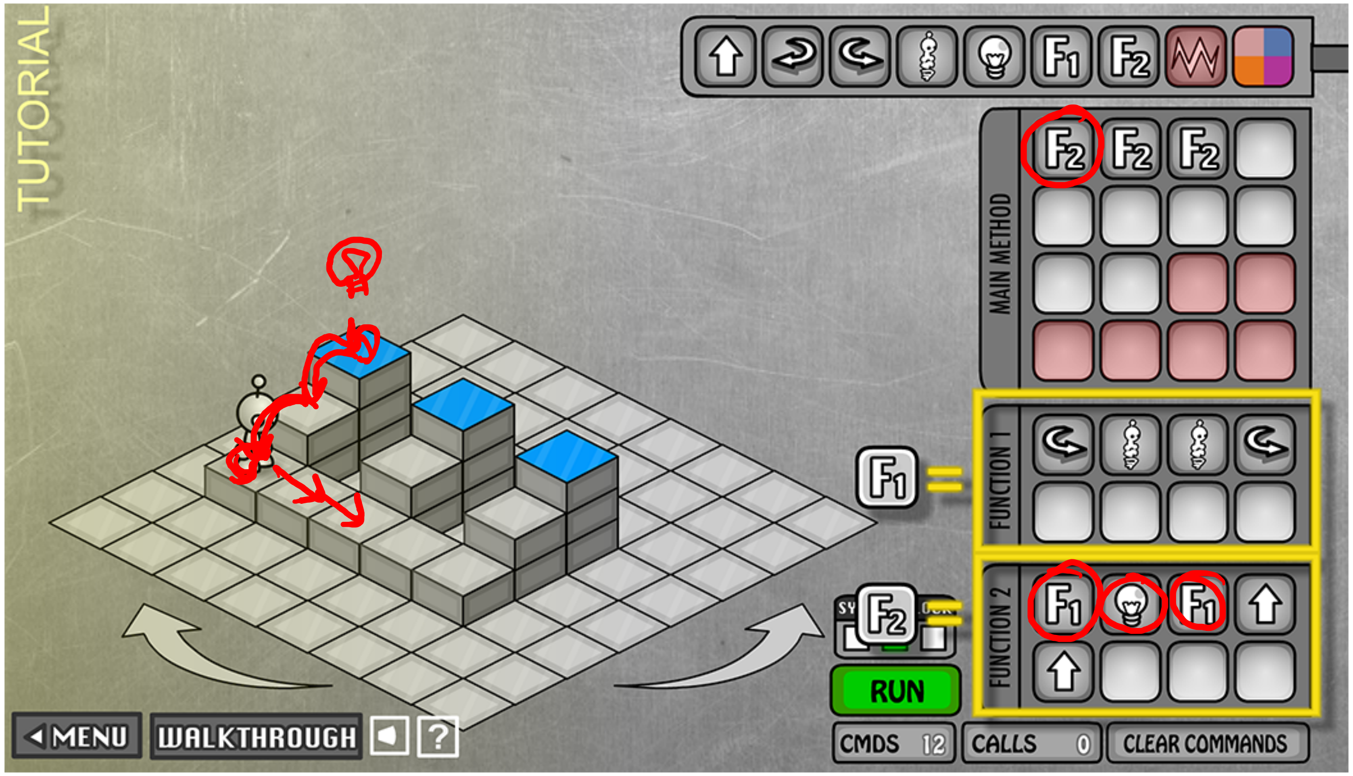


process riser

move to next riser

Choosing Functions

- Yet another solution shown below:
  - 12 commands, 35 calls

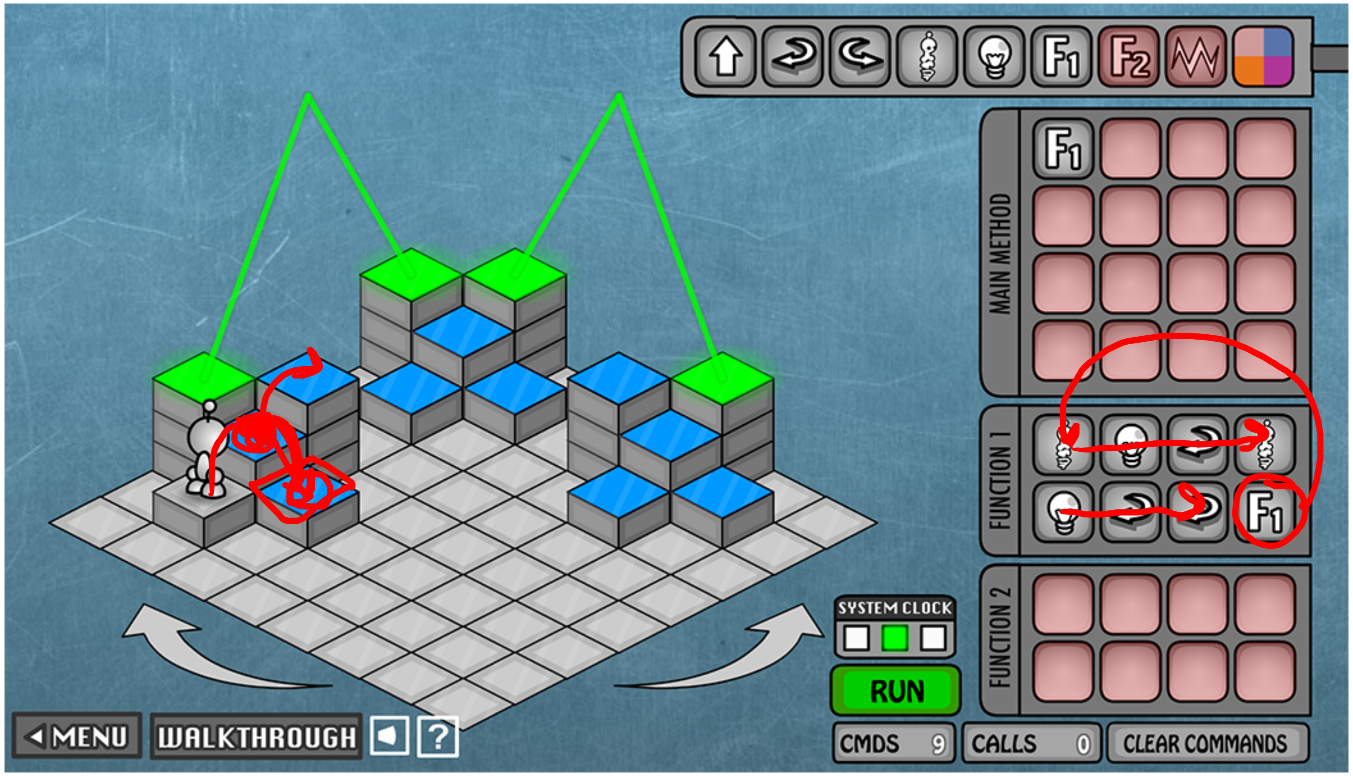


traverse steps

process riser & move to next

Recursion

- Special case where a function calls itself
  - "Conceptual unit" might apply again, immediately



Check-in Question

- Which of the following statements is TRUE?
  - Think for a minute, then talk with your neighbors!

A. An agent can learn new instructions

No, however can be combined

B. It is the agent's fault if the goal is not achieved

no, just following instructions

C. All ways to decompose a problem into functions are equally good

↳ execution time
· program size
· does it work?
· how easy to understand?

[✓ D.] None of the above

E. We're lost…

Functions Summary

- Functions are a foundational idea of computer science
  - Abstraction in action!
- *Functional abstraction* helps us solve problems:
  - Reduce complexity: identify and solve a coherent activity or action (sub-problem) that can be reused
  - Associate these sub-problems with intuitive names
  - Solve the whole problem by composing functions
- There is no "correct" way to abstract!

Looking Forward

- 1st reading check due tomorrow (1/9 before 3:30 pm)
  - ~~Discussion~~ on the reading in section tomorrow

- Continue to explore the concept of programming in the realm of Lightbot
  - Lightbot (checkoff) due tomorrow on 1/9
  - Symbolic Lightbot (checkoff) due by end of 1/10

- Lightbot Functions (submitted) due end of 1/13
  - Create functions using handwritten symbols

```
F.turnaround()  R,R.
```