

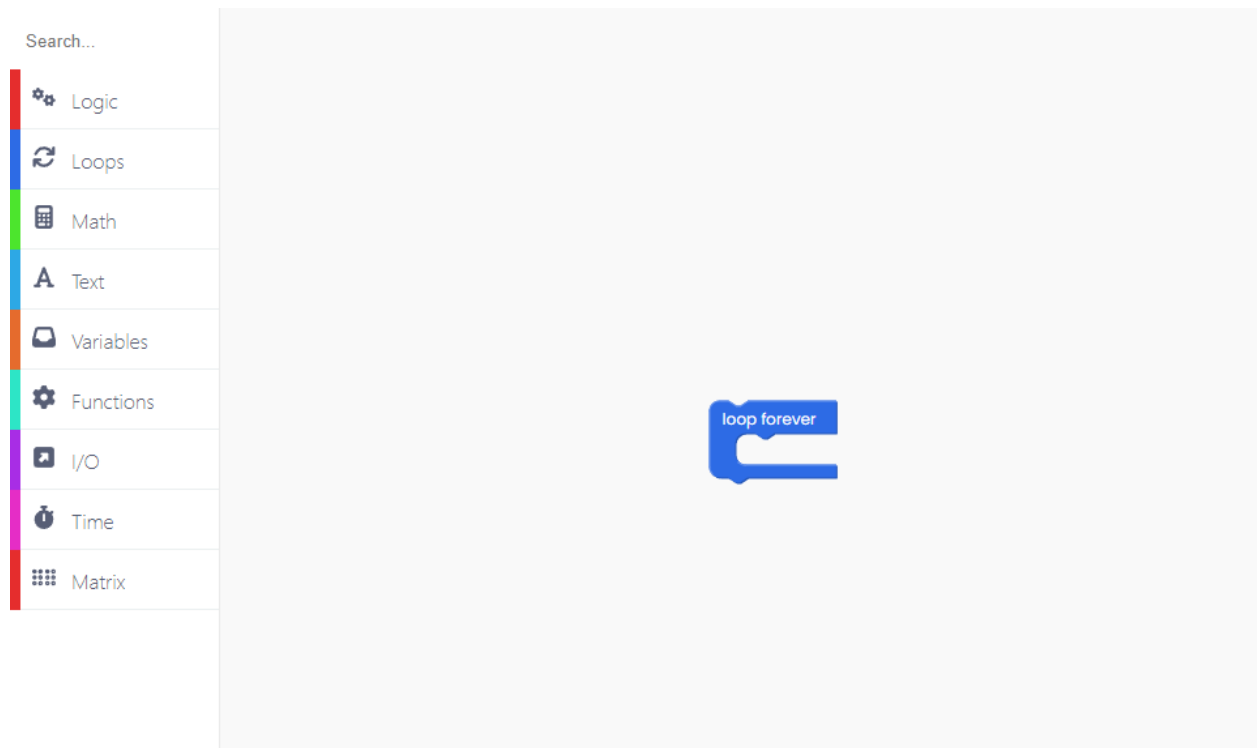
Coding for beginners - how to code your Spencer

Simple timer

We'll start with something simple.

We'll use all of the LEDs on Spencer and make them light up, and turn off in a loop every second.

To begin, drag the "loop forever" block from the "Loops" block section onto the drawing area.

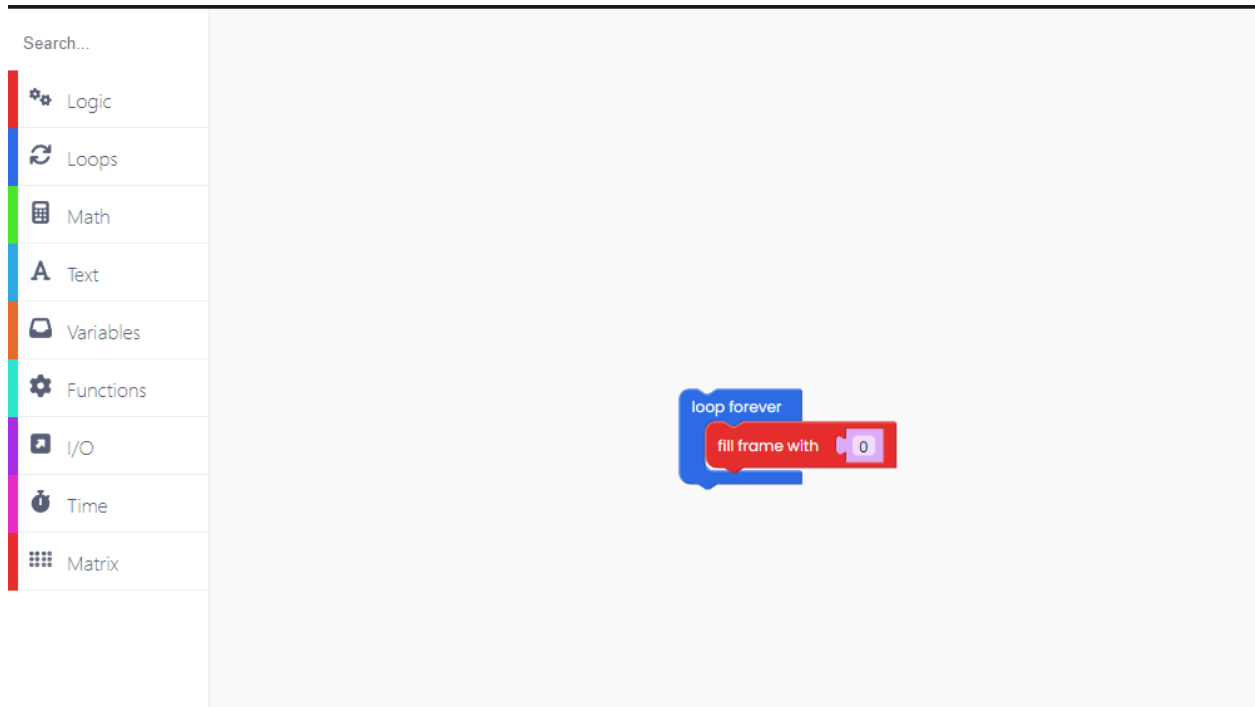


This block will ensure that the code inside it is executed all the time in loops.

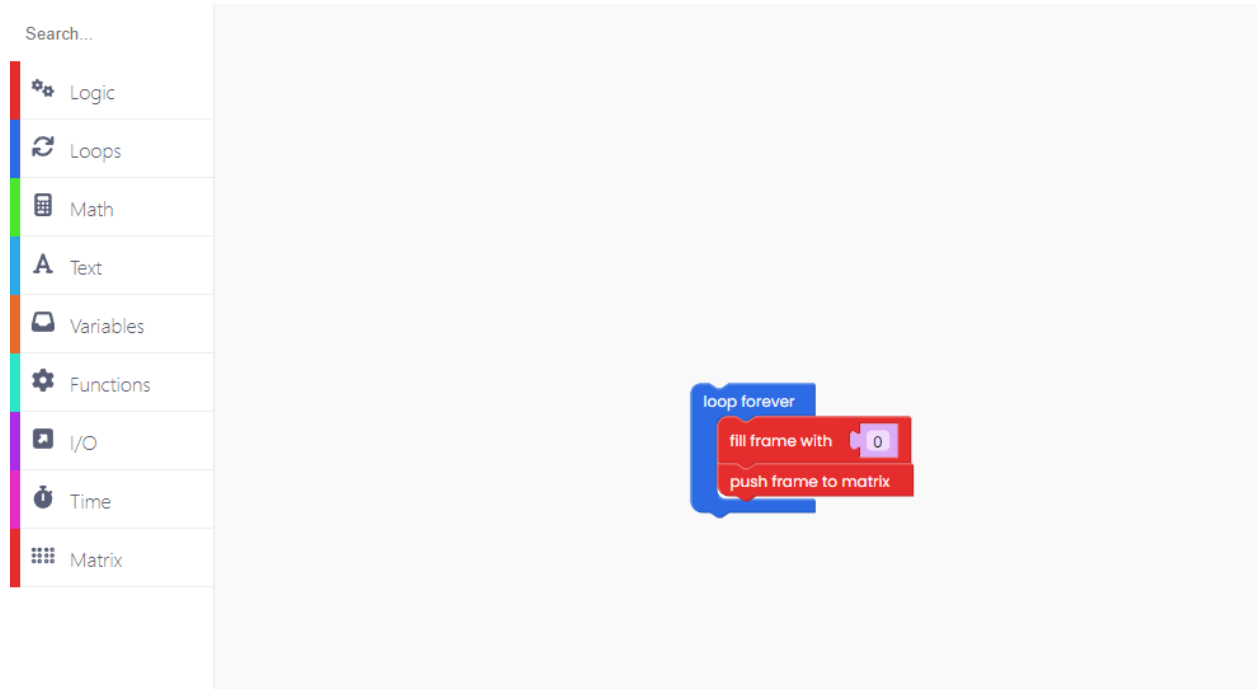
Firstly, we want all of the LEDs to turn off, and for that, we'll need the "fill frame with 0" block

from the "Matrix" block section.

Don't change the 0 for now.

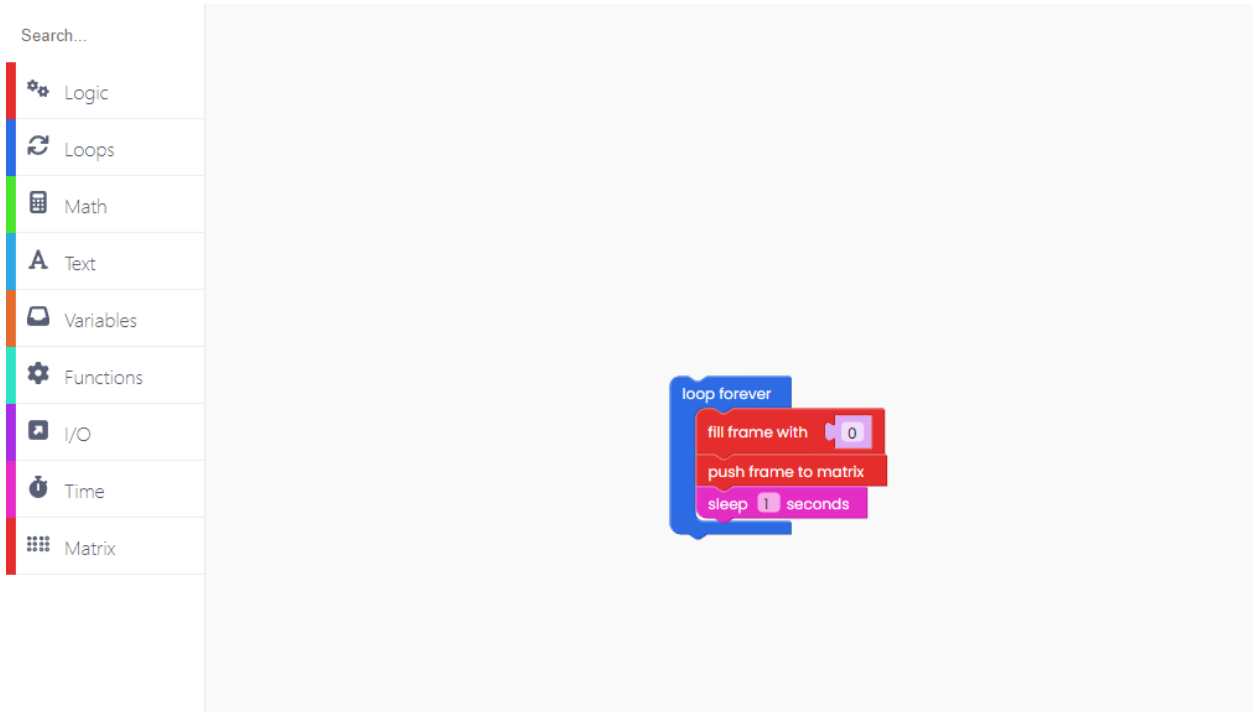


To run the code on the matrix, you must always include the "push frame to matrix" section; otherwise, the code will fail.



