teachers
learning
code

DIGITAL TOOLBOX

Getting Started with Scratch

An educational program by
CANADA LEARNING CODE

Table of Contents

AN INTRODUCTION

LESSON PLAN: WILDLIFE SOUNDSCAPES

LESSON PLAN: CANADA TAKES FLIGHT

EXTRAS

For more great lesson plans, check out our website:
canadalearningcode.ca/lesson-plans

An Introduction to Scratch

WHAT IS SCRATCH?

Scratch is a graphical programming language created by the Lifelong Kindergarten group at the MIT Media Lab that can be used to create animations, games, and interactive artwork.

What sets graphical languages apart is they allow users to drag and drop blocks of instructions to write programs. Scratch eliminates the need to write any fancy code (or syntax) and get bogged down in the nuances of the syntax like colons, commas, and parentheses which can be a distraction to learning key concepts. It's because of this that Scratch is a great tool for beginners and youth setting out to learn code.

In selecting Scratch, our key considerations include:

- The strong and growing Scratch community is comprised of educators from across the world. This means that you never have to feel alone. There is an abundance of free support resources for learners and educators.
- You can use the web-based version with any browser — no need to download and install software.
- There is also an offline version if internet connectivity is an issue or not available. Simply download the Offline Editor onto computers and use it locally.
- It is free to use. This means that learners are able to continue their learning journeys even outside of the classroom without financial cost being a barrier.
- It can be used to code the simplest interaction to the most complex.
- The programming concepts one can learn from Scratch apply to many other programming languages.
- It permits import of audio and image files while also providing the tools to create audio and images within its own environment, thus encouraging learners to create their own original content. Learners can also use the many objects (sprites) found in the libraries within Scratch.

Ultimately, Scratch is designed with learning and education in mind. It is a great collaborative program for making art, playing music, and creating games. It is also a place to experiment with mathematical functions, geometry, graphing, simulations, physics, and more!

Getting Started

Creating Scratch Teacher Accounts

Teacher Accounts are moderator accounts that make it easier for educators (traditional or otherwise) to manage a group of learner accounts, projects, and/or comments within Scratch.

For more information about creating a Teacher Account, please visit

http://bit.ly/teacher-accounts

Classes and Studios

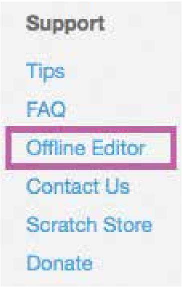
Within your Teacher Account, you can create multiple classes. Classes provide a way for educators to share projects with a specific group of learners. Within each class, there are typically multiple Scratch studios that represent each of the unique projects your learners will be creating or remixing.

We recommend creating one Scratch class per group you will be working with. For example, if you were teaching both an after-school coding club and a history class, we would recommend having a separate Scratch class for each of these groups.

For a refresher on creating Scratch Classes or Studios, please visit:
http://bit.ly/teacher-accounts

Installing the Scratch Offline Editor

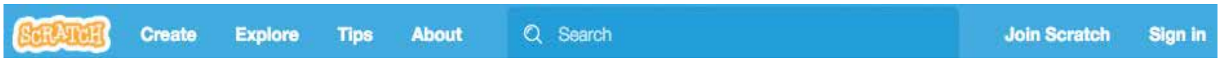
In the event that your WiFi connection goes down, we recommend installing the Scratch Offline Editor. The install files for this program can be found by going to
https://scratch.mit.edu
scrolling to the very bottom of the page, and clicking 'Offline Editor' under 'Support'. You will need to download both Adobe Air and the Offline Editor. We recommend storing these files on several USBs that can be accessed during your coding sessions, in case of WiFi failure.



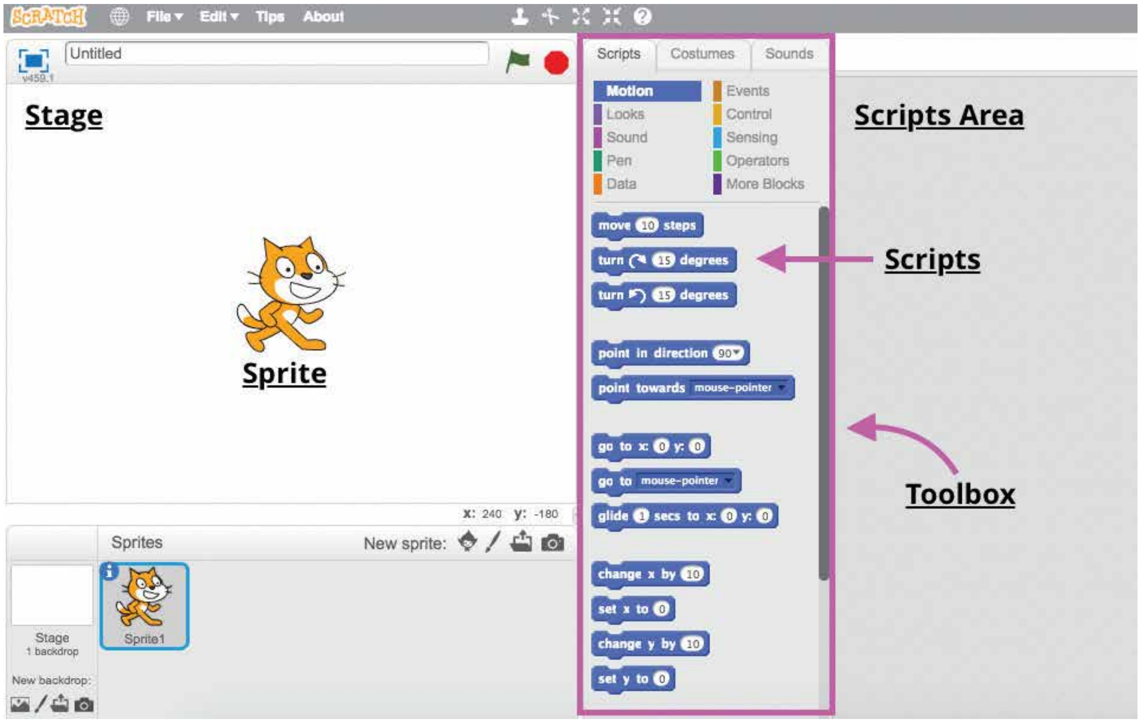
Creating A New Project

Go to **Scratch** and sign in: **https://scratch.mit.edu**

Once you've signed in, you can create a new project by clicking 'Create' in the top tool-bar.



