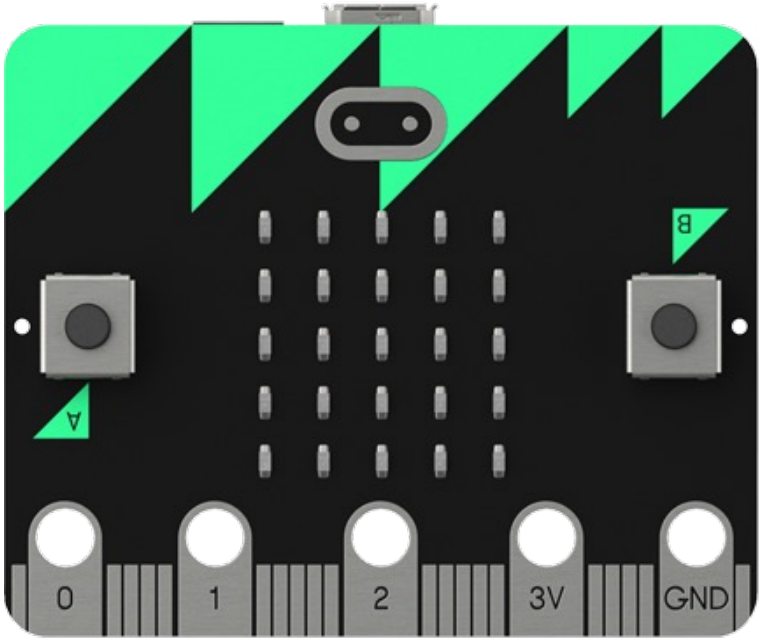
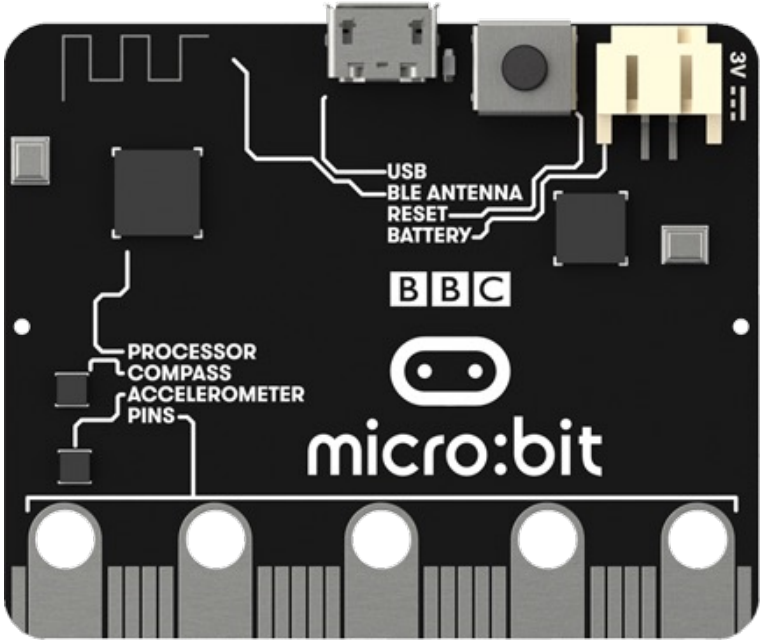
BBC Micro:bit





BBC

micro:bit

Lesson 4

Accelerometer

Gestures

The in built accelerometer can be used to detect movement and direction. This allows the micro:bit to recognise the following gestures.

This example program displays a happy face if the micro:bit is facing up, otherwise it displays a sad face. Try it for yourself.

```
from microbit import *

while True:
    gesture = accelerometer.current_gesture()
    if gesture == "face up":
        display.show(Image.HAPPY)
    else:
        display.show(Image.SAD)
```

```
up
down
left
right
face up
face down
shake
```

Activity 4.1

Create a program that will show different images when different gestures are detected. Use this code as a starting point:

