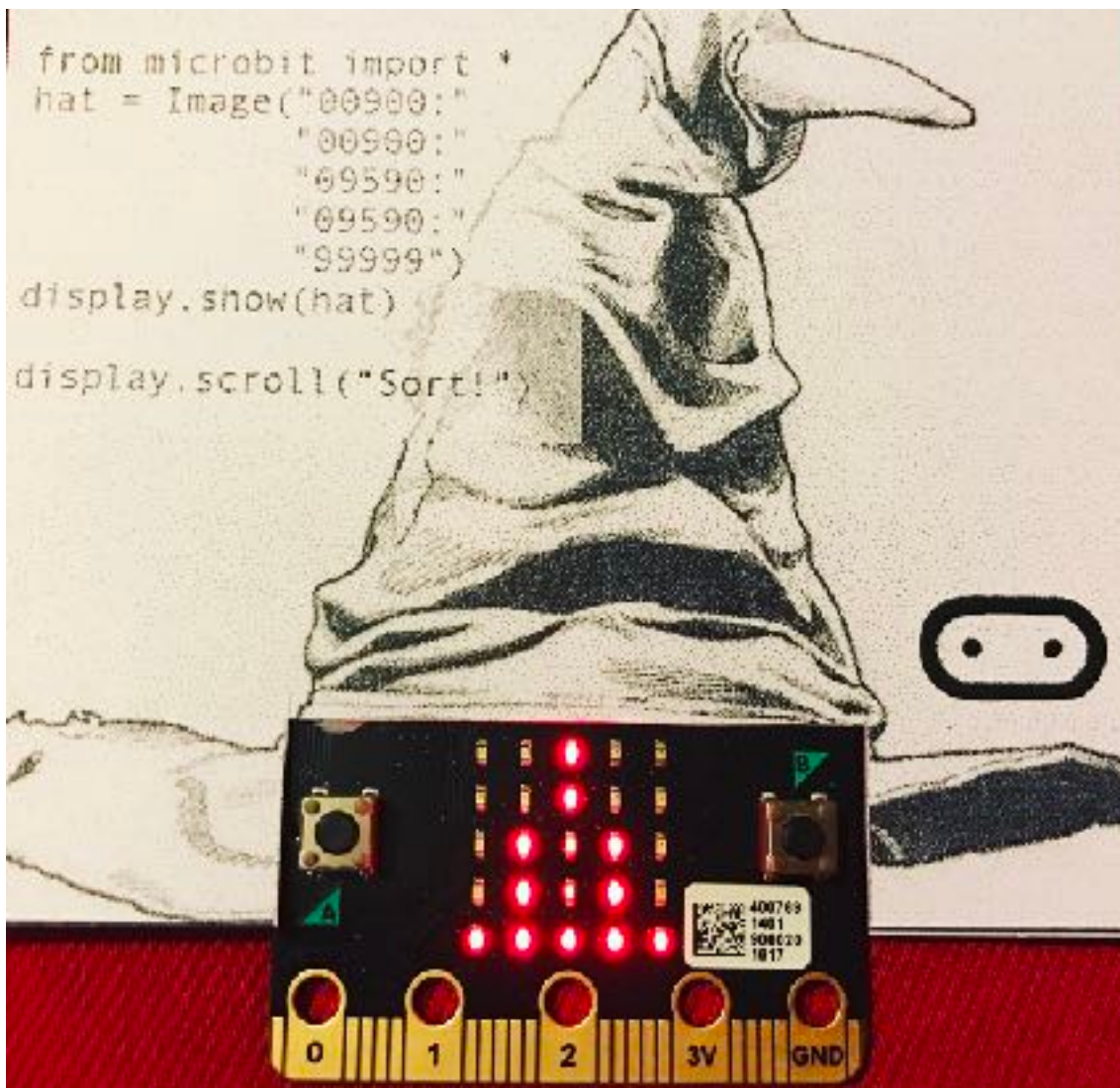


Harry Potter

micro:bit scheme of work



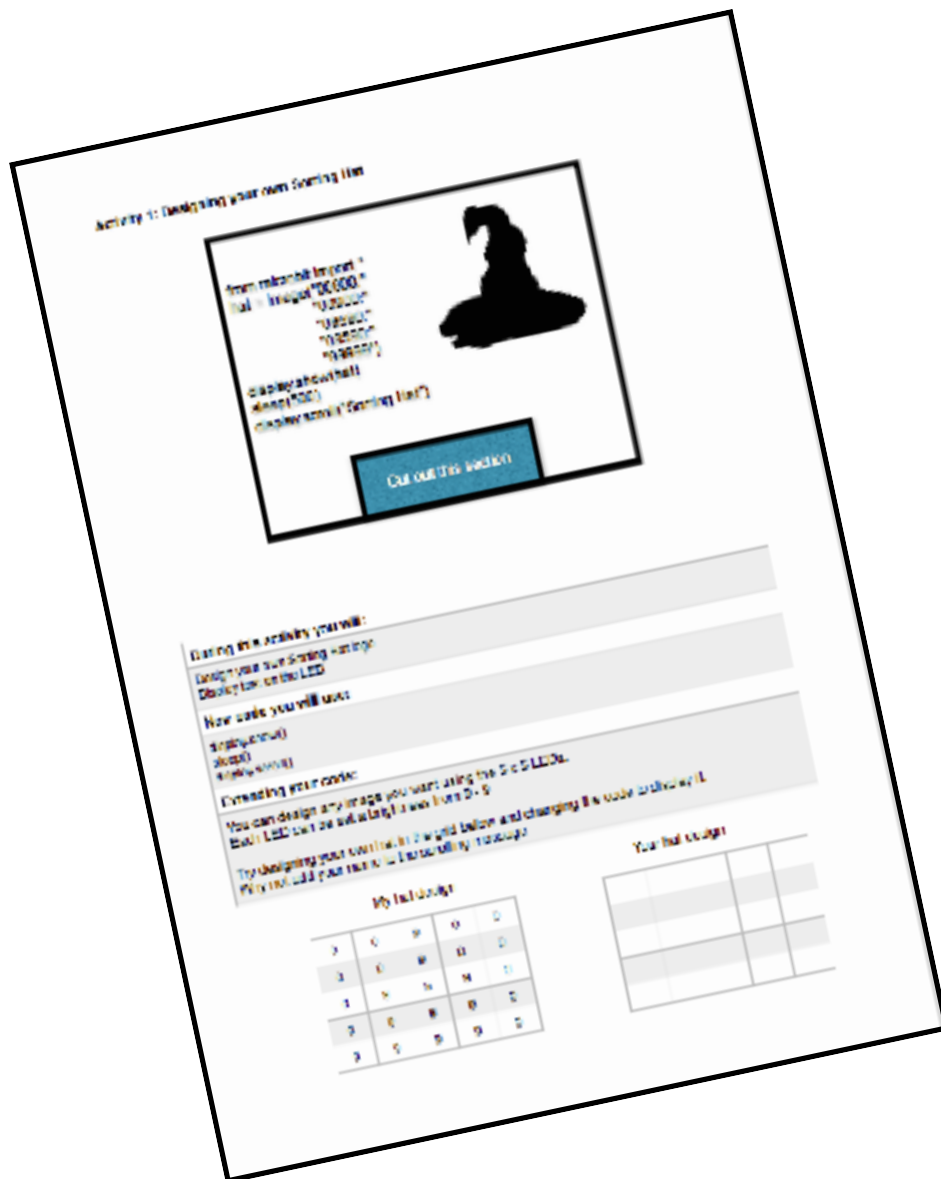
Spencer Organ

King Edward VI Sheldon Heath Academy - Birmingham

February 2017

Join us in the world of Harry Potter as we use the micro:bit to learn some magical lessons.

How to use the resources:



If you want you can cut out the cards from the top of the sheet and blu-tack them onto the micro:bit (like on the front cover) and then follow through the tasks on the sheet. These first 5 lessons get you started with the micro:bit and micro python.

- Activity 1: Designing a sorting hat
- Activity 2: Being sorted into a house
- Activity 3: Potions
- Activity 4: Wand practice
- Activity 5: Charms class

Each of the 5 activities are designed to introduce different functions of the micro:bit and expand on the previous task. There is plenty of scope for developing ideas in each activity.

Activity 1: Designing a sorting hat

- custom image
- `display.show()`
- `sleep()`
- `display.scroll()`

Activity 2: Being sorted into a house

- lists
- `random.choice` (from a list)
- if statements

Activity 3: Potions

- while True:
- `gesture = accelerometer.current_gesture()`
- if statements

Activity 4: Wand practice

- `gesture = accelerometer.current_gesture()`
- if statements


Activity 5: Charms class

- if ... and ... statements
- `button_choice`
- random lists
- `display.clear()`

Activity 1: Designing your own Sorting Hat

```

from microbit import *
hat = Image("00900:"
            "00900:"
            "09590:"
            "09590:"
            "99999")
display.show(hat)
sleep(500)
display.scroll("Sorting Hat")
    
```



Cut out this section

During this activity you will:
Design your own Sorting Hat logo Display text on the LED
New code you will use:
display.show() sleep() display.scroll()
Extending your code:
You can design any image you want using the 5 x 5 LEDs. Each LED can be set a brightness from 0 - 9
Try designing your own hat in the grid below and changing the code to display it. Why not add your name to the scrolling message