Key Components of the Scratch Editor



Sprites: Each object in Scratch is called a Sprite. You can add sprites by choosing from the library, painting your own, uploading an image or taking a picture from a webcam.

Stage: This is where you can preview what your code does. You can add backdrops to your stage. You can 'Start' and 'Stop' the preview at any time using the green flag or red stop symbol at the top of the page.

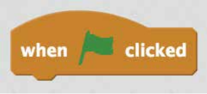
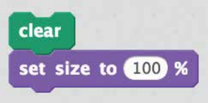
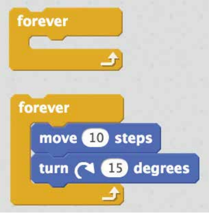
Toolbox: The toolbox contains all scripts, costumes, and sounds.

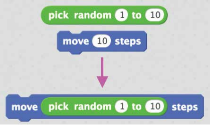
Scripts: Scripts are the commands you will use to code. This is how the magic happens. Students will use scripts to instruct their Sprite to do what they want it to do. From moving forward, to saying 'hello', scripts are arranged in a logical sequence to 'program' the sprite.

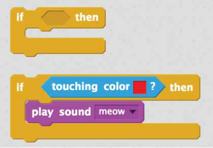
Scripts Area: All of the coding happens here. You can drag and drop scripts from the toolbox into this area. Scripts can be deleted by dragging them back into the toolbox.

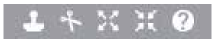
USING

Scripts & Shortcuts

SCRIPTS	DESCRIPTION
 when clicked	**Using Event Blocks to Initiate Code** A sequence of code should always be initiated by an event block. It is critical that learners notice the 'hat'-shaped (ie. the round surface) nature of these blocks. Other blocks can be connected to one another from the top or bottom, but event blocks, like the one to the left, are required to be at the very top — just like a hat sits on top of a person's head! This type of block allows a set of actions to be taken after some event occurs. In this case, any code that comes afterward will be executed after the green flag is clicked.
 clear set size to 100 %	**Connecting Blocks** Blocks can be dragged from their home in the toolbox into your script area where you can connect them to each other. You can also change the value in many blocks by clicking the number and typing a new one. Blocks will only snap together when it makes logical or pragmatic sense to do so.
 forever forever move 10 steps turn 15 degrees	**Using Loops to Make Things Happen More than Once** The forever block is seemingly meaningless on its own, but other blocks can be placed inside of it. We can better organize our code by using loops instead of adding multiples of the same scripts. Make something happen more than once by placing instructions inside of a loop — to make it repeat a set amount of time, or forever!

SCRIPTS	DESCRIPTION
 pick random 1 to 10 move 10 steps move pick random 1 to 10 steps	**Placing Blocks within Blocks** More sophisticated commands can be created by placing blocks within other blocks. For example, an operator block (green) might be placed inside a motion block (dark blue) to make your sprite move a random number of steps each time.


if then
if touching color ? then
play sound meow
	Making Decisions with 'If'
	As students complete more complicated challenges and build more sophisticated projects, conditional blocks like 'if <> then' will become really important. This block tells the sprite to do something if a certain condition is met. If the condition is not met, the sprite with do something else.

	**Using Shortcuts to Simplify Your Coding Experience** Scratch has a few handy shortcuts. To use these, simply click on the shortcut and then click on the sprite, backdrop, or script you would like that action to apply to. Use the stamp to duplicate a sprite or sequence of code. This is really helpful when you want to recreate the same sequence, but don't want to drag-and-drop everything again. Use the scissors to delete a script or sprite. Use the outward-facing arrows to grow. Use the inward-facing arrows to shrink. Use the question mark if you are unsure of what a certain script does. Once you've clicked on the question mark, clicking on any script will create a pop-up of helpful tips specific to that bit of code.

Debugging

Learning to debug code is an important part of every learner's coding journey.

To help you get started, here are some common issues we've come across in Scratch.

How do I delete my blocks?

To delete, drag the code blocks out of your scripts area back into the toolbox. If you want to delete a sprite or backdrop, you can right-click on the object and select 'delete'.

Help! I lost all of my code!

This is common when you start working with multiple sprites. Make sure you've clicked on the sprite in the 'Sprites' area that you're working on. Your code could be hidden in another sprite. Other than this — always remember to save, save, save! Saving periodically while you are coding helps prevent any lost work.

My code won't work!

Code is complex and problems are complex. An important part of computational thinking and coding is being able to break problems down into smaller, more manageable steps, similar to the way we might break down a book report into different sections.

After we break problems into smaller parts, abstraction helps us decide what's important and what's not. It helps manage complexity, like a math problem where we need to decide what information is needed for our equation. Try to identify what isn't working in your project and hone in on the code that affects that particular action.

The best way to prevent errors and silly mistakes is to have neat, organized code, and test your code many times throughout. It may seem extreme, but when you start out, you may want to test your code each time you add a new block or two. Spotting errors early can save you lots of time and trouble later!

Help! My sprite walked off the screen and now I can't get it back!

There are two solutions to try when you encounter this issue:

First, click on the sprite in question and double-click on the 'Show' block under the purple 'Looks' category. Sometimes students accidentally click on 'Hide' which causes their Sprite to become invisible.

