

- Shake the micro:bit to throw the dice
- Will you score a six?



10 mins

1. Overview

Did you know that the micro:bit contains an accelerometer which detects movement? We will use this to create a program which detects when the

micro:bit is shaken; choses a random number; and displays the corresponding number of spots. Just like a real dice, but cooler. And electronic. Which lights up!

2. How our dice will work

We are going to build upon the random number program which we made on the previous worksheet, *Hogwarts Sorting Hat*. We will also program which LEDs will light up. Here's how it will work:

- When the micro:bit is shaken, a random number (between 1 and 6) is generated
- If the random number is 1, one LED is lit
- If the random number is 2, two LEDs are lit
- If the random number is 3... you get the idea!

3. Programming the LEDs

The micro:bit has many built-in graphics, but it can be fun to design your own. The LEDs are arranged in a five-by-five grid, giving a total of 25 LEDs. Each LED's brightness can be set at any number from 0 (off) through to 9 (brightest).

This is how we would create a ghost (with a eyes at half brightness):

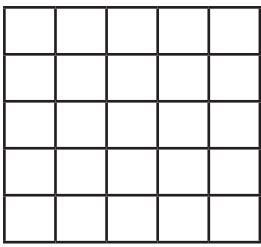
```
from microbit import *  
  
myGhost = Image("09990:"  
                "95959:"  
                "95959:"  
                "99999:"  
                "90909")  
  
display.show(myGhost)
```

You can also save space by writing the LED values like this:

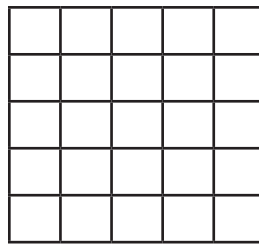
```
myGhost = Image("09990:95959:95959:99999:90909")
```

4. Making the spots

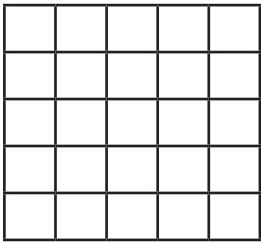
Use these grids to create your own dice layout for numbers 1 to 6. Fill-in the gaps in the code for each grid.



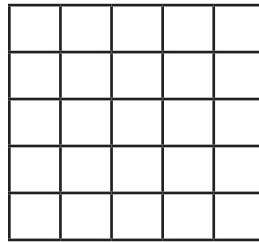
```
one = Image("____:":
            "____:":
            "____:":
            "____:":
            "____")
```



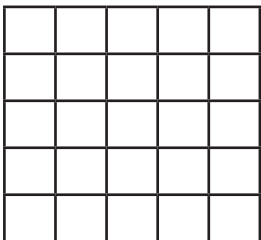
```
two = Image("____:":
            "____:":
            "____:":
            "____:":
            "____")
```



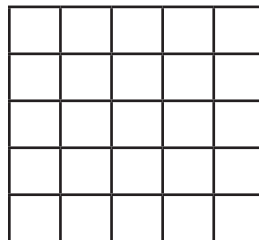
```
three = Image("____:":
              "____:":
              "____:":
              "____:":
              "____")
```



```
four = Image("____:":
             "____:":
             "____:":
             "____:":
             "____")
```



```
five = Image("____:":
             "____:":
             "____:":
             "____:":
             "____")
```



```
six = Image("____:":
            "____:":
            "____:":
            "____:":
            "____")
```

5. Writing the code

Now we need to write the code to react to the shake of the dice and show the spots. We could use a lot of `IF` statements, but instead we're going to use a list to store our images. This is more efficient and requires less typing.

To generate a random number, a library called *random* must be imported

Creates a list called *spots* which your spot graphics are stored. You can add the data in either the long or short format

This line combines different commands to improve efficiency item.

```
from microbit import *
import random

spots = [Image("00000:00000:00900:00000:00000"),
        // Enter your spot data here. The first
        // one has been done for you. Remember
        // to add a comma between each one.
        ]

while True:
    if accelerometer.was_gesture("shake"):
        display.clear()
        sleep(1000)
        display.show(spots[random.randint(0,5)])
```

To call item one in our *spots* list, we would call `spots[1]`. But we can also place our random number command within the brackets instead. This is more efficient than saving our random number in a variable.

The first item in a list is at position zero, so our range will start at zero. Zero to five gives us six possible numbers.