

Unit 3.1: We are programmers

Programming an animation



Software: Scratch (alternative: ScratchJr)

Hardware: Laptop/desktop/Chromebook computers or tablets, cameras and microphones (if needed)

Overview

In this unit, pupils create their own animation in **Scratch**. In:

- **Session 1** they are introduced to Scratch, and explore its tools
- **Session 2** they determine key features of a good animation by looking at examples, and create a **storyboard** for their own
- **Session 3** they plan and **program** character/s and dialogue for their animation
- **Session 4** they begin to animate their character/s by planning and programming movement
- **Session 5** they plan and program switching costumes and backdrops for their animation

- **Session 6** they add sound to their animation before reviewing, **debugging** and improving it.

Alternatives

The unit sessions give step-by-step guidance on using Scratch. However, this unit could also be carried out using ScratchJr. A downloadable version of Scratch is also available – ‘Scratch desktop’ (see *Useful links* on page 12). You can also download this if you prefer to work offline.

Knowledge, skills and concepts

In this unit, pupils will learn to:

- plan and create an **algorithm** for an animated scene in the form of a **storyboard**
- write a **program** in **Scratch** to create the animation, including characters, dialogue, costumes, backdrops and sound
- review their animation programs and correct mistakes.

Progression

In Key Stage 1:

- **Unit 1.1: We are treasure hunters** introduced programming with Blue-bots.
- **Unit 1.5: We are rhythmic** introduced ScratchJr.
- In **Unit 2.1: We are astronauts** and **Unit 2.2: We are games testers**, pupils developed their programming skills further.

In Key Stage 2:

- Pupils fix mistakes in programs in **Unit 3.2: We are bug fixers**.
- Pupils create their own maths game in **Unit 4.1: We are software developers**.

Assessment – by the end of the unit:

All pupils can:

- create an **algorithm** for an animated scene in the form of a **storyboard**
- break the scene down into small sections of action and dialogue
- write a **program** in **Scratch** to create the animation
- put the blocks of their Scratch script into order.

Most pupils can:

- correct mistakes in their animation programs
- create their own sound and graphics for the **sprites** and the backdrop
- explain the connection between their storyboard and the scene they are animating.

Some pupils can:

- use a *repeat* block to switch between costumes to create the illusion of movement
- think logically to detect and correct errors in their animation program
- publish their animations on the Scratch community website
- adapt ideas from others’ work on the Scratch website.

Background information

Programming typically involves taking a way of solving a problem and expressing this as a **sequence** of instructions in a particular programming language for a computer to execute. Computers can solve a vast range of problems, and there are many different programming languages in which these instructions can be written.

In this unit, pupils create simple programs to make an animation on screen, perhaps telling a story or a joke. They use the **Scratch** programming language, in which programs are created by snapping together blocks on-screen rather than typing in instructions. Their programs include the idea of output on the screen, and perhaps through the speakers, sequences of instructions and some repeated instructions.

Key vocabulary

Abstraction: computational thinking approach to managing complexity by simplifying things, through identifying what is important and what detail can be hidden or ignored

Algorithm: a sequence of precise instructions or steps (sometimes a set of rules) to achieve an objective

Bug: an error or mistake in a program or algorithm, causing the computer or robot to behave in a manner that was not originally intended

Code: instructions (or sometimes rules) that can be understood by a computer

Debug: correct mistakes in a computer program or algorithm

Decomposition: breaking a problem down into smaller parts

Event: something that happens within a computer program to cause some particular code to be run, such as an internal message being received, or a sprite being tapped by the user

Iterative development: A trial and improvement approach to programming or other work, in which

each successive version builds on the previous one by the fixing of mistakes or the adding of features

Output: information produced by a computer – in this case, an animation

Parallel processing: when programs run (or appear to run) simultaneously

Program: a sequence of instructions (or sometimes a set of rules) that can be followed by a computer

Repetition: programming construct which allows a group of instructions to be repeated a number of times, or until a certain condition is met

Scratch: simple, block-based programming language in which programs for characters are built by snapping together code blocks

Sequence: placing programming instructions in order, so that each happens one after the other

Sprite: a graphical character in a program that can be given its own sequence of instructions

Storyboard: a visual representation of the key scenes or frames in a video; one way of recording an algorithm visually

Differentiation

See each session (pages 13–18) for ways to increase support and add challenge to this unit.

