# The Finch 2.0 Robot

COR 100.09 Game & Robotics Programming Fall 2023

November 28, 2023

## Programming the Finch 2.0

- You need to match language, purpose and platform (guidance chart).

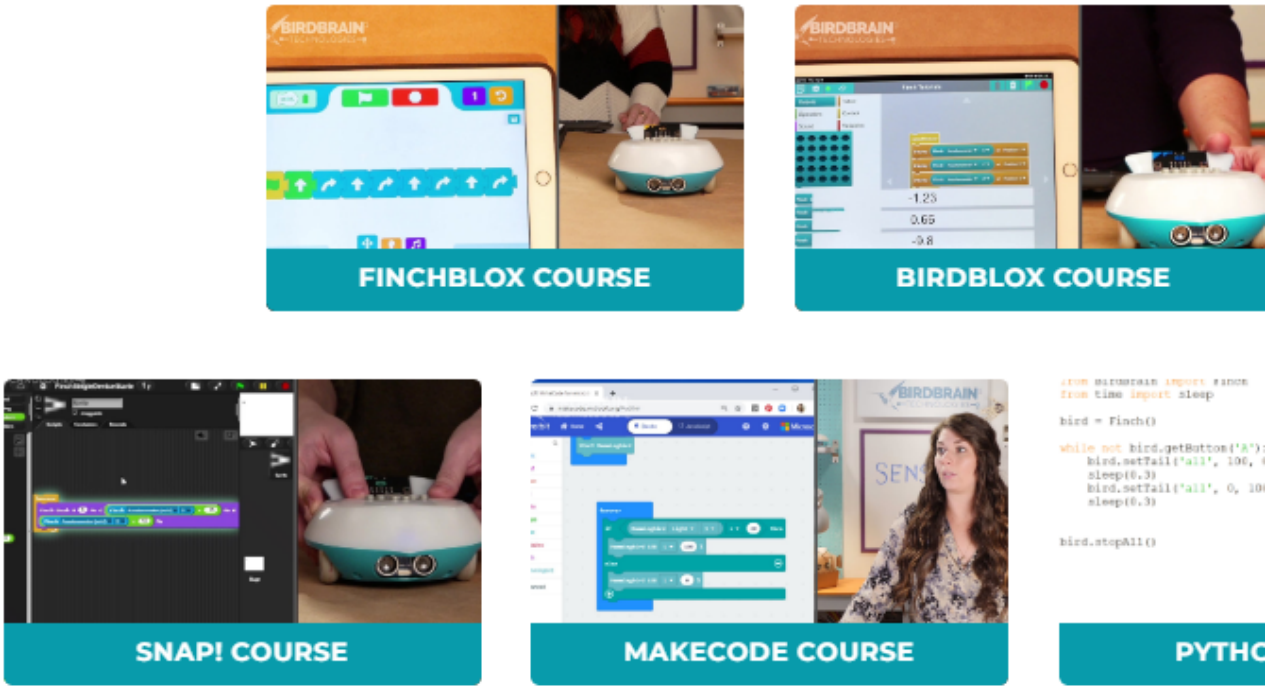



Figure 1: 5 Ways to program the Finch 2.0 robot

- The Finch can actually be programmed in 9 different languages.

## Robots and AI

- **What is Computational thinking?**

    1. Abstraction - taking away detail to see the big picture
    2. Evaluation - measuring impact or performance
    3. Algorithms - set of fixed, formal instructions
    4. Decomposition - taking big problems apart to be able to solve them
    5. Pattern recognition - recognize complex facts among a lot data

- **What is a robot?**


    – The word 'robot' was first used to denote a fictional humanoid in a 1920 Czech-language play R.U.R. (Rossumovi Univerzální Roboti – Rossum's Universal Robots) by Karel Čapek ("The new Adam and Eve")

- Technically, a robot is a non-virtual "rational agent".
- As "Artificial Intelligence", robots are "rational agents":

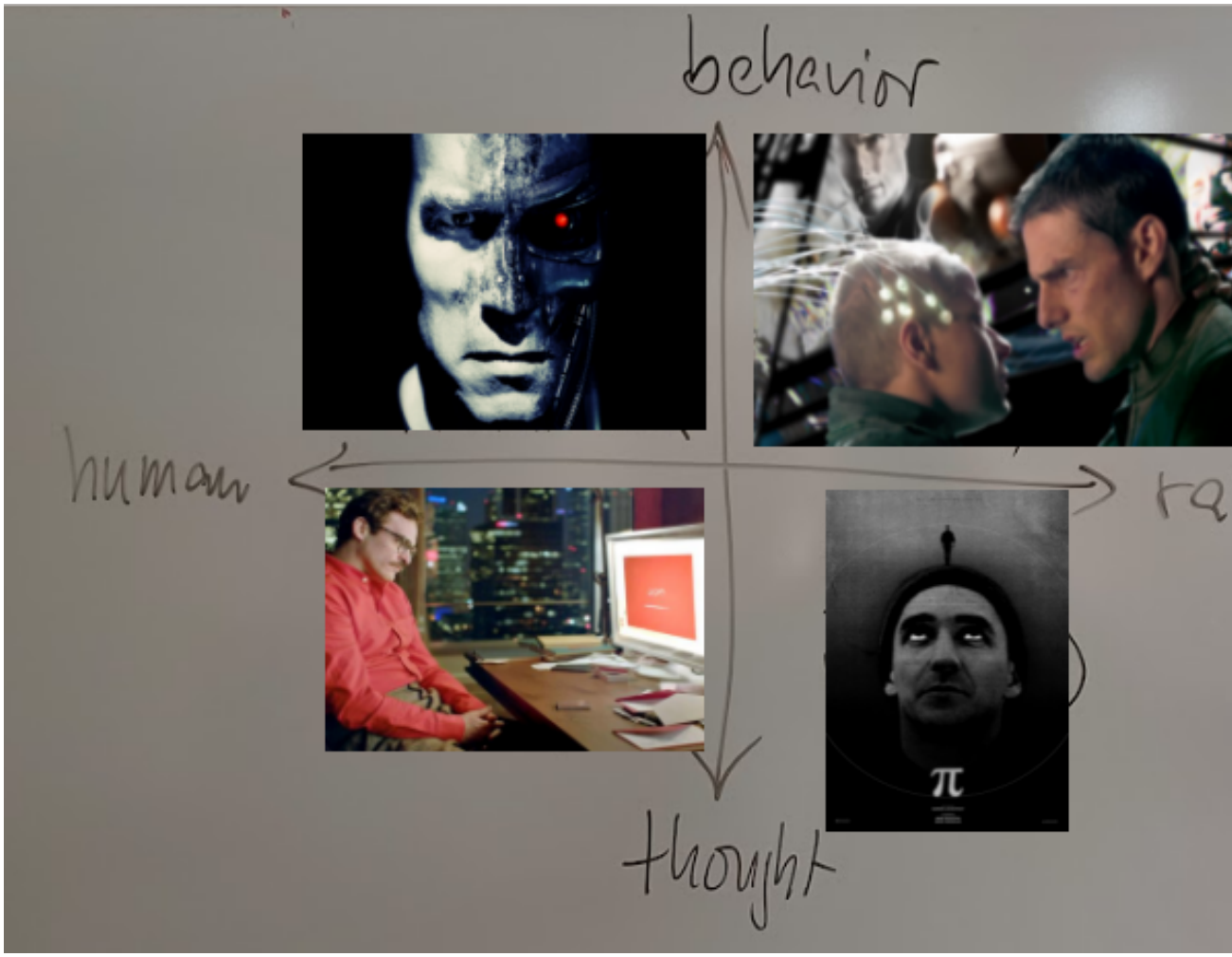
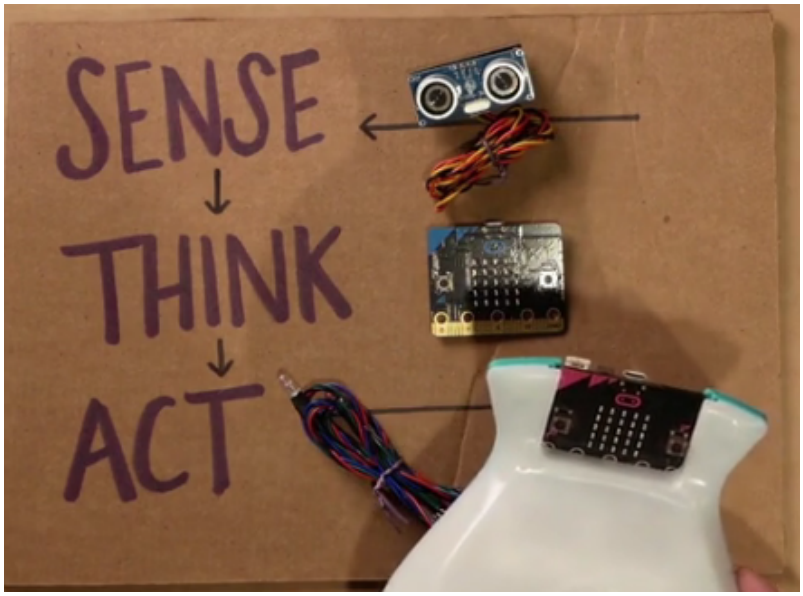



Figure 2: Artificial Intelligence as agents (AIMA, 2021).

