CANADA LEARNING CODE

Canadarm2

By: Caitlin Davey
Duration: 1 hour

LEVEL	SUBJECTS	PROVINCES / TERRITORIES	TOOL
Grades 4-6	Technology Education, Science	Across Canada	Scratch

Overview

In this activity, students will imagine that they are Chris Hadfield remotely controlling the Canadarm2. They will use Scratch to create a simulation in which they put a new module on the International Space Station.

Prep Work

- Review the completed version of the project: bit.ly/canadarm2-example
- Print the Solution Sheet for the main activity: http://bit.ly/canadarm2-solution (see below)

Lesson

This lesson was made in partnership with

CHRIS HADFIELD   let's talk science

Key Coding Concepts

- ✓ Abstraction
- ✓ Conditionals
- ✓ Events
- ✓ Loops
- ✓ Parallel execution
- ✓ Variables

Terminology

Abstraction
Pulling out specific differences to make one solution work for multiple problems

Conditionals
Making decisions based on conditions i.e. if some condition is met do something, else do nothing or something else

Minds On

Watch this video of Chris Hadfield:
bit.ly/clc-chris-hadfield

Ask: What do we know about the International Space Station?

Introduction

Building the International Space Station (ISS) is no easy task! The crews of the ISS have to attach modules weighing tons, extend solar panels longer than a bus, and haul equipment to and from the space shuttle.

Canadarm2 is a Canadian-made robotic arm located on the International Space Station (ISS) that helped astronauts such as Marc Garneau and Chris Hadfield build the ISS in space. Since 2001, the 17-metre long Canadarm2 has been helping move equipment, supplies, astronauts, and even spacecraft such as SpaceX's Dragon capsule around the ISS.

In the International Space Station (ISS), actions have to be done in the correct order to place modules where they need to go on the ISS. It will be a "lock & key" fit, so that the piece must go on properly in order for it to work.

Watch the Hadfield behind the controls of Canadarm2 video (https://chrishadfield.ca/videos/hadfield-behind-the-controls-of-canadarm2/):

- Q: In what ways is the Canadarm2 used on the International Space Station?
- Q: Where is a good place to go to control the Canadarm2?
- Q: Why are there so many cameras to help guide the astronaut when controlling the Canadarm2?
- Q: What are some of the challenges of using a robot such as the Canadarm2?

Events
One thing causing another thing to happen i.e. 'when clicked' block

Loops
Running the same sequence multiple times i.e. repeat or forever blocks

Parallel execution
Making things happen at the same time

Variables
Stores a piece of information i.e. score of a game that increase by 1 value for each goal

Curricular Connections

Science: Canadian contributions to space exploration

Math: geometry - translation, spatial awareness

References

Chris Hadfield - Canadian Encyclopedia
http://thecanadianencyclopedia.ca/en/article/chris-hadfield/

The Amazing Canadarm2
https://science.nasa.gov/science-news/science-at-nasa/2001/ast18apr_1

Canadarm - Canadian Encyclopedia
http://thecanadianencyclopedia.ca/en/article/canadarm/

Using Scratch, we are going to create a game where we control Canadarm2, just as Chris Hadfield did in the video.

Code Along

- Open up a new Scratch project at scratch.mit.edu and click on "create" (top, left corner).
- Point out the main elements: Stage, Sprites, and Scripts. Demonstrate how to drag and connect blocks.
- Give learners a few minutes to click on blocks and explore.
- Go through 1-2 challenges with the group, where learners are tasked with trying to make something happen in Scratch. For example, "Try to make Scratch move" or "Try to make Scratch say something when the space key is pressed" (See the **Code-Along Challenges doc** (bit.ly/scratch-challenge-solutions-doc) for more examples and solutions)

Activity

Show the **example project** (bit.ly/canadarm2-example) so learners know what they are working towards. Ask them what they see - what is happening in this project?

Open the **starter project** (bit.ly/canadarm2-starter) and review the Sprites and backgrounds.

Have learners open the starter project on their screens and click "REMIX." (The remix button will only be visible once they are signed into their accounts.)

