Barefoot

Recommended for
ages 9-11

Viking Raid Animation Activity

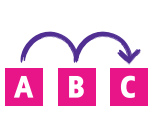
Using sequence in programs

Activity Duration: **1 hour 15 mins**

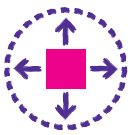
Principal partners

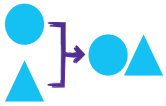
BT Computing at School

Concepts and approaches covered



Sequence


Programming


Algorithms


Tinkering


Debugging


Collaborating

Overview

In this activity pupils program an animation of a Viking raid in Scratch. In doing so they learn that **programming** is the process of implementing **algorithms** as code and about the importance of **sequencing** commands.

Pupil objectives

- [x] **I can write a program with a sequence of instructions**

Introduction (10 mins)

- Recap with pupils what an **algorithm** is and invite pupils to share their own examples of algorithms
- Explain that when we **program** we use an algorithm (which is for humans to follow) to help us write a program in code – a programming language that a computer can follow
- Explain that the programming language pupils will be using in this activity is called **Scratch** which is made up of **commands** and other keywords. Discuss that a **command** instructs us to do something
- Explain that in this lesson we will be programming an animation of a Viking raid using the **commands** in Scratch and that it is important that the events happen in the correct sequence

Recapping about Viking raids (5 mins + depending on pupils' knowledge of Vikings)

- Ask pupils to discuss what they know about Viking raids
- If pupils need to recap their knowledge about Viking raids, see useful links
- Open the Scratch '**Viking raid example animation**' file on your interactive whiteboard
- Explain that within this program the Viking, Monk and Gold Coins are called sprites, which means they are objects we can program. These are the objects we will be writing a sequence of commands to control. Show pupils that each appears in the sprite selection window, as shown below

Note: a number of other sprites also load with the project (Thor, group of monks, axe and a storm cloud). These are for use in the extension exercise and are currently hidden when the project opens.

The Viking raid animation loads with a number of sprites.
Pupils will start by using the Viking, monk and gold in their animation.

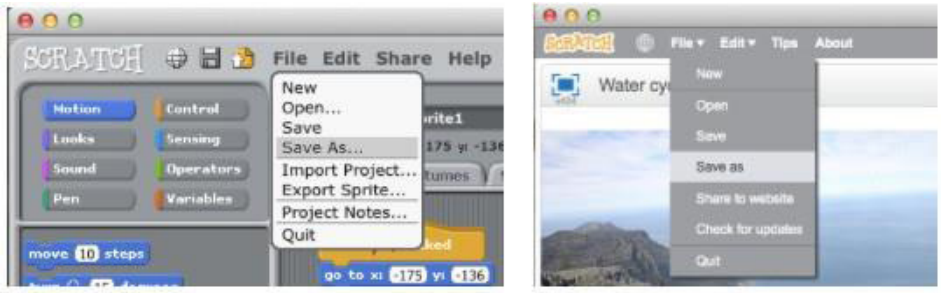
- Click the green flag to watch the animation
- After watching the animation, ask pupils why the **sequence** of this animation was important

Designing the algorithm for their animation (15 mins)

- Explain to pupils that the Scratch file they will use in this activity will load with the monastery background and three sprites (a Viking, monk and a pot of gold coins)
- Display slide 3 of the presentation and explain that the pupils' first task is to design their algorithm. Explain that their animation is to show a Viking raid using the three sprites and it is up to them to choose what happens. Pupils should include details on their algorithm about:
  – When and how the sprites move
  – When and what the sprites say
- Show the example from the model animation (slide 4) to illustrate to pupils one example of how the storyboard might be completed. Remind pupils they are free to choose what happens in their own animation

Introducing the Scratch commands (15 mins)

- Pupils should open the '**Viking raid animation**' Scratch file and save their own copy to an appropriate area on the school network with an appropriate file name



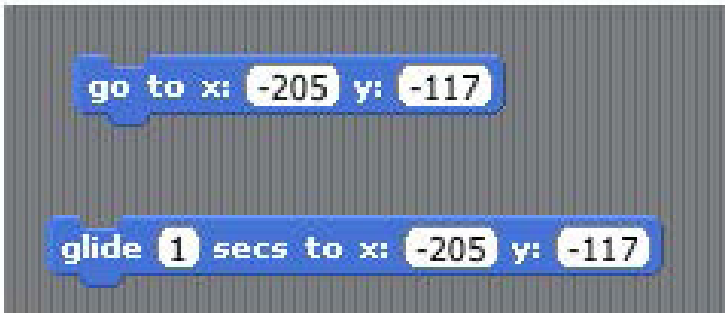
Selecting 'File' then 'Save as' in Scratch 1.4 (left) and 2 (right) to save a copy of the project.

