

Coding Python with the BBC micro:bit

Learning Concepts:

- Import code libraries
- Variables
- Loops
- Images
- If/Else
- Comments/Documentation

Preparation:

- Open a web browser and go to <https://create.withcode.uk/>
- Explore the default code and run it as an example
- For this workshop, we will be coding in Python inside of this coding environment, which includes a micro:bit simulator so you can preview your code while you're working.

Project:

Using the Python programming language and the micro:bit library functionality, code a Rock Paper Scissors game. Using the normal rules of the game, play against another participant in the workshop:

Rock breaks Scissors – Scissors cut Paper – Paper covers Rock

1. Import the needed libraries – these contain pre-written code that adds functionality to your program. All of the needed micro:bit functions that will be used in this workshop will be included.

```
1 from microbit import *  
2 import random
```

2. Create the LED images – the micro:bit includes a 5 x 5 grid of LED lights that can be programmed. The brightness of each LED is on a scale of 0 (zero is off) to 9 (nine is full brightness). Using variables named rock, paper and scissors, create three different images:

```
4 rock = Image("00900:"  
5              "09990:"  
6              "99999:"  
7              "09990:"  
8              "00900:")  
9 paper = Image("99999:"  
10              "90009:"  
11              "90009:"  
12              "90009:"  
13              "99999:")  
14 scissors = Image("99009:"  
15                  "99090:"  
16                  "00900:"  
17                  "99090:"  
18                  "99009:")  
19
```

You can use the paper worksheet to plot out how the image will look in advance.

- Using the variables you just created, now code the opening sequence that will play on the micro:bit when it is powered up. Notice the `display.scroll()` and `display.show()` functions from the micro:bit library. To allow a short delay between each item, the `sleep()` function pauses the program with the number of milliseconds defined (1000 milliseconds equals 1 second).

```
20 display.scroll("ROCK")
21 display.show(rock)
22 sleep(500)
23
24 display.scroll("PAPER")
25 display.show(paper)
26 sleep(500)
27
28 display.scroll("SCISSORS")
29 display.show(scissors)
30 sleep(500)
31
32 display.scroll("SHAKE TO PLAY!")
33 sleep(500)
34
```

- Next, you are going to create a function to provide the game logic. Using a function allows you to reuse this block of code (you'll see later how this comes in handy). In Python, a block of code is defined using indentation. The `def` keyword will define your function and then using a colon (:), tell Python that the following indented block of codes belongs to this function. Notice how the `randint()` function is being used from the random library imported at the beginning.

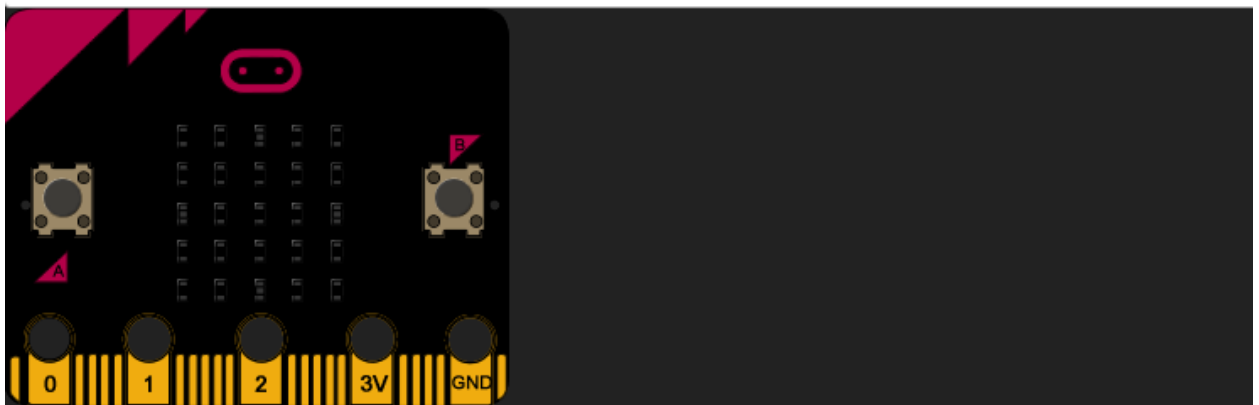
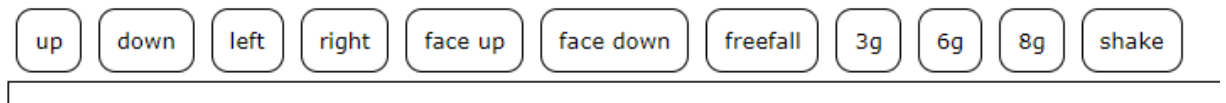
```
35 def gameAction():
36     playItem = random.randint(1,3)
37     if playItem == 1:
38         display.show(rock)
39     if playItem == 2:
40         display.show(paper)
41     if playItem == 3:
42         display.show(scissors)
43
```

5. Create the loop that uses the game logic – while the micro:bit is powered up, you want the game play to activate when the micro:bit sensors detect it has been shaken. The while True statement will run continuously until a condition becomes False (in this case, it never does thus it's an endless loop). Use the accelerometer.was_gesture() function with the 'shake' value, then call the gameAction() you just created:

```
44 while True:
45     if accelerometer.was_gesture('shake'):
46         gameAction()
47
```

6. Run the game in the simulator – click on the + symbol to show the tool buttons. Press the green Play button to open the simulator. Once the opening text has scrolled, click on the Accelerometer tab to access the “Shake” button.

Gestures



7. **Stretch Goal** – Using an *else/if* statement, add the ability to also press the B button to play the game using the *button_b.is_pressed()* function.

```
44 while True:
45     if accelerometer.was_gesture('shake'):
46         gameAction()
47     elif button_b.is_pressed():
48         gameAction()
49
```

Notice how you are able to re-use your *gameAction()* function to duplicate the logic without having to type any additional code!

8. Try the game in the simulator. If working as expected, download the HEX file and load it onto the micro:bit.

Transfer Your Code:

- Plug in your micro:bit using the supplied USB cable. The micro:bit will appear as a drive (either an icon on the desktop when using a Mac or Linux or as a drive letter in Windows).
- Drag the HEX file downloaded in step 13 to the micro:bit – the code will automatically load and run.
- Detach the micro:bit from the USB cable (follow the proper procedure for disconnecting an external drive from your computer). Use a battery pack to power the micro:bit. Challenge one of the other participants to a game of Rock – Paper – Scissors. Have fun!

GNU General Public License v3.0

Permissions of this strong copyleft license are conditioned on making available complete source code of licensed works and modifications, which include larger works using a licensed work, under the same license. Copyright and license notices must be preserved. Contributors provide an express grant of patent rights.