

Section 6 Challenges

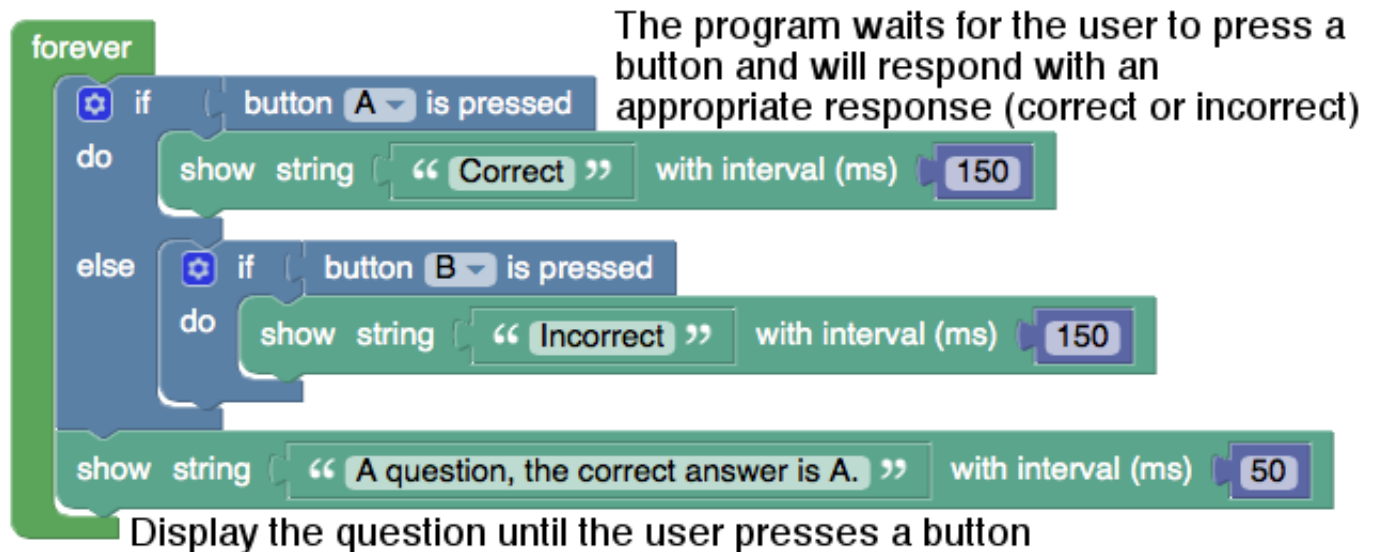
Before we get started

This section details challenges for the **BBC micro:bit** (<https://www.microbit.co.uk/device>). Ensure you have completed all other sections of the **Microsoft Block Editor** (<https://www.microbit.co.uk/blocks/contents>) tutorials before attempting these challenges!

Quiz Challenge [1]