Because the following step will be to turn on the LEDs, we must include a one-second delay between these two activities.

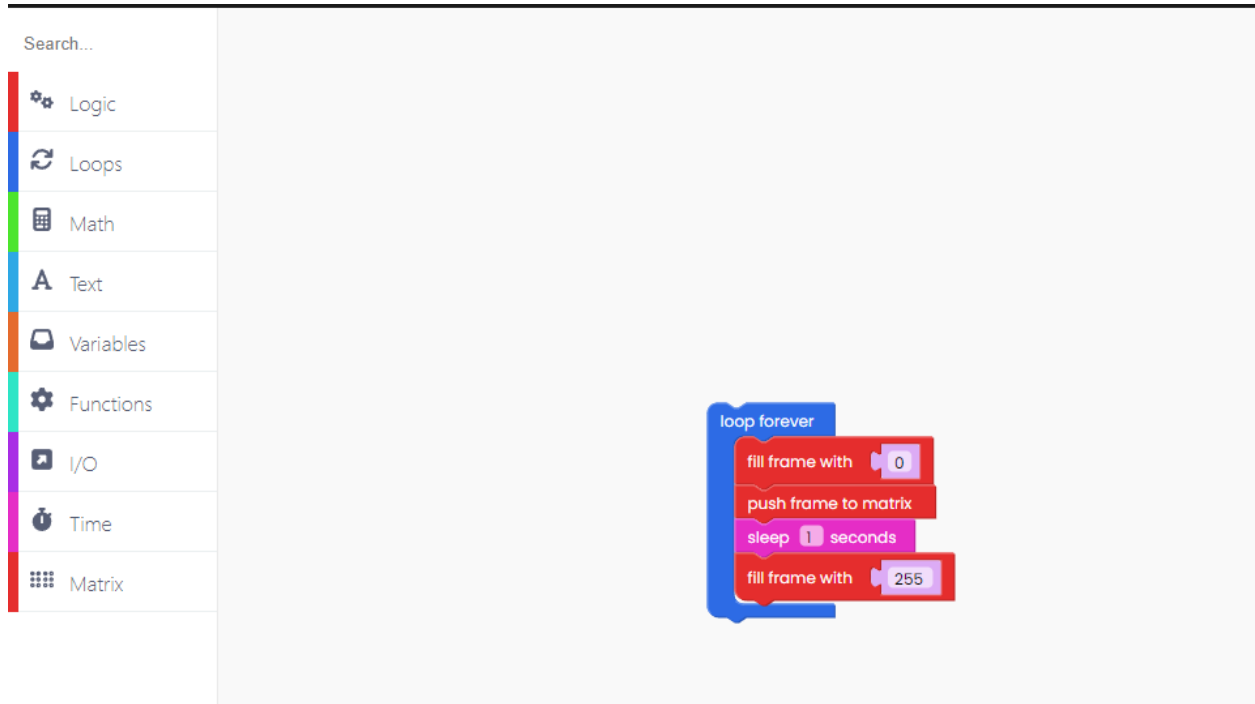
Go to the "Time" block section, and you'll find this block:



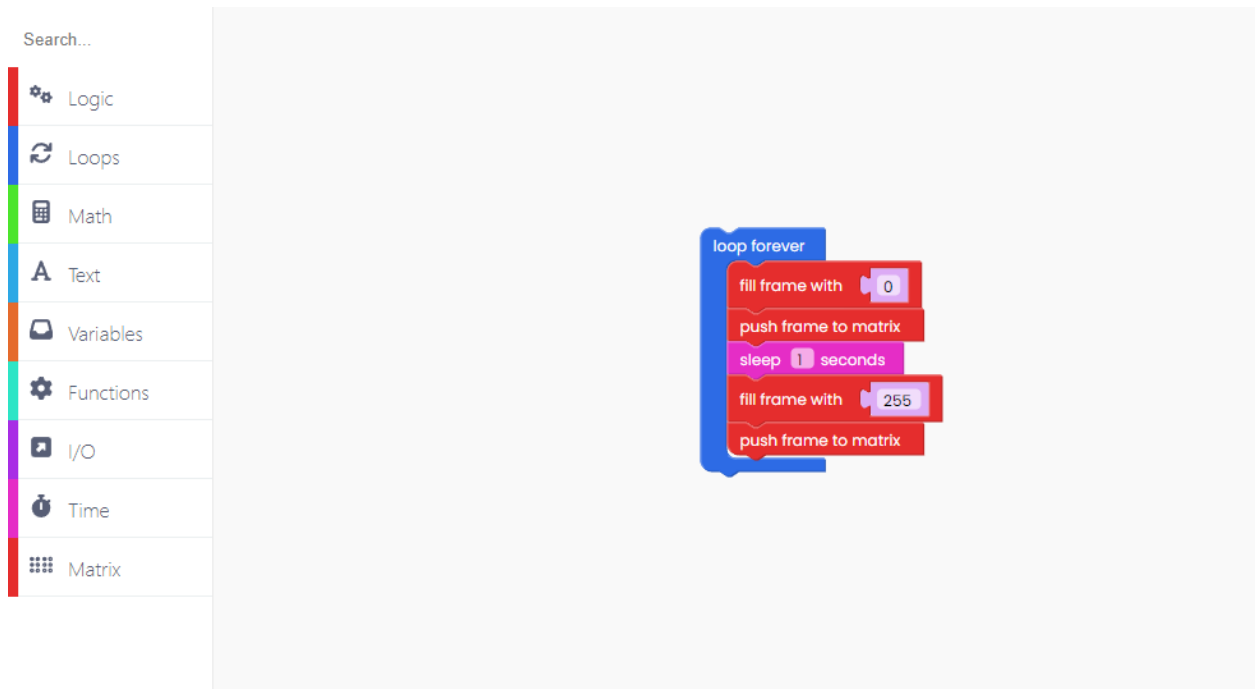
It's time to turn on the matrix.

To do that, duplicate the "fill frame with 0" block.

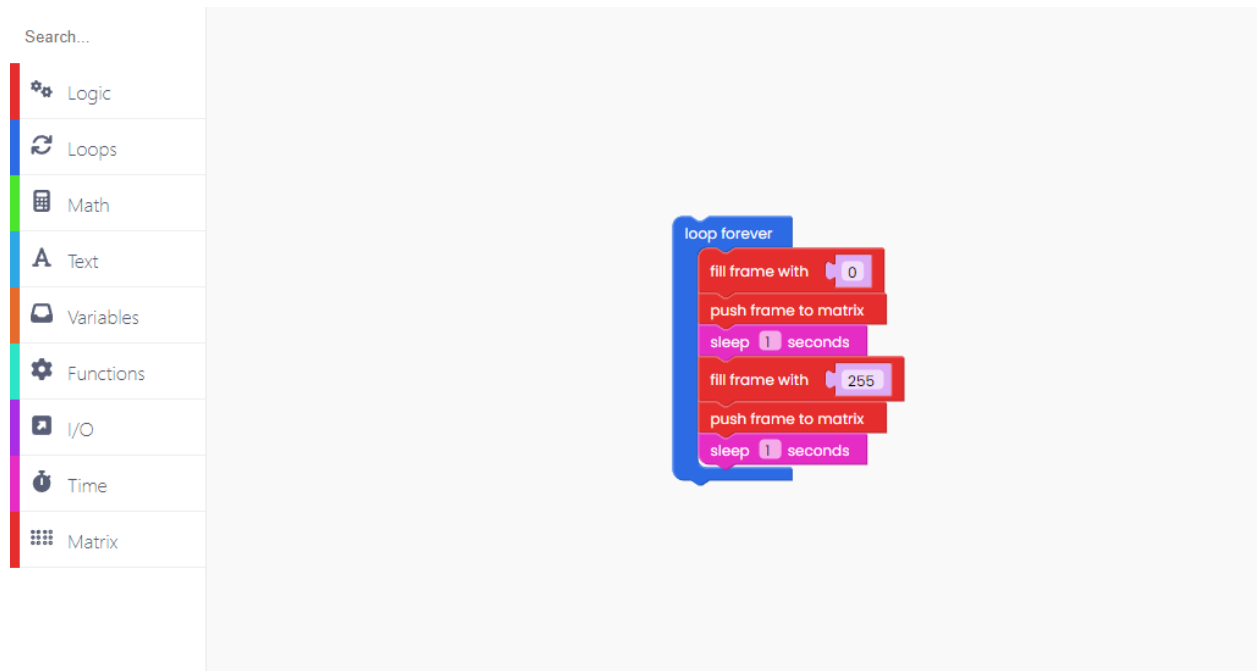
This time, we'll change the 0 to 255.



To ensure that the code runs properly, add another "push frame to matrix" block.



Finally, add the "sleep 1 second" block at the end because that is the amount of time we want to pass before the code runs in loops over and again.



