

Scratch Tinkering

Exploring Scratch

Duration: **40 minutes**
(+ **extra time pupils can tinker**)

Concepts and approaches covered



Tinkering



Programming

Overview

This activity involves your pupils tinkering with Scratch to find out what it does and how to create programs in it.

Tinkering means trying things out; this is the exploratory phase of learning about something.

In order to program pupils should understand how a programming language works. They need to understand what keywords they can use, what they do and in what order and context they can use those keywords. This is much like learning the vocabulary and grammar of a spoken language. In this activity you and your pupils start to understand a simple programming language – Scratch.

This activity involves two main tasks. Firstly, pupils tinker with an existing Scratch program, which is an animation of two characters chatting, before then tinkering with a blank project in Scratch.

It is worth allowing time for children to tinker with any new digital device or software that they have not used before. Tinkering is also a good way to remind.

Pupil objectives

- I can explore Scratch for myself

Before you start

Watch the introductory videos ([Scratch 3.0](#)) to familiarise yourself with the version of Scratch which you will be using in this activity. What might this word mean?

Resources

- Flipchart and pen for class mindmap of ways to learn
- Interactive whiteboard to display and model use of Scratch
- Pupil's can either access the Scratch resources from the downloads folder or by using these links: [Tinkering animation](#), [Tinkering chase game](#) (For more guidance on the use of the [Barefoot Scratch resources please visit here](#))
- Introductory videos for the teacher for [Scratch 3.0](#)

Introduction **5 minutes**

Explain to pupils that in this activity they will learn about a new programming language called Scratch. Explain that technology, including software programs, is always advancing, so it's important to be able to learn how to use new things ourselves. Ask pupils about digital technology they have used and how they or their families found out how to use these things. Mind map for a few minutes onto flipchart paper the different ideas pupils have for how we can find out about a new computer based item. Explain that tinkering is an important skill when learning about computers. Explain it is 'having a play' to find out how it works and what it does. Talk about how if you find out something for yourself, rather than being told, you might remember it more easily. Can anyone think of this happening to them?

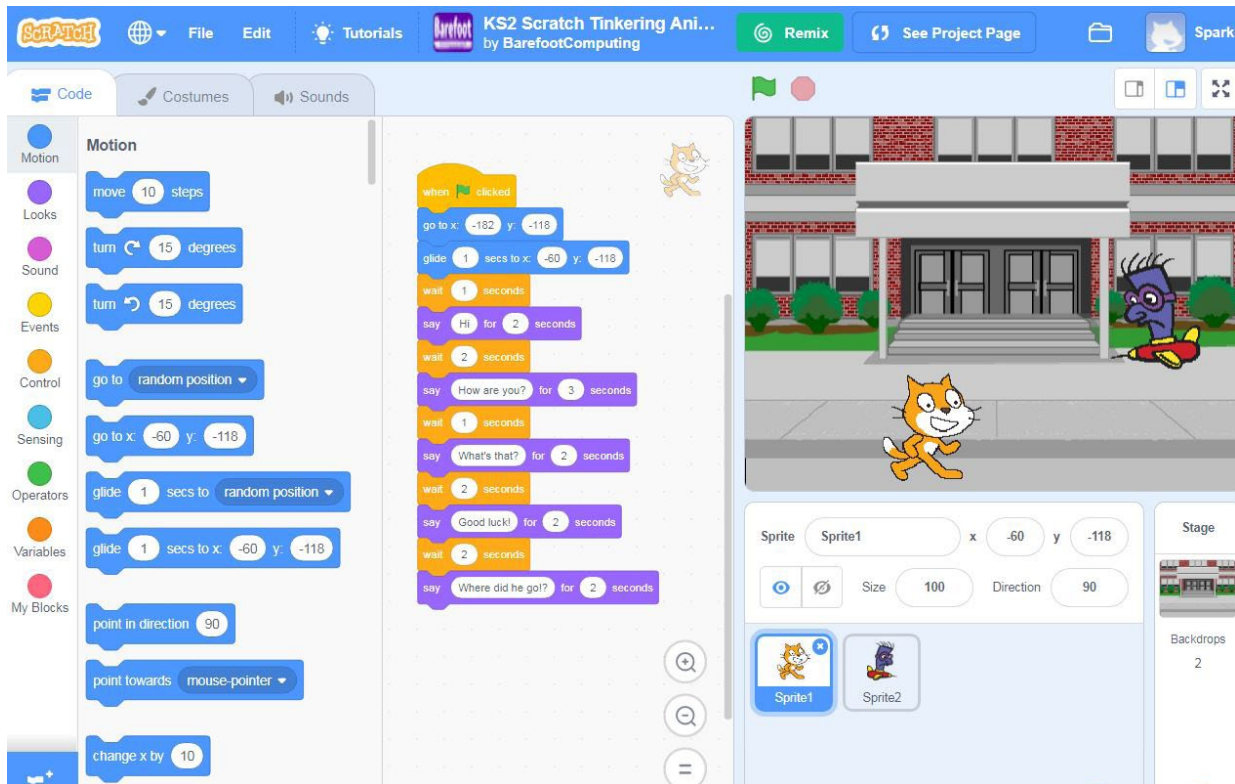
Share the learning objective: I can explore Scratch for myself

Main activity 1 15 minutes

Tinkering with a Scratch project

The first tinkering activity is to provide pupils with an existing program in Scratch and allow them to tinker with this so see how the changes they make change the program.

Model how to open the 'Tinkering animation' on the interactive whiteboard and ask pupils to do the same. (Please refer to [this guide](#) on the ways to download and use our Scratch resources in your school.)



The Scratch animation which pupils can tinker with.