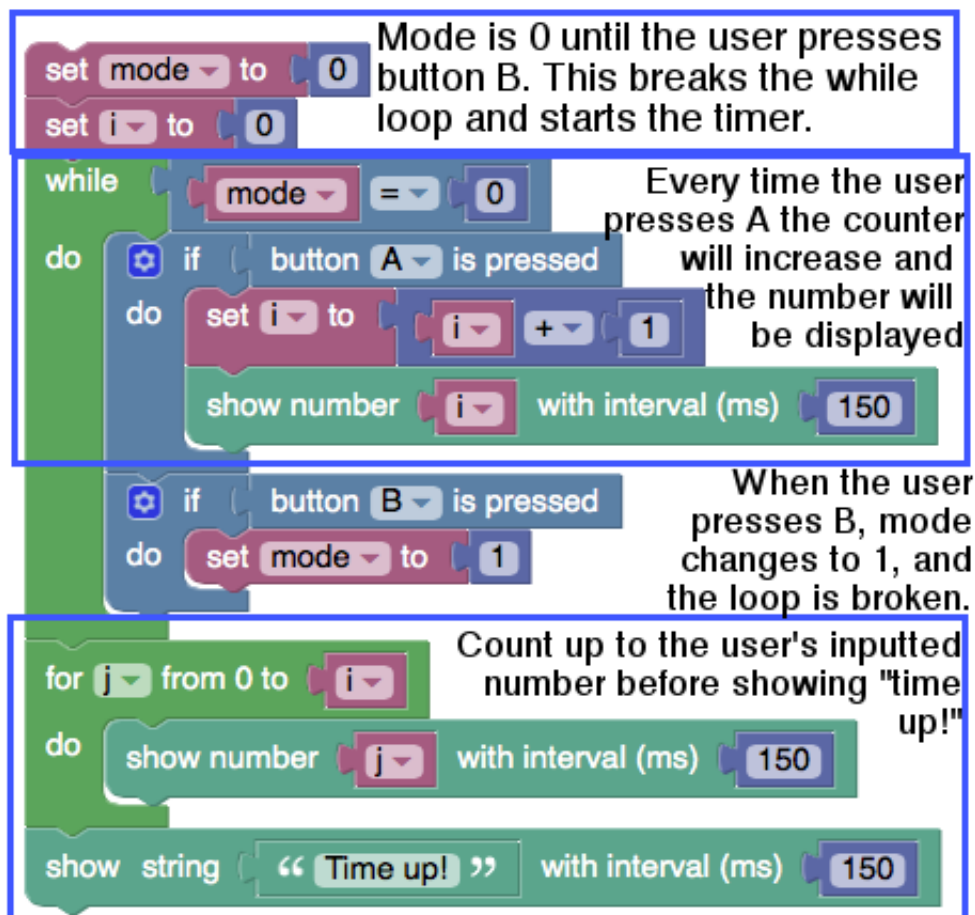
Using **if** (<https://www.microbit.co.uk/td/if>) statements, try to add more statements to create a simple quiz. The user will be told if the question is right or not, and will have two options (button A and button B).

Here is some sample code for a simple quiz:



Timer Challenge [2]

Create a timer that runs out after a certain amount of time (using the count **loop** (<https://www.microbit.co.uk/td/for>)). For an extra challenge, let the user *input* the amount of milliseconds they want the timer to run for using **variables** (<https://www.microbit.co.uk/td/var>) and the **buttons** (<https://www.microbit.co.uk/functions/button-is-pressed>) as *input*. The solution is below.



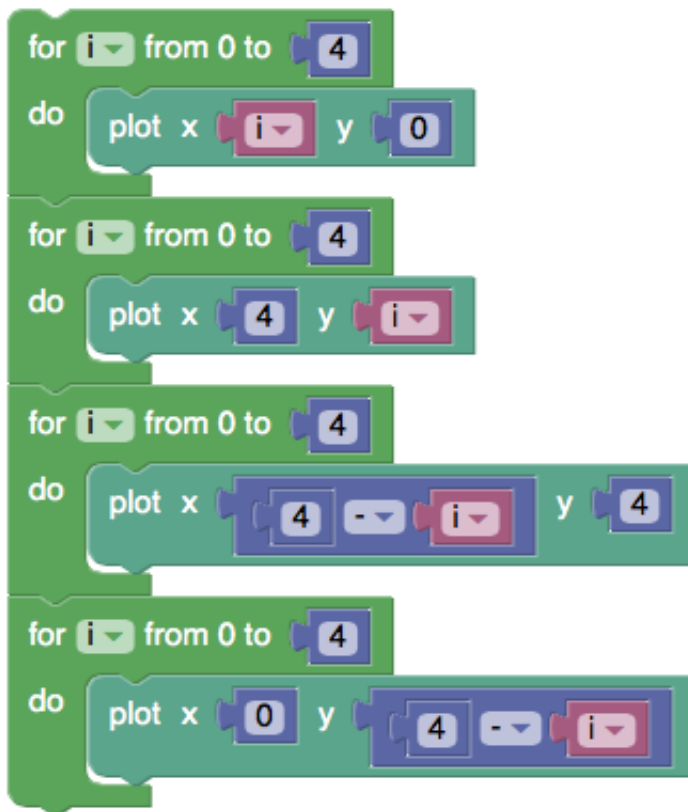
Graphics Challenges [3]