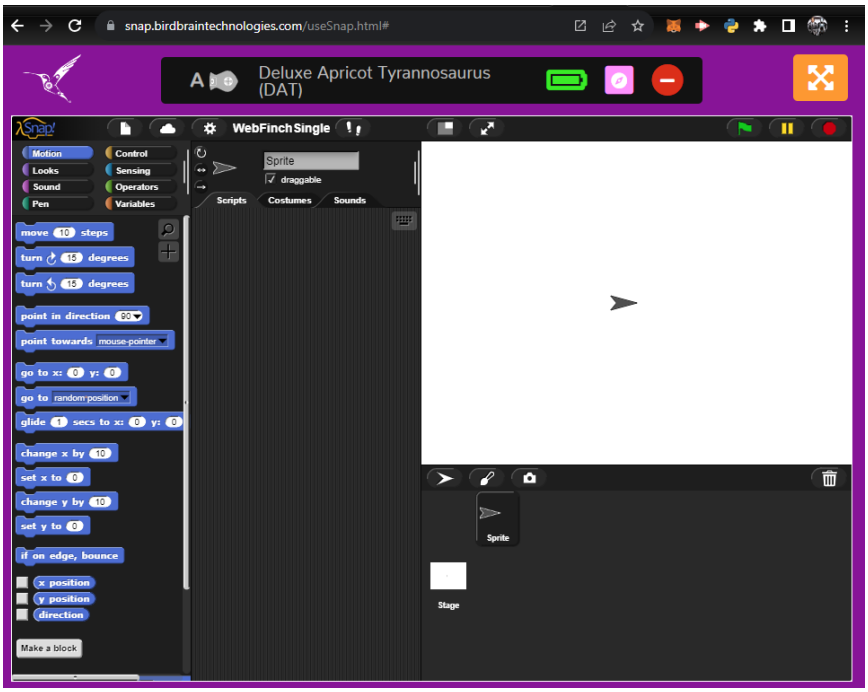
1. Upper left: machines that act like humans ("Turing test")
2. Lower left: machines that think like humans (Neural nets)
3. Upper right: machines that act rationally (unlike humans)
4. Lower right: machines that think rationally (not emotionally) - the robot arrives at conclusions after computing them.

- How does it work with the Finch?



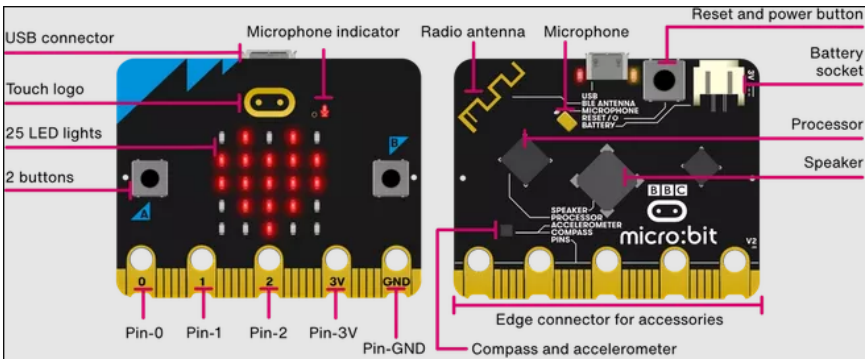

1. "Sense": Finch sensors at the front
2. "Think": Microbit controls actions
3. "Act": LEDs, motors, buzzer, moves, draw etc.

## Setting the Finch robot 2.0 up with Snap!




## Microbit input

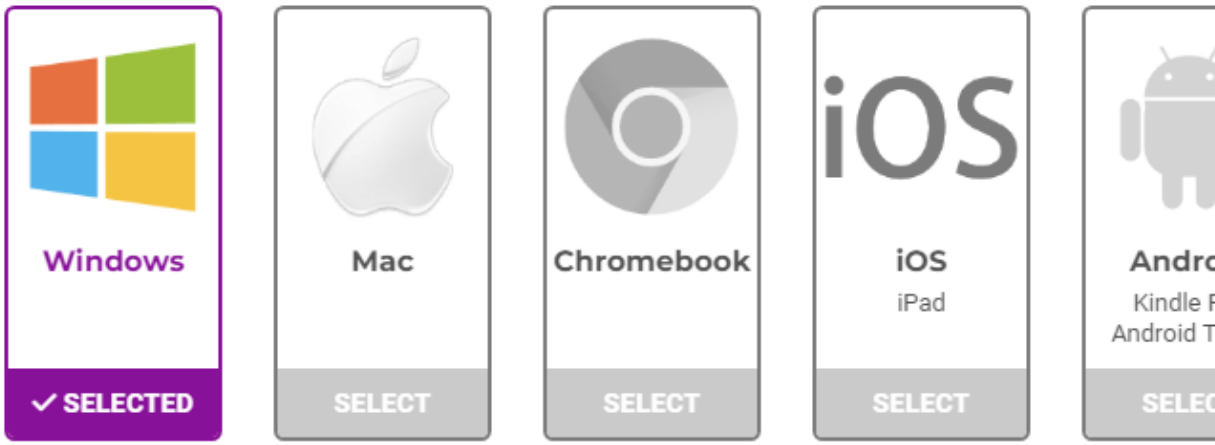
Microbit input: sensor data (light, acceleration, temperature, magnetism, sound) - see user guide.

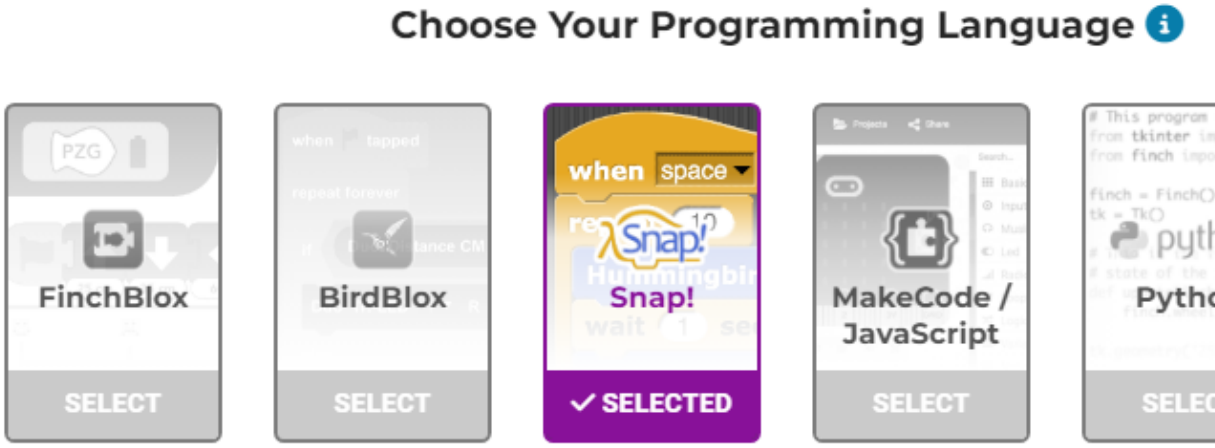



- Install the micro:bit by sliding it into the tail of the Finch and make sure it is charged:

## Choose language and programming platform

- You can choose a platform and a language:

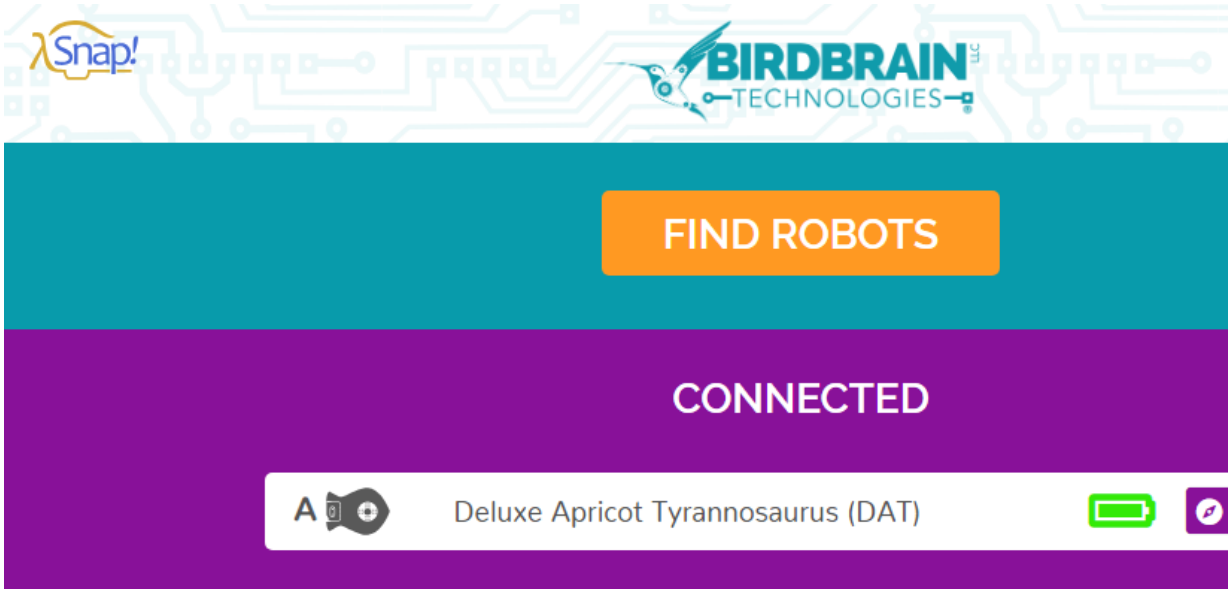



- There are many different activities available:


## Connect Snap! editor to your robot

- With Bluebird Connection Web app or Snap! Web app:
  1. Switch on your (charged) robot (black button at bottom).
  2. Click on `Finding robots` on the web page.
  3. Your robot should be listed - only connect to yours!




- Click on the compass and (re)calibrate until you see a smiley face:


## Installation details

- Installation details (already completed for classroom use):
    1. Full charge lasts ca. 8 hrs and takes 8 hrs to charge. Green lights when turning it on indicate the charging state (4 lights = full).
    2. Update the firmware with a `.hex` file for the micro:bit every 6 months or so (get the file from here). The micro:bit does not have an operating system but it has MicroPython.

