Rover Coder
with Finch robots

Students explore the sensor capabilities of the Finch robots, and code them to respond to obstacles.

AGE
- Gr 4+ (tested on Grades 4-6)
  - For older grades, see note in Additional Resources (tested up to Grade 9)
- Possible for a savvy 3/4 split class

OBJECTIVES
Curricular content:
- Curricular competencies from Science, Math, and ADST across many grade levels
- Science 4: Living things sense and respond

Lesson objectives:
- Students will understand machine inputs & outputs through the lens of animals sensing & responding
- They will be able to code a robot to detect its surroundings and respond to avoid obstacles

MATERIALS
- Finch robots & micro-USB cords, 1 per 2-3 students
- Computer devices, 1 per 2-3 students
- Space for the robots to move around

SET UP
- Have students log on to their devices, in groups of 2-3

ACTIVITY OUTLINE
Overview and Suggested Timeline:

Introduction	10 minutes
Testing Ultrasonic Sensors	15 minutes
Conditionals	25 minutes
Reflection and Wrap-Up	5 minutes

Introduction
- All living things sense and respond. Why? Why do organisms need to sense things in their environment, and how might they respond?
- Machines have a similar functionality with inputs and outputs. Inputs are like senses, it's how the machine takes in information. Outputs are like responses, it's how the machine puts out information.
- We're going to experiment with one of the sensors on these Finch robots, and code it to respond to what it detects.
  - First, we need to learn how to code it!

Additionally, we gratefully acknowledge the support of the YP Heung Foundation, The McLean Foundation, and the Province of British Columbia.

Testing Ultrasonic Sensors

- In makecode.microbit.org, click on "New project"
  - To add the Finch category, go to "Extensions", then search for "Finch", and click on the picture of the robot
- Find the following 5 pieces of code:
  - Basic:
    - On Start
    - Forever
    - Show number
  - Finch:
    - Finch number (cm)
    - Start Finch
- *Take a few minutes to fit the pieces together.*
  - *Then, as a class, discuss how to put them together*
- Download the code onto the microbits. Power on the Finches, and test out the code.
  - What is this code doing? What do the numbers mean? How can you change the number?
- The number indicates the distance in cm from the front of the Finch robot.
  - How does the sensor work?
  - It's an ultrasonic sensor. It sends out a sound that's out of our range of hearing, and then measures the echo.
  - What other things use sound to "see"?
    - Bats! Why do bats use echolocation?
  - Why is it useful to have a robot that can "see" things in front of it?
    - So it doesn't bump into them!
    - Let's code our robots to move around without bumping into things.

Conditionals

- We have to learn how to code our robots to be able to make a decision, to do something different depending on how much space it has to move forward.
- We'll need to use "conditionals". These let the robot check if a condition is true or not, and depending on the answer, it'll pick what to do.
- The condition it needs to check is how much space is in front. Remember, this information is in the oval "Finch Distance (cm)" block. We're going to use that block, plus one from the Logic category that looks like a stretched-out hexagon, that says " 0 = 0 "
  - We need to put the "Finch distance (cm)" block inside one of the 0s.
  - This is the condition that the robot is going to check. We want it to make a decision depending on if it has enough space to move in front, or not.
  - Think about how much space, in cm, it should have for it to move forwards. Put that number in the other 0.
  - Now, click on the "=" sign. It's a drop-down menu, and we want to change it to say greater-than (">") or less-than ("<"). Which symbol you pick will depend on which side of the equation is the Finch Distance, and also what you want the robot to do.
- Now that we have our condition, we need to tell it how to make a decision.
  - From the Logic category, get the block "If true then else" and pop that into the "forever" event block
  - Note the shapes: the stretched hexagon fits on "true"

Additionally, we gratefully acknowledge the support of the YP Heung Foundation, The McLean Foundation, and the Province of British Columbia.

    - We want the robots to move forward if they have space, and do something else if they don't have space, like turn.
    - Grab "Finch move forward 10 cm at 50%" and "Finch turn right 90° at 50%"from the Finch category.
        - These are going to go in the different slots in the if/else block. Where you put them depends on the condition you made.
        - Download the code and test it out!
- Keep testing until it does what you want!
    - What is the relationship between the number on the other side of the equation from "Finch Distance (cm)", and how far the robot moves forward? What happens if one is a lot larger than the other?
    - If the robot is only moving forward when something is blocking it, and turning when it has space in front of it, that means it's doing the opposite of what you want. What do you need to change?