Using the knowledge you have learnt from the **rendering graphics** (<https://www.microbit.co.uk/blocks/book/graphics>) section, try creating an algorithm to draw these shapes. Before you write the code try to figure out how the BBC micro:bit will be thinking to **plot** (<https://www.microbit.co.uk/functions/plot>) these points. For example, with our diagonal line – “count up from 0 to 4 by 1, and plot points $x=i$ and $y=i$ ”.

- Another diagonal line
- A square going around the board
- A filled square
- A square which unplots itself after
- A filled square which then unplots itself

The solutions are below.

Square [3.1]



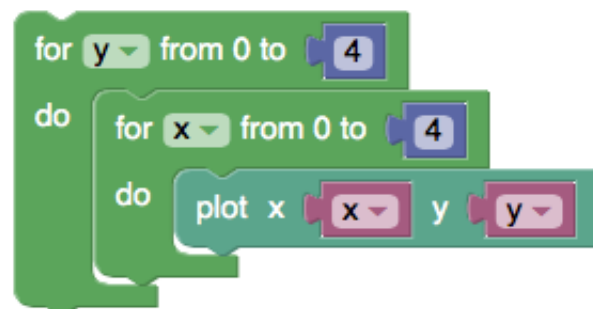
The algorithm:

- Count up from 0 to 4 and plot this along x (to draw a horizontal line at $y = 0$)
- Count up from 0 to 4 and plot this along y (to draw a vertical line at $x = 4$)
- Count up from 0 to 4 and subtract this number from 4 to draw a horizontal line backwards at $y = 4$). This can also be done forwards.
- Count up from 0 to 4 and plot a vertical line at $x = 0$

Filled square [3.2]



This algorithm plots points across x from 0 to 4, and repeats this for all 5 y-values to fill the screen.



This can also be simplified to the algorithm above; for y values from 0 to 4, plot all x values from 0 to 4. This is an example of a nested loop.

Vanishing square [3.3]

Use the same code and algorithm for the square solution, only use the **unplot** (<https://www.microbit.co.uk/functions/unplot>) block to make this LED turn off again. You could also reverse the algorithm.

Vanishing filled square [3.4]

Use the same code and algorithm for the filled square solution, only use the **unplot** (<https://www.microbit.co.uk/functions/unplot>) block to make this LED turn off again. You could also reverse the algorithm.

Animation Challenge [4]

Use your new knowledge of animations and algorithms to program your BBC micro:bit to act human: for example, you could make your BBC micro:bit smile and wink. Remember you can display images with the **show image** (<https://www.microbit.co.uk/functions/show-image>) and **create image** (<https://www.microbit.co.uk/functions/create-image>) blocks. Sample code is below.



Electronic Dice Challenge [5]

Using the code in the Random Numbers tutorial in Section 6, or your own algorithm, create an electronic dice that displays the values appropriate for a dice (so 1 shows a single LED on in the center, two shows two LEDs on at each corner, etc.). You may want to declare image variables to do this, then check what it is equal to using an 'if' statement. Sample code is below.