My hat design

0	0	9	0	0
0	0	9	0	0
0	9	5	9	0
0	9	5	9	0
9	9	9	9	9

Your hat design

Activity 2: Being sorted into a house

```
from microbit import *
import random
houses=["Gryffindor", "Hufflepuff", "Ravenclaw", "Slytherin"]
my_house = random.choice(houses)
display.scroll(my_house)
```



Cut out this section

During this activity you will:

Display built-in images on the micro:bit LED display
Pick a random item from a list

New code you will use:

You may want to keep the sorting hat image from sheet 1 as you start this activity

```
houses = [ ]
random.choice()
```

Extending your code:

Why not redesign an image for each of the houses using the example on sheet 1.
You could then use than **if** statement to control what happens next

Eg,

```
if my_house == "Slytherin":
    display.show(Image.SAD)
```

Can you extend it even further my adding a different image for each house?

Activity 3: Potions

```
from microbit import *  
  
while True:  
    gesture = accelerometer.current_gesture()  
    if gesture == "shake":  
        display.show(Image.HEART)  
    else:  
        display.show(Image.DIAMOND)
```



During this activity you will:

Display images on the micro:bit LEDs
Use the shake gesture to mix up your potions

New code you will use:

```
while True:  
gesture = accelerometer.current_gesture()  
if gesture == "shake":
```

Extending your code:

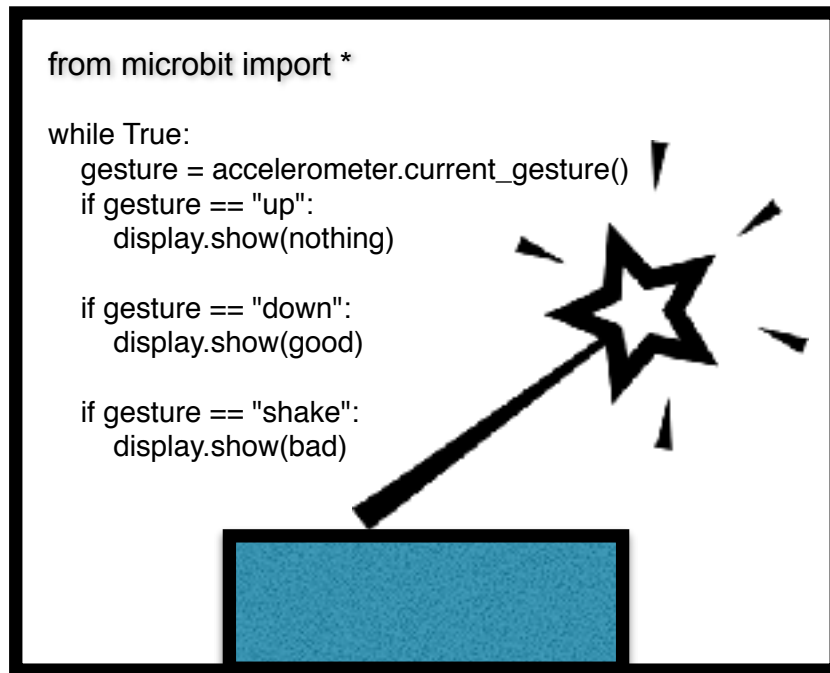
I have used a diamond for the empty cauldron and a heart for when the spell has worked.

Why don't you change the images and see what else you can brew up?

There are many other gestures you can try and we will use these later in charms class!

up, down, left, right, face up, face down, freefall, 3g, 6g, 8g, shake

Activity 4: Wand practice



The code to design the wands is on the back of this sheet and will need to be typed in before the while True section of the code.

During this activity you will:

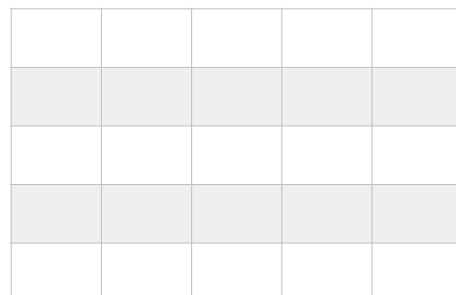
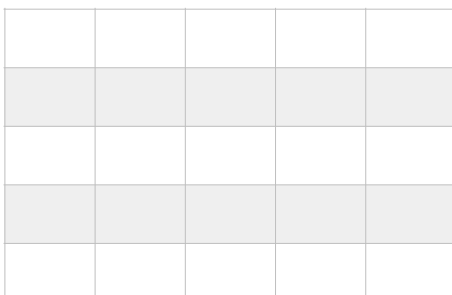
Display images on the micro:bit LEDs
Use the up, down, shake gesture to control what happens
Take care not to hurt yourself or anyone else around you.