If the sprite isn't hidden, it is possible it has traveled outside the stage bounds. This might happen if a student has directed a sprite to move an inordinate amount of steps. Go to the 'Motion' category and double-click on the 'Go to x: 0 y: 0' block. When the X and Y coordinates are both set to 0, the sprite should appear at the centre of the stage.

How can I practice debugging?

Try out some debugging challenges at **http://bit.ly/ScratchDebug**. Click on a project, read the 'Notes and Credits' section to identify desired outcome and current bug, and then click on 'See Inside' to view and edit the code.

LESSON PLAN:

Wildlife Soundscapes

DESCRIPTION

In this lesson, students will explore the sounds of wildlife in protected environments and begin to consider how we can restore and protect animal habitats.

Grade: K-2, 3-5

Length: 60 minutes

Subjects: Science and Technology, Language

Coding Concepts: Algorithms, Events, Sequences

Additional Concepts: Understanding Life Systems, Needs and Characteristics of Living Things, Habitats and Communities, Sustainability and Stewardship

PREP WORK

1. Review the our Teachers Learning Code Digital Toolbox for tips and tricks!
2. Review the example project at **http://bit.ly/soundscape-example**
3. Watch this video about wolves: **http://bit.ly/how-wolves-change-rivers-video**
4. Print the Solution Sheet for the main activity. Available at **http://bit.ly/wildlife-soundscapes-solution** or 'Printables' section in this guide. You may want to print extra copies for learners to share.

The Lesson

**This lesson was made in partnership with the David Suzuki Foundation.*

INTRODUCTION

Watch this video about wolves: **http://bit.ly/how-wolves-change-rivers-video**

✓ **Ask: What does this video tell us about [wildlife/ecosystems]?** (A: That they're all connected!)

Provide some background information on the boreal woodland caribou:

The boreal woodland caribou is threatened with extinction in every province and territory. (Ask: What does 'extinction' mean?) These shy animals need large forests without roads and buildings to survive. At least half of the boreal caribou have been lost as their homes have been destroyed. For example — all of the woodland caribou in oil and grass development areas of Alberta are shrinking in half every 8 years.

We're making a "wildlife soundscape" for the woodland caribou using Scratch. A soundscape is when we take all of the sounds we might hear in a place, and put them together.

✓ **Ask: What is wildlife? What sorts of sounds might we hear in a wildlife soundscape?** (Take note of the answers on a whiteboard or anchor chart paper)

Create a wildlife soundscape with your class!

- Tell learners that we are making a soundscape. Each of them will take turns adding a sound that they would hear in a forest (when you point at them, or when they pass a 'talking stick' item). Remind them that forests are quiet and peaceful, so let's try to make our soundscape sound the same!
- If learners can't think of a sound effect, they can use a sound from the list that you created (from the previous question).
- Go through the entire class until everyone has had a chance to add their sound. Congratulate learners on creating their first soundscape. Now, let's make one using the computer!

CODE ALONG

- Open up a new Scratch project at **http://scratch.mit.edu** and click on "Create" (top, left corner).
- Point out the main elements: Stage, Sprites, and Scripts (especially the "Sounds" category). 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: **http://bit.ly/scratch-challenge-solutions-doc**).

ACTIVITY

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

Open the starter project (**http://bit.ly/soundscape-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:

- Adding animal sounds
- Changing the backdrop
- Showing/hiding animals

Give learners time to work on their projects, and add as many sounds and changes as they like.

Extensions

- Complete a KW(D)L chart with the class (Know, Want to know, (Do), Learned). Ask them what they think they know about habitats (or caribou) at the beginning of the lesson, and what they want to know. After the lesson, regroup and complete the 'learned' column, as well. Brainstorm and research what you can DO as a class. How can you work towards helping save the boreal woodland caribou and their habitat (or endangered species, in general)?
- Use this as an opportunity to learn about onomatopoeia (when words look the way they sound — like 'buzz'). When brainstorming sounds in a wildlife soundscape, have learners think of these words and write them at the front.
- Define "nocturnal" and spend time discussing which animals come out at night, and which are out during the day. Have learners demonstrate their understanding by having animals in the project 'show' and 'hide' according to their backdrop.

Assessment

Learning Outcomes

I can use code to tell the computer what to do

I can use events to control when things happen in my project

I can create a soundscape in Scratch

Success Criteria

I remixed the starter project and renamed it with my first name

All of the animals in my project make a sound

My backdrop changes from day to night (or the other way around)

I made at least 1 animal show, then hide (or the other way around)

Assessment Ideas

- Have students write a reflection on the effects of human activity on animal life.
  - What did they learn from creating a rich soundscape in Scratch?
  - What would their forest sound like if there were fewer or even no animals?
- Show learners how to add a "Say [hello] for ___ secs" script to provide some content for their soundscape. They can edit the 'hello' text to describe what they learned about the caribou and their habitat.

LESSON PLAN:

Canada Takes Flight

DESCRIPTION

In this lesson learners will create a game that tells the story of Bill Boeing and Eddie Hubbard's first international flight to deliver mail from Vancouver to Seattle. Learners will explore several Scratch techniques as they reflect on the history of flight technologies in Canada.

Grade: 3-5, 6-8

Length: 90 minutes

Subjects: Science and Technology, Social Studies

Coding Concepts: Conditional Statements, Random Numbers, Variables, Loops, Algorithms, Events

Additional Concepts: Flight, Forces Causing Movement, Transportation Technology, People and Environments

PREP WORK

- Review our Teachers Learning Code Digital Toolbox for tips and tricks!
- Review the example project at **http://bit.ly/canada-takes-flight-example**
- Watch this video about Bill Boeing and how a little Seattle company began to change the aeroplane industry through their innovations: **http://bit.ly/canada-takes-flight-boeing**
- Review the four forces of flight: lift, weight, drag, and thrust.
- Print the Solution Sheet for the main activity. Available at **http://bit.ly/canada-takes-flight-solution** or 'Printables' section in this guide. You may want to print extra copies for learners to share.

