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 and use Scratch to create a simulation in which they become the controller of Canadarm2 to 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: bit.ly/canada-takes-flight-solution **(see below)**

Lesson

This lesson was made in partnership with

CHRIS HADFIELD   let's talk science

Minds On
Watch this video of Chris Hadfield: https://www.youtube.com/watch?v=BR8svbP1ilw
Ask: What do we know about the International Space Station?

Key Coding Concepts

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

Curricular Connections

Science: Canadian contributions to space exploration

Math: geometry - translation, spatial awareness

References

Learn more about Chris Hadfield in this interview:

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 in order to place modules where it needs 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.

http://thecanadianencyclopedia.ca/en/article/chris-hadfield/

Science At NASA
https://science.nasa.gov/science-news/science-at-nasa/2001/ast18apr_1
Canadarm - Canadian Encyclopedia
http://thecanadianencyclopedia.ca/en/article/canadarm/

Watch the Hadfield behind the controls of Canadarm2 video (or short clip)

- 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?

Using Scratch, we are going to create a game where we control Canadarm2, like in the video of Chris Hadfield.

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 for more examples and solutions)

Activity
Show the example project 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."

Use the Solution Sheet 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

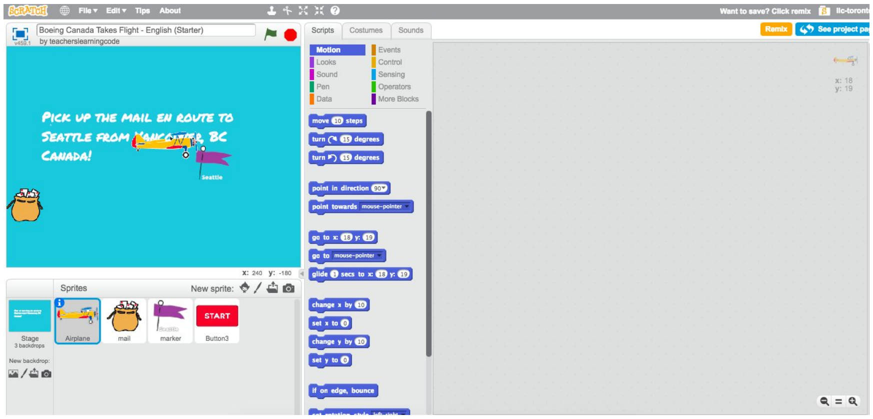
Have students write a first-person account of an astronaut using the Canadarm2 to successfully capture a Dragon or Cygnus capsule.

Watch this video to learn about the 40 years of robotic innovation that Canada has made, including importantly the Canadarm!

Canada Takes Flight

STEP 1: Open the Starter Project

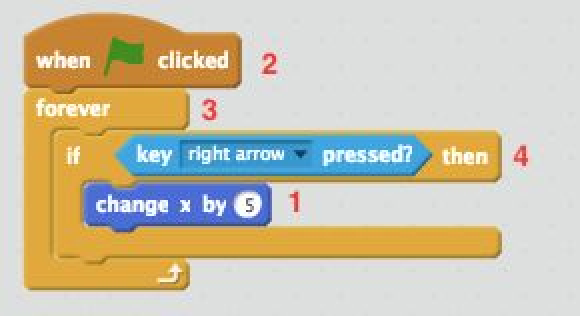
1. Sign in to Scratch
2. Open the starter project: **http://bit.ly/canada-takes-flight-starter**
3. Remix the project & change the project name



*Starter project has 3 backdrops, 4 sprites, and starter code for the bouncing ball

