

Volcano animations

**Lessons:** 5

**Programming languages:** MakeCode

**Target age:** 7-11 yrs

**Subjects & topics:**

* Computational thinking: Decomposition, Algorithms, Flowcharts
* Geography: Physical geography
* Programming: Iteration
* Design & technology: Product design

# Unit of work summary

This series of five lessons is aimed at students aged 8-9 years. Through a mixture of fun unplugged and programming activities related to animations, pupils develop their understanding of decomposition, flowchart algorithms and repetition. They then write, program and test an animation showing volcanic eruption using the LEDs on the BBC micro:bit.

## Overall key learning

* can analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems
* can evaluate and apply information technology, including new or unfamiliar technologies, analytically to solve problems
* are responsible, competent, confident and creative users of information and communication technology

## Additional skills

Creative thinking, collaboration, problem-solving, debugging

## Lesson 1: Animation & decomposition

In this 'unplugged' lesson pupils are introduced to the theme of animation and produce a thinking map based on the subject. They then develop their understanding decomposition by planning a dance sequence and creating a flipbook animation.

**Key learning:**

* To understand decomposition
* To use decomposition to create a dance sequence
* To create a flipbook animation of a dance sequence

## Lesson 2: Flowcharts & repetition

Pupils write a simple flowchart algorithm for their dance sequence. They then follow their algorithm to write a program using the BBC micro:bit MakeCode editor and explore how repetition can be used to create an animation.

**Key learning:**

* To understand the use of repetition
* To write simple flowchart algorithms using repetition
* To write programs based on algorithms using repetition

## Lesson 3: Planning eruption animations

In this ‘unplugged’ lesson pupils plan a volcanic eruption animation using the BBC micro:bit. They decompose the process into stages and write a simple flowchart algorithm.

**Key learning:**

* To decompose a process into stages
* To construct simple flowchart algorithms
* To use repetition in algorithms

## Lesson 4: Programming eruption animations

Pupils follow their algorithm and write a program to create their volcanic eruption animation using the MakeCode editor.

**Key learning:**

* 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 5: Reflection & review

ln this ‘unplugged’ lesson, pupils reflect on their learning by decomposing the steps they took to create their animation and record this using a thinking map.

**Key learning:**

* To understand and explain decomposition
* To use decomposition to review learning
* To reflect on learning

# Curriculum links

## England National Curriculum

#### KS2 computing curriculum

Curriculum aims:

* can analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems
* can evaluate and apply information technology, including new or unfamiliar technologies, analytically to solve problems
* are responsible, competent, confident and creative users of information and communication technology

Students should be taught to:

* design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts
* use sequence, selection, and repetition in programs; work with variables and various forms of input and output
* use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs

[Read the full KS2 computing curriculum](https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/239033/PRIMARY_national_curriculum_-_Computing.pdf).

#### KS2 geography curriculum

Students should be taught to:

* describe and understand key aspects of physical geography, including: volcanoes and earthquakes

[Read the full KS2 geography curriculum](https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/239044/PRIMARY_national_curriculum_-_Geography.pdf).

#### KS2 DT curriculum

* evaluate their ideas and products against their own design criteria and consider the views of others to improve their work
* apply their understanding of computing to program, monitor and control their products

[Read the full KS2 DT curriculum](https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/239041/PRIMARY_national_curriculum_-_Design_and_technology.pdf)

## Scotland Curriculum for Excellence

#### Technologies

* I can explore and comment on processes in the world around me making use of core computational thinking concepts and can organise information in a logical way (TCH 1-13a)
* I understand the instructions of a visual programming language and can predict the outcome of a program written using the language (TCH 1-14a)
* I understand the operation of a process and its outcome. I can structure related items of information (TCH 2-13a)
* I can explain core programming language concepts in appropriate technical language (TCH 2-14a)
* I can create, develop and evaluate computing solutions in response to a design challenge (TCH 2-15a)

[Read the full technologies curriculum](https://education.gov.scot/Documents/Technologies-es-os.pdf)

#### Social studies

* I can describe the physical processes of a natural disaster and discuss its impact on people and the landscape (SOC 2-07b)

[Read the full social studies curriculum](https://education.gov.scot/Documents/social-studies-eo.pdf)

#### Expressive arts

* I have the opportunity to choose and explore an extended range of media and technologies to create images and objects, comparing and combining them for specific tasks (EXA 2-02a)

[Read the full expressive arts curriculum](https://education.gov.scot/Documents/expressive-arts-eo.pdf)

## Northern Ireland Curriculum - Primary

#### Using ICT across the curriculum

* evaluate - talk about, review and make improvements to work, reflecting on the process and outcome

#### KS1 - suggested curriculum ideas for the world around us

* design and make simple models

#### KS2 - suggested curriculum ideas for the world around us

* design and make simple models

[Read the full Northern Ireland curriculum - primary](https://ccea.org.uk/learning-resources/northern-ireland-curriculum-primary)

#### KS1 & 2 - requirements for using ICT

* evaluate - talk about, review and make improvements to work, reflecting on the process and outcome and consider the sources and resources used, including safety, reliability and acceptability