Use the **Solution Sheet** (bit.ly/canadarm2-solution) to guide learners through the following steps:

- Make the arm move
- Reset the arm's position
- Open and close the claw
- Pick up the modules
- Place modules on the port
- Reset the modules' positions
- Copy instructions to remaining modules
- Any Add-Ons, if time

Assessment

Learning Outcomes
I can create algorithms in Scratch
I can use conditionals to control what happens in my project
I can use loops to make things happen more than once
I can use events to control when things happen in my project

I can use broadcast messages to communicate between sprites in my project

Success Criteria
I remixed the starter project and renamed it with my first name.
My Canadarm2 can be controlled by the player.
My Canadarm2 can collect and sort modules.
I used broadcast messages to communicate between elements in my game.

Assessment Ideas
Use Comments in Scratch for learners to explain their steps (right click > add comment).

Students could create a flowchart or sequence of steps using mathematical language before building the game in Scratch (e.g., turn 30º clockwise, travel forward 3 cm, turn 60º counterclockwise, etc.)

Extensions

Have students write a first-person account of an astronaut using the Canadarm2 to successfully capture a Dragon or Cygnus capsule.

Watch this video (bit.ly/canadian-space-agency-canadarm) to learn about the 40 years of robotic innovation that Canada has made, including the Canadarm!

Canadarm2

## STEP 1: Open the Starter Project

1. Sign in to Scratch: scratch.mit.edu
2. Open the starter project: bit.ly/canadarm2-starter
3. Remix the project & change the project name
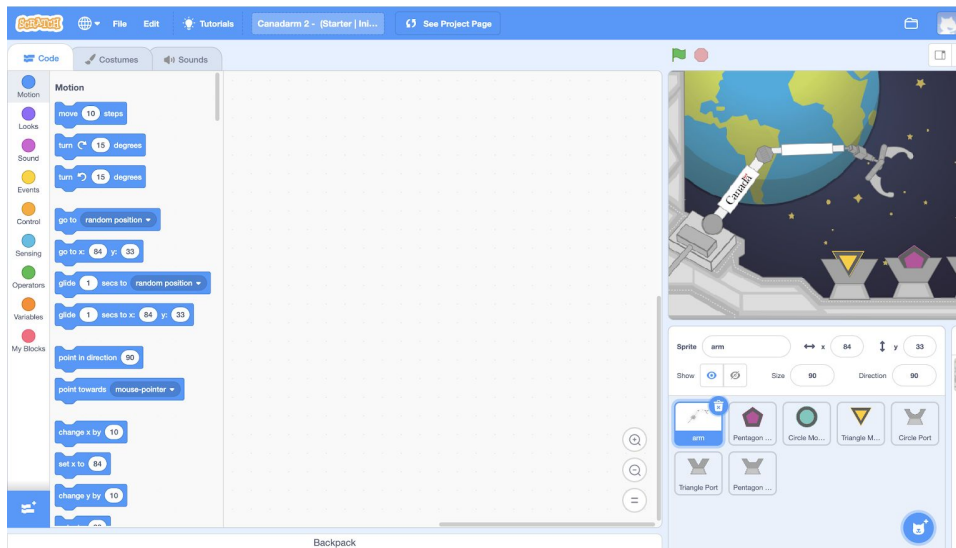

STEP 2: Make the arm move

1. Explain how the X,Y axis works (See lesson: Stretch & Scuttle!)
2. Select the arm sprite
3. Make it move right (x axis)
4. Make this happen when the right arrow key is pressed
5. Make the sprite move left (x axis) when the left arrow key is pressed
6. Make the sprite move up (y axis) when the up arrow key is pressed
7. Make it move down (y axis) when the down arrow key is pressed
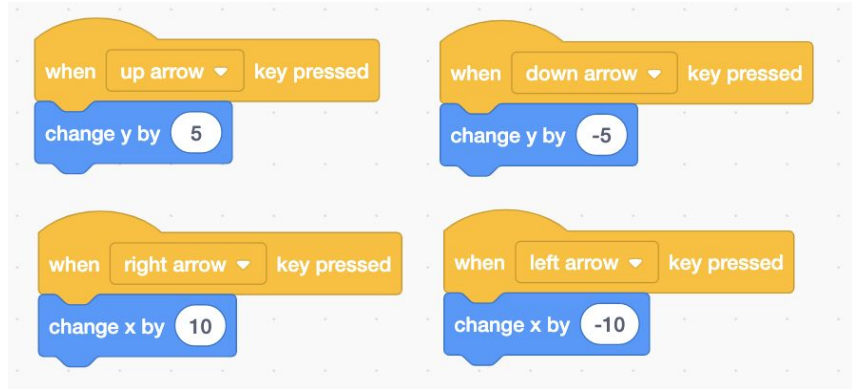
STEP 3: Reset the arm's position