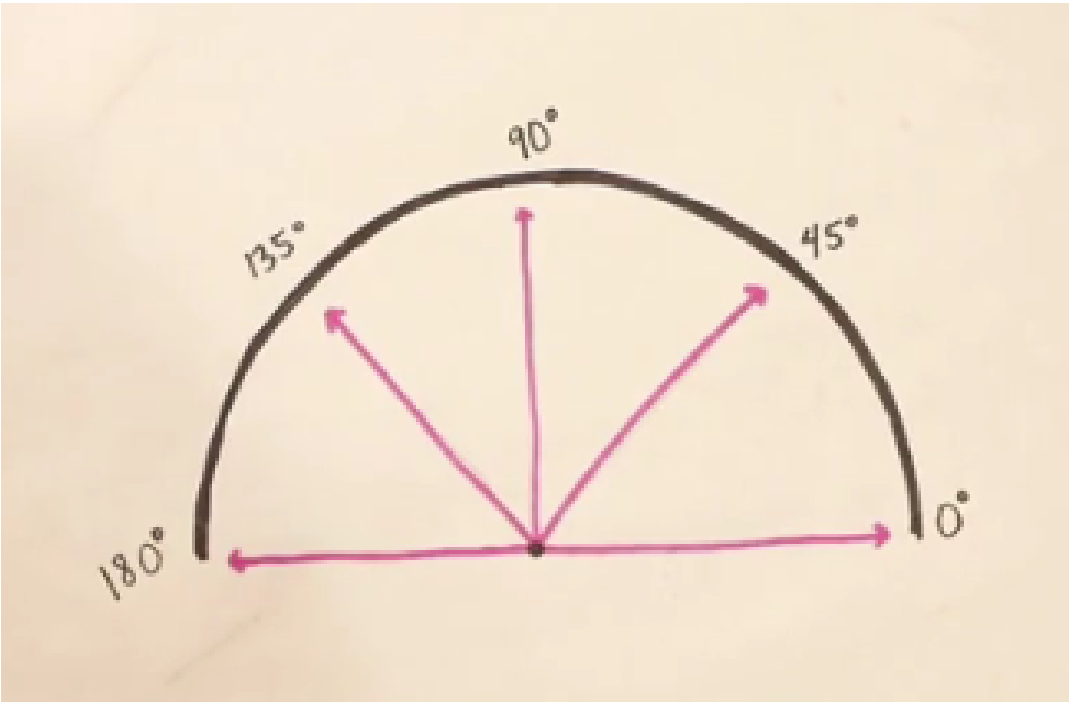
3. Once uploaded, the micro:bit responds directly by giving three letters (DAT - "Deluxe Apricot Tyrannosaurus") followed by pound # sign and its number, e.g. FNC3929.

- Change the turtle with the Finch Costume: download this file and `import` it to Snap!: tinyurl.com/finchCostume

- Try moving the Finch after connecting to it (don't let it go):


- If you're interested: there are 14 additional programming videos.

## Moving and turning



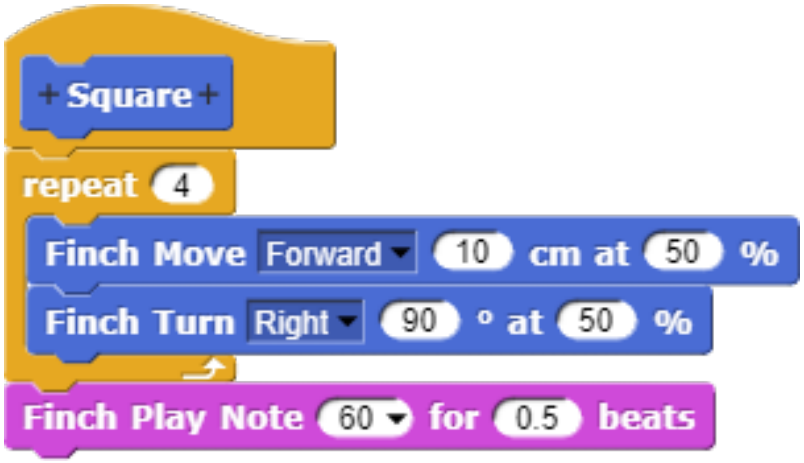

- Move forward 10 cm at 25% full speed, then backward by 20 cm at full speed:


- Turn right by 20 degrees at 50% full speed, then left by 90 degrees at 75% full speed:

- Repeat the turning motion 5 times:


- Practice:

  1. Make the Finch move in a square, then play a sound.
  2. Put this program into a `Motion` block `Finch square`:
     - In the `Motion` menu, `Make a block`
     - Inside the block, add the code below
     - Click OK
     - Drag the new block `Square` in the script editor
     - Run it (make sure the Finch is safe to move)



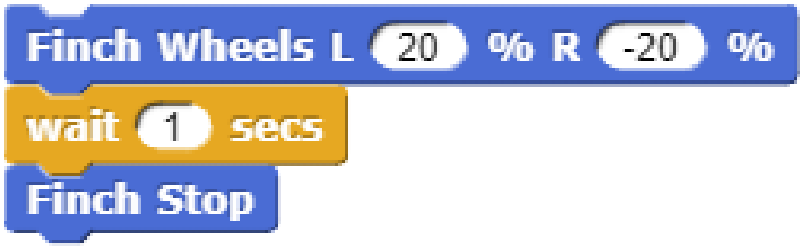

## Controlling wheels

- For a counter-clockwise spinning motion, make the left wheel go backward and the right wheel go forward:

Figure 3: Muriel Long with bicycle decorated for street procession.


- For a clockwise spinning motion, make the right wheel go backward and the left wheel go forward:

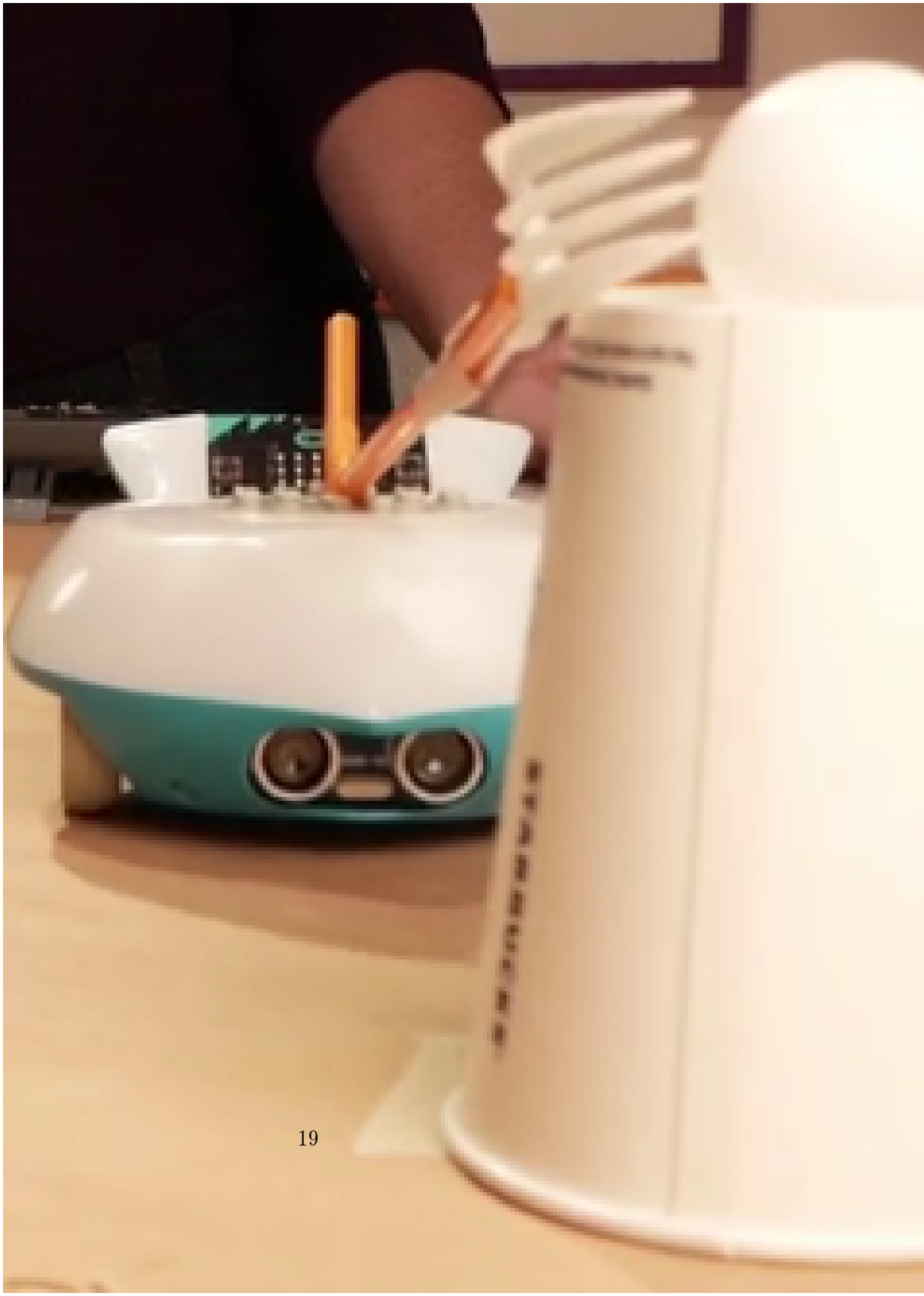

- To stop, use the "Finch Stop" block or the red STOP editor button:




- To make the Finch move around in a circle, make the wheels go at different speeds while moving forward:

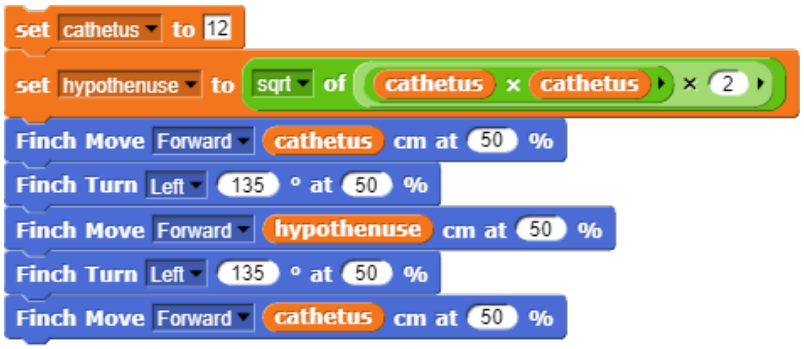

## Finch movement activities (example)

- Activity idea: finch jousting - known the ball off a cup using e.g. a fork on a straw:



## Drawing shapes with the pen mount

- Draw a right triangle (use Pythagoras to compute the length of the hypothenuse):