The Lesson

INTRODUCTION

From canvas and wood to materials for space, Boeing has continued to invent and engineer ways to extend human abilities. Programming, engineering, and mechanics are all ways that we can extend our abilities and make the impossible possible.

The first international airmail was delivered by Boeing in a flight from Vancouver to Seattle in May 1919. Throughout the 1920s Boeing played a significant role in expanding the airline industry through developing multiple airplane models including fighter planes, transports, and more.

After a hundred years of flight, Boeing continues to innovate the aerospace industry. Boeing is now looking towards the sky to take people not only all across planet Earth, but maybe to new planets as well!

Discover more at Boeing's History Page:
> http://www.boeing.ca/boeing-in-canada/history.page

CODE ALONG

- Open up a new Scratch project at **http://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: **http://bit.ly/scratch-challenge-solutions-doc**).

ACTIVITY

In this game, you'll be on board with William Boeing and Eddie Hubbard on the flight to deliver the first international airmail.

Show the example project (**http://bit.ly/canada-takes-flight-example**) so learners know what they are working towards. Ask them what they see/hear — what is happening in this project?

Open the starter project (**http://bit.ly/canada-takes-flight-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:

- Making the Plane Move
- Adding Other Directions
- Collecting Mail
- Wind/Gravity Effect
- Flying to Seattle
- Win Condition

Now you're ready for takeoff! Give the game a whirl.

Extensions

- Can learners find a way to keep score within their game?
- Use the backdrops tab to draw a map of your neighbourhood and set it as the background for a new level.
- Have learners create a success screen that describes this historic flight.
- Identify how the four forces of flight are incorporated within our code (weight, lift, thrust, drag).

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

Success Criteria

I remixed the starter project and renamed it with my first name

My sprite moves in all four directions (up, down, right, left)

I can control my game using keys on my keyboard

I used broadcast messages to communicate between elements in my game

My game has a Win condition

Assessment Ideas:

- Use comments in Scratch for learners to explain their steps (right click > add comment).
- Have learners use an attached 'flight log' sheet to document their findings.

Resources

SCRATCH

Debugging Scratch: https://en.scratch-wiki.info/wiki/Debugging_Scripts

Debugging Scratch Practice: http://bit.ly/ScratchDebug

Scratch: https://scratch.mit.edu

Scratch Cards: https://scratch.mit.edu/info/cards

Scratch Code Along Challenges: http://bit.ly/scratch-challenge-solutions-doc

ScratchEd: http://scratched.gse.harvard.edu

Scratch FAQ: https://scratch.mit.edu/info/faq

ScratchJr: http://scratchjr.org

Scratch Offline Editor: https://scratch.mit.edu/scratch2download

Scratch Teacher Accounts: http://bit.ly/teacher-accounts

Scratch Wiki: http://wiki.scratch.mit.edu

WILDLIFE SOUNDSCAPES LESSON

How Wolves Change Rivers Video: http://bit.ly/how-wolves-change-rivers-video

Wildlife Soundscapes Example Project: http://bit.ly/soundscape-example

Wildlife Soundscapes Solution Sheet: http://bit.ly/wildlife-soundscapes-solution

Wildlife Soundscapes Starter Project: http://bit.ly/soundscape-starter

CANADA TAKES FLIGHT LESSON

Boeing History Page: http://www.boeing.ca/boeing-in-canada/history.page

Boeing Video: http://bit.ly/canada-takes-flight-boeing

Canada Takes Flight Example Project: http://bit.ly/canada-takes-flight-example

Canada Takes Flight Solution Sheet: http://bit.ly/canada-takes-flight-solution

Canada Takes Flight Starter Project: http://bit.ly/canada-takes-flight-starter

ADDITIONAL RESOURCES

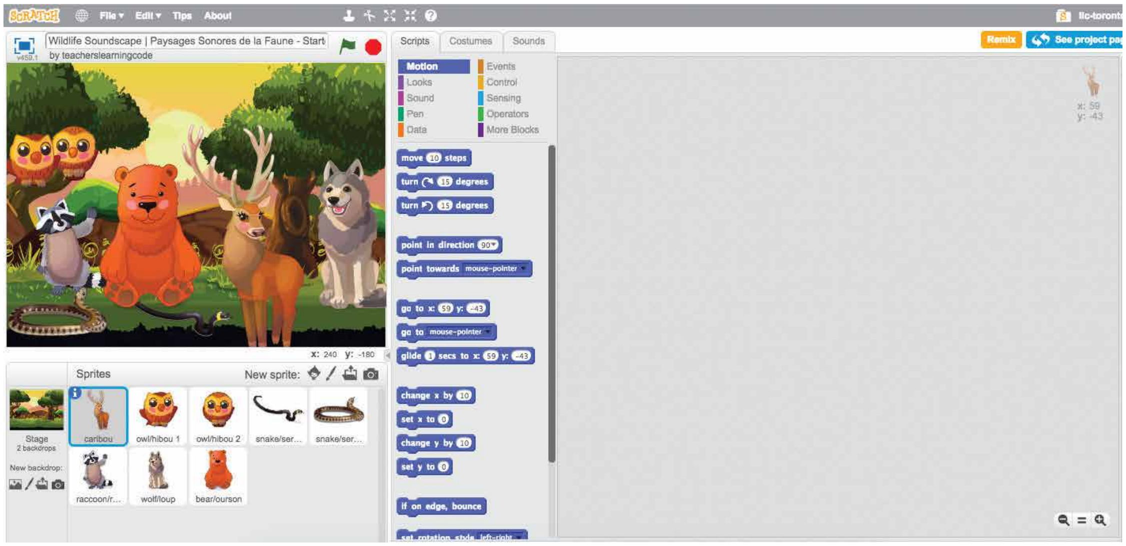
Teachers Learning Code Website (for additional lesson plans): http://teacherslearningcode.com

ANIMATING WITH SCRATCH SOLUTION SHEET

Wildlife Soundscapes

STEP 1: OPENING THE STARTER PROJECT

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

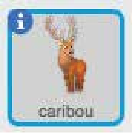


*Starter project has 2 backdrops and 8 sprites with sounds.