New code you will use:

```
if gesture == "up":
```

Extending your code:

You will need to design some basic wands at the start of the activity. You can always improve on my designs if you want!

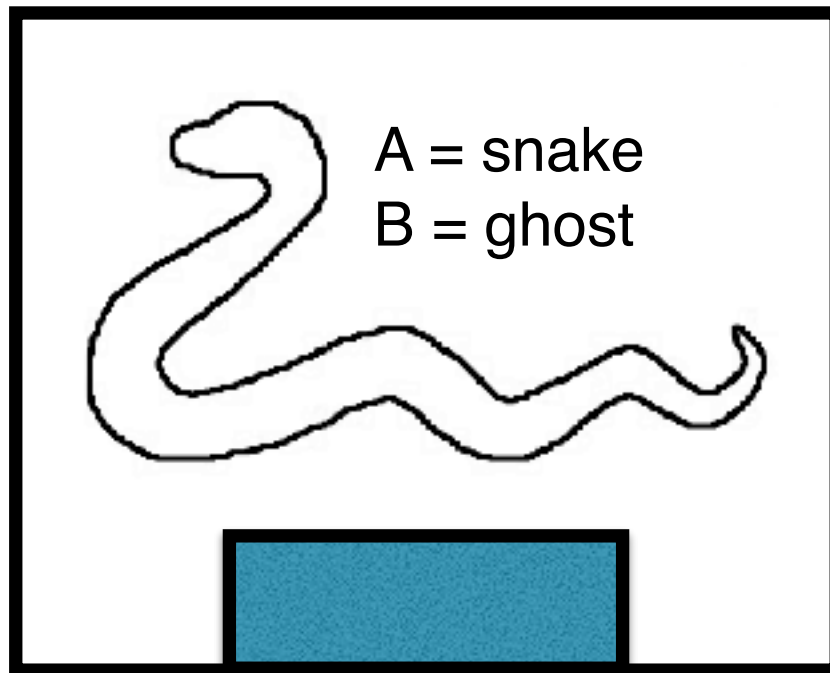


```
nothing = Image("00900:"  
               "00900:"  
               "00900:"  
               "00900:"  
               "00900")
```

```
good = Image("00009:"  
            "00090:"  
            "00900:"  
            "09000:"  
            "90000:")
```

```
bad = Image("22222:"  
           "22222:"  
           "22292:"  
           "22919:"  
           "19222:")
```


Activity 5: Charms class



During this activity you will:

Press button (A) or (B) to respond to random names shown on the screen.

This activity uses lots of If ... and statements.

This is a long piece of code and you will need to type it in very carefully

New code you will use:

```
If ..... and .....  
button_choice
```

Extending your code:

This could easily be the basis of a game.

You could make the game harder by replacing `display.scroll(button_choice)` with an image.

Try replacing the `display.scroll(button_choice)` with: (A) = snake, (B) - ghost

```
if button_choice == "A":  
    display.show(Image.SNAKE)  
else:  
    display.show(Image.GHOST)  
sleep(500)
```

<— the `sleep(500)` gives you time to see it

```
from microbit import *
import random
while True:
    options=["A","B"]
    button_choice = random.choice(options)
    display.scroll(button_choice)
    display.clear()
    while True:
        if button_choice=="A" and button_a.is_pressed():
            display.show(Image.HAPPY)
            sleep(500)
            break
        if button_choice=="A" and button_b.is_pressed():
            display.show(Image.SAD)
            sleep(500)
            break
        if button_choice=="B" and button_b.is_pressed():
            display.show(Image.HAPPY)
            sleep(500)
            break
        if button_choice=="B" and button_a.is_pressed():
            display.show(Image.SAD)
            sleep(500)
            break
        else:
            display.show(Image.DIAMOND)
    display.clear()
```

```
from microbit import *
import random
while True:
    options=["A","B"]
    button_choice = random.choice(options)
    if button_choice == "A":
        display.show(Image.SNAKE)
    else:
        display.show(Image.GHOST)
    sleep(500)
    display.clear()
    while True:
        if button_choice=="A" and button_a.is_pressed():
            display.show(Image.HAPPY)
            sleep(500)
            break
        if button_choice=="A" and button_b.is_pressed():
            display.show(Image.SAD)
            sleep(500)
            break
        if button_choice=="B" and button_b.is_pressed():
            display.show(Image.HAPPY)
            sleep(500)
            break
        if button_choice=="B" and button_a.is_pressed():
            display.show(Image.SAD)
            sleep(500)
            break
        else:
            display.show(Image.DIAMOND)
    display.clear()
```