Module 3: Interacting Sprites

Investigation 1
Animating Sprites

Investigation 2
Encountering Conditions

Investigation 3
Broadcasting Messages

Investigation 4
Interactive Stories

Introduction to Module 3

Module 3 is focused around building interactive behaviours between multiple sprites firstly using conditions and then broadcasting. This module could potentially be linked with several different areas of the Key Stage 2 curriculum including art and design and English.

Art and Design: Animation

The first two investigations introduce animation in Scratch and could be be linked with exploring different forms of animation within art and design.

English: Poetry and Storytelling

There are multiple links to the English curriculum throughout this module. The first and fourth investigations both involve elements of narrative creation and storytelling. The unplugged activity in the third investigation also introduces the concept of broadcasting through rhyme or poetry.

Key Vocabulary and Concepts Covered by Module 3

Scratch	Computing	Mathematics
▶ **when this sprite clicked** block	▶ Multiple actors	▶ Coordinates
▶ **hide** and **show** blocks	▶ Events	▶ Positive and negative numbers
▶ Graphic effects block	▶ Parallel behaviours	▶ Multiplication and division
▶ **change by …** and **set to …** blocks	▶ Repetition	▶ Factors
▶ **forever** block	▶ Multiple costumes and Animation	▶ Random numbers
▶ **if on edge, bounce** block	▶ Conditions and Conditional loops	▶ Problem solving
▶ **point towards…** block	▶ Basic Selection	▶ Mathematical modeling
▶ **repeat until...**, **touching…** blocks	▶ Expressions	
▶ **if … then …** block	▶ Broadcasting and Receiving messages	
▶ **… < …**, **… > …** blocks		
▶ **broadcast** blocks		
▶ **say…** blocks		

Map Of Module 3

	Activity 1	Activity 2	Activity 3	Activity 4
Investigation 1 **Animating Sprites** 	**Multiple Sprites** Starter project: **31-Multiple Sprites**	**Teleporting Nano** Continue with **31-Multiple Sprites**	**Jumping Tera** Continue with **31-Multiple Sprites**	**Walking Pico** Continue with **31-Multiple Sprites**
Investigation 2 **Encountering Conditions** 	**Repeat until** Continue with **31-Multiple Sprites** or start with **32-Multiple Sprites**	**Touching Colour?** Continue with **31-Multiple Sprites**	**Walking in the air** Continue with **31-Multiple Sprites**	**Unplugged: True or False?**
Investigation 3 **Broadcasting Messages** 	**Unplugged: Broadcast & Receive**	**Introduction: One to One** Continue with **31-Multiple Sprites** or start with **33-Multiple Sprites**	**Come to Tera: One to Many** Continue with **31-Multiple Sprites**	
Investigation 4 **Interactive Stories** 	**Unplugged: Reading Scripts**	**The Story of the Sprites** Continue with your **31-Multiple Sprites** or start with **34-Multiple Sprites**		

The **red** dashed line indicates the **core** activities which are important to complete before moving on to the next module.

For activities which require pupils to continue with a project from a previous lesson you can alternatively use the suggested 'INT' (intermediate) project for those pupils who do not have a project to continue with or if you wish all pupils to begin from the same point.

Contents Of Module 3

Text

MODULE 3: Interacting Sprites

Important note about Starter Projects for the module

The same starter project, **31-Multiple Sprites**, is used throughout Module 3 and pupils are expected to build on their project during each investigation. This enables pupils to develop their initial scripts into more complex behaviours over multiple lessons, but there may also be issues ensuring everyone is at the same point if pupils miss lessons or lose their work. Therefore we recommend that you also have a starter project that includes all of the scripts from the previous investigations available for those pupils who have not got their own version of the project. See below for further details.

Investigation 1
Animating Sprites

All pupils should start this investigation with the **31-Multiple Sprites** project which contains the four sprites and no scripts.

Please note that the starter projects (32-34) below include all scripts from the previous investigation(s) including extensions.