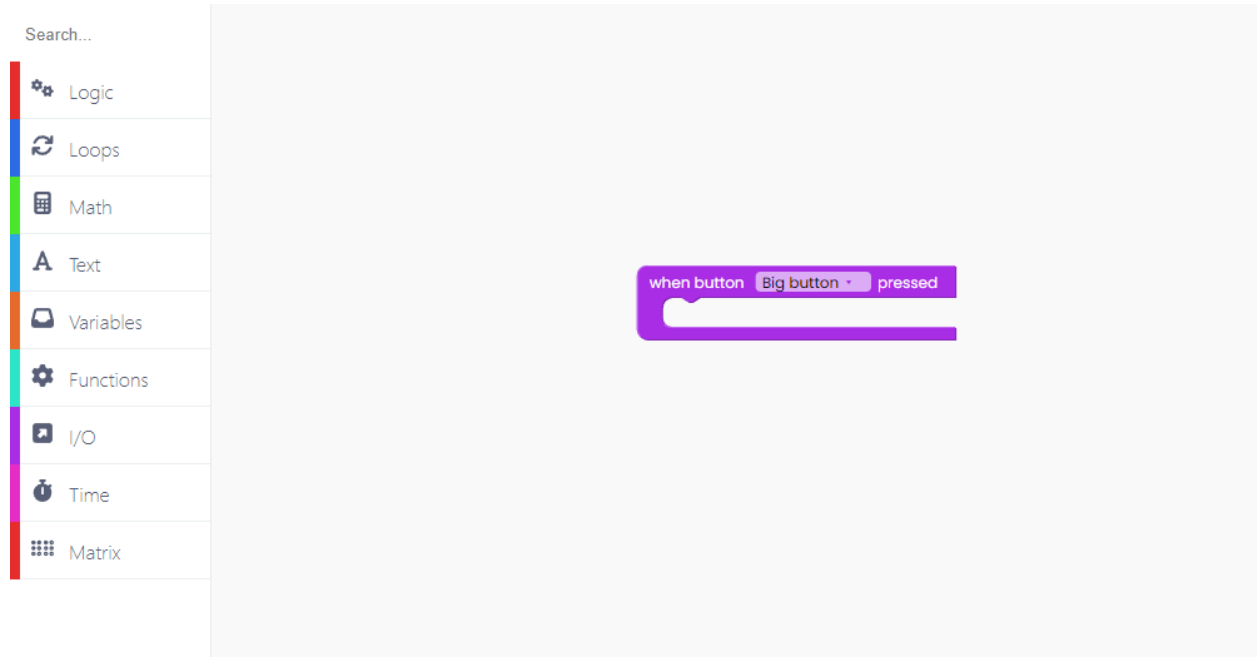
Click on the Run button, and check the code!

Play with the red button!

This time, we'll do something similar to the Simple timer example. The code, on the other hand, will not execute in the loops but will begin when you click on the big red button.

If you wish to code something with the buttons, head to the I/O block section.

To begin, drag this "When big button pressed" block onto the drawing area.

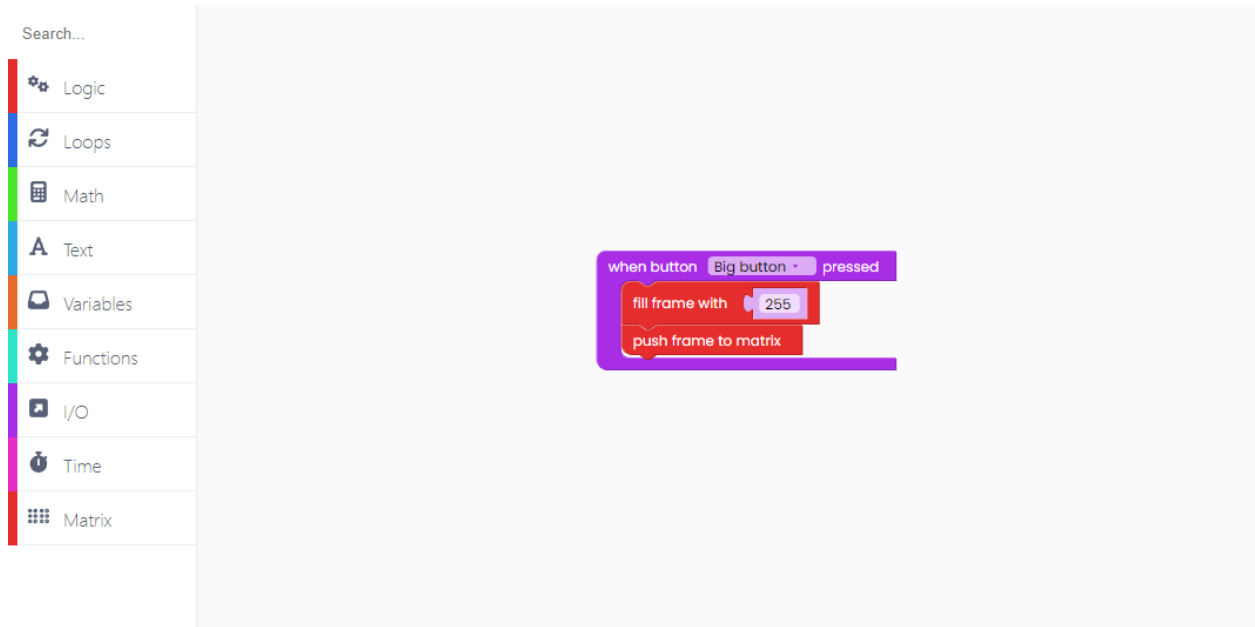


Inside this I/O block will be the code for what should happen once you hit the button.

We want the matrix to light up when we press the button and switch off when we release it.

Change the 0 in the "fill frame with 0" block to 255 to ensure it lights up.

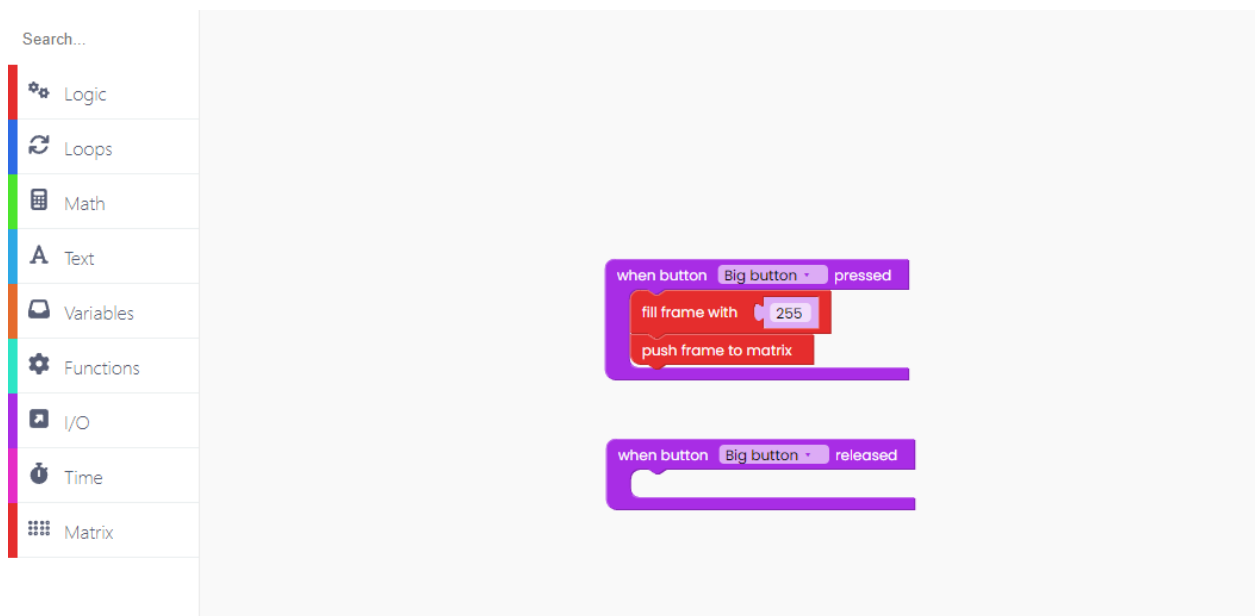
Also, don't forget the "push frame to matrix" so your code is executed properly.



Now, let's code what'll happen once we release the button.

Return to the I/O block section and look for the "when big button released" block.

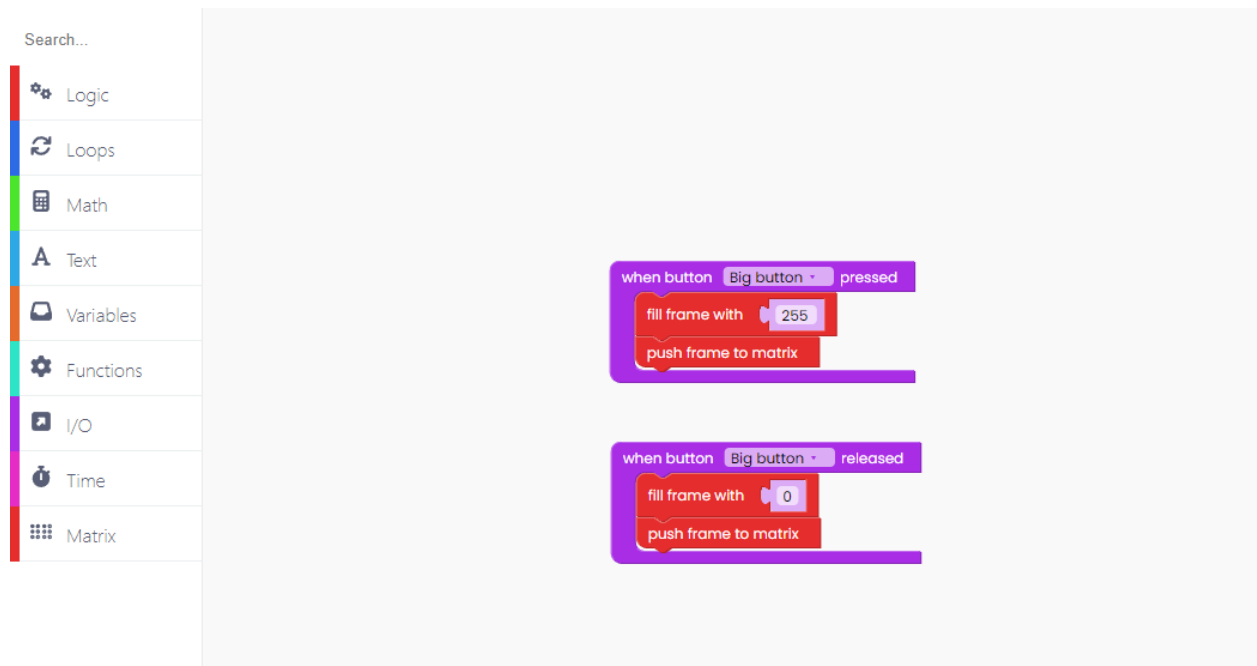
Drag and drop it onto the drawing area.



You can duplicate the same "matrix" blocks inside it.

This time, make sure to replace 255 to 0.

