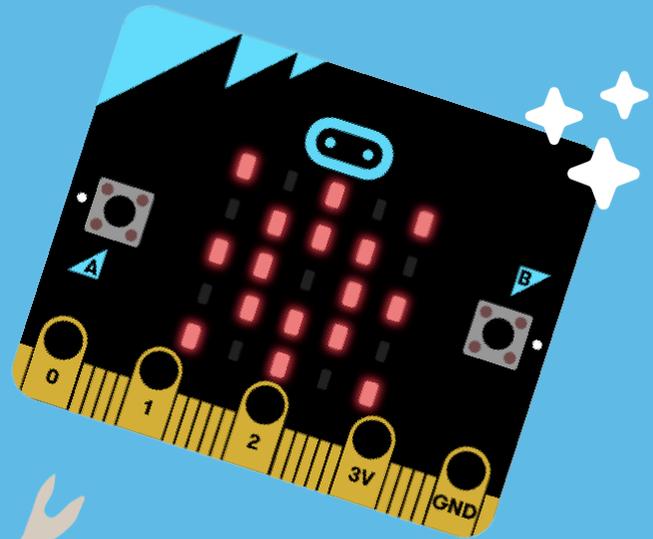


# Hour of Code

## Merry Micro:bits



[kidscodejeunesse.org](https://kidscodejeunesse.org)

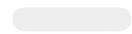
[@kidscoding](https://twitter.com/kidscoding)





We are a bilingual, Canadian not for profit organization dedicated to empowering children and educators, with a particular focus on girls and children from disadvantaged groups, with the skills we all need to thrive in a technology driven society. Since 2013, we have reached more than 4300 teachers and over 150,000 students.

We're igniting young minds across Canada by teaching kids to code. Learning computational thinking, coding, and physical computing provides youth with the skills needed to create, communicate, and innovate with the tools of the 21st century. And by introducing code into classrooms and communities, we're able to create a new generation of thinkers, creators and doers.





# Table of Contents

ACTIVITY: Hour of Code - Micro:bit Snow Globe ..... 2

STRATEGIES TO GET UNSTUCK ..... 13

ACTIVITY CARDS ..... 14

CERTIFICATES ..... 23

RESOURCES ..... 24



# Hour of Code

# Micro:bit Snow Globe

## Length:

60 minutes

## Learning Objectives:

- Familiarity with physical computing through micro:bit.
- Familiarity with basic concepts like sequences, modularization and repetition.
- Familiarity with computational practices like planning and debugging

## Materials:

- Online Editor [makecode.microbit.org](https://makecode.microbit.org)
- One micro:bit with its USB cable (optional)

## ACTIVITY 1 My LED Snow Globe



The class will use MakeCode blocks to create a “snow globe” animation on the micro:bit.

This activity uses blocks from the Basic, Input, Led, and Functions categories. Notice that the categories and blocks are colour-coded, which makes it easy to follow the screenshots below and find the blocks students will need.

Follow the steps below and help students create their own sparkling snow globe!

# Hour of Code

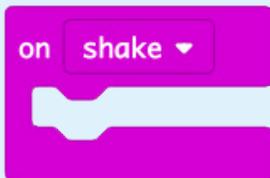
## Micro:bit Snow Globe



### STEP 1 Basic: create the shape of the globe

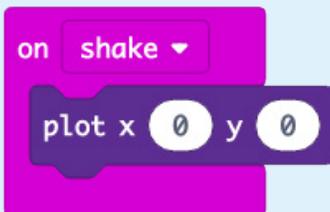


### STEP 2 Input: choose the trigger button “on shake”



### STEP 3 LEDs on and off

A) Inside the **on shake** block, add a **plot x 0 y 0** block. Try it in the emulator, clicking the SHAKE button. Where does it light up? It's outside the snow globe, not where we want it.



# Hour of Code

## Micro:bit Snow Globe



Each LED light on the micro:bit has a code, just like coordinates on a graph . On the micro:bit, the codes start at the top left (x 0, y 0) and work their way across to the right (x) and down (y).