If you have pupils who struggle to read the **Scratch** command blocks or find the interface too confusing, consider using ScratchJr on tablets or Chromebooks, as this provides a simple, icon-based tool for learning to **program**.

Pupils learning with EAL can program scratch in their first language using the globe icon in the Scratch editor.

Pupils can be challenged to make their animations accessible to as many people as possible, e.g. by providing both audio and text for dialogue, or creating subtitles in another language for spoken dialogue or narration.

Cross-curricular opportunities

Art and design: Pupils design characters and backdrops using drawing, painting or sculpture.

Dance and drama: Pupils could create animations for a dance or drama performance (e.g. by projecting them onto a screen).

English: Pupils could make an animation based on a traditional tale or picture book.

History, geography and science: Pupils create an animation based on their current topic.

Languages: Pupils could write/record dialogue for their character in a foreign language.

Maths: Pupils could create an animation to explain a concept such as symmetry or area; or they create a reasoning question.

Music: Pupils compose and record backing music for their cartoons.

Preparation for teaching the unit



Things to do

- Check you have access to the Scratch website and online editor.
- Work through the unit yourself, so you know what is expected of pupils.
- Choose some Scratch animations to show to your pupils. There are example animations for this unit of work. Otherwise, look at the 'Featured Projects' and 'Featured Studios' collections on the Scratch home page (see *Useful links*). Search for an animation based on a relevant topic, or create your own.
- If pupils are to share their work with the worldwide Scratch community, or save their work online, they will need accounts setting up for them. If they are using Scratch Desktop, they can save their work to a network folder you can set up in advance.



Resources needed

- **Software:** Scratch (see *Alternatives* on page 10)
- **Hardware:** Laptop/desktop/Chromebook computers cameras and microphones (if needed)



Online resources provided

Session resources

- Worksheet 3.1a: *Jack be Nimble* handout
- Worksheet 3.1b: Storyboard three frame template
- Worksheet 3.1c: Storyboard six frame template
- Worksheet 3.1d: End-of-unit quiz
- Worksheet 3.1e: Pupil self-assessment
- Teaching slides 3.1a–3.1f
- Walkthrough videos 3.1a–3.1f
- Interactive end-of-unit quiz 3.1

Scratch Animations

- Chicken crossing the road 1: [scratch.mit.edu/projects/298882859/](https://scratch.mit.edu/projects/298882859)
- Jack be Nimble: [scratch.mit.edu/projects/318594242/](https://scratch.mit.edu/projects/318594242)
- Stone Age: [scratch.mit.edu/projects/315735083/](https://scratch.mit.edu/projects/315735083)
- Chicken crossing the road 2: [scratch.mit.edu/projects/411789246/](https://scratch.mit.edu/projects/411789246)

Additional resources

- Software in 60 seconds: Scratch 1–5
- CPD video: Instructions for sprites
- CPD video: Sequencing instructions
- CPD video: Multithreading



Online safety

- If pupils plan to upload their animations to the Scratch website, you or they will need to create accounts; this requires parental permission.
- Exploring online animation galleries may expose pupils to inappropriate content. Talk about what to do if they see something inappropriate – turn their iPads over (or turn screens off/close laptop lids) and tell a teacher/adult.
- Review the appropriateness of any animations you show, including the related comments.



Collaboration

The planning here is based on pairs working collaboratively to discuss and decide the key features of a good animation, create a storyboard and plan the dialogue for the animation. Partners will work together to program character and dialogue for their animation. Finally, with other pairs and in groups, they will review, debug and improve their animation.