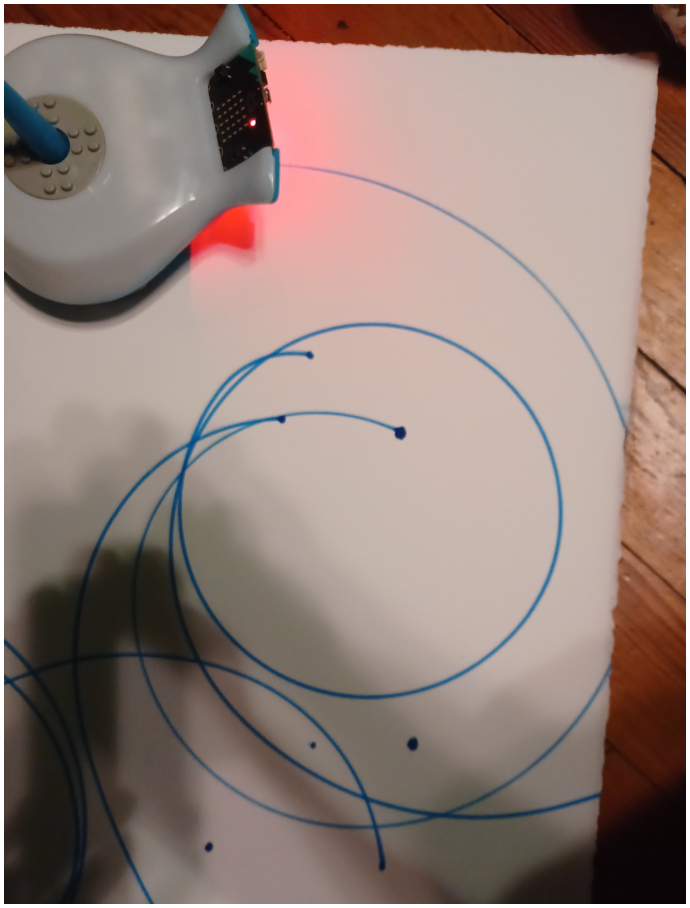


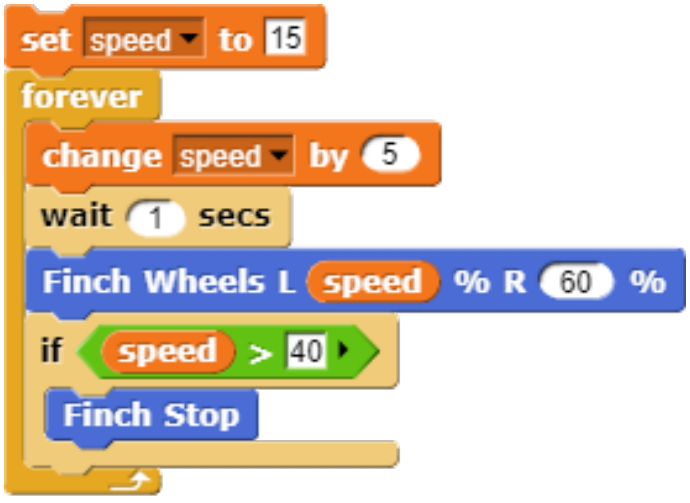

- Drawing shapes activity:

  1. decomposition (drawing the triangle step by step)
  2. evaluation (testing to see if you got what you wanted)

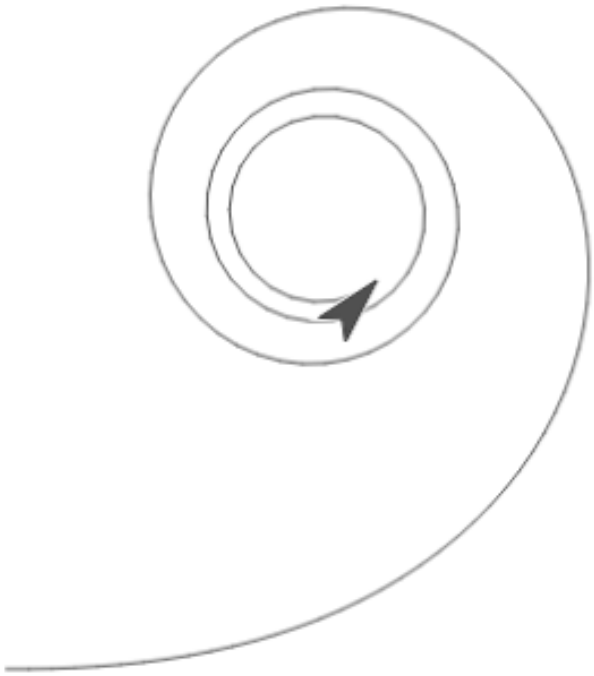
## Practice: Drawing a spiral

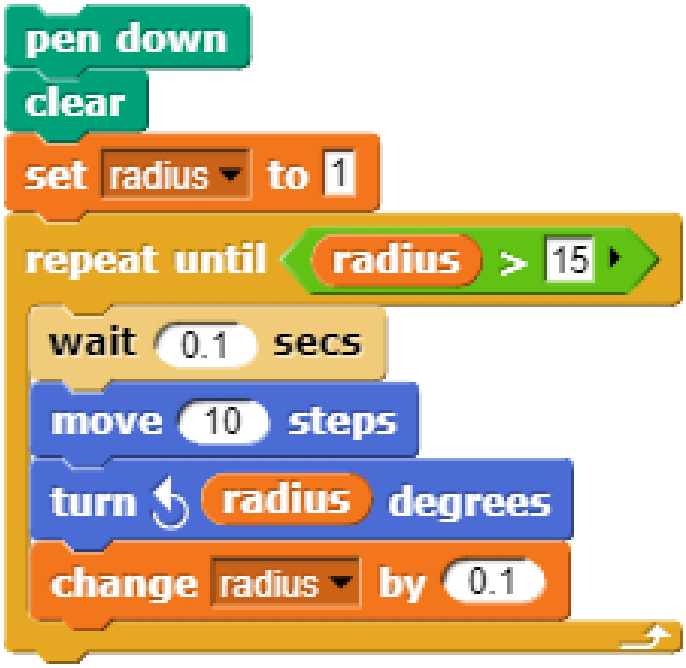
1. When the Finch's two motors move at the same speed, it moves in a straight line. When they move at different speeds, it moves in a circle. The following script works more or less - not quite sure about the `speed` values - but it produces a spiral.



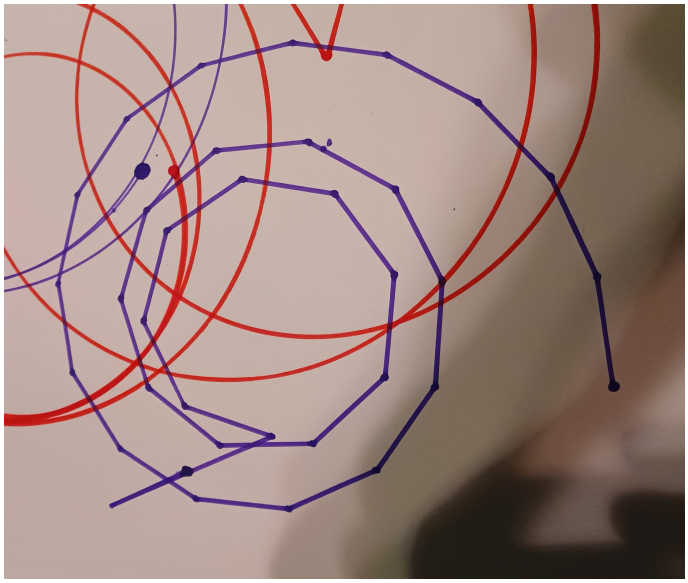

set speed to 15
forever
change speed by 5
wait 1 secs
Finch Wheels L speed % R 60 %
if speed > 40
Finch Stop

2. An alternative is to use the move + turn commands - this script draws a spiral for the turtle:

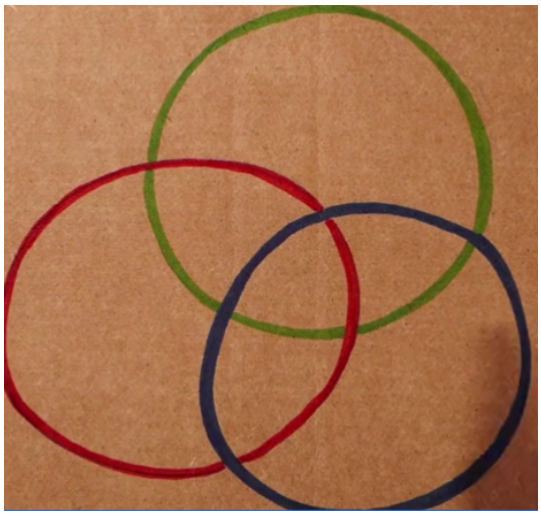




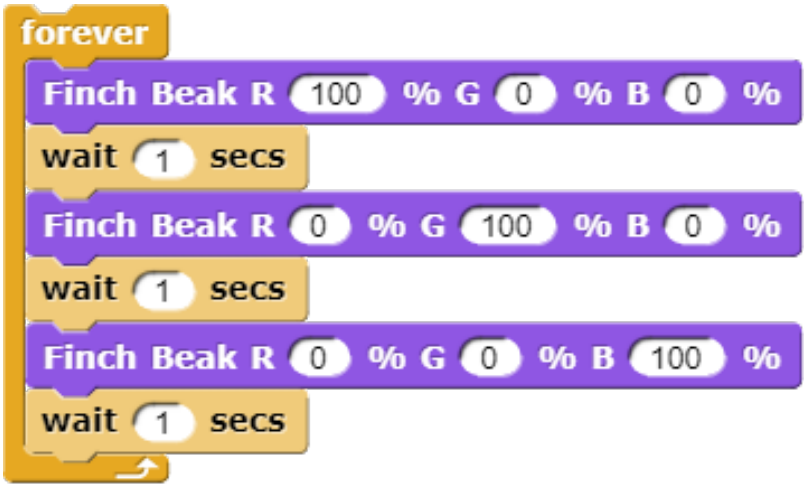

3. This will produce a spiral:




## Finch LEDs



- Red, blue and green are the primary colors of (white) light.
- Turning the Finch's beak red, green then blue forever:




- Why do blue and green make yellow?