Optional Extensions

- Random turning
    - From the Math category, get the oval "pick random 0 to 10"
    - Put it in where it says 90° on "Finch turn 90° at 50%"
    - What do you have to change in the random block, to make to robot turn a random direction when it encounters an obstacle?
- While condition
    - Instead of using "if true then else," from logic, use the "while false do" from loops. This repeats whatever is inside that block while the condition is met
    - *This is slightly more challenging than if/else because instead of just swapping where things are, students have to really understand what the condition block (the stretched hexagon) is doing*
- Alarm
    - Instead of just Finch movements, have the microbit make a sound (from the Music category) when it encounters an obstacle

Reflection and Wrap-Up

- Put away robots before asking reflection questions
- Share challenges and successes from the activity with each other
- Ask the reflection questions (below)

REFLECTION QUESTIONS

- What is the purpose of the < or > signs in the code? How does knowing math help us to code?
- In what other situations would we want a robot to be able to do something different depending on what its sensors are detecting? Where might we use robots that can move around without bumping into obstacles?
- Why is it important to test out our code as we go along? Why did we start by testing out the ultrasonic sensor?

Additionally, we gratefully acknowledge the support of the YP Heung Foundation, The McLean Foundation, and the Province of British Columbia.

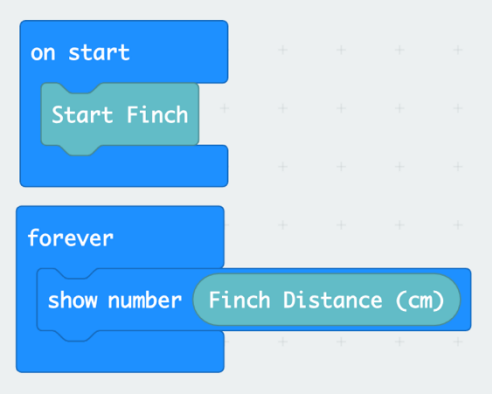
TROUBLESHOOTING TIPS

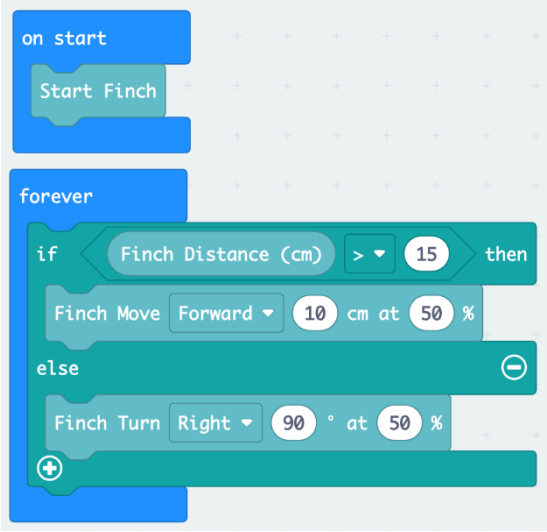
- Familiarize yourself beforehand with loading code from makecode.microbit.org onto a microbit, using the devices that your students will have access to. Instructions differ depending on the device.
  - See microbit.org/start for instructions and a guide to using microbits.
  - You will need to show students how to access the code, and how to load the code onto the microbits
- Make sure that the logic comparison block isn't the one with 0 in quote marks. ("0" = "0")
- If the robot is moving forward when it's blocked, and turning when it isn't, there are 3 things you can change to make it do the opposite:
  - Flip the direction of the > or < sign
  - Move the "Finch Distance (cm)" oval into the other side of the equation
  - Swap the movement blocks in the if/else code
- The students must remove the microbit from the robot before plugging it in to the computer.

ADDITIONAL RESOURCES

- Older students may benefit from going a bit less step-by-step than this lesson. For them, still work with the different lesson sections, but try just showing them the blocks they'll be using and let them experiment with how they fit together, trying to figure out how to make the code do what they want.
  - Use the Optional Extensions for the lesson for older students, or groups who solve the challenges a bit faster and need something new to try