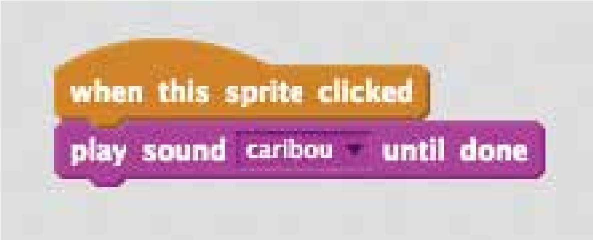
Solution Sheet created for Canada Learning Code. More info on Scratch (by MIT Media Lab): scratch.mit.edu

STEP 2: ADDING ANIMAL SOUNDS

1. Select the caribou sprite

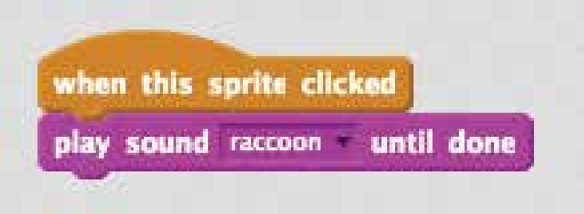


2. Make the caribou say something (play sound until done will make sure that the sound doesn't get cut off)



```
when this sprite clicked
play sound caribou ▾ until done
```

3. Make this happen when the sprite is clicked
4. Show learners how to select another animal sprite and add the same instructions — to make a sound when the sprite is clicked
5. Give them 5 extra minutes to add sounds to the other animals in the project. For example, the raccoon:



```
when this sprite clicked
play sound raccoon ▾ until done
```

Solution Sheet created for Canada Learning Code. More info on Scratch (by MIT Media Lab): scratch.mit.edu

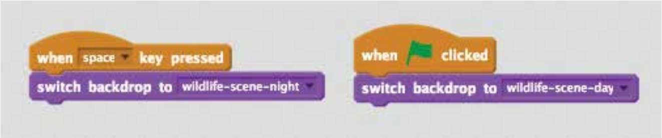
STEPS 3&4: ANIMATING WITH SCRATCH SOLUTION SHEET

Wildlife Soundscapes

STEP 3: CHANGING THE BACKDROP

1. Select the stage

2. Use the 'switch backdrop' script to change the backdrop to the night scene
3. Make this happen when we press the space key
4. Make the backdrop reset to the day scene when the green flag is pressed



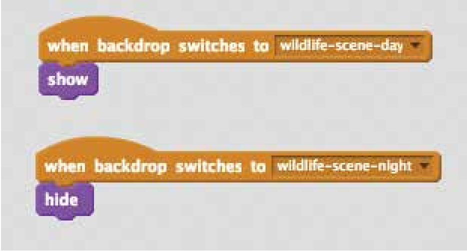
```
when space key pressed
switch backdrop to wildlife-scene-night
```

```
when ⚑ clicked
switch backdrop to wildlife-scene-day
```

STEP 4: SHOWING/HIDING ANIMALS

1. Select the Caribou sprite

2. Make the sprite hide
3. Make this happen when the backdrop changes to the night scene
4. Make the caribou show when the backdrop changes to the day scene



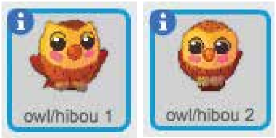
```
when backdrop switches to wildlife-scene-day
show
```

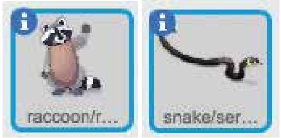
```
when backdrop switches to wildlife-scene-night
hide
```

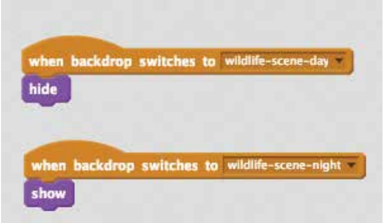
Solution Sheet created for Canada Learning Code. More info on Scratch (by MIT Media Lab): scratch.mit.edu

ADD-ON: NOCTURNAL OR DIURNAL?

1. Discuss which animals would come out at night (nocturnal) or during the day (diurnal)
2. For each animal that comes out at night, make them:
   a. Hide when it switches to the day scene backdrop
   b. Show when it switches to the night scene backdrop
3. For each animal that comes out during the day, make them:
   a. Show when it switches to the day scene backdrop
   b. Hide when it switches to the night scene backdrop







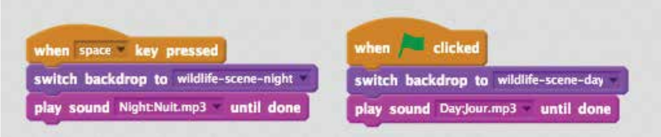
```
when backdrop switches to wildlife-scene-day
hide

when backdrop switches to wildlife-scene-night
show
```

ADD-ON: BACKGROUND SOUNDS

1. Select the stage

2. Make the night time sounds play after we switch to the night scene
3. Make the day time sounds play after we switch to the day scene



```
when space key pressed
switch backdrop to wildlife-scene-night
play sound Night:Nuit.mp3 until done
```

```
when [flag] clicked
switch backdrop to wildlife-scene-day
play sound Day:Jour.mp3 until done
```