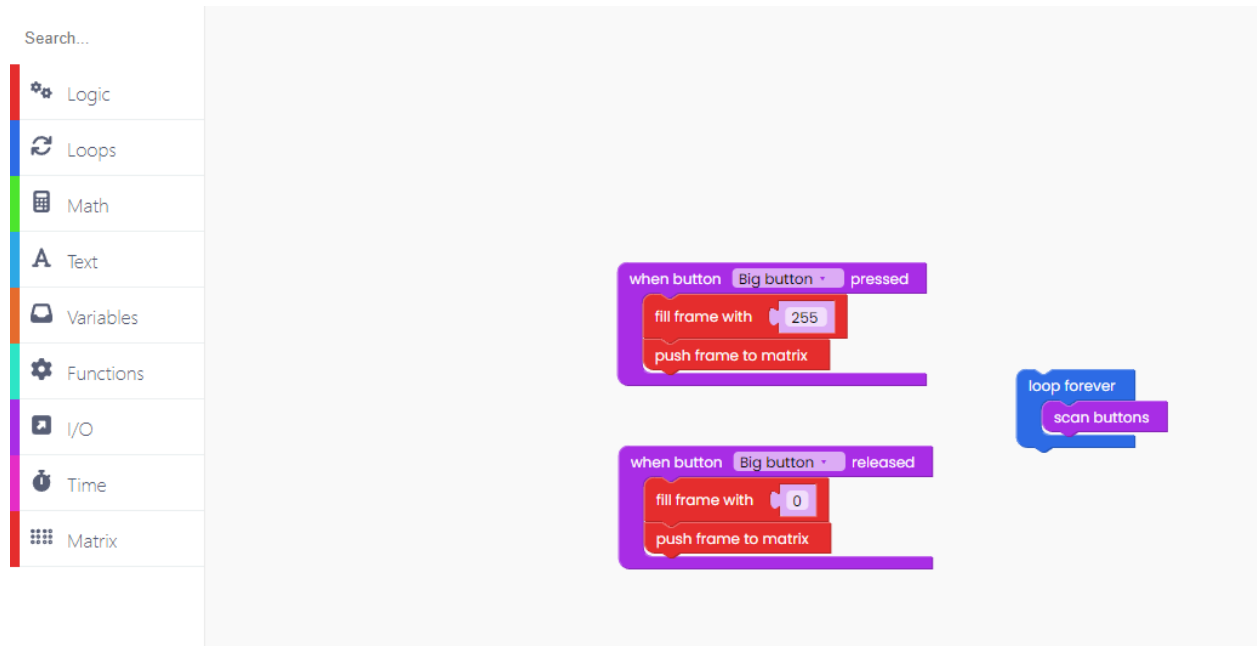
Here's how your code should look:



Your code is almost finished!

The only thing that's missing is the "scan buttons" block inside the "loop forever" block.

It is critical to include these two blocks whenever you are coding anything with buttons; else, your code will fail.



Click on the Run button, and start playing with the big red button.

Write something!

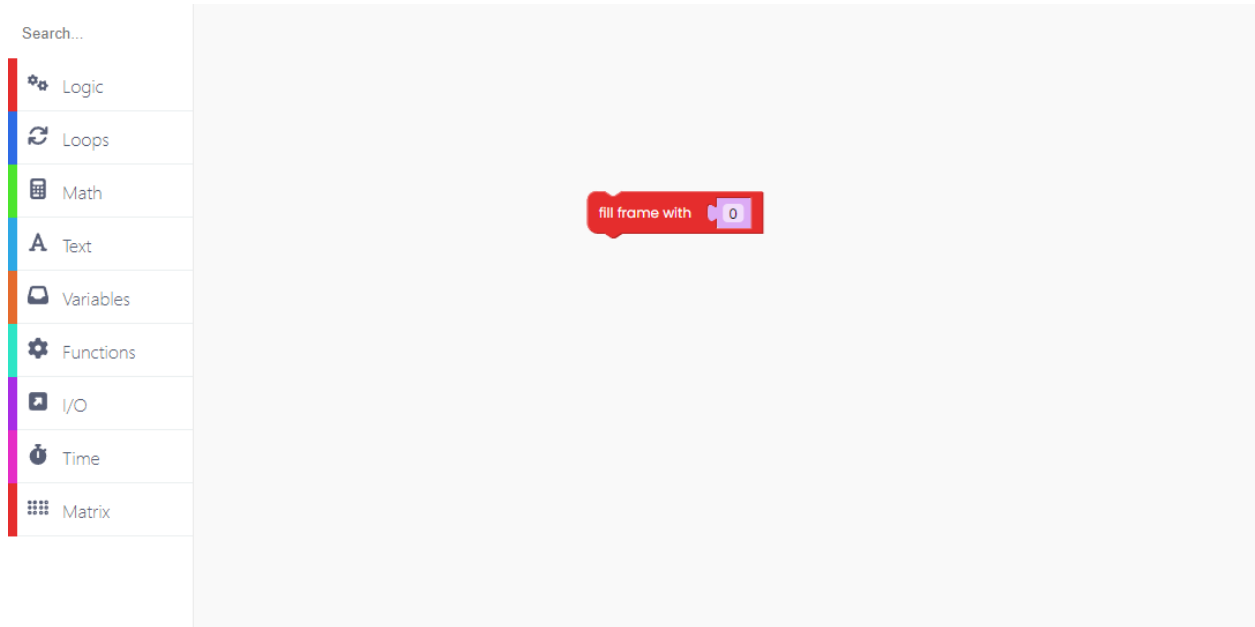
You can write any kind of short message on Spencer's matrix board.

We'll try that out right now.

The goal is to write "ON" when you press the big red button and "OFF" when you release it.

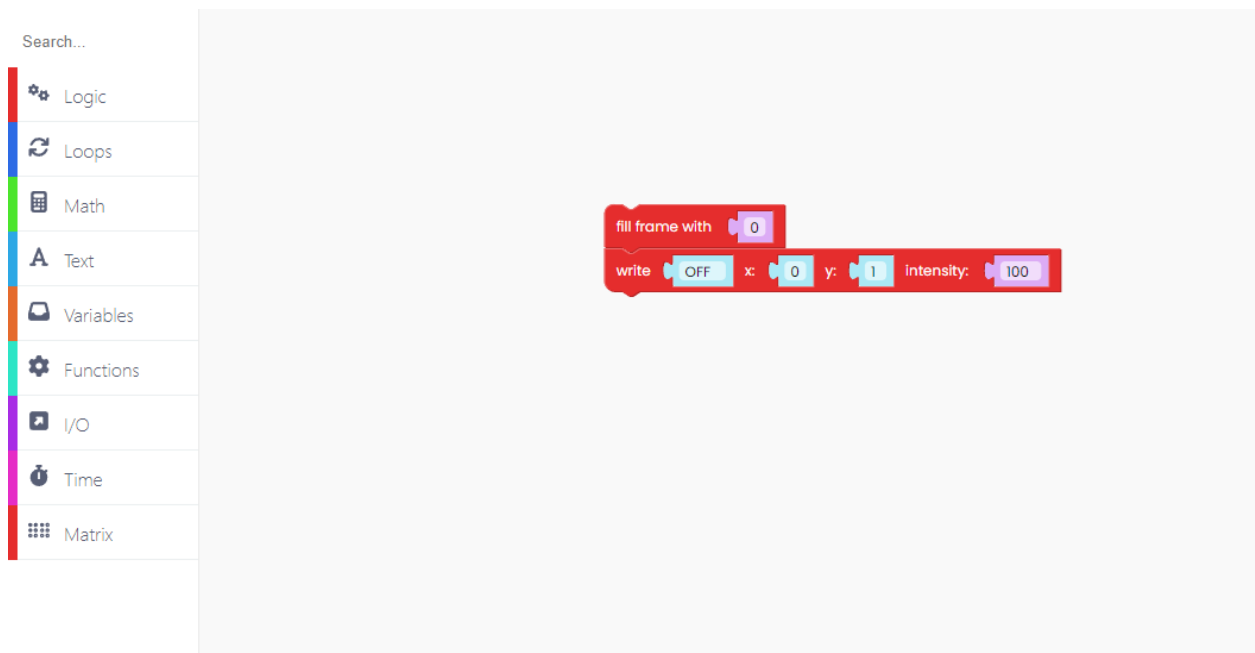
Let's first code what'll happen once you run the code.

We want all of the LEDs on the matrix to turn off, and for that we need the "fill frame with 0" block.



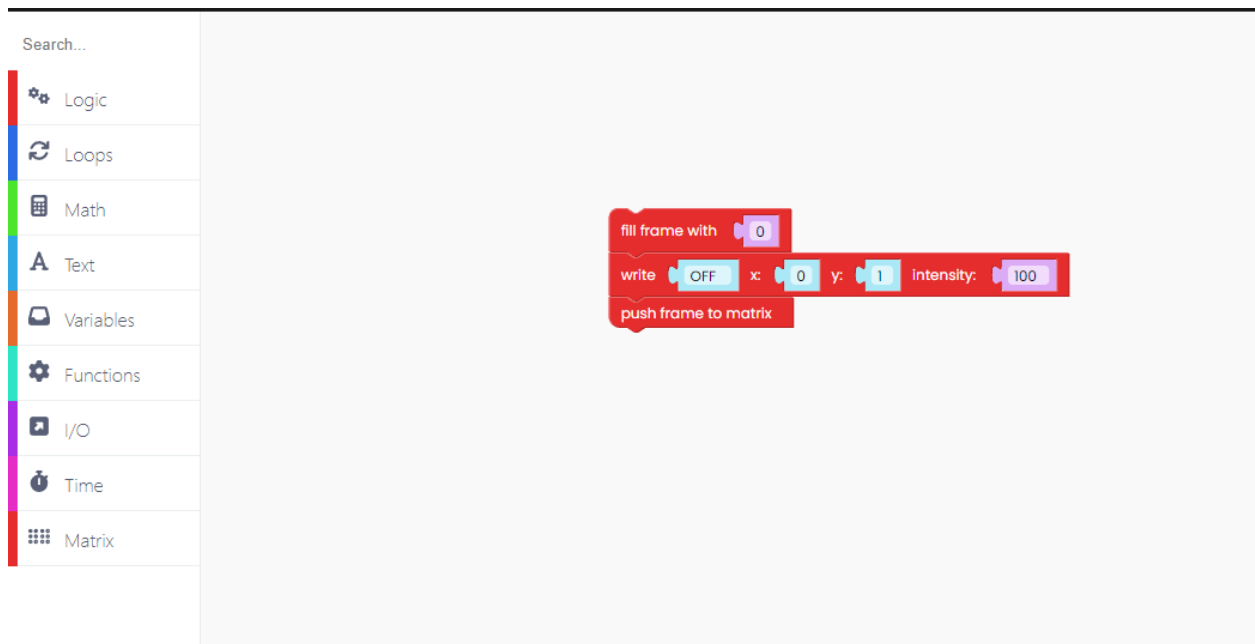
Then, we want to write "off", since the "on" message will not appear until you press the button.

The writing block can be found within the "Matrix" block area.