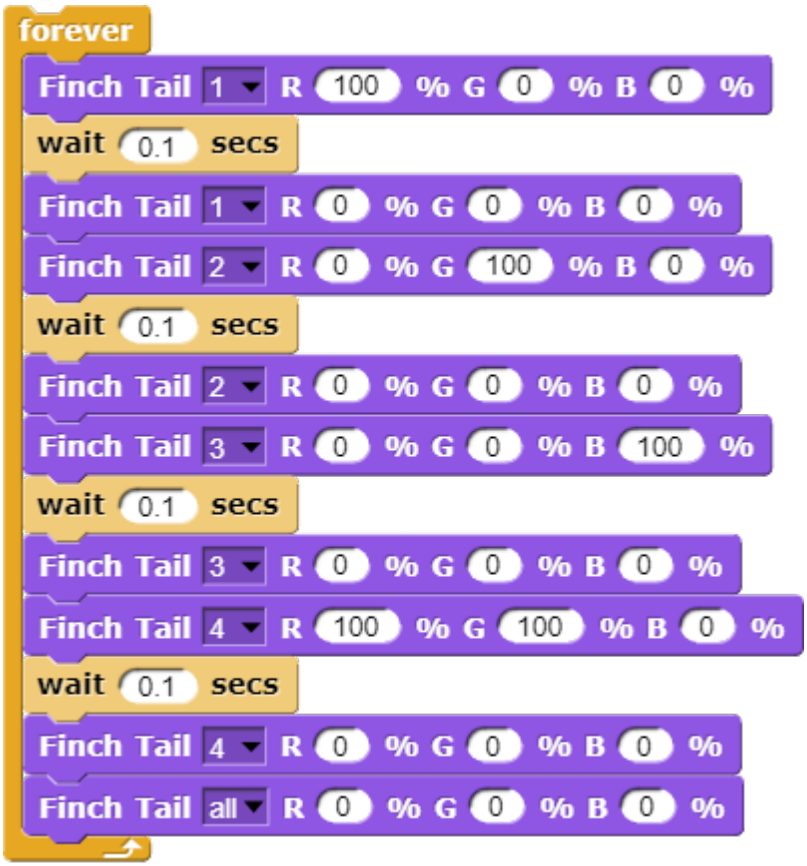
  The combination of red and green to make yellow can be explained in the context of additive color mixing, which is the way colors of light combine. This is the method used by digital displays like computer monitors, televisions, and projector screens.

  In the additive color model, the primary colors are red, green, and blue. When red light and green light combine, they create yellow light. This may seem counterintuitive if you're used to the subtractive color model (like mixing paints), where red and yellow make orange.

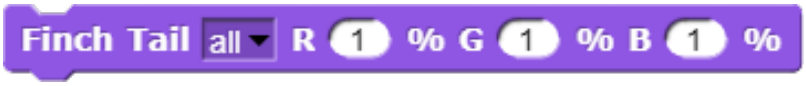
  The reason this happens has to do with how our eyes perceive color. We have three types of color receptors, or cones, in our eyes that are sensitive to short (blue), medium (green), and long (red) wavelengths of light. When we see yellow, it's usually because an object is reflecting both red and green light to our eyes, stimulating both the long and medium wavelength cones. Our brains interpret this combination as the color yellow.

  So, when a screen wants to create the perception of yellow, it emits both red and green light. Our eyes see this combination of red and green light, and our brain interprets it as yellow.

- Create an LED disco effect (see video): make LEDs 1,2,3,4 (from the right)
- Solution code:




- You get white light by mixing all colors with the same weight - the lower the number the fainter the light: try setting it to 1%




- How does a prism work?

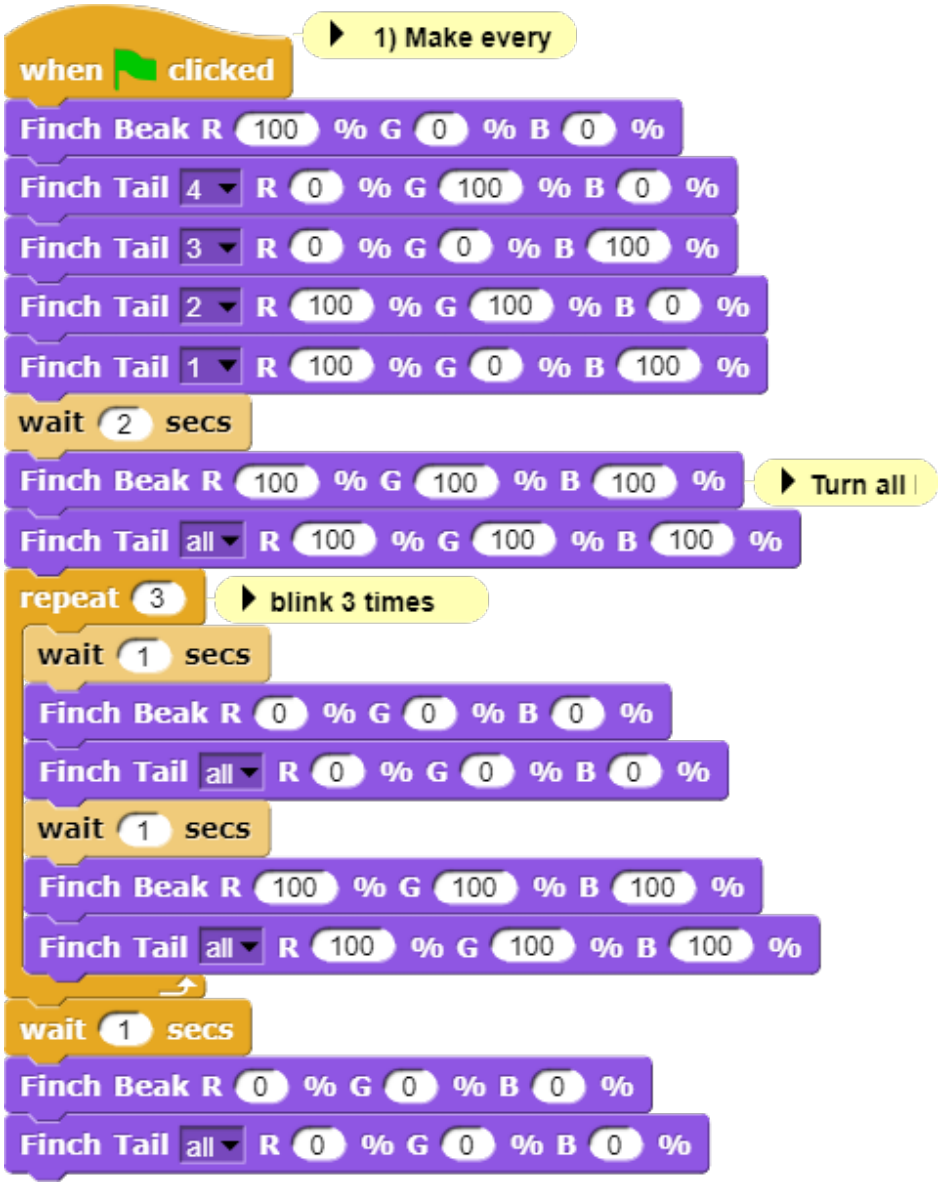
  As sunlight (white light) enters a prism, its components travel with different speeds (wavelengths) leading to refraction - as it exits, it is dispersed - cp. Snell's law, which

describes the relationship between the angles of incidence and refraction for light or other waves passing between two different isotropic media (e.g. water, glass, air where no direction is privileged over another).

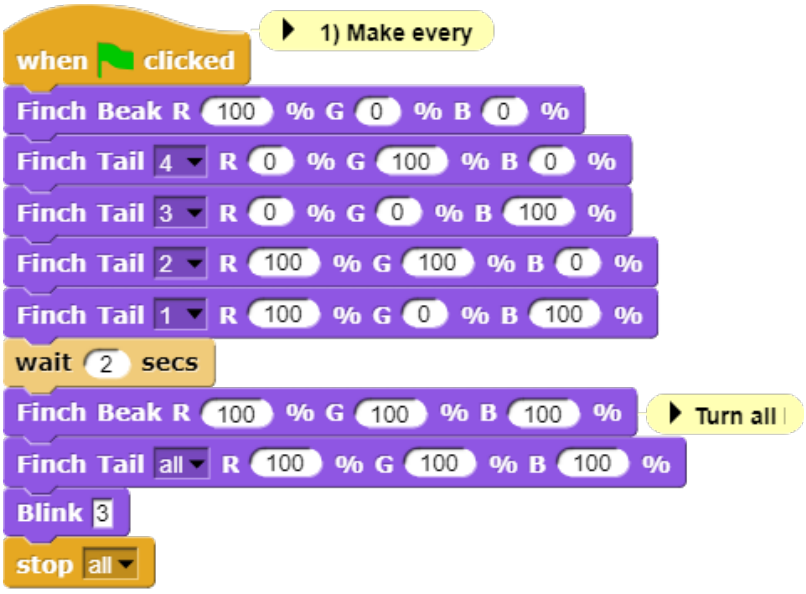
- Challenge:

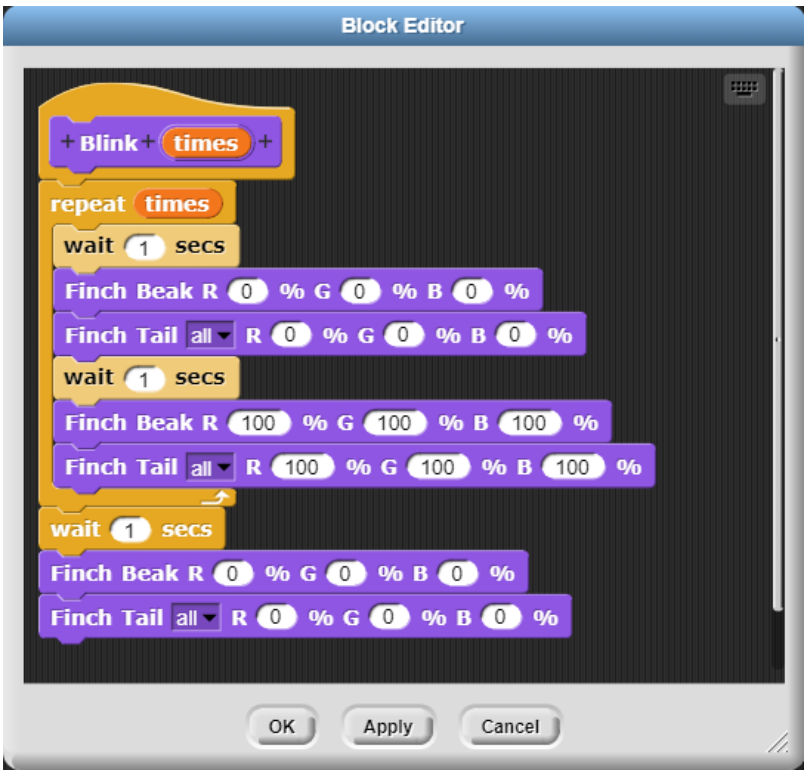
  1. Make every LED in tail and beak show a different color
  2. Turn all LED's white at the same time
  3. Make them blink off/on three times before turning all of them off.