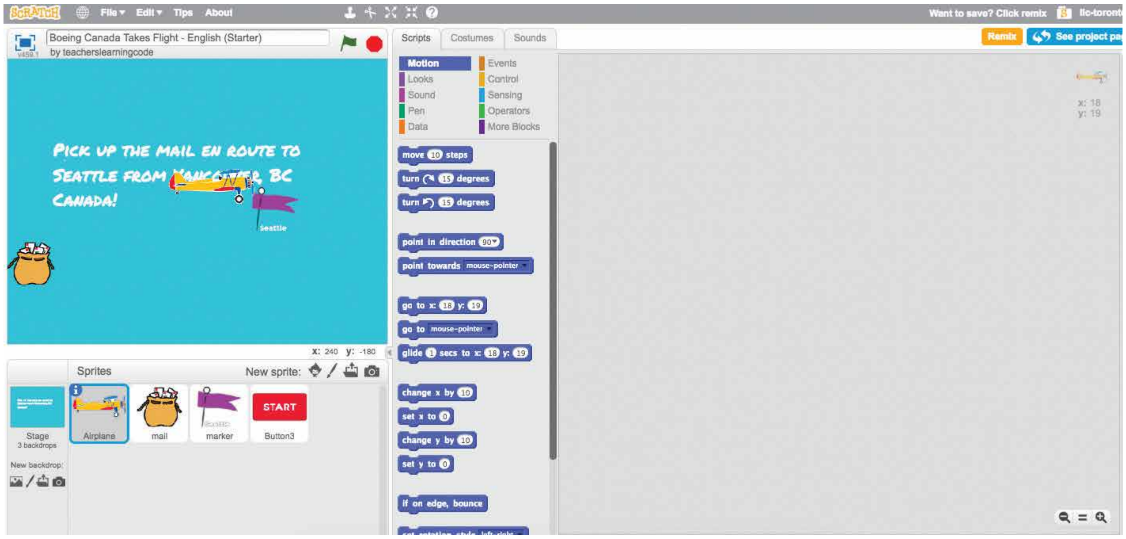
Solution Sheet created for Canada Learning Code. More info on Scratch (by MIT Media Lab): scratch.mit.edu

ANIMATING WITH SCRATCH SOLUTION SHEET

Canada Takes Flight

STEP 1: OPENING 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.

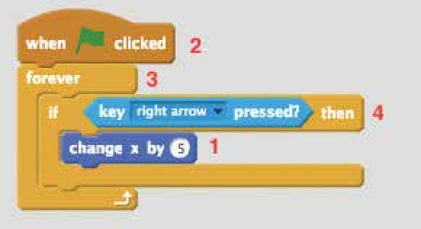
Solution Sheet created for Canada Learning Code. More info on Scratch (by MIT Media Lab): scratch.mit.edu

STEP 2: MAKE THE PLANE MOVE

1. Explain how the X,Y axis works (Stretch & Scuttle!)

   https://www.canadalearningcode.ca/lessons/stretch-and-scuttle/

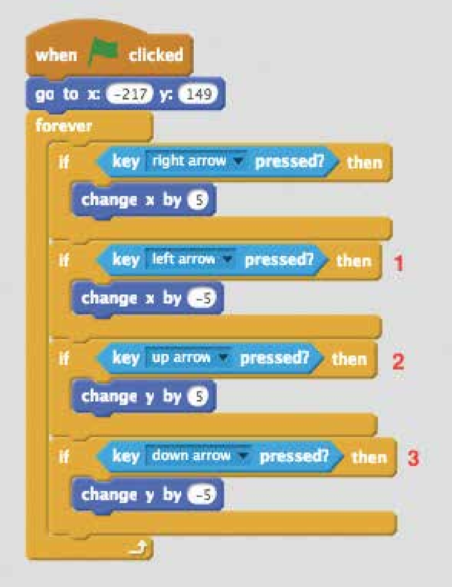
2. Make the plane move right (along x axis)
3. Make this happen when the green flag is clicked
4. Make this happen forever
5. Make this happen if the right arrow key is pressed



```
when 🟩 clicked 2
forever 3
    if key right arrow pressed? then 4
        change x by 5 1
```

STEP 3: ADDING OTHER DIRECTIONS

6. Duplicate the IF statement — make the plane move left if the left arrow key is pressed
7. Duplicate again — make the plan move up (along the y axis) if the up arrow key is pressed
8. Duplicate one last time — make the plan move down if the down arrow key is pressed
9. 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 1
        change x by -5
    if key up arrow pressed? then 2
        change y by 5
    if key down arrow pressed? then 3
        change y by -5
```

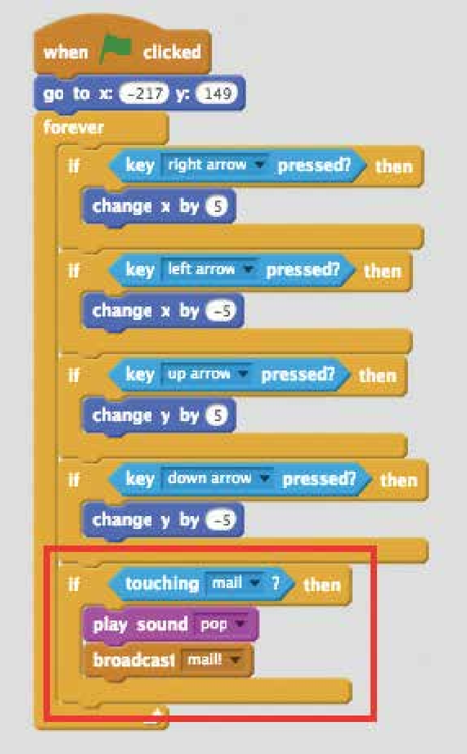
Solution Sheet created for Canada Learning Code. More info on Scratch (by MIT Media Lab): scratch.mit.edu

ANIMATING WITH SCRATCH SOLUTION SHEET

Canada Takes Flight

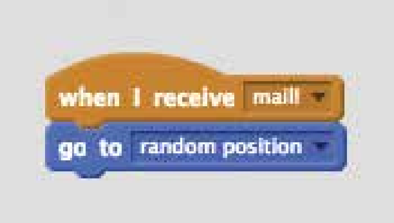
STEP 4: COLLECTING MAIL

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