- Ask pupils what two different actions they were asked to show on their algorithm design? **Answer: Sprites' movement and speech.** Explain that pupils are going to explore a selection of Scratch's movement and speech **commands** so they can turn their algorithms into programs within Scratch

Getting the sprites moving

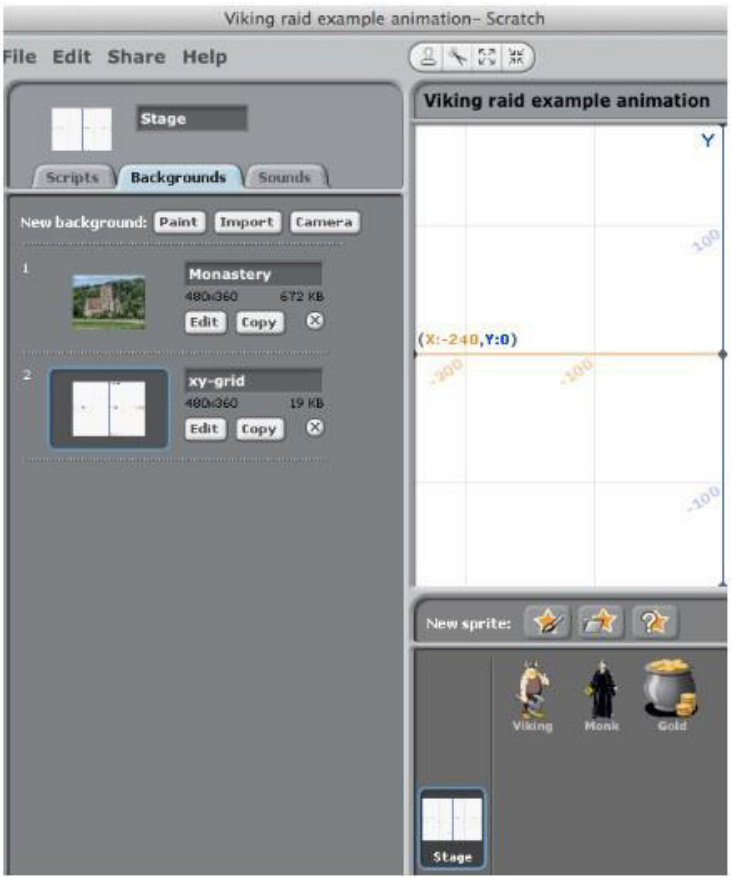
'go to X: Y:' & 'glide _ secs to X: Y:

- Explain to pupils that both the 'go to X: Y:' and 'glide _ secs to X: Y:' commands makes a sprite move to the coordinates entered. Show these two commands within the 'Motion' commands



Show the 'go to X: Y:' and 'glide _ secs to X: Y:'

- Recap that pupils will have used coordinates in mathematics. Note that pupils without a good knowledge of coordinates in four quadrants can still use these commands (see below) and this will help develop their understanding of coordinates. The 'Viking animation example' Scratch file loads with an alternative grid background to show the coordinate system. To show this to pupils, select 'stage' then 'background' and switch to X-Y grid, as shown on next page:

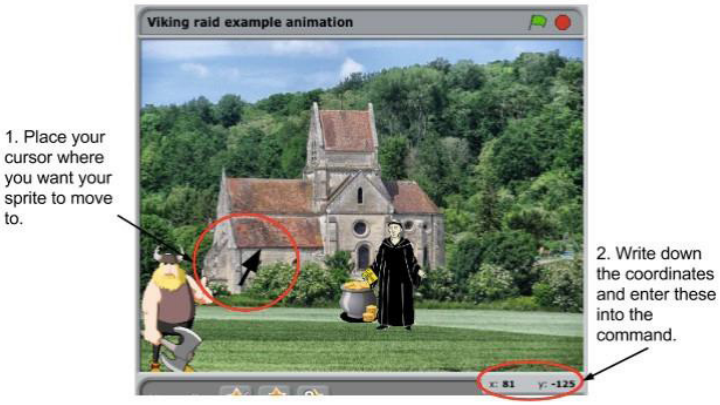


Switching to display the coordinate background

Finding the coordinates for the movement blocks (Scratch 1.4 and 2)