1. Make the arm go back to the bottom, left corner using 'go to xy'
2. Make this happen when the game starts - when the green flag is clicked
3. Make the arm start off closed (switch costume)
4.
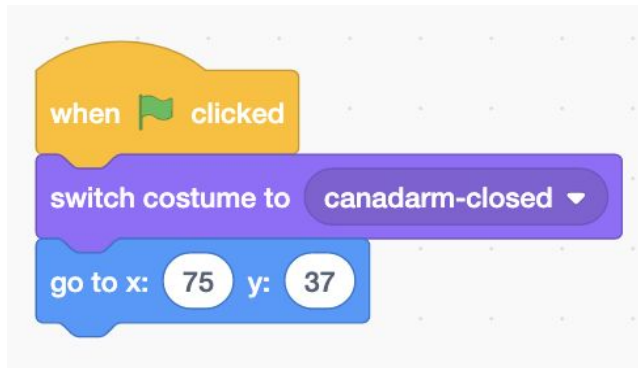
STEP 4: Open and close the claw

1. Make the arm look like it's open (switch costume)
2. Then, make it switch back to the closed costume
3. Make it wait a second in between (so you can see it happen!)
4. Make this happen when the space key is pressed
5. Test it out - does it pick up the shapes? (nope! - we need to communicate with the other sprites to tell them when to follow the arm)
6. Create a new broadcast message called "Open" and broadcast this after the costume switches to open
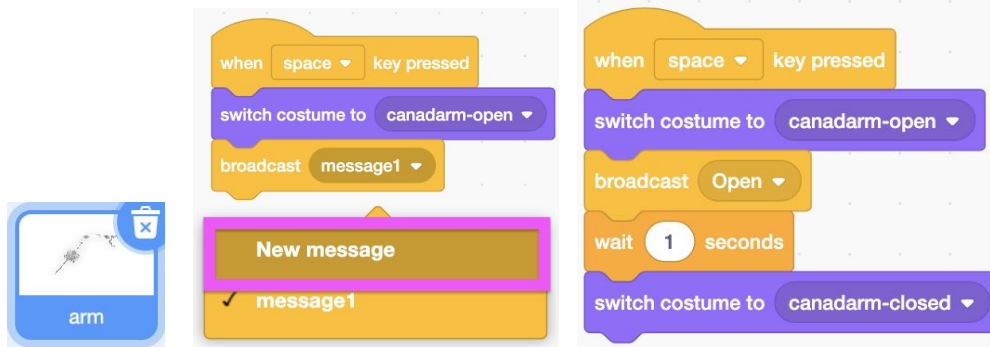
STEP 5: Pick up the modules

1. Select the first (pentagon) shape sprite
2. Make the sprite go the the arm
3. Make it do this when it receives the "Open" message
4. Make it only do this IF it is touching the arm
5. Make it keep following the arm by adding a forever block

STEP 6: Place modules on the port

1. Make the sprite go to the pentagon port
2. Make it do this when it receives the "Open" message
3. Make it only do this IF it is touching the pentagon port
4. Stop the other scripts on the sprite so that the sprite doesn't keep following the arm
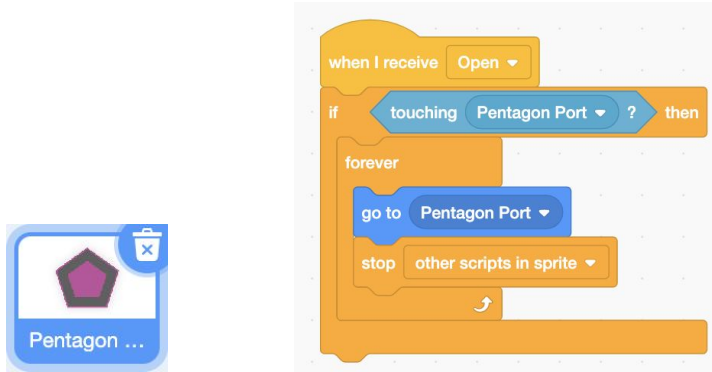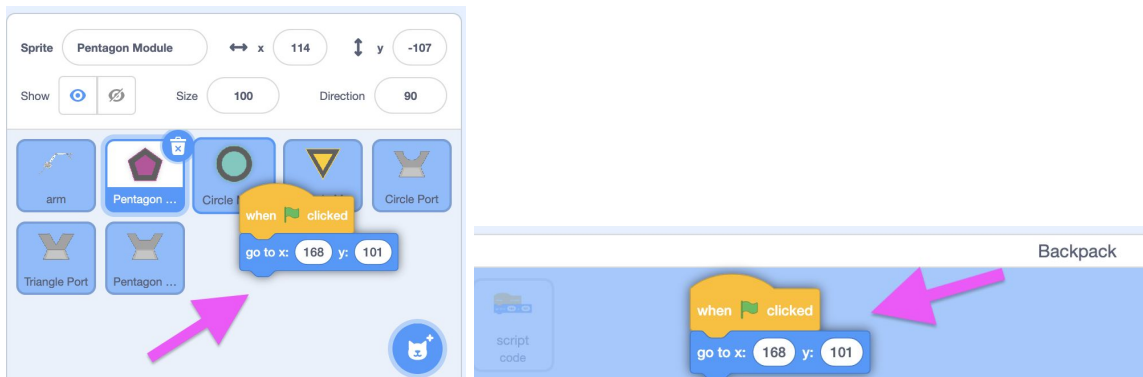
STEP 7: Reset the modules' positions