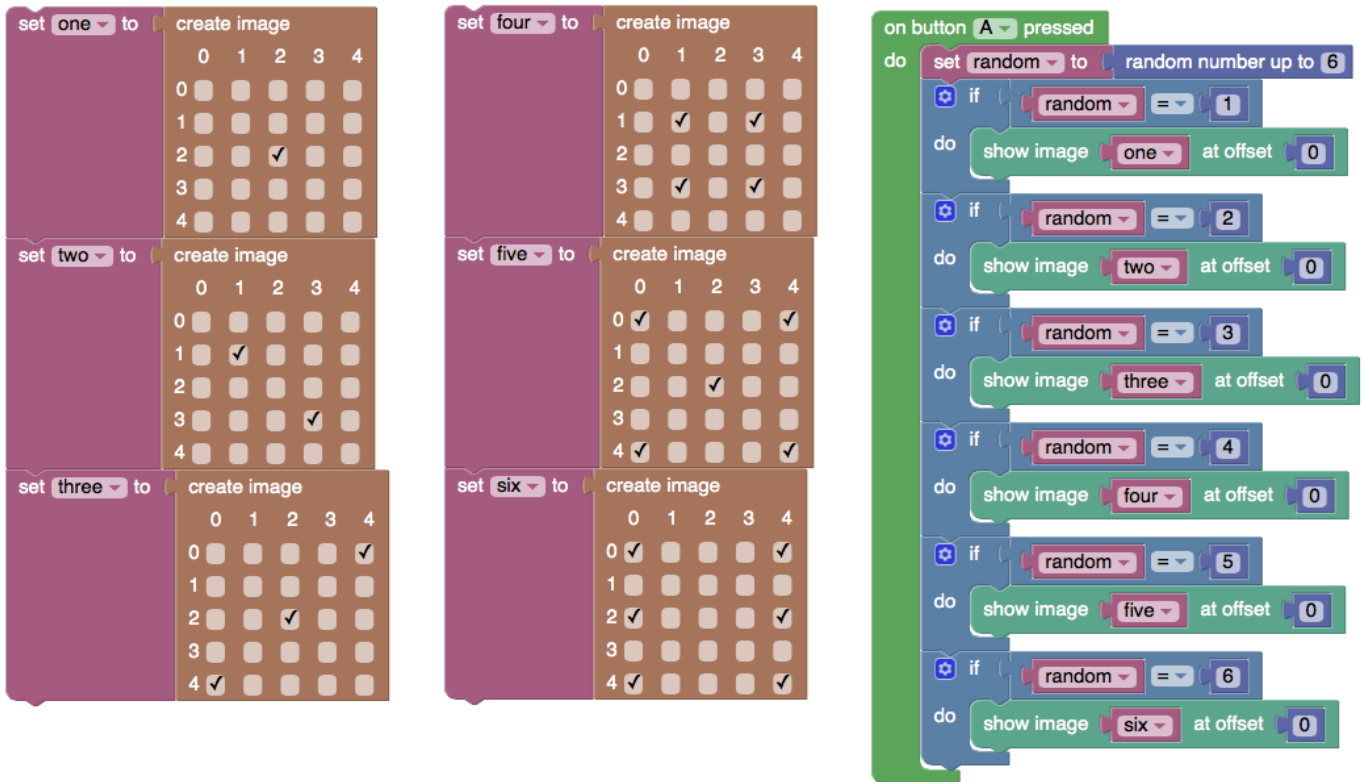
Calculator Challenge [6]

Using your knowledge of loops, counters and math, create a calculator.

The calculator should:

- Count the amount of times the user presses the left button before pressing the right button (this is the first value, or valueOne)
- Count the amount of times the user presses the left button before the right button again (this is the second value, or valueTwo)
- Scroll through operations (+, -, x and divide) until the user presses the right button to make a choice
- Perform the calculation
- Show the entire calculation, for example: 5 + 10 = 15

Sample code is below.



Smart watch Challenge [8]

Create a smart watch using the BBC micro:bit. Create a menu where the user presses one button to cycle through options and another button to choose this option. Add applications to this smart watch:

- Calculators
- Games
- Random number generators

And any other applications you can think of.

Where next?

Section 5 Rendering Graphics

(<https://www.microbit.co.uk/blocks/book/graphics>)

Section 6 Challenges

(<https://www.microbit.co.uk/blocks/book/challenges>)

Table of Contents (<https://www.microbit.co.uk/blocks/book>)

