

Snow Globe

Simulate a snow globe with a simple animation effect.

There are several built-in images for the micro:bit. For a complete list and some additional details, visit the MicroPython documentation (<http://microbit-micropython.readthedocs.io/en/latest/tutorials/images.html>) .

A custom image can be created by coding a value for each LED in the array using values from zero (0 for full off) to nine (9 for full on) or any value in between for different levels of brightness.

For this program, we're going to use both a built-in image for the house and then a sequence of custom images to represent the falling snow. We will also scroll some text as instructions.

```
from microbit import *  
  
display.scroll("SHAKE FOR SNOW")  
display.show(Image.HOUSE)
```

Next, for the custom images, use the Image() function and populate the values for each LED using five groups of five numbers separated by a colon (ie. 5 rows of 5 LED brightness values). You can use the LED designer worksheet to plan out how you want to code your images. We are using different brightness levels to enhance the animation effect.

```
snow1 = Image("00900:90909:09090:90909:00000")  
snow2 = Image("00500:90909:09090:90909:05050")  
snow3 = Image("00300:50505:09090:90909:09090")  
snow4 = Image("00100:10101:05050:90909:09090")  
snow5 = Image("00000:00000:01010:50505:05050")  
snow6 = Image("00000:00000:00000:10101:01010")  
snow7 = Image("00000:00000:00000:00000:01010")
```

Now that the custom images have been created, we are going to group them all together into a list. A list is created when you start and end with square brackets. We'll include the house image at the beginning and end of the sequence.

```
snow_globe = [Image.HOUSE, snow1, snow2, snow3,  
              snow4, snow5, snow6, snow7, Image.HOUSE]
```

Now all the pieces are in place to create the animation. Using a while True loop, when the micro:bit detects it has been shaken, show all the images in order using a delay between each image:

```
while True:
```

```
if accelerometer.was_gesture('shake'):
    display.show(snow_globe, delay=1000)
```

The end effect is a light show that appears like falling snow!