- Demonstrate to pupils that to find the coordinates that you want the sprite to move to, you follow the simple steps below:

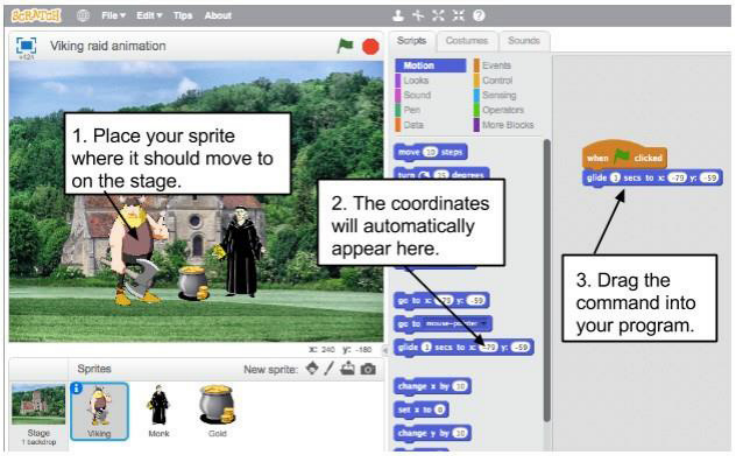
1. Add a movement command to your program ('go to X: Y:' or 'glide _ secs to X: Y:')
2. Place the cursor where you would like the sprite to move to within the stage.
3. Read the X and Y coordinates shown for that position.
4. Enter these into the movement command block you have added.



Two simple steps for finding the coordinates for use in the 'go to' and 'glide to' commands (Scratch 1.4 & 2)

Finding the coordinates for the movement blocks (Scratch 2 only)

- If your pupils are using Scratch 2, demonstrate there is a second way for determining the coordinates. Firstly, place the sprite in the location you wish it to move to, highlight that the coordinates in the movement commands automatically update so you can just drag the command into the script area ready to use



Two simple steps for finding the coordinates for use in the 'go to' and' glide to' commands (Scratch 1.4 & 2)

- **Pupils tinkering:** Give pupils a few minutes to tinker with adding a few instances of the movement commands so they become familiar with using these
- Lead a discussion with pupils to determine how the sprite moves differently with the 'go to X: Y:' and 'glide _ secs to X: Y:' commands. Discuss that the 'go to X: Y:' command causes a sprite to move instantly to that position, which makes it ideal for setting the starting position of each sprite. Show that the program for the monk and coins sprite in the example animation uses a 'go to X: Y:' command to set the initial position of the sprite
- Explain to pupils that ensuring sprites start in the correct position is one example of the need to set initial conditions in a program. Add that this is important to remember since programs that don't do this may run differently each time. For example, if we didn't reset a variable being used for a score in a game, we'd find when we played we were continuing with the previous player's high score

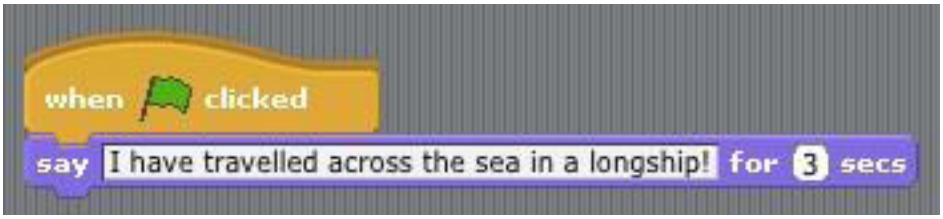
Getting the sprites to talk

'Say _ for _ secs':

- Show pupils the 'Say_for_secs' command from within the 'Looks' command blocks, as shown below