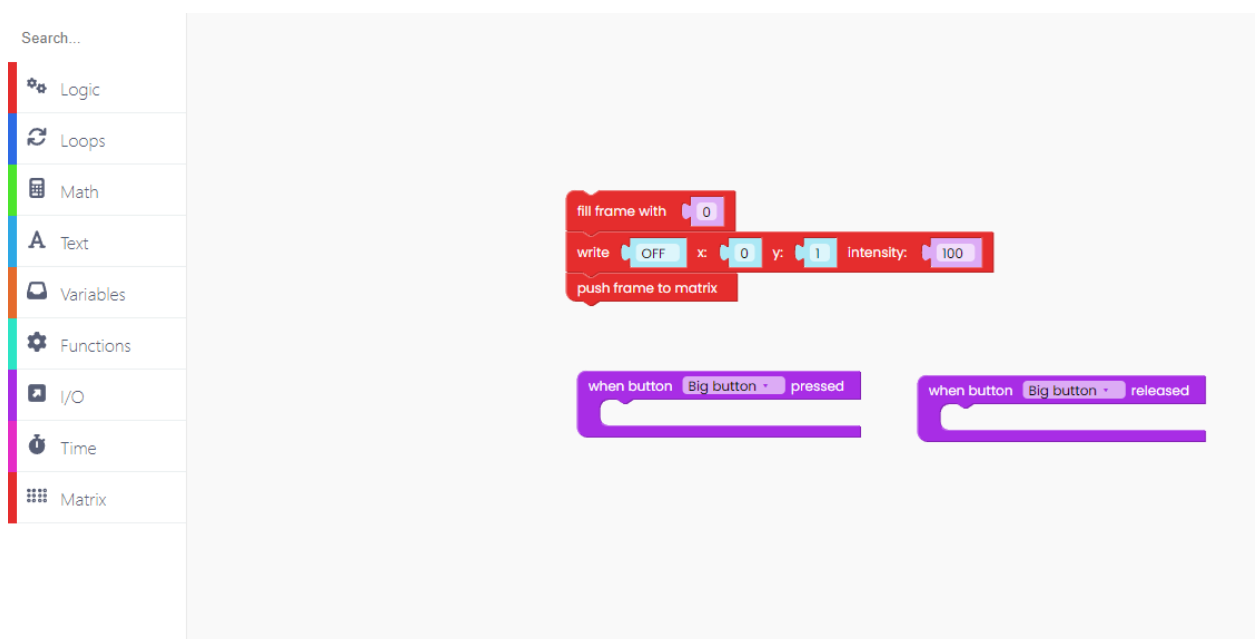
As you can see, the text we want to appear is "OFF", and we want the coordinates to be x: 0, and y: 1. The intensity level will be set to 100.

As you already know, the matrix code will not function without the "push frame to matrix" section at the end.

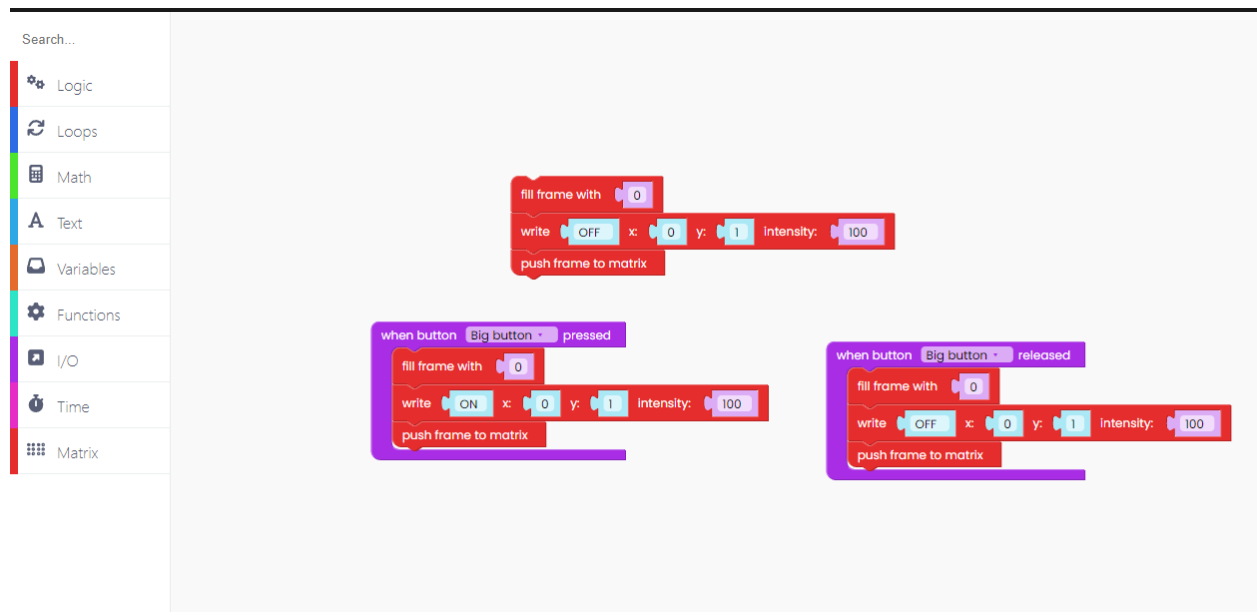


Now, let's check what actions we want the big red button to trigger.

You'll need the I/O block section's "when big button pressed" and "when big button released" blocks for this.



You'll put whatever you want to happen when you press and release the big button inside these purple blocks.

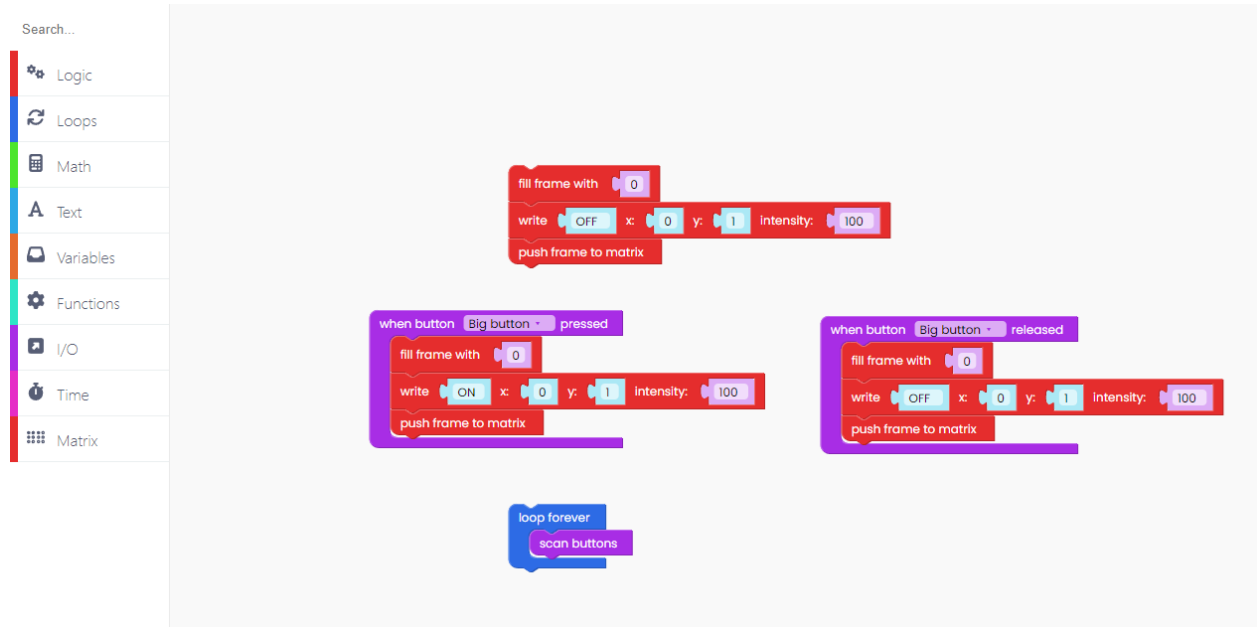


So, in both cases we want the matrix to turn off.

When we press the button, the text "ON" appears, and when we release it, the text "OFF" appears.

Both will have the coordinates x: 0 and y: 1, with an intensity of 100.

Because we're working with buttons, we'll need to add the "scan buttons" and "loop forever" buttons at the end.



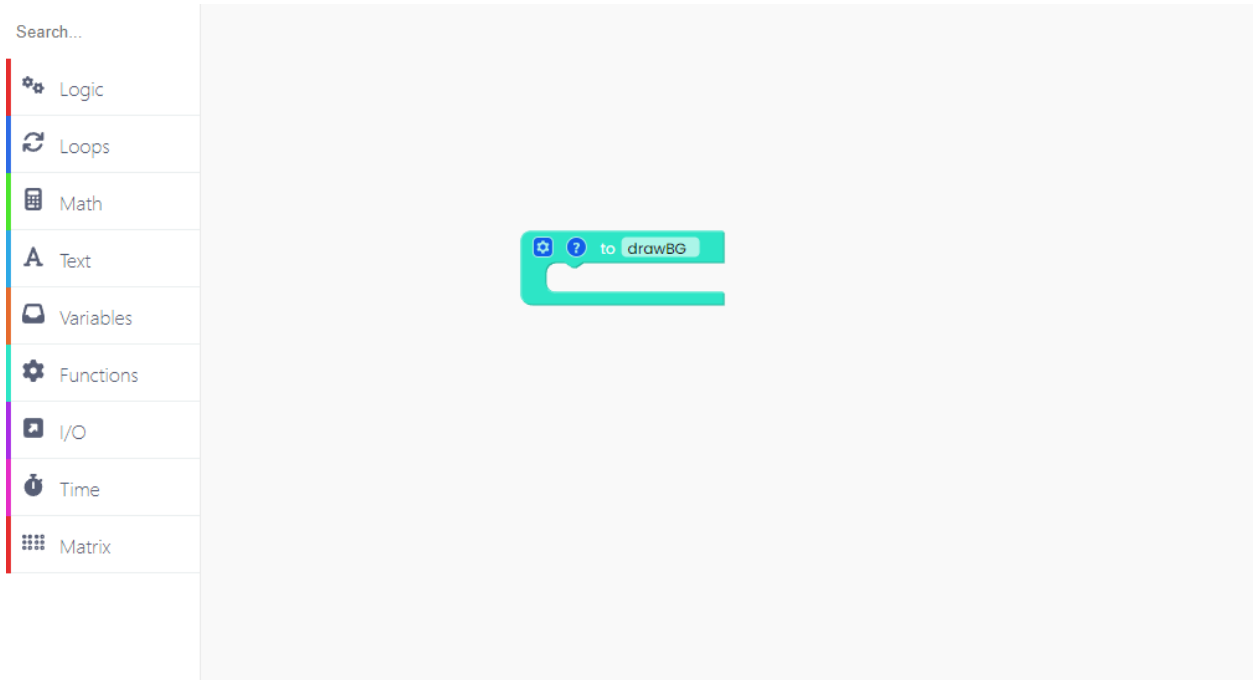
Now, click on the Run button, start playing with the button and check the text.