What are some x,y coordinates that would be inside the globe? Have the class guess a few and test them out in the code. SHAKE the emulator to verify the result.

Here is an LED that works inside the globe: **plot x 1 y 1**.

```
on shake  
  plot x 1 y 1
```

**B)** To make this behave more like a snowflake, have the class add an unplot with the same coordinates.

```
on shake  
  plot x 1 y 1  
  unplot x 1 y 1
```

# Hour of Code

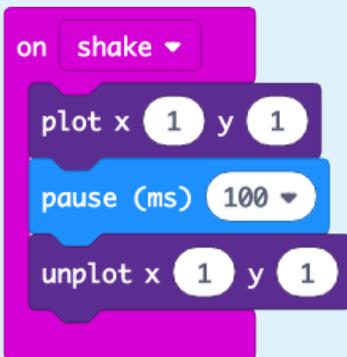
## Micro:bit Snow Globe



Test it. The code is right, but it looks like nothing is happening. Can the class guess what's wrong? The micro:bit is turning on and off so quickly, we can't see it.

Use the “snail” button  on the emulator, to slow the pace.

A **pause** block is used to slow down the program in real time.

A Scratch code block for a snowflake. It starts with an 'on shake' event block. Inside, there are three blocks: a 'plot x 1 y 1' block, a 'pause (ms) 100' block, and an 'unplot x 1 y 1' block. The 'plot' and 'unplot' blocks are purple, and the 'pause' block is blue.

Students could add more snowflakes with more **plot**, **pause**, and **unplot** blocks. However, adding lots of blocks gets tedious and ultimately makes the script hard to read.

# Hour of Code

## Micro:bit Snow Globe

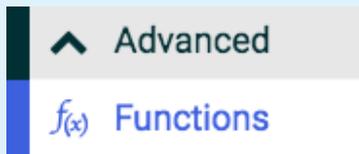


### STEP 4 Functions

The class is now ready to give the basic **plot-pause-unplot** steps a shorthand name and then run them by “calling” the name. This simple but powerful technique will help make the program easier to read and easier to extend with more snowflakes.

A) In the Advanced category, create a function and name it “snowflake1” (“snowflake” and the number “1”).

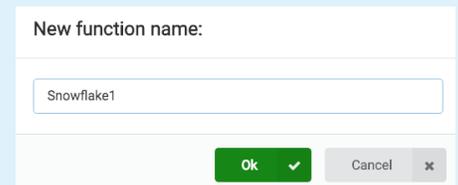
1.



2.



3.



4.



# Hour of Code

## Micro:bit Snow Globe



B) Drag the first set of **plot-pause-unplot** instructions into the **snowflake1** function.

```
function Snowflake1
  plot x 1 y 1
  pause (ms) 100
  unplot x 1 y 1
```

With the **snowflake1** function “defined,” open the **Functions** blocks, that the **call snowflake1** block out and put it into the **on shake** block.

Use the snail button  to demonstrate how the function is “called” with the “call function” block. In snail mode, each block is highlighted as the program runs each snowflake sequence.

```
on shake
  call function Snowflake1
```

Test this in the emulator. The single snowflake should turn on and off as before.

# Hour of Code

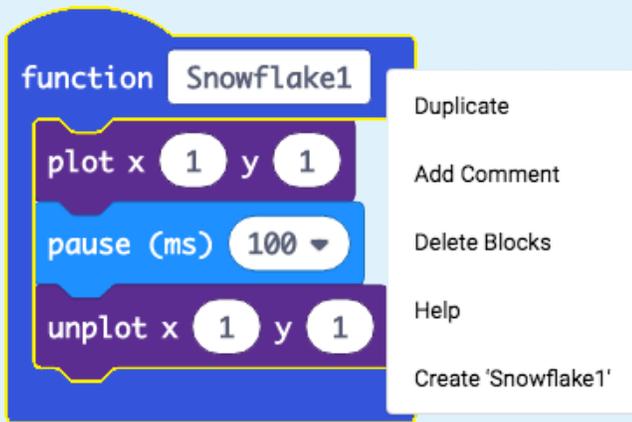
## Micro:bit Snow Globe



### STEP 5 More functions

Next step is to add more snowflakes using the same function technique.