Pupils should be given time to tinker with the animation. Pupils could tinker with Scratch either independently, in pairs or in small groups. You might like to display the following points to guide pupils' tinkering with the program if some or all pupils need more structure.

- Can you change what the sprites say?
- Can you change when the sprites talk?
- Can you add extra dialogue (talking) between the sprites?
- Can you change how and when the sprites move?

Whilst pupils are tinkering, use open questions to support them in developing their understanding, such as:

- What do you think this command does? How do you know?
- Have you changed anything? What happened when you changed...?
- Why do you think it changed like that?
- Are there any commands you recognise? What do they do?
- What have you learnt from tinkering? How have you learnt that?
- How could you learn more?

Main Activity 2 15 minutes

Tinkering with a blank project

Give pupils 15 mins to tinker with a blank Scratch project. You may like to write up the following points to loosely guide pupils' time tinkering with the program if some or all pupils need more structure.

- Can you make the sprite move?
- Can you make the sprite say/think something?
- Can you make the sprite change size?
- Can you make the sprite draw something?
- Can you add more sprites?

Pupils could tinker with Scratch either independently, in pairs or in small groups. As they are tinkering, use open questions to encourage pupils to develop their understanding, such as those from the first activity, and also:

- What have you created? How have you created it?
- What does it do? How does it do that?
- What else does it do? How?
- Why have you used that? How have you used that?
- What can it not do? Why? How could it be changed to do that?
- What could make it better? Why? How?
- What have you learnt?
- How could you learn more?

Also remind pupils of the different ways we thought of that we could learn. Who could they ask? Could they use a search engine to find information on using Scratch? Could they find tutorial videos to watch online?

Extension

An additional Scratch program called [Tinkering chase game](#) has been included in this activity. This program incorporates programming concepts such as repetition, selection and variables. This can be provided to more confident pupils to tinker with to see how they can further extend their understanding of Scratch.

Plenary 5 minutes

Pair pupils and get them to play a verbal tennis match in which they take it in turns to say one thing each that they have learnt about Scratch from their tinkering session. They have 1 minute to say as many points each as possible.

- Get pupils to swap partners and repeat the game several times
- Ask a selection of pupils to share what they have learnt to guide a discussion as a class
- Ask pupils to talk to a partner to consider: What could you use Scratch for based on what you have learnt that it can do? Share a selection of pupils ideas

Differentiation

Support

Some pupils may ask you what to do, as they are not used to being given responsibility for driving their own learning, or may be nervous to break the thing they are exploring. Pupils requiring more support in tinkering may work in a small group, with an additional adult if they are available. It may be that these pupils initially undertake a more guided activity, perhaps all altering one aspect of the Scratch animation and discussing as a group the changes this creates.

Stretch & Challenge

An extension activity has been provided above for more confident pupils.

Assessment opportunities

The open questions provided above and used throughout this tinkering activity will both help encourage, and gauge, pupils' developing understanding of Scratch.

- Observe pupils in their exploration. Are pupils confident to tinker independently or do they wait for instruction? Do they copy others or try new and novel ways of using the equipment?
- Listen to their discussion about tinkering with their peers, do they ask others questions about how they did things, do they suggest ideas? Are they open to new ideas and build on others ideas to discover more

Teaching notes

Concepts and approaches



Tinkering

Tinkering means trying things out, this is the explorative phase of learning about something. We often tinker when we encounter something new to find out what it does and how it works, for example when given a new mobile phone we might try out all the features of it and then more purposefully work out what is useful for us. Tinkering should be fun, free, creative and full of questions and surprises. Open ended themes, questions, playful challenges engender confidence, encourage reflection, highlight diverse ideas and build a joyful, risk taking 'let's have a go' attitude.



Programming

Programming is the act of translating ideas for doing something (algorithms) into instructions (code) that can be followed by a computer; it is a creative, problem solving process. To be able to program effectively pupils require an understanding of commands in the programming language they are using. In this activity pupils are developing their understanding of the commands in Scratch by tinkering. As they combine the commands they are programming, however they are not necessarily programming with a purpose as they aim to solve a problem, but rather are programming experimentally to see what they can work out about the function of some commands.

Curriculum links

Please refer to the resource overview page on the website, to understand how the learning objectives covered in this lesson relate to the curriculum in your country.

Taking this further

The Scratch website (www.scratch.mit.edu) hosts a community whereby Scratch users can host, share and remix others' Scratch projects. Using this pupils can tinker with a whole range of Scratch projects of varying complexities. Whilst you need an account and to be signed in to save any changes you make, you can still make changes and explore the outcome of these changes without an account. This is explain in more detail here.

Pupils can also tinker with other programming languages, such as [Kodu](#) or apps such as [ScratchJr](#), [Hopscotch](#)

Reading

There are a wealth of resources available online to use with Scratch, including:

[Scratch curriculum](#)

[Scratch cards](#)

[Simon Haughton's Scratch resources](#)

Related activities

[ScratchJr tinkering activity](#)

[Bee-Bot tinkering activity](#)

[Programming activities](#)

Barefoot

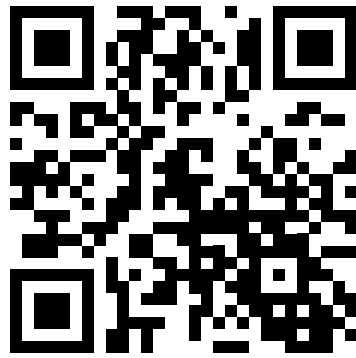
 Computing at School

**We love seeing our resources
being used in the classroom.**

Please share your comments and pictures via our social channels:

 @BarefootComp  /barefootcomputing  @barefootcomputing

Scan the QR code to download
more **FREE** lesson plans!



enquiries@barefootcomputing.org | barefootcomputing.org

Supported by



With help from