Drawing shapes

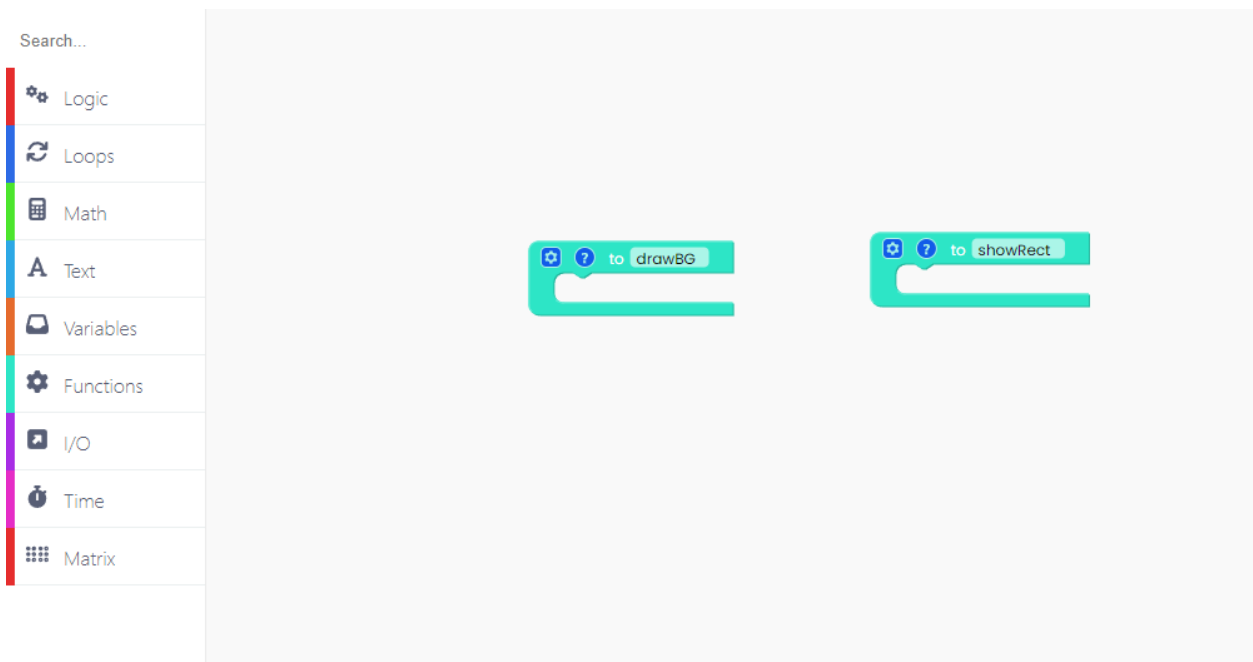
For the end, let's code something more complicated.

The matrix board can be used to draw a variety of shapes.

To draw shapes, we'll make two functions. We'll name one of them "drawBG" and use it to clear the background and draw two crossing lines in the middle of the matrix.



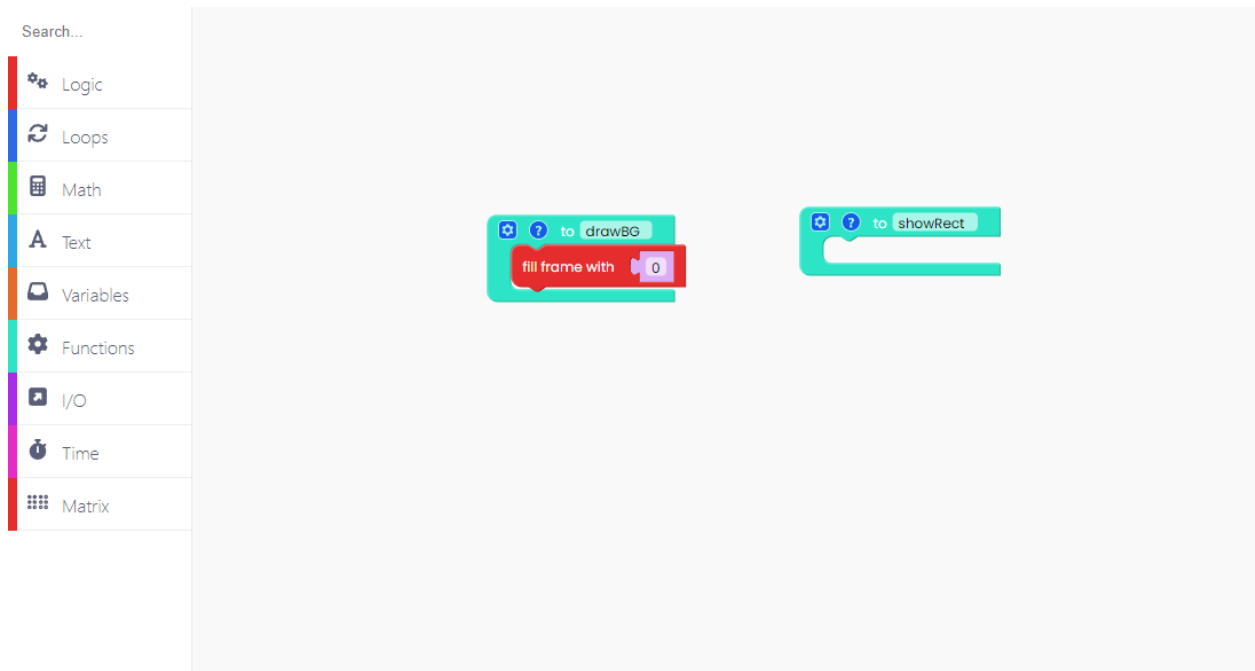
The other one will be called "showRect" and will be used to draw a rectangle.



Let's play with the first one for a second.

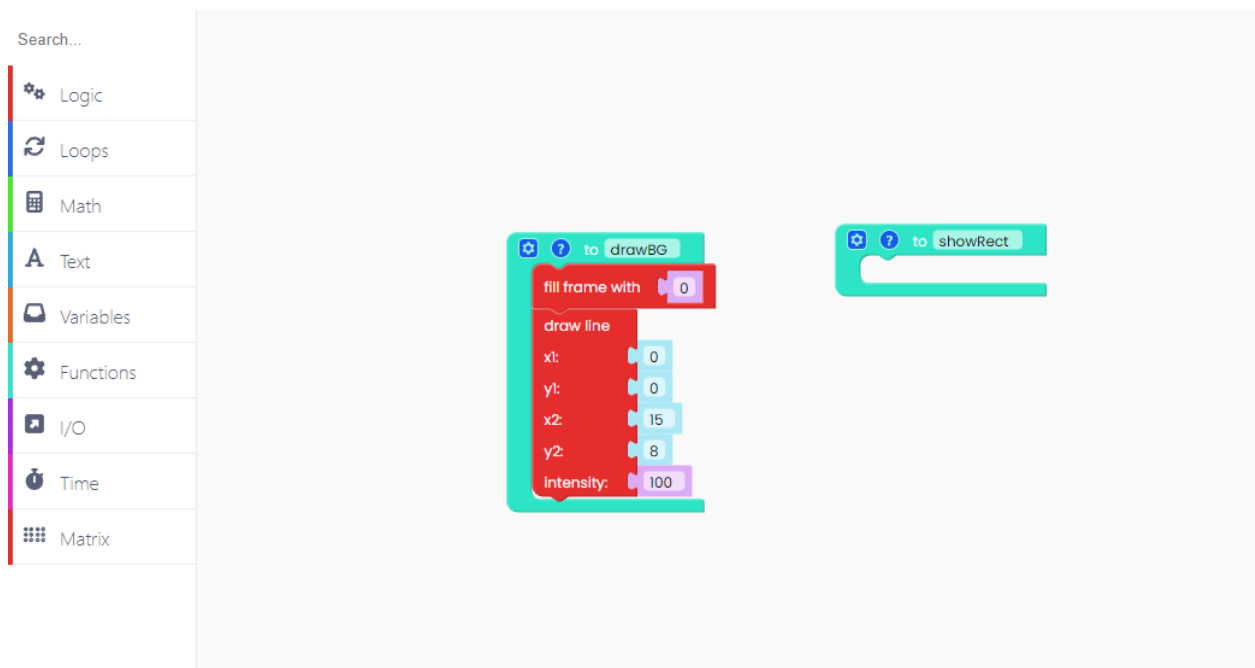
First, we want to clear the background, or turn off anything on the matrix.

We'll need the "fill frame with 0" block for this.



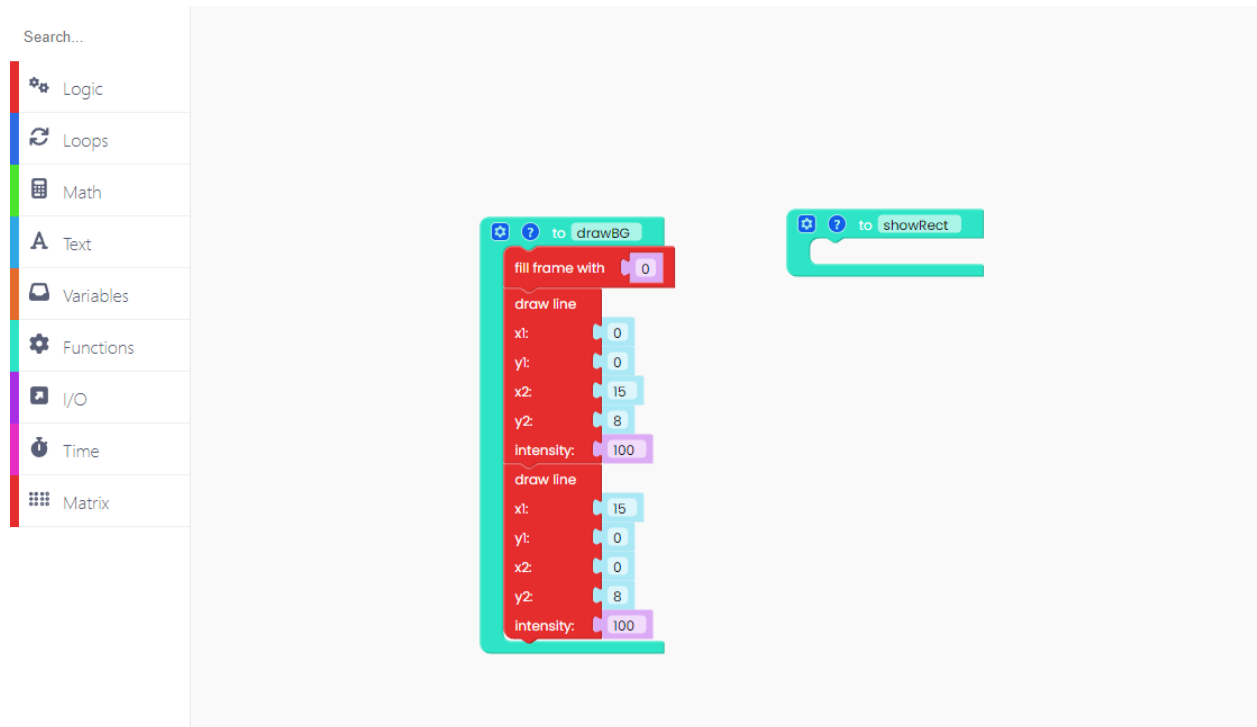
Let's start with the first line.

The block for doing so can be found in the "Matrix" block section.