STEP 2: Make the Plane Move

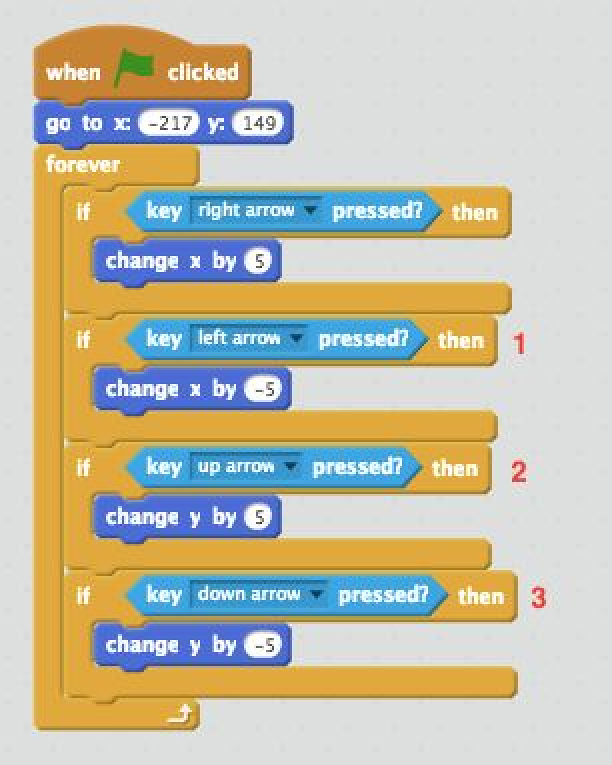
1. **Explain how the X,Y axis works (Stretch & Scuttle!)**
   http://teacherslearningcode.com/en/lessons/stretch-and-scuttle
2. Make the plane move right (along x axis)
3. Make this happen when the green flag is clicked
4. Make this happen forever
5. Make this happen if the right arrow key is pressed



```
when [flag] clicked  2
forever  3
    if <key (right arrow) pressed?> then  4
        change x by (5)  1
```

STEP 3: Adding Other Directions

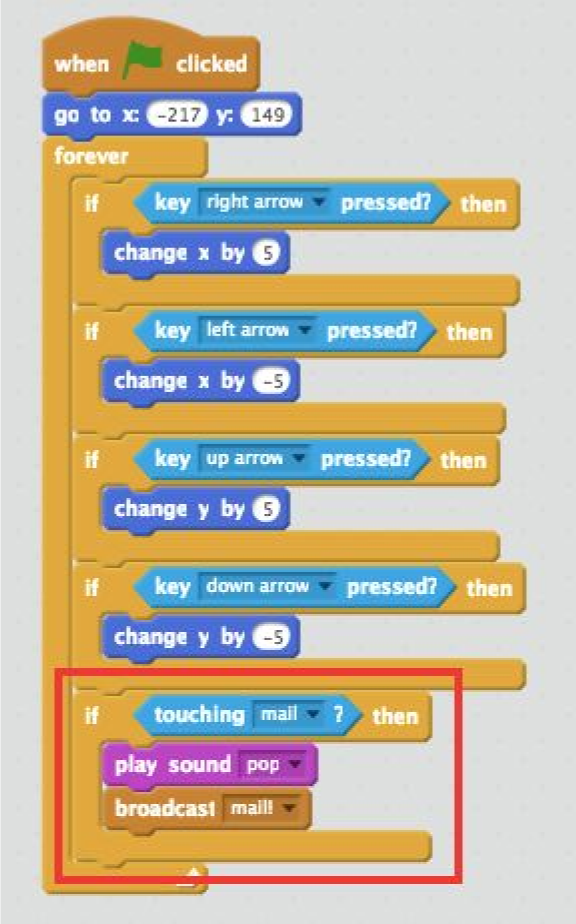
1. Duplicate the IF statement - make the plane move left if the left arrow key is pressed
2. Duplicate again - make the plan move up (along the y axis) if the up arrow key is pressed
3. Duplicate one last time - make the plan move down if the down arrow key is pressed
4. Make the plane go back to the top by adding 'go to x,y' (move the plane to the starting point, then drag the block over)



```
when ⚑ clicked
go to x: -217 y: 149
forever
    if key right arrow pressed? then
        change x by 5
    if key left arrow pressed? then
        change x by -5
    if key up arrow pressed? then
        change y by 5
    if key down arrow pressed? then
        change y by -5
```

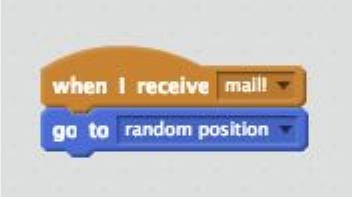
STEP 4: Collecting Mail

1. Create a new broadcast message called "mail" to communicate with the mail sprite
2. Make the plane broadcast this message if touching the mail
3. Put this in the forever loop so the program always checks if they are touching
4. Play a sound before broadcasting



```
when [flag] clicked
go to x: -217 y: 149
forever
    if key right arrow pressed? then
        change x by 5
    if key left arrow pressed? then
        change x by -5
    if key up arrow pressed? then
        change y by 5
    if key down arrow pressed? then
        change y by -5
    if touching mail ? then
        play sound pop
        broadcast mail!
```