The 'Say _ for _ secs' command

- **Pupils tinkering:** Give pupils a few minutes to tinker with the 'Say _ for _ secs' command so they become familiar with using this. If you feel your pupils require further support, model adding this command within the Viking's script area beneath a 'When green flag clicked' command. This will illustrate how to add the text you wish the sprite to say and change the duration they will say it for, as shown below



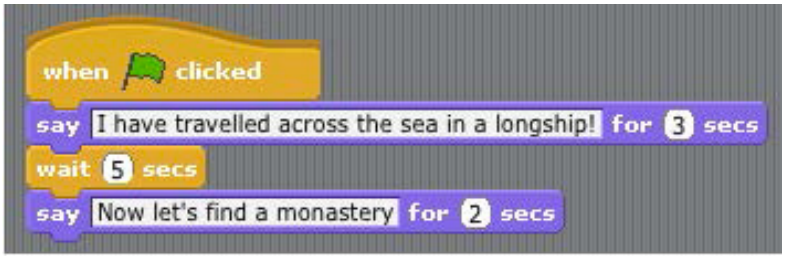
Model how to use the 'Say for _ secs' command in the Viking sprite.

'Wait _ secs'

- Show the 'wait_secs' command from the control blocks palette, as shown below

The 'wait_secs' command

- **Pupils tinkering:** Give pupils a few minutes to tinker with adding a few instances of the wait command (with other commands) so they become familiar with using this
- If you feel pupils need further support, demonstrate that it causes the program to wait for the time entered before moving on to the next command. This can be demonstrated by adding the command beneath the 'Say _ for _ secs' command you have just added to the Viking sprite with another instance of the 'Say _ for_ secs' beneath, as shown below. By running this program several times with different 'wait times' entered you can illustrated how the command holds the program for the time entered

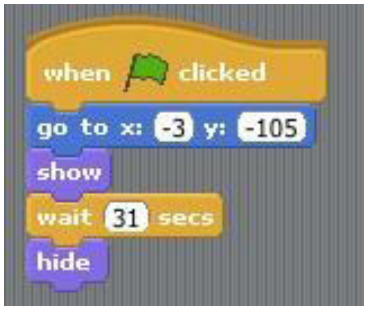


Illustrate the 'wait_sec' command.

- Explain that this command is useful in ensuring that the programs for different sprites are correctly sequenced so they move and speak at the right time

Hide & show

- Finally, give pupils time to tinker with using the 'hide' and 'show' commands from the 'Looks' command menu. If they require further support explain to pupils that in the example animation program, the 'hide' command was used to make the coins sprite disappear so that it looked like the Viking had collected it. This can be shown within the coin sprite's program, as shown below



Show the use of the 'show' and 'hide' sprite within the coin sprite.

Pupils program their animations (20 mins)

- Pupils should work in pairs to write their Viking raid animation as a program within Scratch
- A Scratch commands help sheet has been provided to support pupils during this part of the activity if they require it (see 'Resources')
- Remind pupils that it can be useful to run their program as they are writing it to check it matches their algorithm design and **debug** it as necessary

Plenary (10 mins)

- Give pupils time to move around the class and watch a selection of each others' animations. When sharing their animations with others, pupils should firstly show their algorithm to illustrate how they used it to help create their program. After watching an animation pupils should feedback two things they think pupils have done well and one point for improvement. Slide 5 of the presentation has a range of questions to support pupils in giving this feedback
- Use think-pair-share to guide a class discussion with the questions on slide 6

Differentiation

Support

- Mixed ability pairing should be used to ensure less confident pupils are encouraged by their peers. An additional adult, if available, may work with a small group of pupils to provide additional support. They may plan the algorithm together as a group and work through implementing this as a more guided activity

Stretch & Challenge

- An extension activity sheet has been provided which encourages pupils to tinker with using repetition to control the movement of additional sprites within their animation

Assessment opportunities

- Informal teacher assessment of progress during the activity. Focus on: pupils' understanding of importance of correctly sequencing algorithm and code; their ability to correctly use the Scratch commands introduced; and that they can logically 'think through' their code to debug if required
- Formal, summative assessment of Scratch projects if required (in pairs)
- Peer assessment and feedback on each others' animations during the plenary

Teaching notes

Concepts and approaches

- Pupils write the **algorithm** for their animation. In doing so they focus on correctly **sequencing** the events
- Pupils implement their algorithm as code in Scratch and in doing so are **programming**
- Pupils **tinker** with code to build up their understanding of different Scratch commands
- When the output of their program doesn't match their algorithm pupils **debug** their code
- Pupils **collaborate** with a partner as they create their programs in pairs

Curriculum links

Computing

- use sequence in programs

History

- Pupils should be taught about the Vikings. This could include Viking raids (non-statutory).