[Read the full KS1 & KS2 requirements for using ICT](https://ccea.org.uk/downloads/docs/ccea-asset/Curriculum/Curriculum%20Requirements%20for%20Using%20ICT.pdf)

#### KS2 - the world around us

* features of, and variations in places, including physical, human, climatic, vegetation and animal life
* how movement can be accelerated by human and natural events such as wars, earthquakes, famine or floods
* how change is a feature of the human and natural world and may have consequences for our lives and the world around us
* ways in which change occurs over both short and long periods of time in the physical and natural world

[Read the full KS2 requirements for the world around us](https://ccea.org.uk/downloads/docs/ccea-asset/Curriculum/Key%20Stage%202%20Statutory%20Requirements%20for%20The%20World%20Around.pdf)

#### Primary using ICT - desirable features - computational thinking and coding

**Level 4**

Pupils should:

* create a more sophisticated coding project using a broad range of commands; and/or
* solve a given problem using commands in a programming environment.

**Programmable devices (such as Parrot Drone, micro:bit or Sphere)**

* look at and talk about examples of coding projects, including the use of motion, looks, lights or sounds, sensors, control and events such as ‘if...then’ and ‘loop until’ (or equivalent) that make the code more efficient;
* recognise that these projects are composed of different components and break the task into smaller manageable tasks (decomposition);
* in small groups, plan and storyboard their own coding project, working out what different parts of the program must do, using logical reasoning to discuss and compare the commands that are required for their algorithm;
* use a range of commands to create a project including triggering commands such as ‘if...then’ and ‘loop until’ to facilitate a more efficient method of interaction;
* test and debug at regular intervals and collaborate with others to solve problems as they arise;

**Finally**

* share their work (possibly using digital tools), respond to feedback and comment on others’ work; and
* organise files and export work in an appropriate format so that others may view it.

[Read all Primary using ICT desirable features](https://ccea.org.uk/downloads/docs/ccea-asset/Curriculum/Primary%20Using%20ICT%20Desirable%20Features%20Update%202019.pdf)

## Curriculum for Wales

#### Science and technology

Progression step 2 - design thinking and engineering offer technical and creative ways to meet society's needs and wants:

* I can safely use a range of tools, materials and equipment to construct for a variety of reasons
* I can use computational thinking techniques, through unplugged or offline activities
* I can create simple algorithms and am beginning to explain errors
* I can follow algorithms to determine their purpose and predict outcomesI can follow instructions to build and control a physical device

Progression step 3 - design thinking and engineering offer technical and creative ways to meet society's needs and wants:

* I can identify repeating patterns and use loops to make my algorithms more concise
* I can explain and debug algorithms

[Read the full science and technology curriculum](https://hwb.gov.wales/curriculum-for-wales/science-and-technology/descriptions-of-learning/)

#### Humanities

Progression step 3 - Our natural world is diverse and dynamic, influenced by processes and human actions:

* I can give simple descriptions of the processes that lead to change in the natural world

[Read the full humanities curriculum](https://hwb.gov.wales/curriculum-for-wales/humanities/descriptions-of-learning/)

#### Digital competence framework

Progression step 1 - data and computational thinking - problem-solving and modelling:

* I can recognise and follow instructions in the appropriate order to perform a task.
* I can organise, select and use simple language to give instructions to others.
* I can control devices giving instructions.
* I can identify errors in simple sets of instructions (algorithm).

Progression step 2 - data and computational thinking - problem-solving and modelling:

* I can detect and correct mistakes which cause instructions (a solution) to fail (debug).
* I can create and record verbal, written and symbolic instructions to test ideas, e.g. the order of waking up through a diagram or flowchart.
* I can change instructions to achieve a different outcome.
* I can identify repetitions or loops in a sequence, e.g. identify where to shorten a set of instructions by repeating steps, for instance when learning a new song.

Progression step 3 - data and computational thinking - problem-solving and modelling:

* I can create and refine algorithms and flowcharts to solve problems, making use of features such as loops, Boolean values and formulae.
* I can understand the importance of the order of statements within algorithms.

Progression step 1 - producing – creating digital content:

* I can explore and use animation and video.
* I can create simple digital work.

Progression step 1 - producing - evaluating and improving digital content:

* I can comment on work in relation to a single success criterion.
* I can give an opinion about my own work and suggest improvements based on the success criteria.

[Read the digital competence framework](https://hwb.gov.wales/curriculum-for-wales/cross-curricular-skills-frameworks/digital-competence-framework)

## USA Code.org

#### CS Fundamentals

Courses A and B
Concepts included:

* computational thinking
* sequencing
* events
* flowchart algorithms
* loops

[Read the full CS Fundamentals curriculum](https://code.org/educate/curriculum/elementary-school).

## USA CSTA Standards

#### Grades 3-5

* 1B-CS-01 - Describe how internal and external parts of computing devices function to form a system.
* 1B-CS-02 - Model how computer hardware and software work together as a system to accomplish tasks
* 1B-CS-03 - Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies.
* 1B-AP-10 - Create programs that include sequences, events, loops, and conditionals.
* 1B-AP-11 - Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.
* 1B-AP-12 - Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.

[Read the CSTA Standards in full](https://csteachers.org/k12standards/ ).

This content is published under a [Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0)](https://creativecommons.org/licenses/by-sa/4.0/) licence.