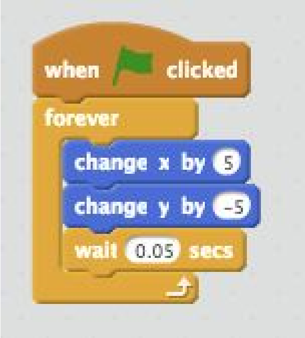
1. Select the mail sprite
2. Make it go to a random position
3. Make this happen when it receives "mail"



```
when I receive mail!
go to random position
```

STEP 5: Wind/Gravity Effect

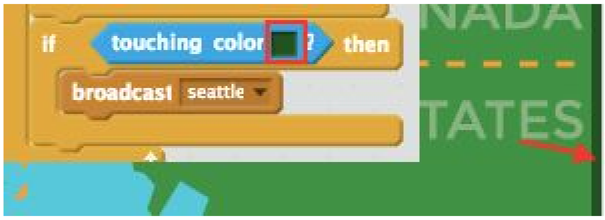
1. Select the plane sprite
2. Make the plane look like it's falling - make it move down (y axis)
3. Make the plane look like it's flying forwards - make it move to the right (x axis)
4. Make this happen forever, when the green flag is clicked
5. Make it wait for a short amount of time before looping (to add a choppy effect)



```
when [flag] clicked
forever
    change x by 5
    change y by -5
    wait 0.05 secs
```

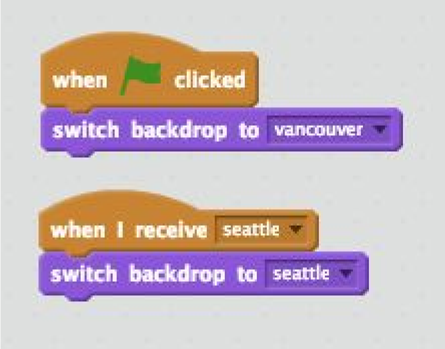
STEP 6: Flying to Seattle

1. Make a new broadcast message called "seattle" to communicate with the other sprites, and tell them when to switch to the next screen
2. Make the plane broadcast this message when it is touching the dark green colour on the right edge
3. Be sure to click on the colour square and then select the dark green by clicking on that area of the backdrop in the stage (the mouse pointed will look like a glove until you select a colour)



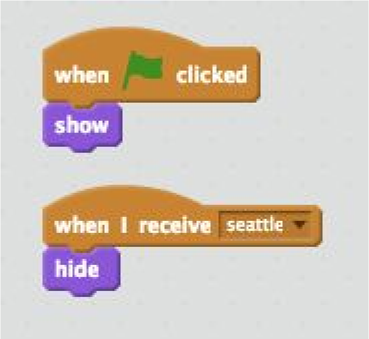
```
if touching color [ ] ? then
    broadcast seattle
```

1. Select the Stage
2. Make the backdrop switch to seattle when it receives the "seattle" message
3. Make the backdrop switch back to vancouver when the green flag is clicked



```
when [flag] clicked
switch backdrop to vancouver

when I receive seattle
switch backdrop to seattle
```

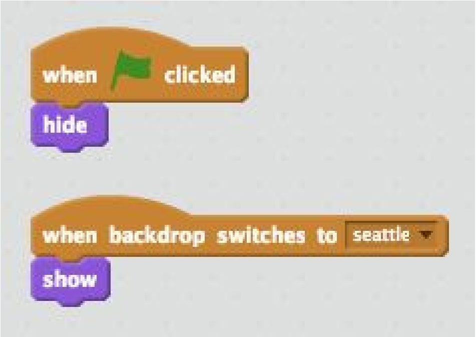
1. Make the mail sprite hide when seattle is broadcast
2. Make the mail show again when the green flag is clicked



```
when [flag] clicked
show

when I receive [seattle ▾]
hide
```