Resources

- MIT's **Scratch 1.4** or **Scratch 2.0** (See **this guide**)
- Interactive whiteboard
- Viking raid presentation (see 'downloads')
- Viking raid algorithm design sheet (see 'downloads')
- Scratch commands help sheet (see 'downloads')
- Viking raid extension (see 'downloads')
- Pupil access to the Scratch resources. Either downloaded from the download link at the bottom of this webpage or within the Scratch 2 online editor from these links: **Viking raid example animation**; **Viking raid animation**

Web pages on Viking raids

BBC Schools web page about the Vikings

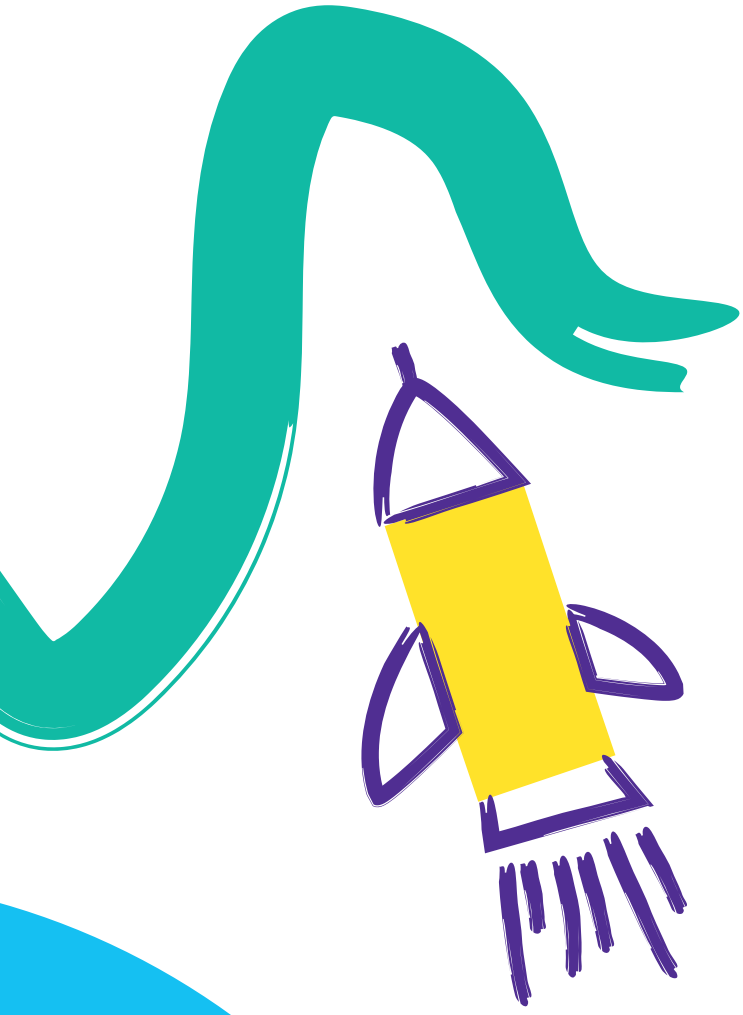
Horrible Histories song about the Vikings

Related activities

Lower KS2 dinosaur fossil animation sequence activity

KS2 Shapes and Crystal Flowers repetition activity

Get more **Barefoot**



Have you had a great Barefoot workshop, or delivered a fun computer science lesson? Send us your comments and pictures via our social channels to help get more teachers involved!

@BarefootComp

/barefootcomputing

barefootcomputing.org

You can start your journey here

Principal partners

BT · Computing at School

Supported by

bcs · openreach · Department for Education · Education Scotland Foghlam Alba · Gweithio mewn partneriaeth gyda Llywodraeth Cymru Working in partnership with Welsh Government · Department of Education www.education-ni.gov.uk

Version 1118 Expires Dec 2021