- Solution code:




- Or with a "blink N times" Finch block:


when clicked
1) Make every
Finch Beak R 100 % G 0 % B 0 %
Finch Tail 4 R 0 % G 100 % B 0 %
Finch Tail 3 R 0 % G 0 % B 100 %
Finch Tail 2 R 100 % G 100 % B 0 %
Finch Tail 1 R 100 % G 0 % B 100 %
wait 2 secs
Finch Beak R 100 % G 100 % B 100 %
Turn all
Finch Tail all R 100 % G 100 % B 100 %
Blink 3
stop all


Block Editor
+ Blink + times +
repeat times
wait 1 secs
Finch Beak R 0 % G 0 % B 0 %
Finch Tail all R 0 % G 0 % B 0 %
wait 1 secs
Finch Beak R 100 % G 100 % B 100 %
Finch Tail all R 100 % G 100 % B 100 %
wait 1 secs
Finch Beak R 0 % G 0 % B 0 %
Finch Tail all R 0 % G 0 % B 0 %
OK
Apply
Cancel

- Activity: "Finch parade" - turn a Finch in a parade float using Lego bricks, blinking lights and movement mixes.
- Computational thinking:
    1. algorithm = order of things
    2. decomposition = parallel blocks
    3. evaluation = match goal to result

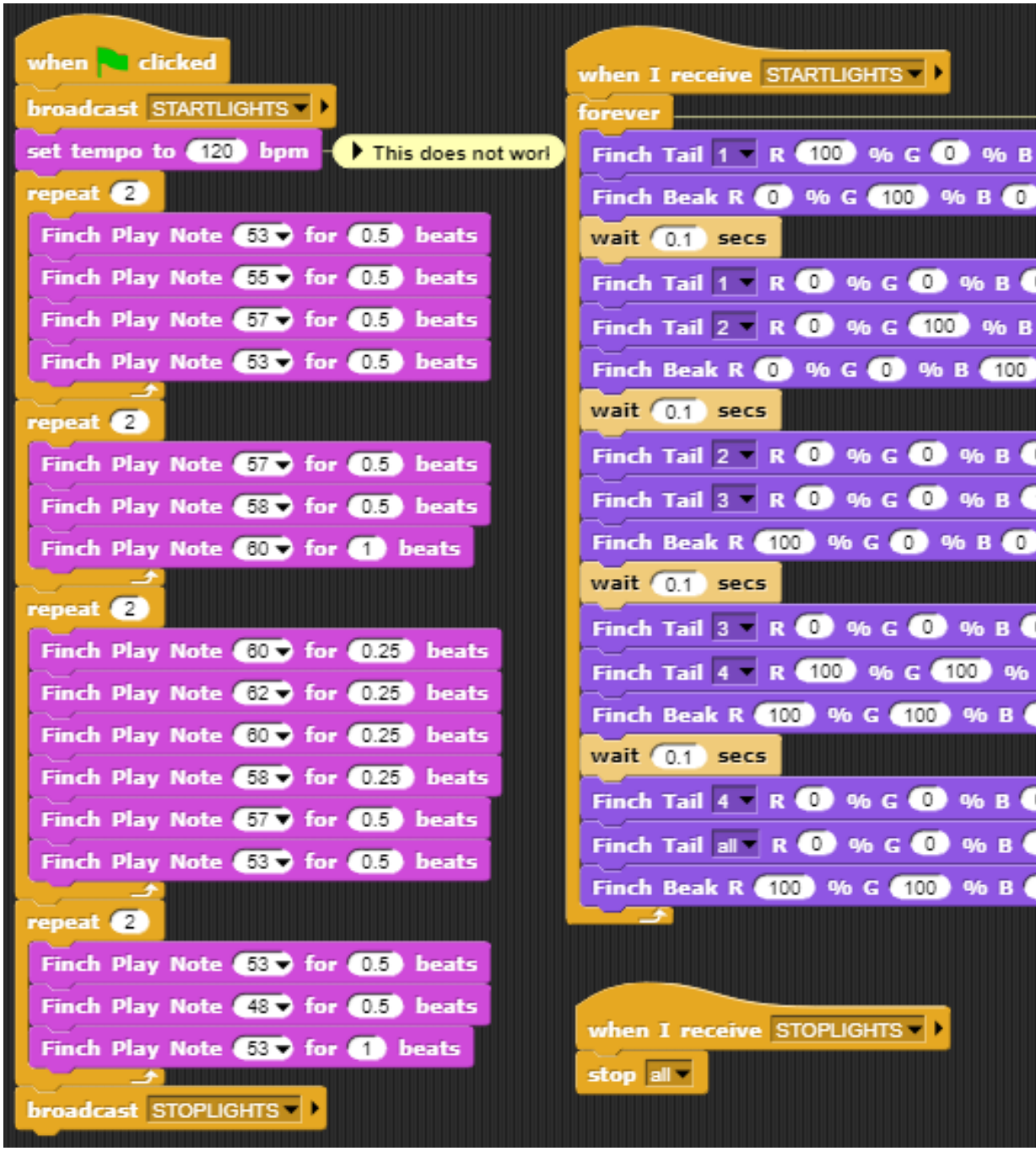
## Finch sounds

- There is only one Finch-specific sound command:


- The notes corresponds to a MIDI keyboard:


A MIDI (Musical Instrument Digital Interface) keyboard is an electronic instrument that sends MIDI signals to other devices like computers, synthesizers, or drum machines. It doesn't produce sound itself but triggers sounds stored in digital devices. MIDI keyboards come in different sizes, can have additional control features like pads and knobs, and usually connect via USB or MIDI cables. They are essential tools for digital music production and performance.

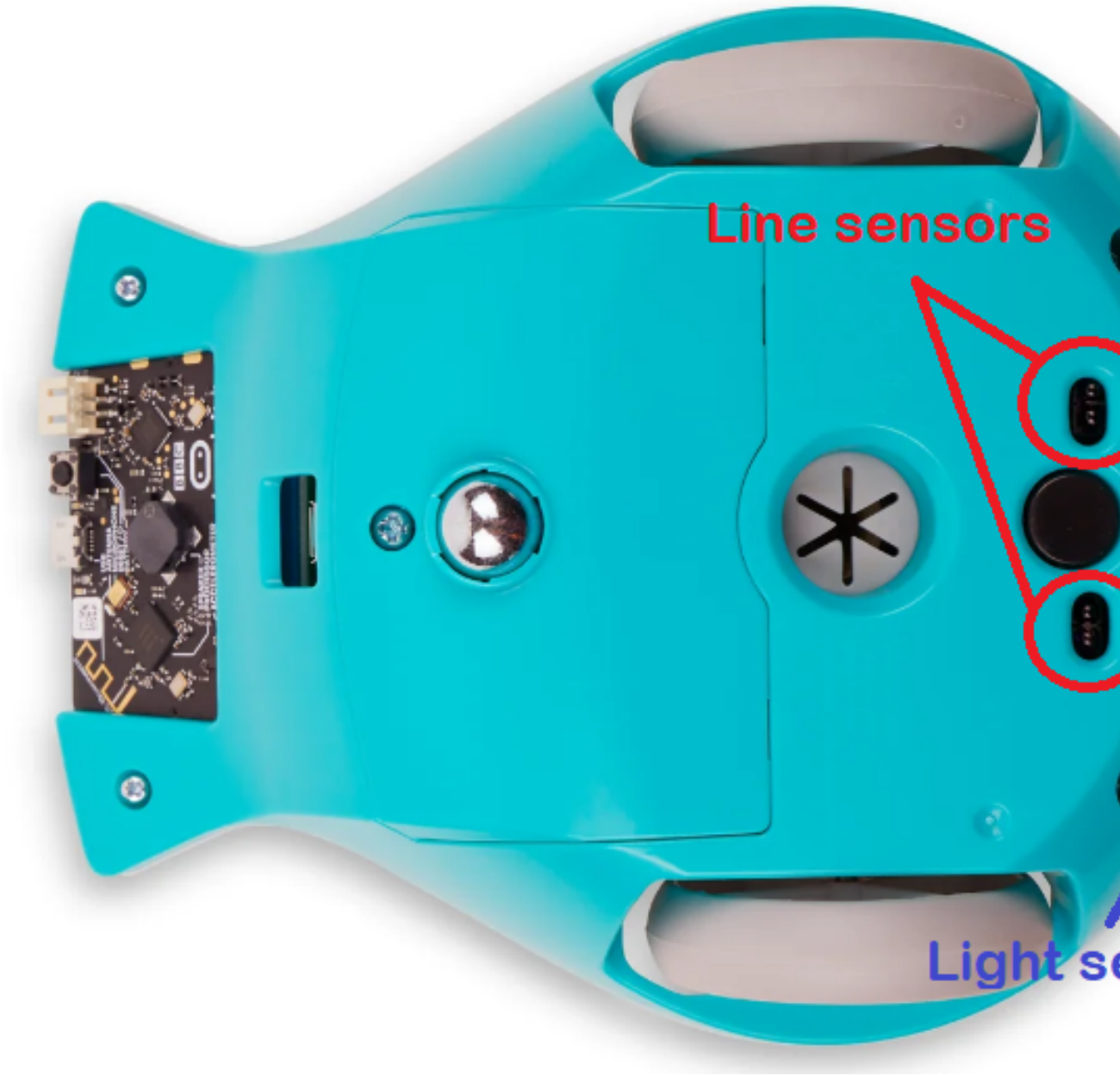
- Practice: create the 'Frere Jacques' with disco lights (for a video, see: tinyurl.com/frereJacquesDisco) - lyrics are here: tinyurl.com/BrotherJack.
- Solution (cloud file):