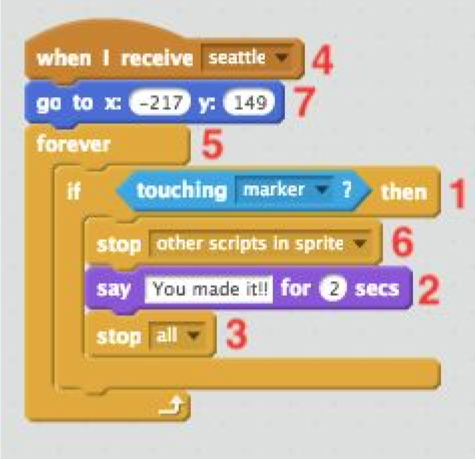
STEP 7: Win Condition

1. Select the marker sprite
2. Make the marker show when it receives the “seattle” message
3. Make the marker hide again when the green flag is clicked



```
when [flag] clicked
hide

when backdrop switches to [seattle ▾]
show
```

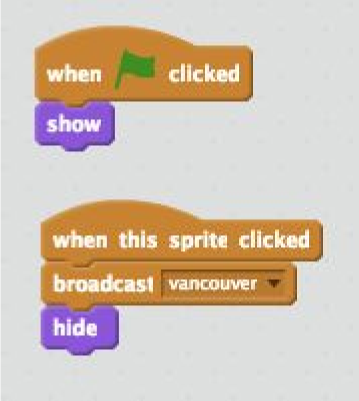
1. Select the airplane
2. If the plane is touching the marker, make it do something
   E.g. say “You made it!”
3. Then, make it stop all of the scripts (end the game)
4. Make this happen when the plane receives “seattle”
5. Add a forever loop, so the program always checks for this
   (Test it out!)
6. To make the plane stop doing everything else (besides this script) add a “stop other scripts in sprite” before the ‘say’ block
7. Make the plane reset to the top using “go to x,y”



```
when I receive [seattle ▾]
go to x: (-217) y: (149)
forever
    if <touching [marker ▾] ?> then
        stop [other scripts in sprite ▾]
        say [You made it!!] for (2) secs
        stop [all ▾]
```

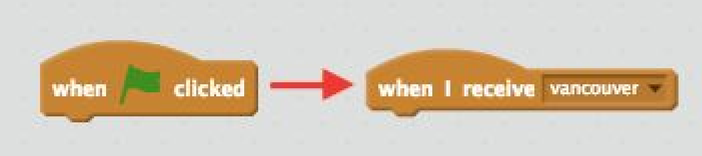
ADD-ON: Opening Screen

1. Select the Start button
2. Make the button show when the green flag is clicked
3. Make the button hide when it is clicked
4. Make it broadcast a new message called ‘vancouver’ before hiding

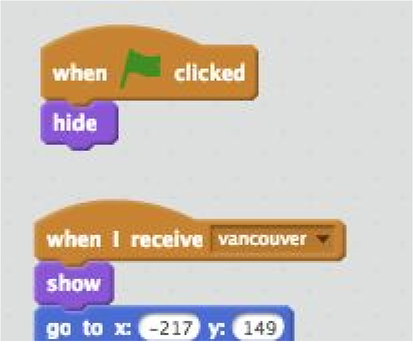


```
when ⚑ clicked
show

when this sprite clicked
broadcast [vancouver ▾]
hide
```

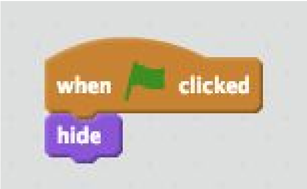
1. Make the game wait until the button is clicked to start (instead of the green flag): Replace all of the existing “when green flag clicked” scripts with “when I receive vancouver” scripts for the airplane and mail
2. Make the airplane hide when the green flag is clicked, then show when it receives the ‘vancouver’ message
3. Make the mail hide when the green flag is clicked



```
when ⚑ clicked → when I receive [vancouver ▾]
```