A) Right-click the **snowflake1** function and **Duplicate** it.



This will automatically create a new function with a number. (**snowflake2**, **snowflake3**, etc.)

# Hour of Code

## Micro:bit Snow Globe



B) For each of the new functions, change the coordinates (x,y) to create the effect of snow sparkling inside the globe. Here are two coordinates that work (students can choose their own).

```
function Snowflake2
  plot x 2 y 2
  pause (ms) 100
  unplot x 2 y 2

function Snowflake3
  plot x 1 y 3
  pause (ms) 100
  unplot x 1 y 3
```

C) Finally, students use the **on shake** block to **call** all three functions.

```
on shake
  call function Snowflake1
  call function Snowflake2
  call function Snowflake3
```

# Hour of Code

## Micro:bit Snow Globe



### STEP 6 Check and continue

A) The class should now have all these elements in their programs.

They may be arranged differently on each student's canvas. This is fine; it has no impact on how the program runs. Students can drag the stacks around as they like to make their programs easy to read.

```
on start
  show leds
  [5x5 grid of white and blue squares]
```

```
on shake
  call function Snowflake1
  call function Snowflake2
  call function Snowflake3
```

```
function Snowflake1
  plot x 1 y 1
  pause (ms) 100
  unplot x 1 y 1
```

```
function Snowflake2
  plot x 2 y 2
  pause (ms) 100
  unplot x 2 y 2
```

```
function Snowflake3
  plot x 1 y 3
  pause (ms) 100
  unplot x 1 y 3
```

B) Make more snow!

# Hour of Code

## Micro:bit Snow Globe



From here, the class can start to play. Have students continue making more snowflake functions, turning more LEDs on and off.

They may change the coordinates inside each function to make the snow “look” the way they want. Invite them to switch the order in which they call the functions, or call the same function multiple times. Have the class look inside the other block categories for a block that **repeats** the code a bunch of times in a single **shake**.

### ACTIVITY 2 Let's download our animation to the micro:bit



#### Optional

Transferring your program to the micro:bit is as simple as saving a file to a drive on your computer.

- **Connect** the micro:bit to your computer with the USB cable.
- **Name** your program and click **Download** (a .hex file will be saved in your Downloads)
- **Drag** the .hex file from your computer onto your micro:bit.



The participants will practice for a few minutes.

# Hour of Code

## Micro:bit Snow Globe



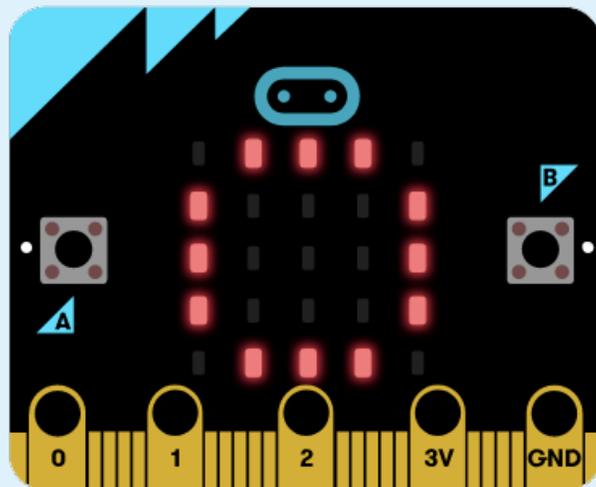
### QUESTIONS?

The following link has instructions for the browser that you are using (Safari, Chrome, Firefox, etc.)

<https://makecode.microbit.org/device/usb>

Challenge alert! Can you find more ways to animate your globe? Post pictures, videos, or even a link to your project on social media and tag [@kidscoding](#) so we can see!

Have fun!



# Top 5 Strategies to Get Unstuck When Coding (According to Kids)



## 1. Read through your code

Go back to the beginning. Read it as if it was normal text. Be critical and analytical of what you've built.