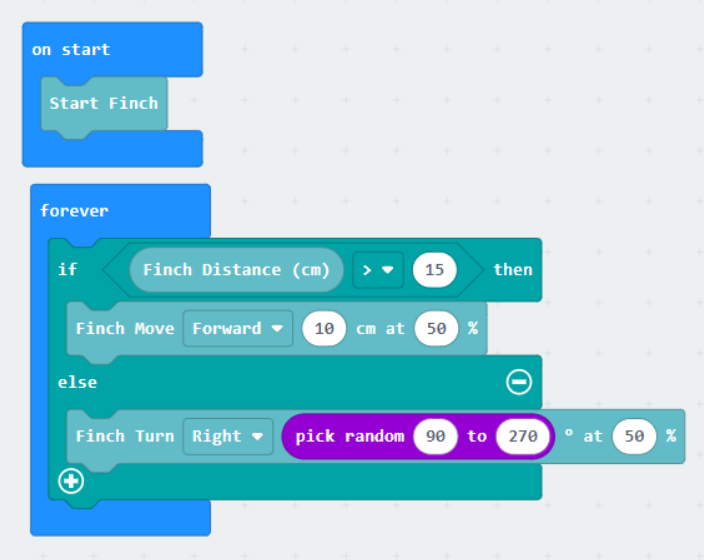
SAMPLE SOLUTIONS

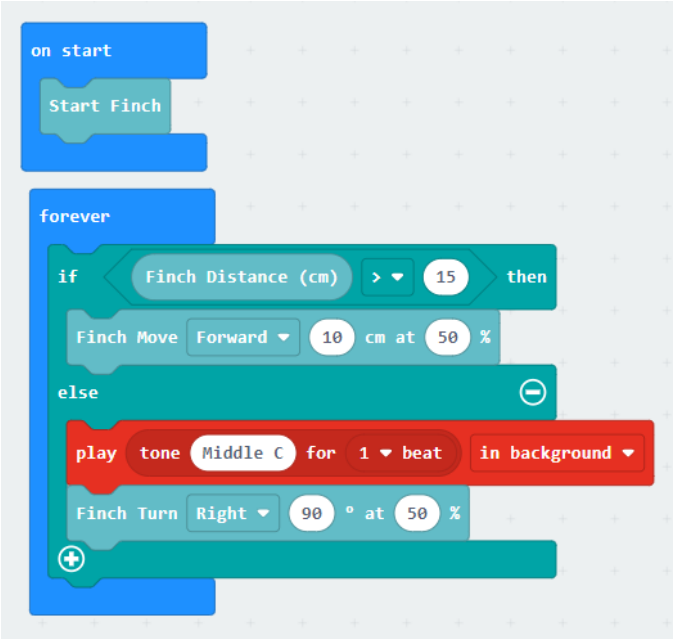


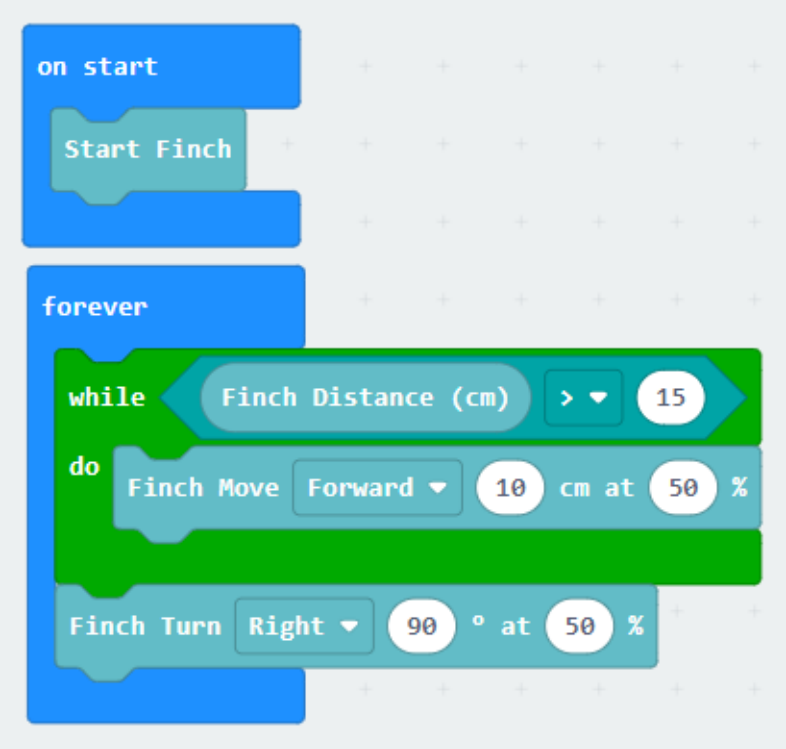


Additionally, we gratefully acknowledge the support of the YP Heung Foundation, The McLean Foundation, and the Province of British Columbia.

Optional extensions:







Additionally, we gratefully acknowledge the support of the YP Heung Foundation, The McLean Foundation, and the Province of British Columbia.