**Volcano animations**

**Lesson 4: Programming eruption animations**

**Introduction**

In this lesson pupils use the algorithms they created in the previous lesson to program their BBC micro:bit volcanic eruption animation using the MakeCode editor. Once completed, they consider their opinions on different aspects of learning in the unit.

**Time:** @60 minutes

**Materials needed:** lesson presentation, pupils’ algorithms from previous lesson, [**missing-from-repetition**](https://makecode.microbit.org/#pub:_Eq5Y6hHdFRbi) and [**microbit-example-of-animation**](https://makecode.microbit.org/#pub:_52URfWgywg5H) hex files, printouts of slide 7, computers/laptops with access to [the MakeCode editor](https://makecode.microbit.org/#editor), physical micro:bits and USB leads (if you have them).

**Learning objectives**

* To follow an algorithm accurately to write a program
* To use repetition in a program effectively
* To test and debug programs and algorithms
* To review learning

**Lesson summary**

* Introduction: going loopy (10 minutes)
* Making micro:bit animations (35 minutes)
* Reviewing animations (15 minutes)

**Introduction: going loopy (10 minutes)**

* Upload and open the [**missing-from-repetition**](https://makecode.microbit.org/#pub:_Eq5Y6hHdFRbi) hex file to the MakeCode editor (or use link in **slide 3** to published project).
* Using the simulator run the program and ask pupils to identify how many times the animation is repeated and how many images are contained within the repeat (there are four images in the sequence and the sequence is repeated six times).
* Invite suggestions on why the final image, despite being part of the planned sequence, was only shown once (it has not been placed inside the repeat block). Show pupils the program and identify that it was not placed within the repeat block. Invite pupils to debug the program and re-run it to see if it works.

**Making micro:bit animations (35 minutes)**

* Invite pupils to describe what they did in the previous lesson (created algorithms) and how they will make use of these in this lesson (**slide 4**).
* Model how to use the MakeCode editor to construct a program with more than one repeating section by inviting pupils to demonstrate on the whole class display (example [**microbit-example-of-animation.hex**](https://makecode.microbit.org/#pub:_52URfWgywg5H) is included in the lesson downloads and published in the link on **slide 5**).
* Give pupils time to work in their pairs, or small groups, to follow their algorithms to write their programs using the MakeCode editor (**slide 5**). You may wish them to make use of paired programming (**slide 6**).
* Remind pupils to test and debug their programs as they work and, when successful changes are made, to document this by annotating their algorithm.
* If you have access to physical micro:bits, download and transfer the programs.
* Encourage pupils to share their animations with other groups.

**Reviewing Animations (15 minutes)**

* Display **slide 7** and give copies to pupils.
* Explain that the table will be used to record their opinions of the unit so far. Highlight the terms *process* and *product* and invite pupils’ ideas on what these refer to - the process: the different activities/steps they undertook to make their animations; the product: the final animation.
* Explain to pupils that they are going to identify any aspects of the process and the product they are pleased with/enjoyed and any that they didn’t. Identify where on the table different opinions would be placed (if pupils have experience of Carroll diagrams in maths, link to this).
* Give pupils time to complete their own copy of the *Reviewing your Learning* sheet.

**Extension ideas:**

* Pupils could create a video of their micro:bit animations and add narration using movie-making software to create an explanation video.

**Differentiation**

**Support:**

* Pupils will have planned less stages to program in the previous lesson and should be given the support needed to program their animation successfully.

**Stretch & challenge:**

* Pupils will have planned how to include words and/or numbers into the animation in the previous lesson and can program these into their animation. They can add comments to their program to explain their code.

**Opportunities for assessment:**

* Informal observations of pupils’ understanding of repetition, debugging and programming.
* More formal assessment of pupils’ completed programs.