## 2. Experiment with your code

Take blocks out. Add blocks in. Tinker with it. Play with it.

## 3. Look for examples

How are your friends doing this? While micro:bit does not yet have the same bank of projects as Scratch, there are many example of projects on the landing page that might give you ideas for how to resolve a problem. Look for different examples to inspire yourself and build something new.

## 4. Work with someone else

Collaboration and consultation are essential. You will be able to build more together than when you are alone. Effective teamwork is key to attaining growth and success.

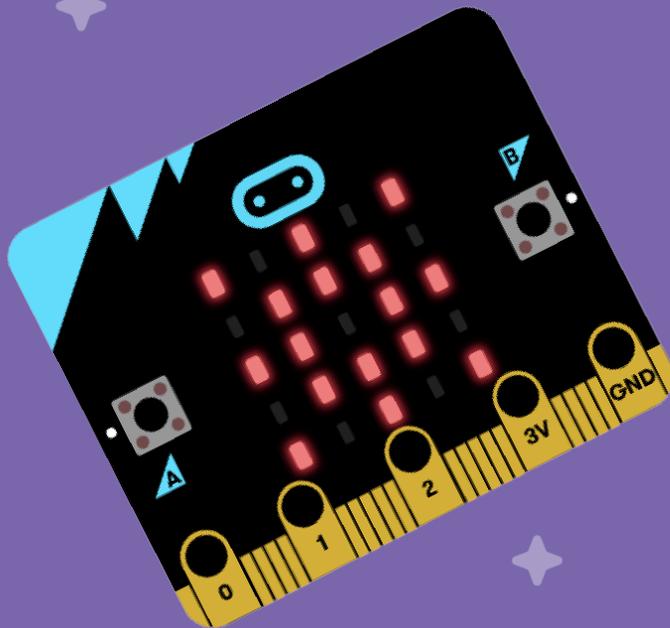
## 5. Be persistent but know when to take a break

Persistence is a quality found in all innovators. When you are unable to solve a problem after working on it for a long time, take a break. This will allow you to see the bigger picture and view the problem from different perspectives to find a solution or a better idea.

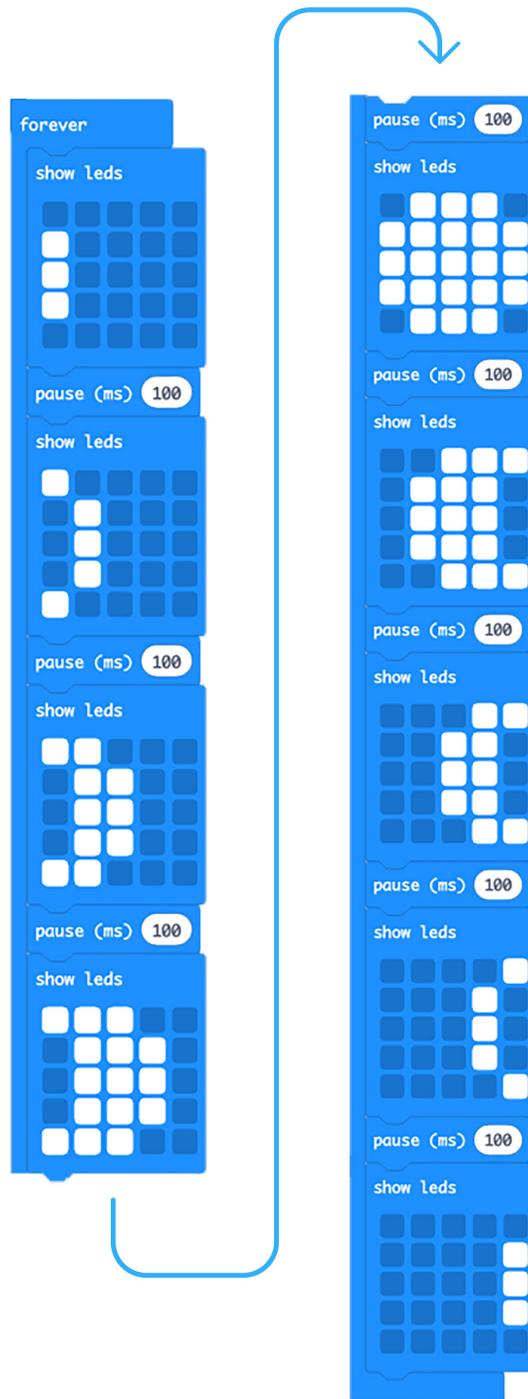
Survey conducted by Harvard Graduate School of Education  
See "Getting Unstuck: Karen Brennan's '8 for 8'" video:  
[https://youtu.be/c\\_AdWB1GkRw](https://youtu.be/c_AdWB1GkRw)