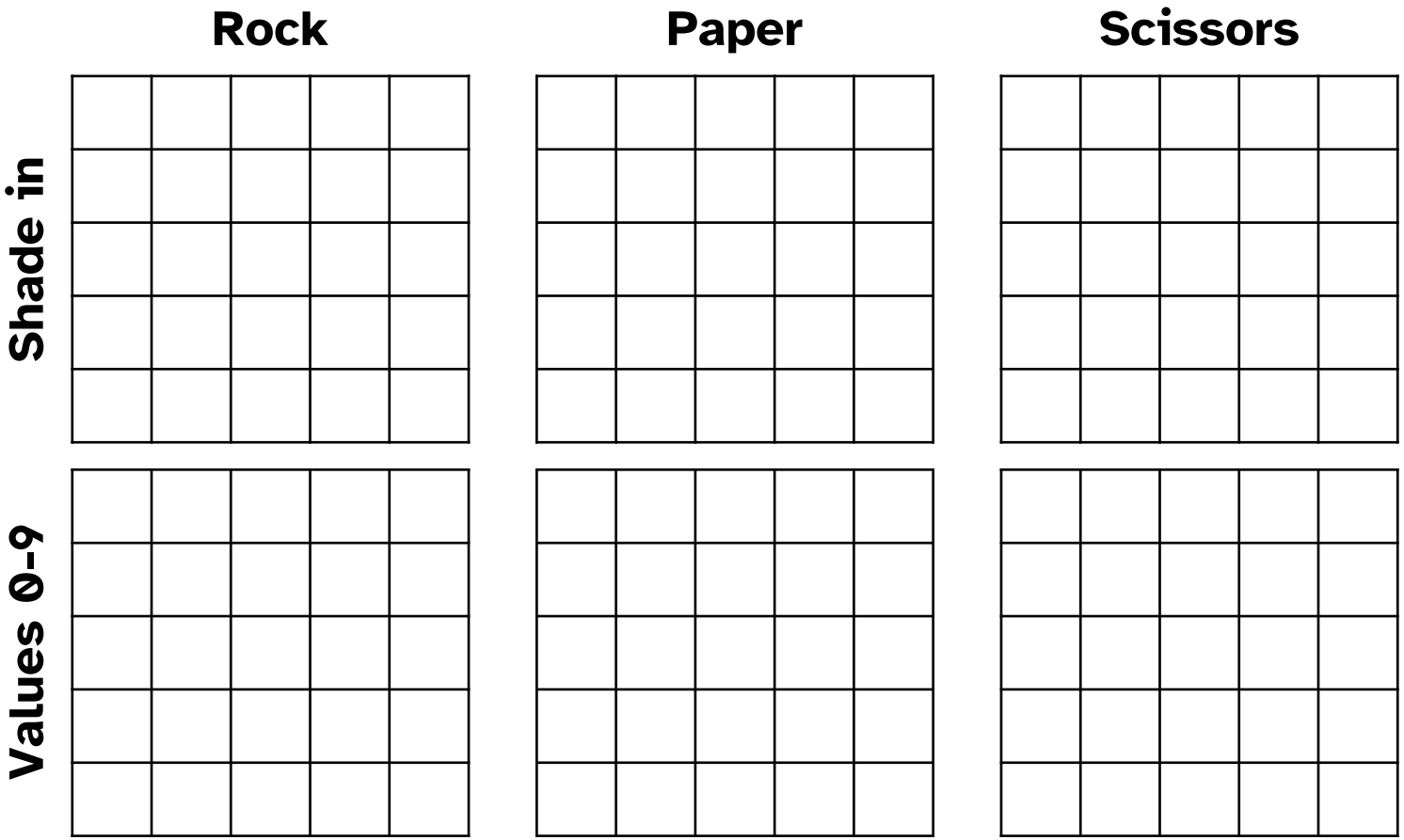
```
from microbit import *

while True:
    gesture = accelerometer.current_gesture()
    if gesture == "left":
        display.show(Image.HAPPY)
    elif gesture == "right":
        display.show(Image.SAD)
```

Place a screenshot of your code here.

Activity 4.2

You are going to create a rock, paper, scissors game using your micro:bit. The first thing you need to do is design the images.



Activity 4.3

This is the incomplete code for the rock, paper, scissors game. You need to replace the comments with appropriate lines of code.

```
from microbit import *
import random

#define your custom rock, paper, scissor images

#create a list containing your images

while True:
    if accelerometer.was_gesture("shake"):
        #display a random image from the list
```

Place a screenshot of your code here.

Accelerometer

The accelerometer can be used to detect precise movement along three axes:

X – tilting from left to right
Y – tilting forwards and backwards
Z – moving up and down

When the reading is 0 you are “level” along that axis.

This program acts like a basic spirit level. Displaying – if the micro:bit is level, < if it is tilted to the left and > if it is titled to the right.

Try it out for yourself.

```
from microbit import *

while True:
    reading = accelerometer.get_x()
    if reading > 20:
        display.show(">")
    elif reading < -20:
        display.show("<")
    else:
        display.show("-")
```

Running Time

The micro:bit keeps track of the amount of time it has been running.

This example program has a while loop that will keep running until button a is pressed. The running time is then calculated in seconds and displayed on the screen.

The break command is used to break out of a loop.

Try it out for yourself.

```
from microbit import *
import math

while True:
    if button_a.is_pressed():
        time=round(running_time()/1000)
        break

display.scroll(str(time))
```

Activity 4.4

This is the incomplete code for a game. The aim is to hold the micro:bit level for as long as possible. It's incomplete, you need to replace the comments with appropriate lines of code.

```
from microbit import *
import math

#show a happy face

while True:
    reading = accelerometer.get_x()
    if reading > 45 or reading < -45:
        time=round(running_time()/1000)
        #show a sad face
        break

#pause for 2 seconds
display.scroll(str(time))
```

Place a screenshot of your code here.

Extension

Change the sensitivity of the game to make it harder.

Place a screenshot of your code here.