



Activity 4: Repelling the shockwave

PXT - JavaScript

Teacher note – Curriculum links:

- Learning about the hardware and software components that make up computer systems and how data and instructions are stored and executed within a computer system
- Understand how computers collect data from various input devices, including sensors

In our **Doctor Who and the micro:bit Live Lesson**, we'll be asking you to help us repel the shockwave engineered by the Daleks using your BBC micro:bit.

In order to join in with this activity, you'll need to flash the hex file to your micro:bit before or during the Live Lesson.

Step 1: Download the hex file and join in live

Right-click on the **Repelling the shockwave hex file** link on the Live Lessons website and 'save as' to download the hex file. Plug in your BBC micro:bit, hit 'compile' and drag the hex file onto your BBC micro:bit.

You can now join in with our **Repelling the shockwave** activity alongside our Live Lesson. Simply watch the demonstration on screen and shake your micro:bit when asked to by the presenters.

When you've shaken the micro:bit enough to help repel the shockwave, your micro:bits LEDs will flash and light up fully!

Step 2: Understanding the code

After the Live Lesson, you may want to investigate how the program actually works. On the Live Lesson website, click on **Repelling the shockwave – PXT editor** to view the code on the micro:bit website.

Here is a quick explanation of how each part of the code in PXT JavaScript works. To see how the code in PXT Blocks works, refer to the worksheet **Repelling the shockwave – Block Editor**.

```

1 let brightness = 0
2 basic.forever(() => {
3   led.setBrightness(brightness)
4 })

```

Introducing the variable 'brightness'

Here, we've introduced a **variable**, **brightness**, and set its original value to 0.

```

5 input.onGesture(Gesture.Shake, () => {
6   basic.clearScreen()
7   brightness += 20
8   basic.showLeds(`
9     . . . . .
10    . . . . .
11    . . # . .
12    . . . . .
13    . . . . .
14    `)
15   basic.showLeds(`
16     . . . . .
17     . # # # .
18     . # # # .
19     . # # # .
20     . . . . .
21    `)
22   basic.showLeds(`
23     # # # # #
24     # # # # #
25     # # # # #
26     # # # # #
27     # # # # #
28    `)
29   if (brightness >= 240) {
30     basic.clearScreen()
31     led.setBrightness(255)
32     basic.showLeds(`
33       # # # # #
34       # # # # #
35       # # # # #
36       # # # # #
37       # # # # #
38      `)
39     basic.showLeds(`
40       . . . . .
41       . . . . .
42       . . . . .
43       . . . . .
44       . . . . .
45      `)
46     basic.showLeds(`
47       # # # # #
48       # # # # #
49       # # # # #
50       # # # # #
51       # # # # #
52      `)
53     basic.showLeds(`
54       . . . . .
55       . . . . .
56       . . . . .
57       . . . . .
58       . . . . .
59      `)
60     basic.showLeds(`
61       # # # # #
62       # # # # #
63       # # # # #
64       # # # # #
65       # # # # #
66      `)
67     basic.pause(10000)
68     pins.digitalWritePin(DigitalPin.P0, 1)
69   } else {
70     basic.clearScreen()
71   }
72 })
73

```

Introducing the 'forever' loop

Here, we've introduced a **loop** that goes on forever. Within the loop, we've set the brightness of the micro:bit's LEDs to a **variable** named **brightness**.

What happens when you shake your micro:bit

Here, we've said that when you shake your micro:bit, the LED screen is cleared.

Then, 20 is added to the variable **brightness**, and a little animation is displayed.

When you've shaken your micro:bit enough

In this part of the program, we've introduced a conditional statement, which checks if the variable **brightness** is more than or equals to 240.

If it is, then the screen is cleared again, and the micro:bit's LEDs are set to 255 (full brightness). The LEDs then flash in a short animation, before the program writes '1' to Pin 0 – this would turn on any additional LED or buzzer that is connected to the micro:bit at Pin 0. There's then a 10 second (10000 millisecond) pause in the program.

If **brightness** is less than 240, then the screen is cleared, and nothing else happens until you shake your micro:bit again. This way, each time you shake your micro:bit, 20 is added onto the variable until the condition is fulfilled.

Step 3: Test and play

Now that you understand the code, you can adapt it to create your own version of the **Repelling the shockwave** game.

A simple thing to experiment with is changing the images or animations that your BBC micro:bit displays. Simply click on the 'basic.showLeds' segments and change the patterns within the brackets and quote marks to whatever you like.

Another thing you can do easily is change the difficulty of the game. You do this by changing how much you're adding to the variable **brightness** with each shake, or by changing the conditional statement.

Currently, we're adding 20 to brightness with each shake. You can make the game easier by clicking on the line that says '**brightness += 20**' and making the number bigger, or make the game harder by making the number smaller.

Alternatively, you could click on the line that says 'if (**brightness >= 240**)' and make the number smaller, to make it easier.