# Activity Cards

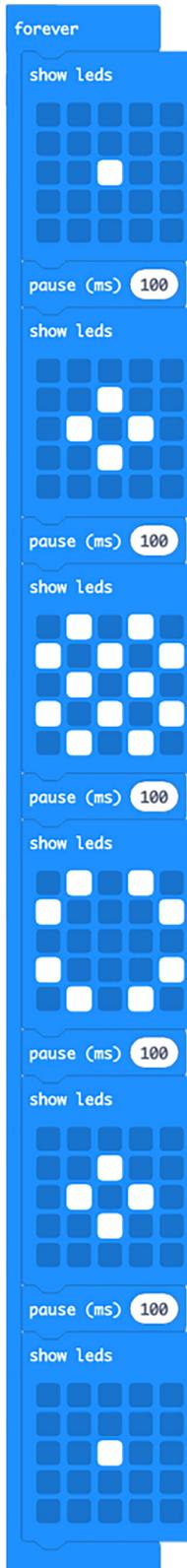
Go further with the micro:bit!



# Moon Phases



```
forever
  show leds
  pause (ms) 100
  show leds
  pause (ms) 100
```



The code block is a vertical stack of Scratch blocks. It starts with a 'forever' loop block. Inside the loop, there are eight pairs of blocks. Each pair consists of a 'show leds' block followed by a 'pause (ms) 100' block. The 'show leds' blocks display different patterns of white squares on a 5x5 grid. The patterns are: 1. A single white square in the center. 2. A 3x3 grid of white squares. 3. A 5x5 grid of white squares. 4. A 3x3 grid of white squares with a white square in the center. 5. A 5x5 grid of white squares with a white square in the center. 6. A 3x3 grid of white squares with a white square in the center. 7. A 5x5 grid of white squares with a white square in the center. 8. A 3x3 grid of white squares with a white square in the center.

# Starry Lights

```
forever
  plot x pick random 0 to 4 y pick random 0 to 4
  unplot x pick random 0 to 4 y pick random 0 to 4
```

# Spinning Dreidel

```
on start
  show string "Shake!"
  show leds
```



```
on shake
  set spin to pick random 1 to 4
  if spin = 1 then
    show leds
  +
  if spin = 2 then
    show leds
  +
  if spin = 3 then
    show leds
  +
  if spin = 4 then
    show leds
  +
```



**Tip:** Duplicate functions to create more candles that scroll across the screen.

```

on button A pressed
  repeat 3 times
  do
    call function Men1
    call function Men2
    call function Men3
    call function Men4
    call function men5
    call function men6
    call function Men7
    call function men8
    call function men9
  clear screen
  pause (ms) 1000
  
```

```

function Men1
  show leds
  repeat 2 times
  do
    toggle x 0 y 1
    pause (ms) 100
    toggle x 2 y 1
    pause (ms) 100
    toggle x 4 y 0
    pause (ms) 100
  
```

```

function Men2
  show leds
  repeat 2 times
  do
    toggle x 1 y 1
    pause (ms) 100
    toggle x 3 y 0
    pause (ms) 100
  
```

```

function Men3
  show leds
  repeat 2 times
  do
    toggle x 0 y 1
    pause (ms) 100
    toggle x 2 y 0
    pause (ms) 100
    toggle x 4 y 1
    pause (ms) 100
  