Useful links

Software and tools

Scratch is available at: scratch.mit.edu. It is free to use, and no registration is required (although accounts can be created for pupils, allowing their work to be stored and shared online). Pupils can access the site from home. A downloadable version is available at: scratch.mit.edu/download ScratchJr is an app for iOS, Android and Chrome devices, providing a simpler introduction to block-based programming: scratchjr.org

Online tutorials

- A guide to Scratch for educators: scratch.mit.edu/educators
- Scratch tutorials are part of the editor
- ScratchJr: scratchjr.org/learn/tips

Information and ideas

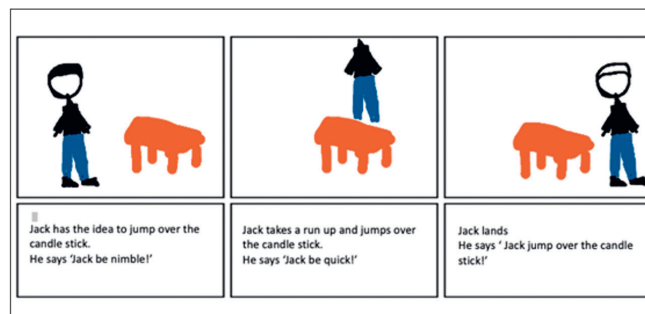
- Specific images: www.pixabay.com
- Sites that demonstrate animation and videos: www.bbc.com/bitesize/levels/zbr9wmn www.brainpop.co.uk
- Advice on sourcing licensed backing music: www.creativecommons.org/legalmusicforvideos
- Example Scratch animations: scratch.mit.edu/explore/projects/animations

Unit outcomes

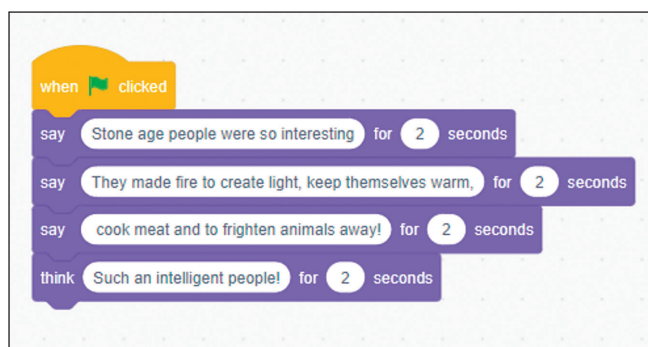
Below are some examples of the outcomes you could expect from this unit.



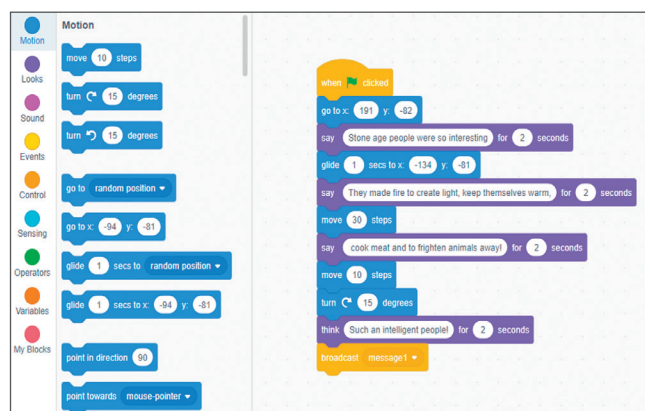
Session 1: Exploring the tools in Scratch to create *Jack be Nimble*



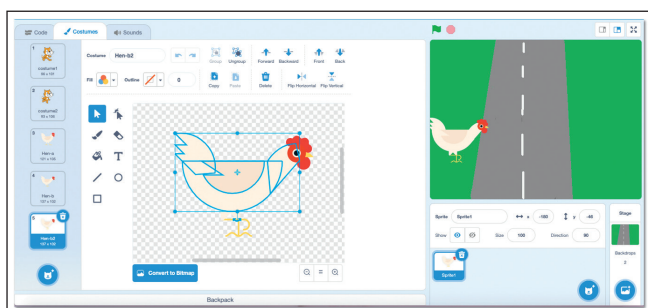
Session 2: Creating a storyboard



Session 3: Starting to program dialogue using *Looks* blocks



Session 4: Animating the sprites by using *Motion* blocks



Session 5: Duplicating and editing costumes



Session 6: Adding and editing sound