- Challenge: make the Finch play the song and dance to it. Here is a

fancy example, the "Finch chorus line" - https://youtu.be/qouVW_oVDPs

**Sensors: distance, lights, lines (numeric)**



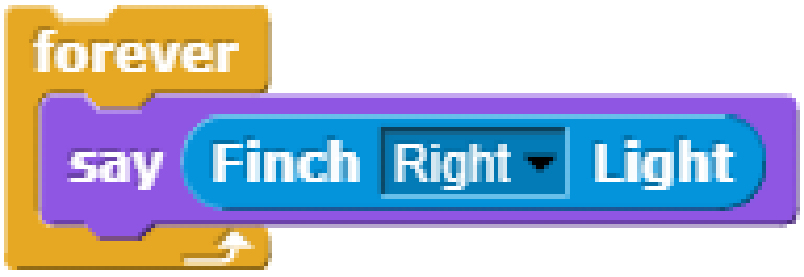

- Finch sensors include: light, line, distance, buttons A/B, accelerome-

ter, compass, temperature, orientation, and sound.

- The Finch can measure DISTANCE using the two sensors in front - click on this command to get the current reading:

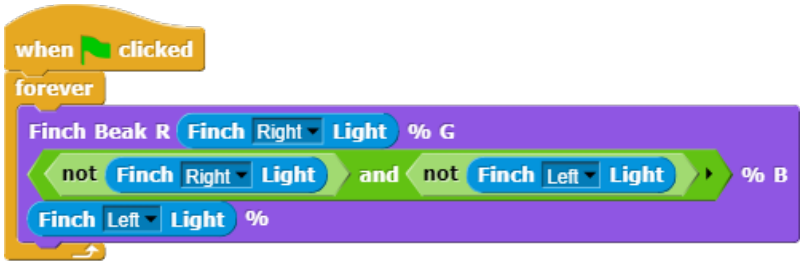

- To continuously update the distance reading, use the output as the string in a `say` command and wrap it in a `forever` loop:


- Data gathering mission: use the other blocks inside this code.

- Sensor overview video on the Snap/sensor programming page - especially for registering the minimum/maximum values.
- PRACTICE: use the value of the light sensor as a dimmer for LED where the LED gets brighter in proportion to increasing light falling on one of the sensors.
- Write a script that turns the beak BLUE when the left, and RED when the right light sensor are covered.
- Solution:

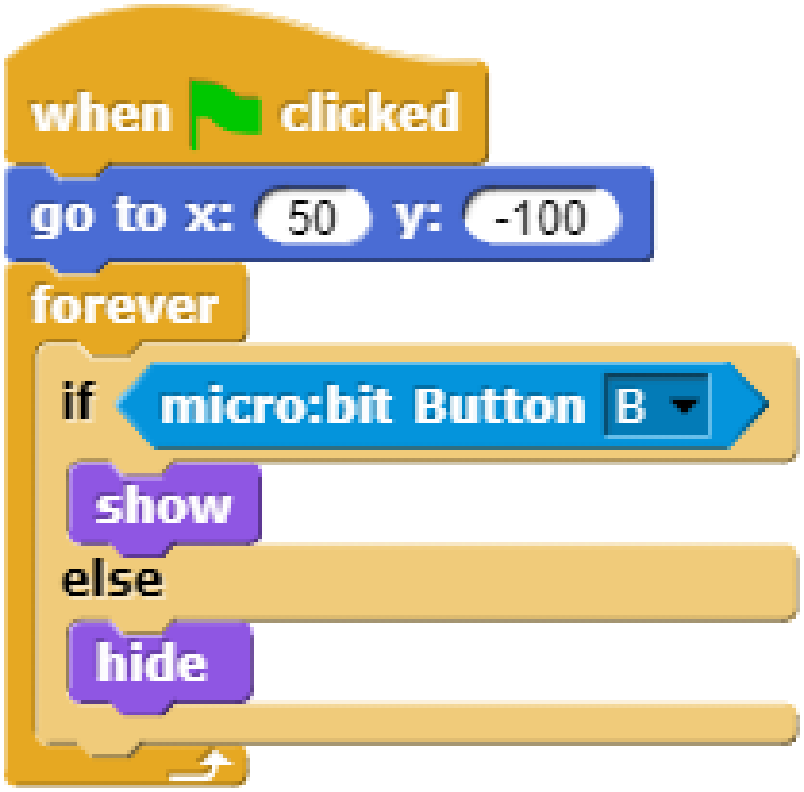

- PRACTICE: Modify the script to include that the beak goes GREEN when BOTH light sensors are covered.
- Solution:



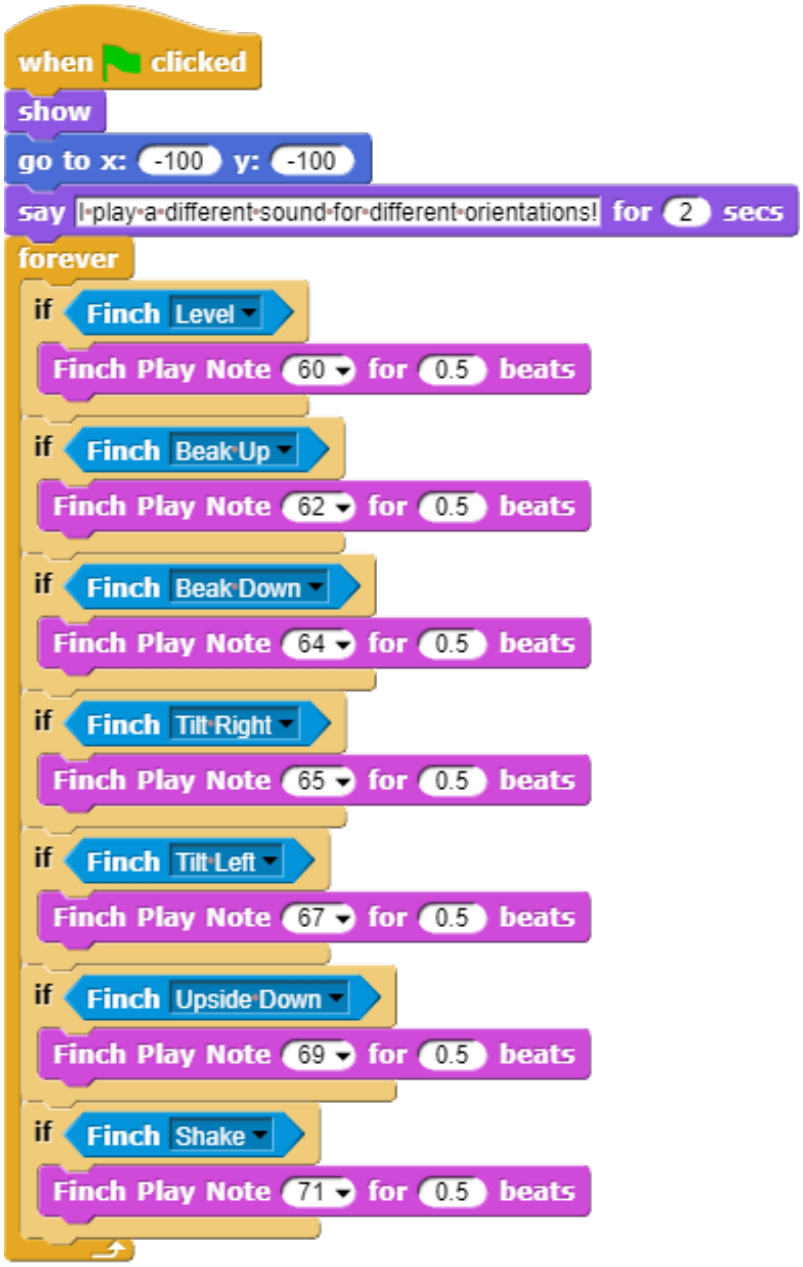
## Sensors: buttons, orientation (Boolean)


- Practice: write a script that shows a button "A" whenever the left A-button of the Finch is pressed, and a button "B" whenever the right B-button of the Finch is pressed.
- Solution:


when clicked
go to x: -100 y: -100
forever
if micro:bit Button A
show
else
hide

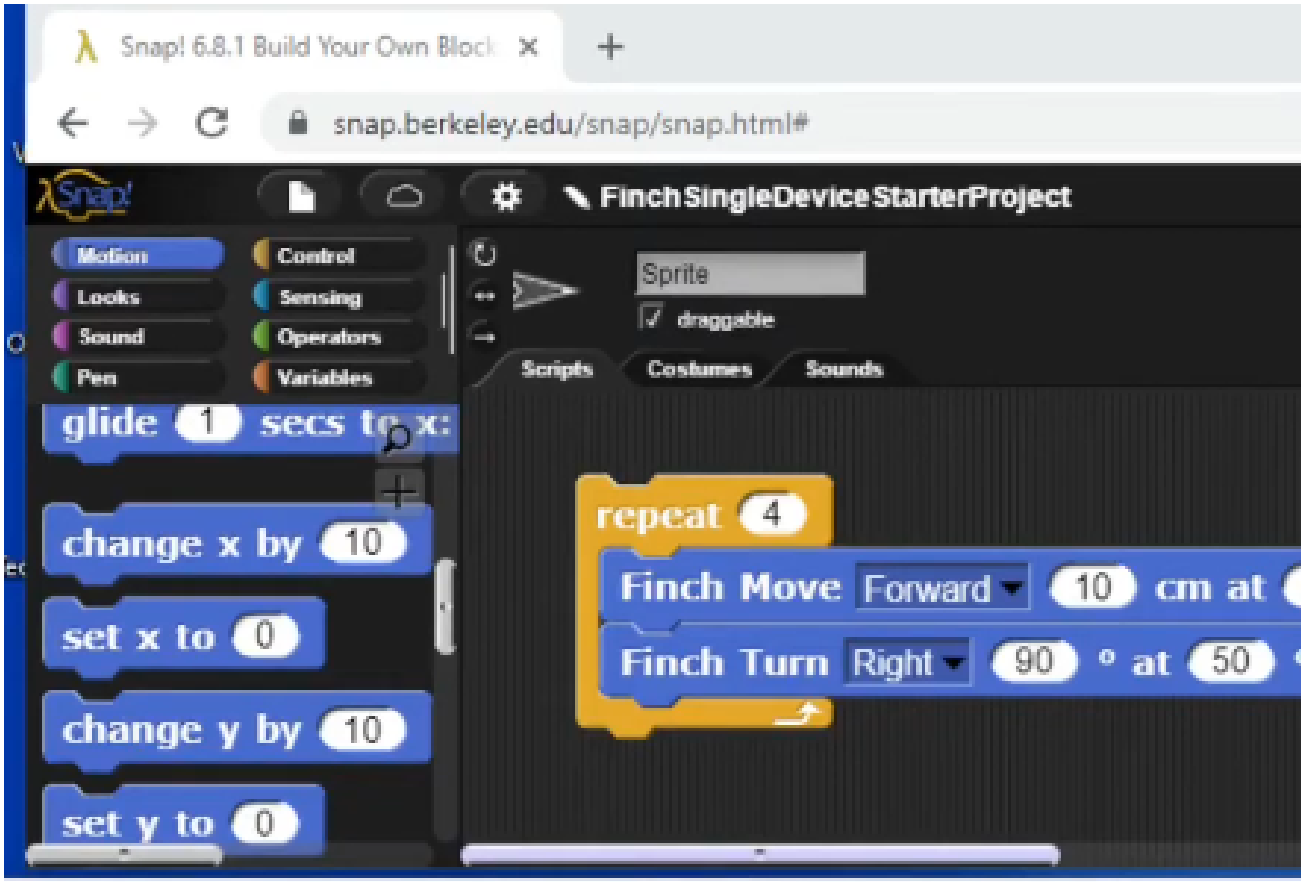


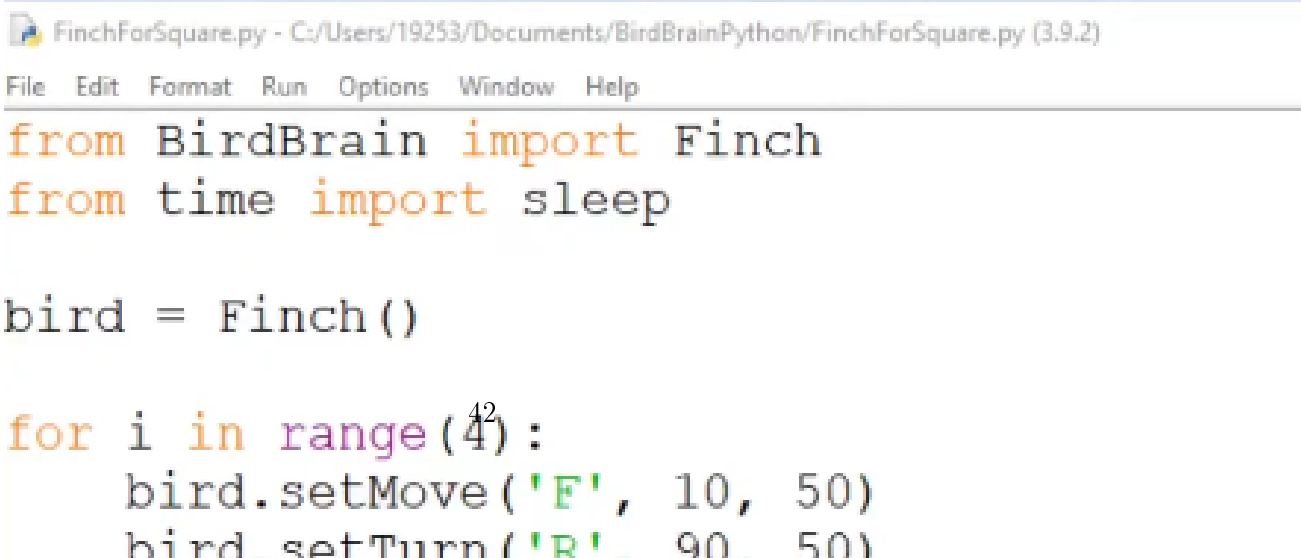

- Practice: write a script that plays the different sounds of a scale depending on the orientation of the Finch's beak, and try to play "Frère Jacques" with the Finch.
- Solution: the playing reminds me of the "bamboo Katana challenge" in "Ghost of Tsushima":


when clicked
show
go to x: -100 y: -100
say I play a different sound for different orientations! for 2 secs
forever
if Finch Level
Finch Play Note 60 for 0.5 beats
if Finch Beak Up
Finch Play Note 62 for 0.5 beats
if Finch Beak Down
Finch Play Note 64 for 0.5 beats
if Finch Tilt Right
Finch Play Note 65 for 0.5 beats
if Finch Tilt Left
Finch Play Note 67 for 0.5 beats
if Finch Upside Down
Finch Play Note 69 for 0.5 beats
if Finch Shake
Finch Play Note 71 for 0.5 beats

**TODO** Compass, acceleration, sound, temperature (numeric)

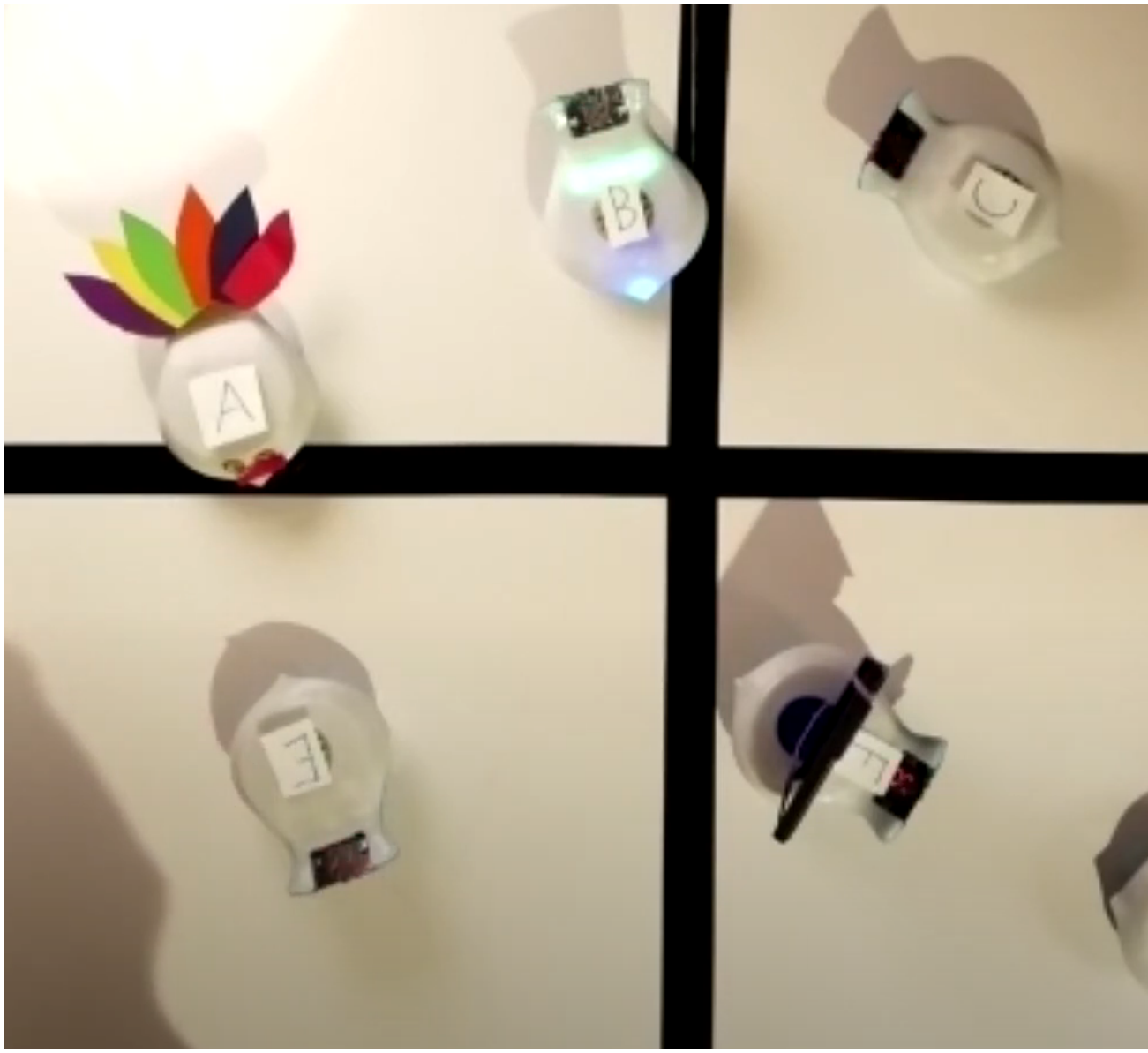
**Finch with Python - what changes?**



- Different setup: use this web app to connect.
- Different course to learn how to (must register).
- More and different ways to work with numeric and Boolean data.
- Good for using the Finch as a data gathering device (data science).

## SOMEDAY Play with the Finch robot remotely




- I will set up a server for remote Finch manipulation (see here).
- Let me know if you want to test that for me in July/August.

- Instead of spending 150$ on a Finch + micro:bit you get to use the robot for free at a distance and also learn client/server computing.
- There is no guarantee that this will succeed (depends on local infrastructure/firewalls) but I'll give it a try.
- This is how it looks (Finch Robot 2.0 Playground - 11/30/20).