```

```

function Men4
  show leds
  repeat 2 times
  do
    toggle x 1 y 0
    pause (ms) 100
    toggle x 3 y 1
    pause (ms) 100
  
```

```

function men5
  show leds
  repeat 2 times
  do
    toggle x 0 y 0
    pause (ms) 100
    toggle x 2 y 1
    pause (ms) 100
    toggle x 4 y 1
    pause (ms) 100
  
```

```

function men6
  show leds
  repeat 2 times
  do
    toggle x 1 y 1
    pause (ms) 100
    toggle x 3 y 1
    pause (ms) 100
  
```

```

function Men7
  show leds
  repeat 2 times
  do
    toggle x 0 y 1
    pause (ms) 100
    toggle x 2 y 1
    pause (ms) 100
  
```

# Square Dance

```
forever
  for index from 0 to 3
  do
    toggle x index y 0
    pause (ms) 100
    toggle x 4 y index
    pause (ms) 100
    toggle x 4 - index y 4
    pause (ms) 100
    toggle x 0 y 4 - index
    pause (ms) 100
```

## Square Dance 2

```
forever
  for vertical from 0 to 4
  do
    for horizontal from 0 to 4
    do
      plot x horizontal y vertical
      pause (ms) 100
    do
      for horizontal from 0 to 4
      do
        for vertical from 0 to 4
        do
          unplot x horizontal y vertical
          pause (ms) 100
        do
```

```
forever
  show leds
  for index from 0 to 3
  do
    plot x index y 0
    plot x 0 y 4 - index
    plot x 4 - index y 4
    plot x 4 y index
    pause (ms) 200
    unplot x index y 0
    unplot x 4 - index y 4
    unplot x 0 y 4 - index
    unplot x 4 y index
    pause (ms) 200
```

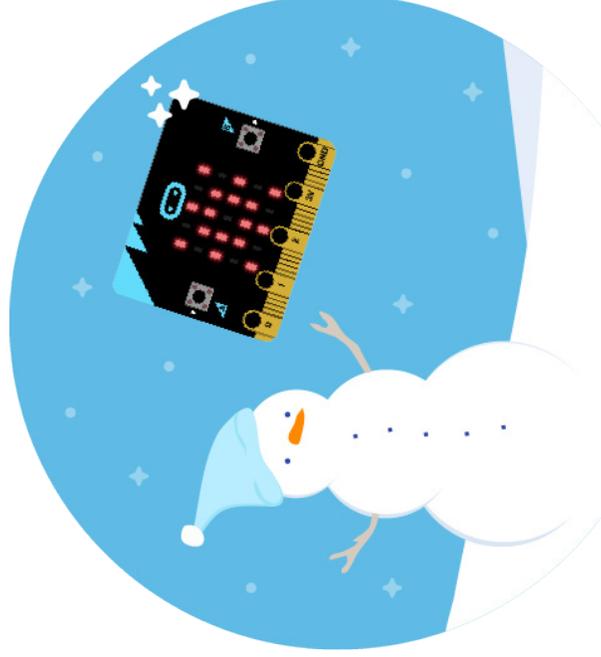


# Certificate of Achievement

presented to:

\_\_\_\_\_

for completing KCJ's Merry Micro:bits project for Hour of Code!



DATE: \_\_\_\_\_



# Resources

Microbit resources for teachers include lesson plans, curriculums, project ideas:

<https://microbit.org/teach/>

The KCJ art:bit facilitates animation and makes using the micro:bit easier for younger students. It is currently available for chromebooks and soon to be available for iPads.

<https://kidscodejeunesse.org/artbit.html>

Code Club micro:bit projects are great for stretch challenges:

<https://codeclubprojects.org/en-GB/microbit/>

We want to see your kids' coding skills!

Make sure to share pictures, videos, quotes and screen captures on social media. Tag us on social media and use the hashtag [#kidscodingmicrobits](#) so that we can share with the rest of our online community.

 /kidscodejeunesse

 @kidscoding