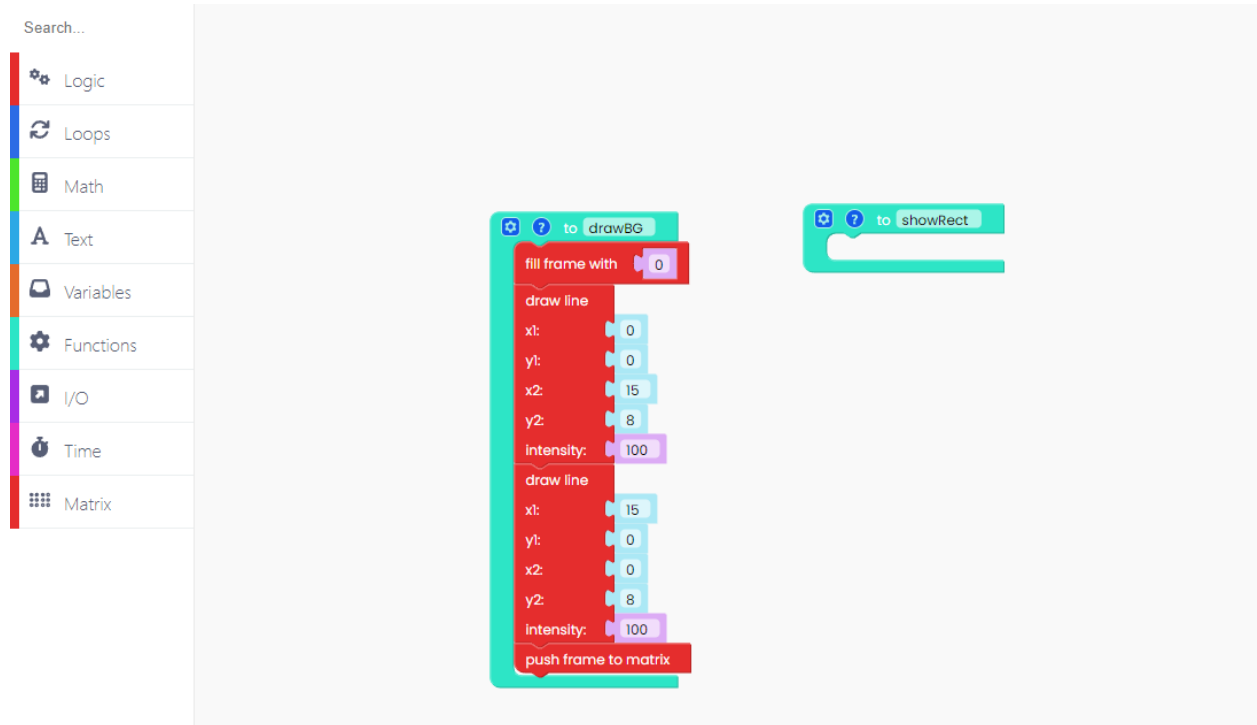
Check the coordinates we put. The intensity level is set to 100.

Let's draw the second line now. By right-clicking on this block, you can duplicate it.



Make sure the coordinates for the second line are changed.

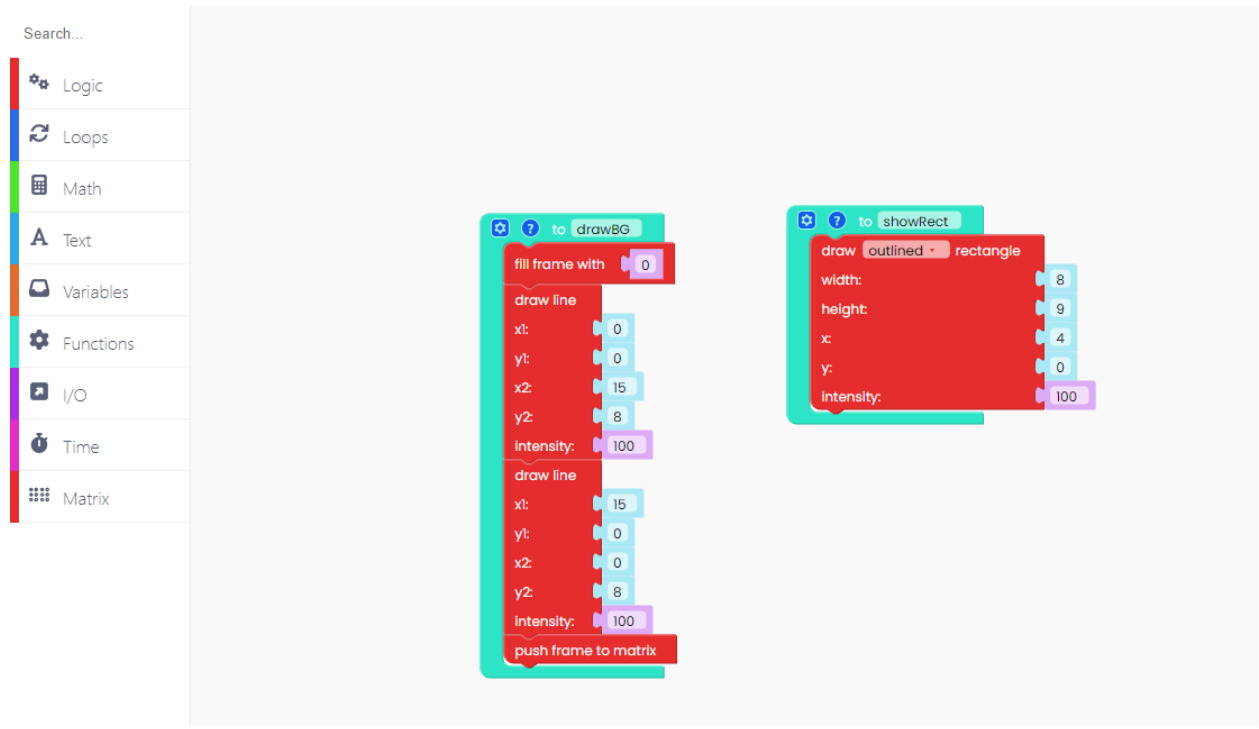
Don't forget to include the "push frame to matrix" block at the end, or your code will fail.



Good job!

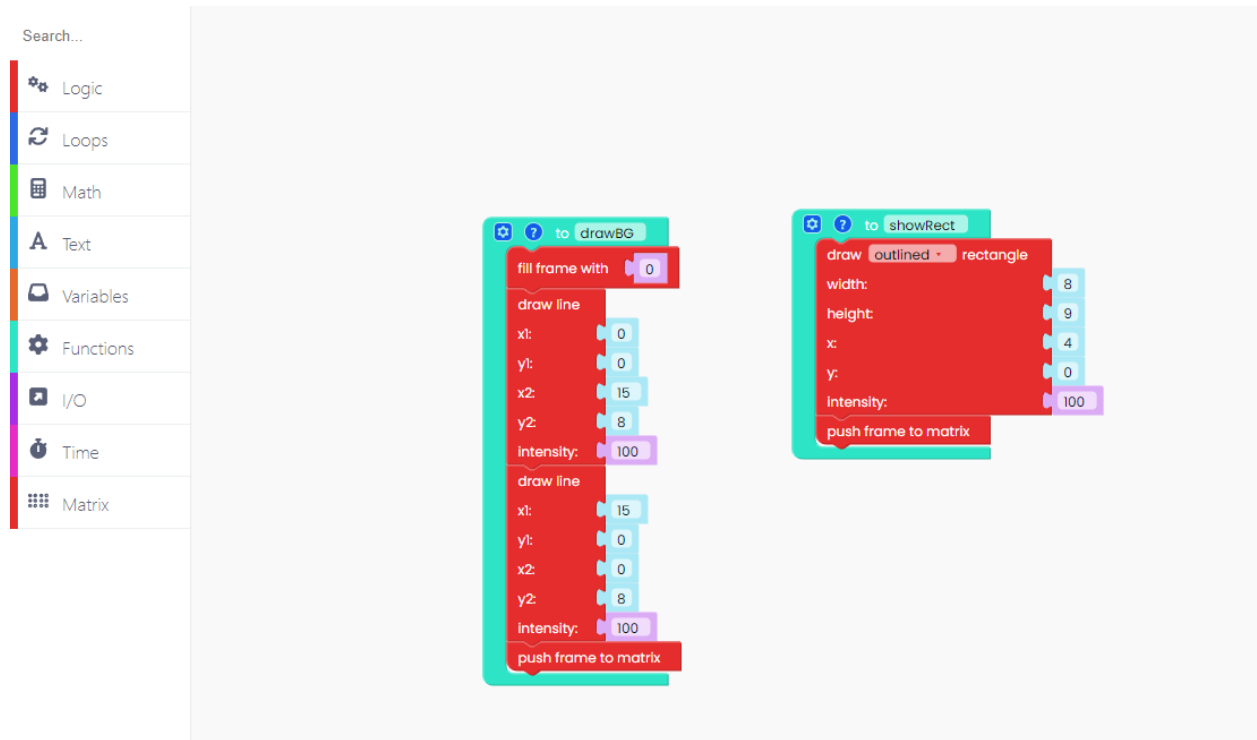
Now, inside the "showRect" function, write the code.

In the "Matrix" block section, look for the "draw outlined rectangle" block.



Feel free to play about with the coordinates and rectangle size.

Remember to include the "push frame to matrix" block once more.



Let's make those shapes appear in loops one after the other.

Find the "loop forever" block in the "Loops" block section to do so.

Search...

- Logic
- Loops
- Math
- Text
- Variables
- Functions
- I/O
- Time
- Matrix

The image shows two separate function blocks in a Scratch environment. The first block, titled 'to drawBG', contains the following code:
fill frame with 0
draw line
x1: 0, y1: 0, x2: 15, y2: 8, intensity: 100
draw line
x1: 15, y1: 0, x2: 0, y2: 8, intensity: 100
push frame to matrix

The second block, titled 'to showRect', contains the following code:
draw outlined rectangle
width: 8, height: 9, x: 4, y: 0, intensity: 100
push frame to matrix

Below these blocks is a blue 'loop forever' block.