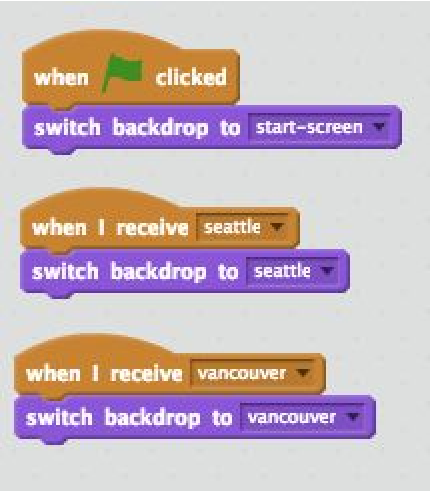


```
when [flag] clicked
hide

when I receive vancouver
show
go to x: -217 y: 149
```



```
when [flag] clicked
hide
```

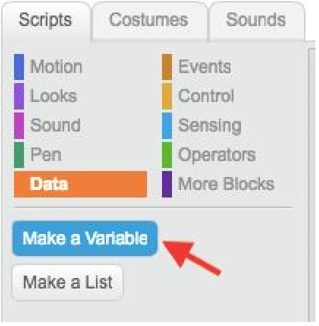
1. Select the Stage
2. Make the game begin with the ‘start-screen’ backdrop
3. Make it switch to the Vancouver backdrop when it receives ‘vancouver’

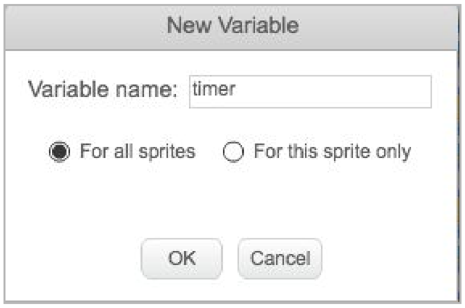


```
when [flag] clicked
switch backdrop to start-screen

when I receive seattle
switch backdrop to seattle

when I receive vancouver
switch backdrop to vancouver
```

ADD-ON: Timer

1. Make a new variable called ‘timer’
2. Make the timer increase by 1
3. Make this happen forever, once ‘vancouver’ has been received
4. Make the timer wait 1 second before increasing
5. Make the timer reset to 0 when vancouver is received (aka button is clicked)
6. Make the airplane say the time when it reaches the marker

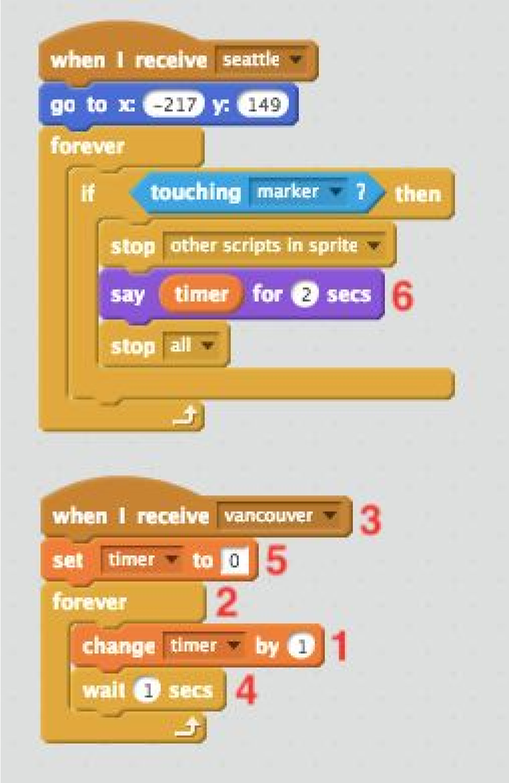




New Variable

Variable name: timer

● For all sprites ○ For this sprite only



```
when I receive seattle
go to x: -217 y: 149
forever
    if touching marker ? then
        stop other scripts in sprite
        say timer for 2 secs 6
        stop all

when I receive vancouver 3
set timer to 0 5
forever 2
    change timer by 1 1
    wait 1 secs 4
```

Additional Extension Ideas

- Edit the plane sprite costume or replace it with another sprite to change the story
- Use variables to keep track of the score (e.g. amount of mail collected)
- Use maps to draw a map of your neighbourhood and the looks block 'broadcast' to create new levels!
- Have students create a success screen that describes this historic flight.

See extended example for possible solutions: https://scratch.mit.edu/projects/160217740/