```
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
    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"

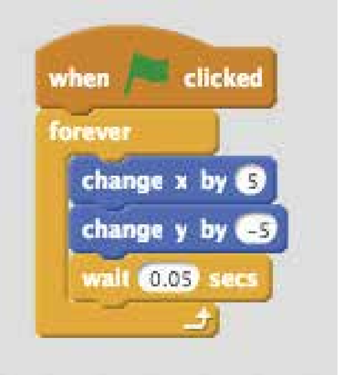


```
when I receive mail!
go to random position
```

Solution Sheet created for Canada Learning Code. More info on Scratch (by MIT Media Lab): scratch.mit.edu

STEP 5: WIND/GRAVITY EFFECT

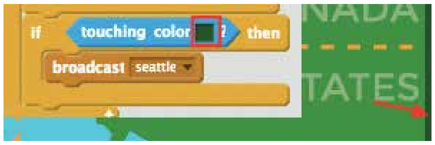
4. Select the plane sprite
5. Make the plane look like it's falling — make it move down (y axis)
6. Make the plane look like it's flying forwards — make it move to the right (x axis)
7. Make this happen forever, when the green flag is clicked
8. Make it wait for a short amount of time before looping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



```
when 🏴 clicked
forever
    change x by 5
    change y by -5
    wait 0.05 secs
```

STEP 6: FLYING TO SEATTLE

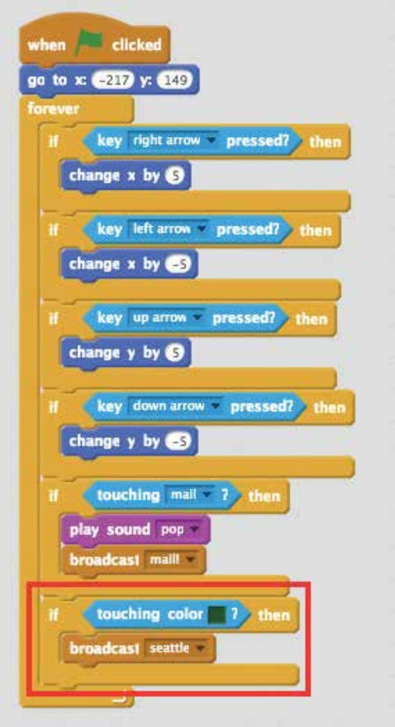
9. Make a new broadcast message called "seattle" to communicate with the other sprites, and tell them when to switch to the next screen



```
if touching color ■ ? then
    broadcast seattle
```

10. Make the plane broadcast this message when it is touching the dark green colour on the right edge

11. 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)



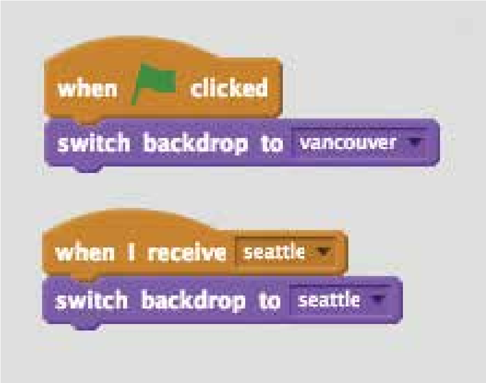
```
when 🏴 clicked
go to x: -217 y: 149
forever
    if key right arrow pressed? then
        change x by 5
    if key left arrow pressed? then
        change x by -5
    if key up arrow pressed? then
        change y by 5
    if key down arrow pressed? then
        change y by -5
    if touching mail ? then
        play sound pop
        broadcast maill
    if touching color ■ ? then
        broadcast seattle
```

Solution Sheet created for Canada Learning Code. More info on Scratch (by MIT Media Lab): scratch.mit.edu

Canada Takes Flight

STEP 6: FLYING TO SEATTLE

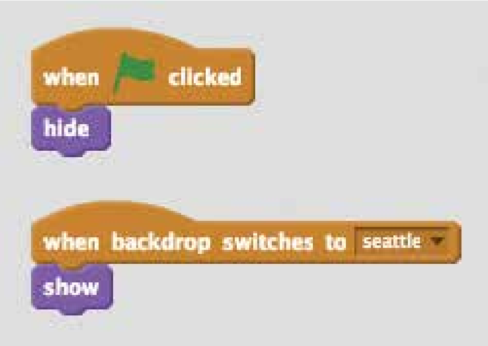
1. Select the Stage
2. Make the backdrop switch to seattle when it receives the "seattle" message
3. Make the backdrop switch back to vancouver when the green flag is clicked



```
when 🏁 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



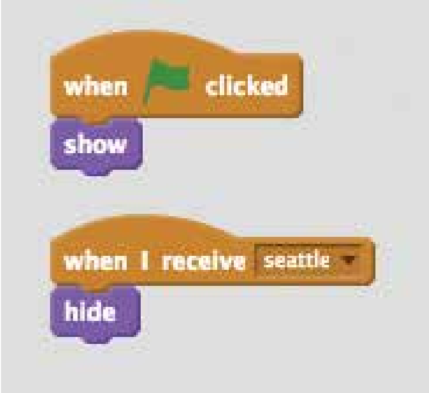
```
when 🏁 clicked
hide

when backdrop switches to seattle ▾
show
```

Solution Sheet created for Canada Learning Code. More info on Scratch (by MIT Media Lab): scratch.mit.edu

STEP 7: WIN CONDITION

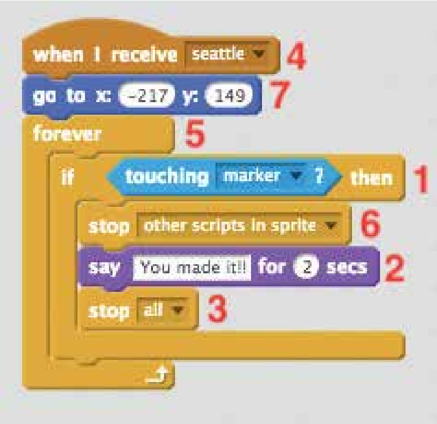
3. Select the marker sprite
4. Make the marker show when it receives the "seattle" message
5. Make the marker hide again when the green flag is clicked