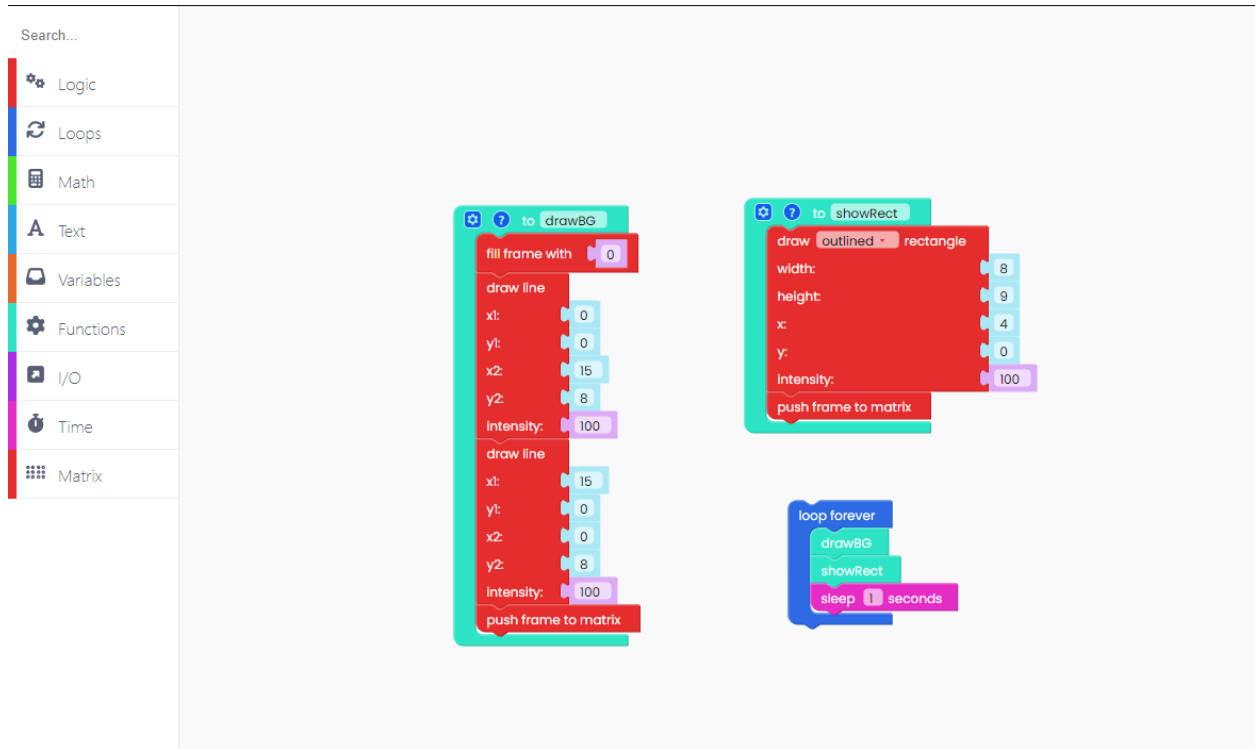
Put both functions within the blue block.

Search...

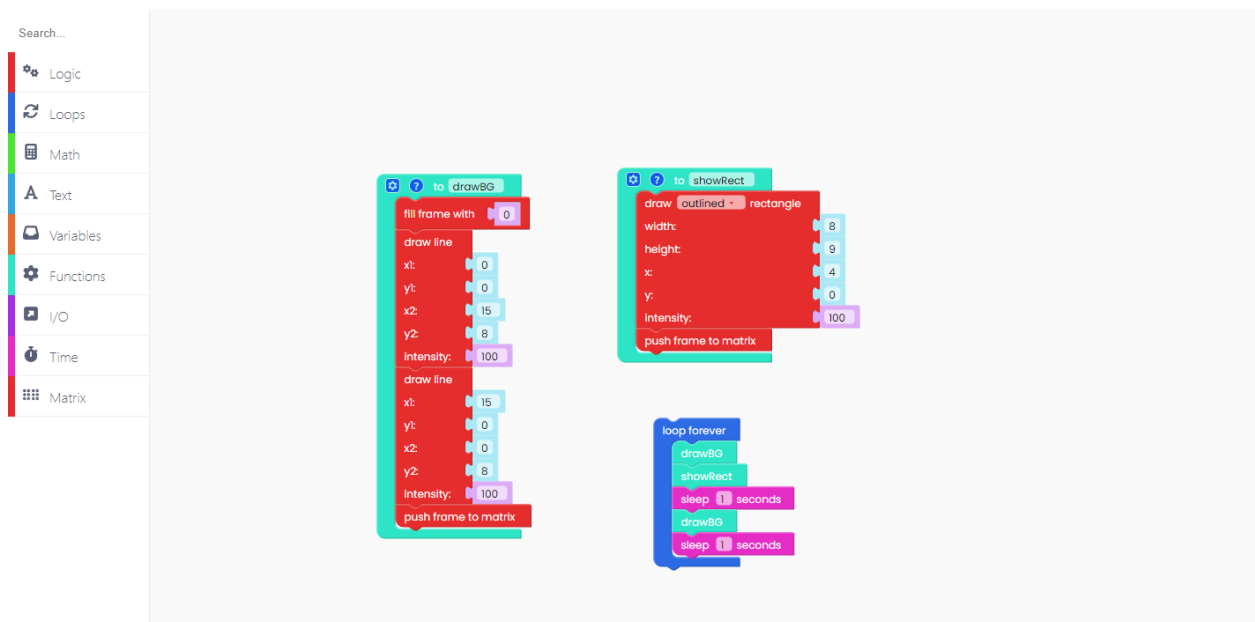
- Logic
- Loops
- Math
- Text
- Variables
- Functions
- I/O
- Time
- Matrix

The image shows the same Scratch environment as above, but with the 'drawBG' and 'showRect' function blocks now nested inside the 'loop forever' block. The 'loop forever' block contains:
drawBG
showRect

To create a one-second delay, add the "sleep 1 second" block.



Now, add another "drawBG" and "sleep 1 second" blocks at the end.



That's it.

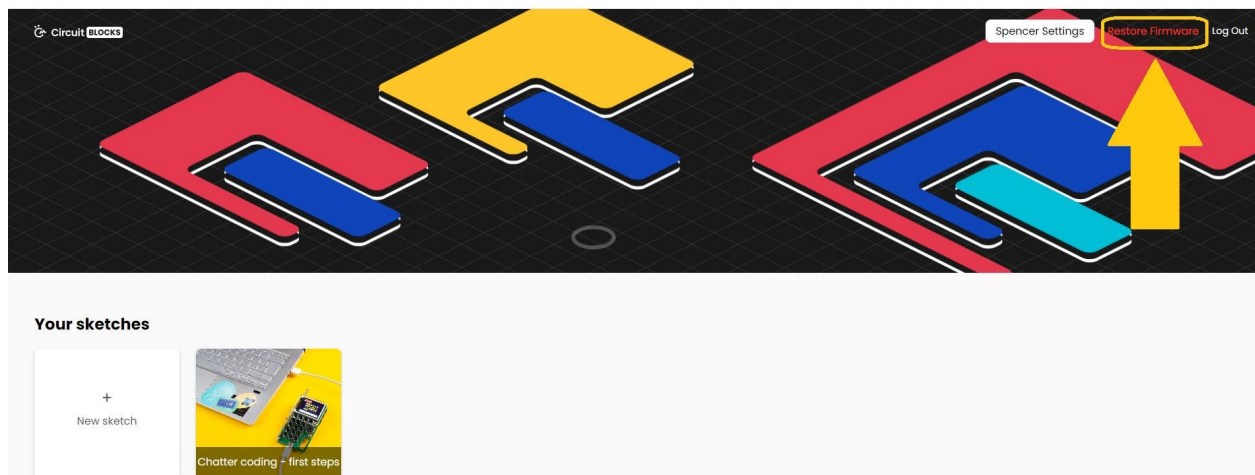
Great job!

Click on the Run button and check your code.

Restore Spencer's firmware

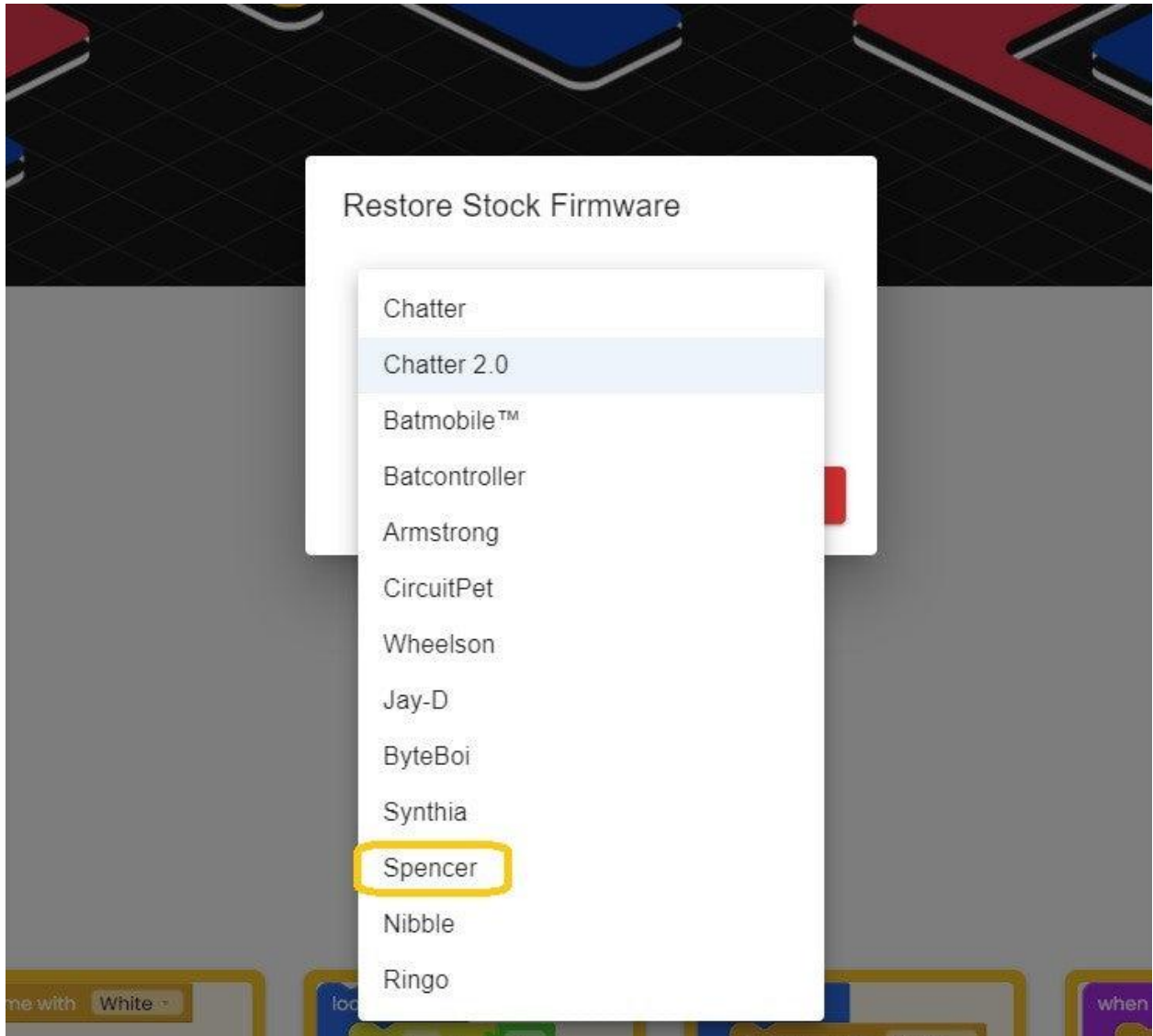
If, for any reason, you want to restore Spencer's firmware, follow these steps.

This is quite simple, just connect your Spencer to the USB port of your computer and press the "Restore firmware" button on the top right.



You will be prompted with a window where you need to choose the device that you are restoring the firmware for.

Choose Spencer, of course.



Wait for a few seconds, and your Spencer will be back and running like usual.