1. Make the sprite go to a set position
2. Make this happen when the green flag is pressed
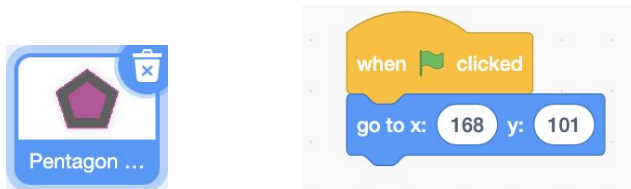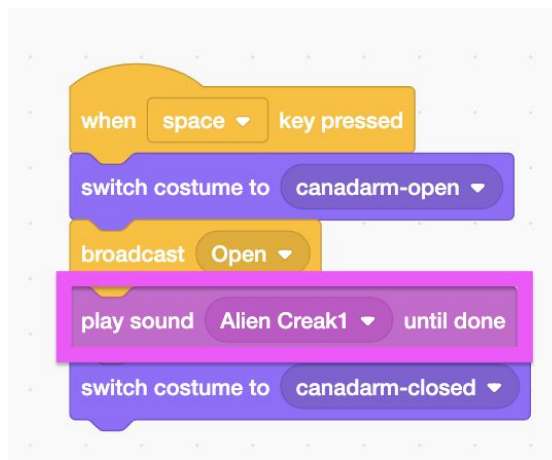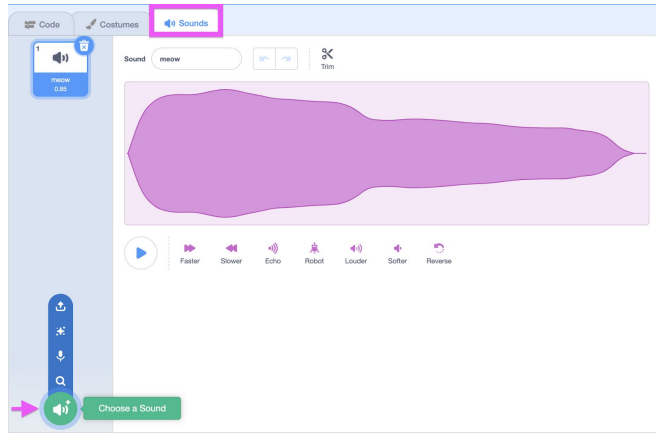

STEP 8: Copy instructions to remaining modules

Add the same code to each of the shapes by clicking and dragging it over to the sprite, or using the backpack.

Add-On: Add sound effects

1. Add a "play sound until done" into your code to add sound effects - E.g. when the claw opens and closes
2. Click the dropdown arrow > "record" to record a sound effect

Add-On: Make the modules float around

1. Play around with "glide" and "go to" blocks to make the game more challenging.
   E.g. Make the module glide to a random position, hide, move to another random
   position, then show again

when 🚩 clicked

go to x: 168 y: 101

repeat until < touching arm ▾ ? > or < touching Pentagon Port ▾ ? >

glide 3 secs to random position ▾

hide

go to random position ▾

show

Pentagon ...