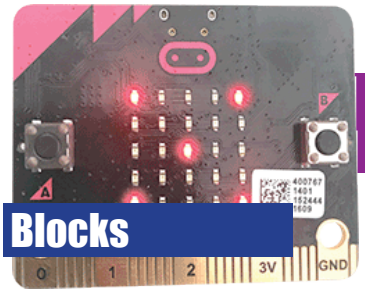
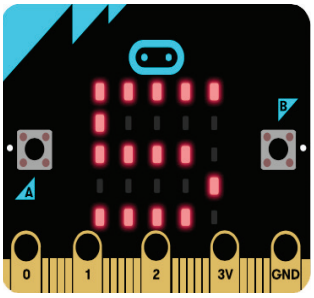


Guessing Game 03

- The Micro:bit will choose a random number
- Can you guess it?

10 mins

1. Objective

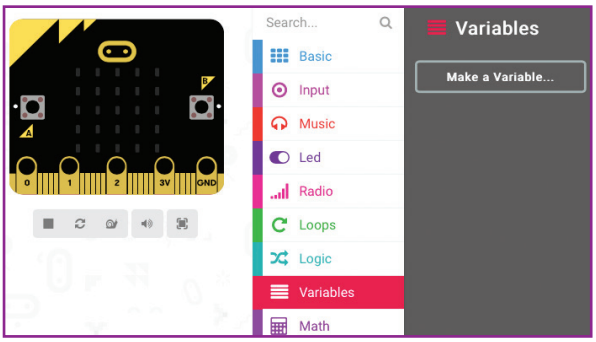


The Micro:bit will choose a random number, between 0 and 10. Will you be able to guess it?

The Micro:bit doesn't have a keyboard, so how will we be able to enter our guesses?

Hint: it may not have a keyboard, but it does have `A` and `B` buttons.

2. Storing data

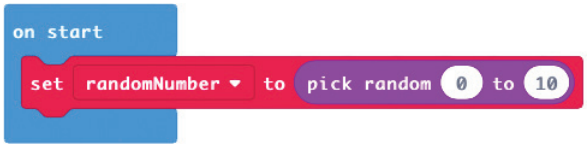


Computers need an area of memory to store numbers and other data. We give each of these areas a name so that we can refer back to them. They are called *variables*. We need a variable in which to store the number which the Micro:bit will choose.

Lets make our first variable!

1. Click on the red variables tab
2. Click the Make a Variable... button
3. A text box will appear. Give the variable a meaningful name (e.g. randomNumber). Note that variables typically cannot contain spaces or punctuation, nor must they start with a number.

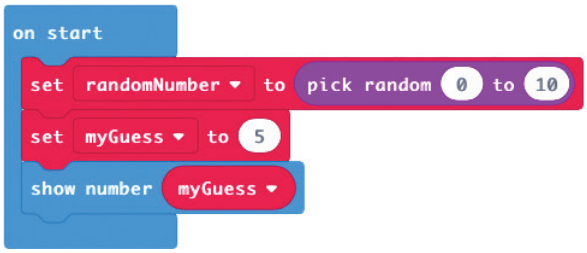
3. Choosing a random number



Now we have a variable, the Micro:bit can choose a random number and *assign* it to the variable.

1. New blocks will have appeared in the variables tab. Drag the `set randomNumber to 0` block into the `on start` block.
2. Now, we need to generate a random number. This block is located in the *math* tab. Drag it into the previous block, as shown
3. Now when the Micro:bit starts, a random number will be chosen and assigned to the *randomNumber* variable
4. Nothing is being displayed in the virtual Micro:bit. Can you work out why?

4. Storing our guess



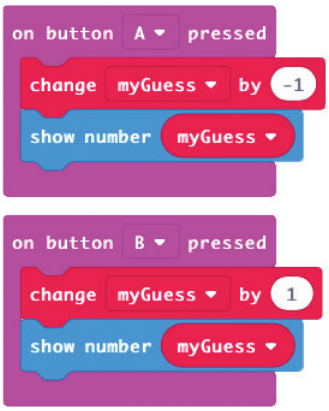
Did you manage to solve the earlier problem of how to input our guess?

There is more than one solution, but one way is to use the A button to decrease our guess number; B button to increase our guess number; and both buttons to submit our guess.

If you have thought of a different way, try to do it your way instead.

1. Create a second variable in which to store your guess. In the example, it's called `myGuess`
2. Add set `myGuess to 5` into the `on start` block
3. We also need to display what our current choice is. Add the `show number` block, and drag the `myGuess` variable into it. This will show whatever number is stored in the variable

5. Changing our guess number



Our default guess number is 5. We need to be able to change that to a number of our choosing.

Lets add the code which will allow us to reduce this number.

1. Add the blocks as shown. Each time the A button is pressed, the number stored in `myGuess` is reduced by 1, then shown on the LEDs.
2. Add another set of blocks to control what happens when button B is pressed. Can you see what the difference between them is?

6. Is our guess correct?

The Micro:bit needs to compare what is stored in each variable. If both variables store the same number, display "Correct. Well done!", or else print "Wrong. Try again".

1. The `if` block is stored in the *Logic* tab
2. Add the `equals` comparison, also stored in the *Logic* tab
3. Press the *plus* icon to add an `else` statement
4. Add the messages as shown
5. Test your program. Can you guess the number?

7. Challenge activities

- Modify your program to count how many attempts you take to get the correct answer
- Limit the number of guesses to three
- You can currently submit numbers outside the guessing range? How can you prevent this?