```
when [flag] clicked
show

when I receive seattle
hide
```

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



```
when I receive seattle 4
go to x: -217 y: 149 7
forever 5
    if touching marker ? then 1
        stop other scripts in sprite 6
        say You made it!! for 2 secs 2
        stop all 3
```

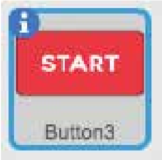
Solution Sheet created for Canada Learning Code. More info on Scratch (by MIT Media Lab): scratch.mit.edu

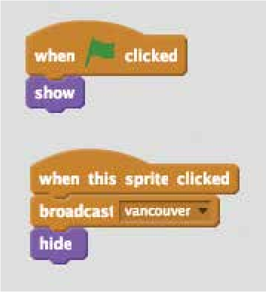
STEP 6: ANIMATING WITH SCRATCH SOLUTION SHEET

Canada Takes Flight

ADD-ON: OPENING SCREEN

10. Select the Start button
11. Make the button show when the green flag is clicked
12. Make the button hide when it is clicked
13. Make it broadcast a new message called 'vancouver' before hiding

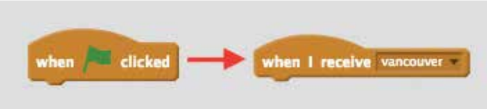




```
when ⚑ clicked
show

when this sprite clicked
broadcast vancouver ▾
hide
```

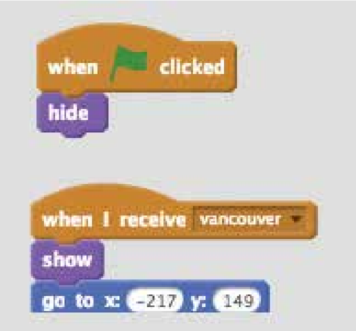
1. Make the game wait until the button is clicked to start (instead of the green flag): Replace all of the existing "when green flag clicked" scripts with "when I receive vancouver" scripts for the airplane and mail
2. Make the airplane hide when the green flag is clicked, then show when it receives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



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

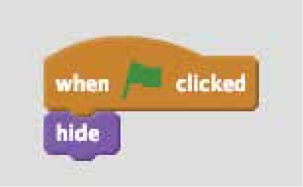
Solution Sheet created for Canada Learning Code. More info on Scratch (by MIT Media Lab): scratch.mit.edu

ADD-ON: OPENING SCREEN



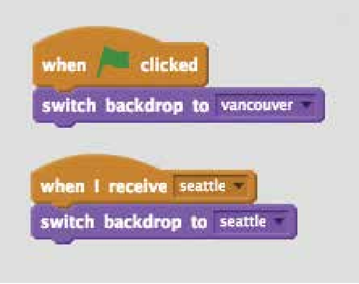
```
when 🏳 clicked
hide

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



```
when 🏳 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 🏳 clicked
switch backdrop to vancouver ▾

when I receive seattle ▾
switch backdrop to seattle ▾
```

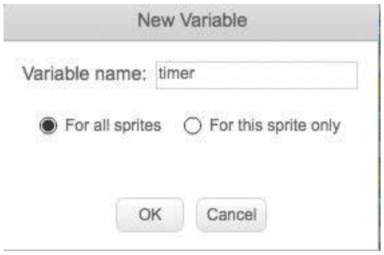
Solution Sheet created for Canada Learning Code. More info on Scratch (by MIT Media Lab): scratch.mit.edu

STEP 6: ANIMATING WITH SCRATCH SOLUTION SHEET

Canada Takes Flight

ADD-ON: TIMER

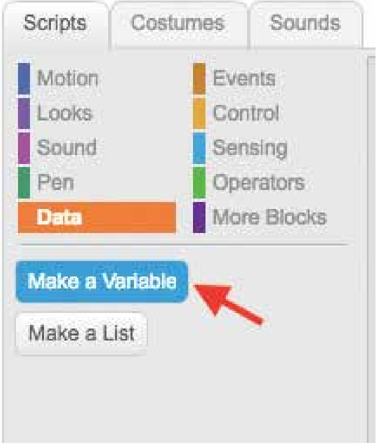
4. Make a new variable called 'timer'
5. Make the timer increase by 1
6. Make this happen forever, once 'vancouver' has been received
7. Make the timer wait 1 second before increasing
8. Make the timer reset to 0 when vancouver is received (aka button is clicked)
9. Make the airplane say the time when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

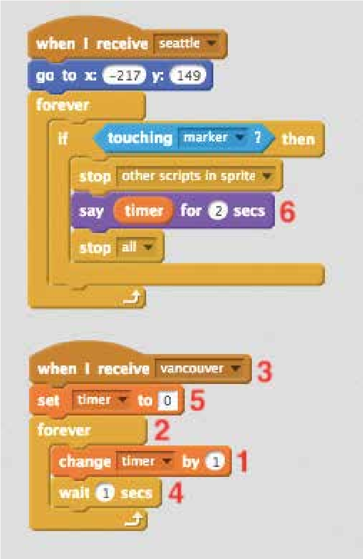


New Variable

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

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

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

Solution Sheet created for Canada Learning Code. More info on Scratch (by MIT Media Lab): scratch.mit.edu

DATE:

DATE:

GET IN TOUCH

Teachers Learning Code
129 Spadina Ave, Unit 501
Toronto, ON M5V 2L3

info@teacherslearningcode.com
teacherslearningcode.com

※teachers learning code