Investigation 2
Encountering Conditions

Pupils who did not complete investigation 1 should start this investigation with the **32-Multiple Sprites2** project which contains the following scripts:

- Nano: *Setup script; Teleporting script*
- Tera: *Setup script; Jumping script*
- Pico: *Setup script; Walking script*
- Giga: *Setup script*

Investigation 3
Broadcasting Messages

Pupils who did not complete investigation 2 should start this investigation with the **33-Multiple Sprites** project which contains the following scripts:

- Nano: *Setup script; Teleporting script*
- Tera: *Setup script; Jumping script*
- Pico: *Setup script; Walking script with conditions*
- Giga: *Setup script; Walking script with conditions*

Investigation 4
Interactive Stories

Pupils who did not complete investigation 3 should start this investigation with the **34-Multiple Sprites** project which contains the following scripts:

- Nano: *Setup script; Teleporting script with broadcasts*
- Tera: *Setup script; Jumping script with broadcasts*
- Pico: *Setup script; Walking script with conditions/broadcasts*
- Giga: *Setup script; Walking script with broadcasts*

MODULE 3: INVESTIGATION 1

Animating Sprites

OVERALL LEARNING OBJECTIVE: Explore projects with multiple sprites and define scripts to run their individual behaviours, initiated by different events (sequential and parallel) and bridging to knowledge of coordinates. Use multiple costumes to animate a sprite.

This investigation introduces the concept of multiple sprites and parallel scripts, events, hide and show plus other graphic effects as well as using repeat to control the speed of the sprite's behaviour. It also introduces sprites jumping and walking in the stage, using different costumes to animate their shapes.

- **Activity 3.1.1** – Multiple Sprites
- **Activity 3.1.2** – Teleporting Nano
- **Activity 3.1.3** – Jumping Tera
- **Activity 3.1.4** – Walking Pico

Activity 3.1.1	Activity 3.1.2	Activity 3.1.3	Activity 3.1.4
10-15 mins	20-25 mins	20-25 mins	20-25 mins

We recommend allowing **80 to 100 minutes** for this investigation.

Scratch starter project **31-Multiple Sprites**

LINKS TO PRIMARY NATIONAL CURRICULUM

CURRICULUM OBJECTIVES

Computing

Design, write and debug programs that accomplish specific goals.

Solve problems by decomposing them into smaller parts.

Use repetition in programs.

Mathematics

Describe positions on the full **coordinate grid** (all four quadrants).

Identify **factors** of a number; use **negative numbers** in context.

(KS3) Work with experiments that involve **randomness**.

LINK WITH SCRATCHMATHS

- Pupils are required to build scripts to ensure their sprites react in a certain way to specific events;
- Pupils are required to build the behaviour of each sprite in multiple small parts;
- Pupils are required to experiment with the **repeat** block to alter the speed of the sprite's behaviour.
- Pupils are required to use their knowledge of the coordinate grid to control where a sprite can appear on the stage;
- Pupils are required to use their knowledge of negative numbers as well as the different factors of a number to control the speed of a graphic effect and movements of sprites;
- Pupils are required to work with random numbers to tell a sprite to move to a random position on the stage.

Module 3 • Investigation 1 • Activity 3.1.1
Multiple Sprites

Learning Objectives

Explore how to make a sprite react to being clicked.
Explain two different ways of executing a script.

Activity Instructions

1. Pupils open project **31-Multiple Sprites**, save as a copy (online)/save as (offline) and rename.
2. Pupils explore the project and discuss what they can see (see the top three discussion points below). They click each of the sprites and discuss what happens and why.
3. Pupils then try clicking on the green flag and discuss what happens and why (note – there is no script to define any behaviour, so nothing happens whatever is clicked).

The **31-Multiple Sprites** project will be used throughout the whole of Module 3, building behaviours for the different sprites. Therefore a *setup script* needs to be built to return each sprite to their **initial positions** when the green flag is clicked, i.e. the positions they started in when the project was opened (more blocks will be added to these scripts later in the module).

4. Pupils add the block **when green flag clicked** to each of the sprites and snap the **go to x: … y: …** block to this hat block, ensuring the x and y values are set to the correct initial position.
5. **[Extension]** Alternatively, pupils may replace the **go to x: … y: …** block with the **glide … secs to x: … y: …** block with the same coordinates. They can experiment with different times (e.g. 0.5, 1 or 2 secs), dragging the sprites from their initial positions then clicking the green flag.

These four sprites are going to pop up in the activities throughout the module, so first we should create a back story about who they are and how they know one another.

6. As a class decide on a short back story for the four sprites: Where are they from? How do they know the other sprites? What do they like doing?

Vocabulary

The **when this sprite clicked** block is a **hat block** and represents an **event**: when its sprite is clicked within the stage area the script that is attached to this hat block will be run.

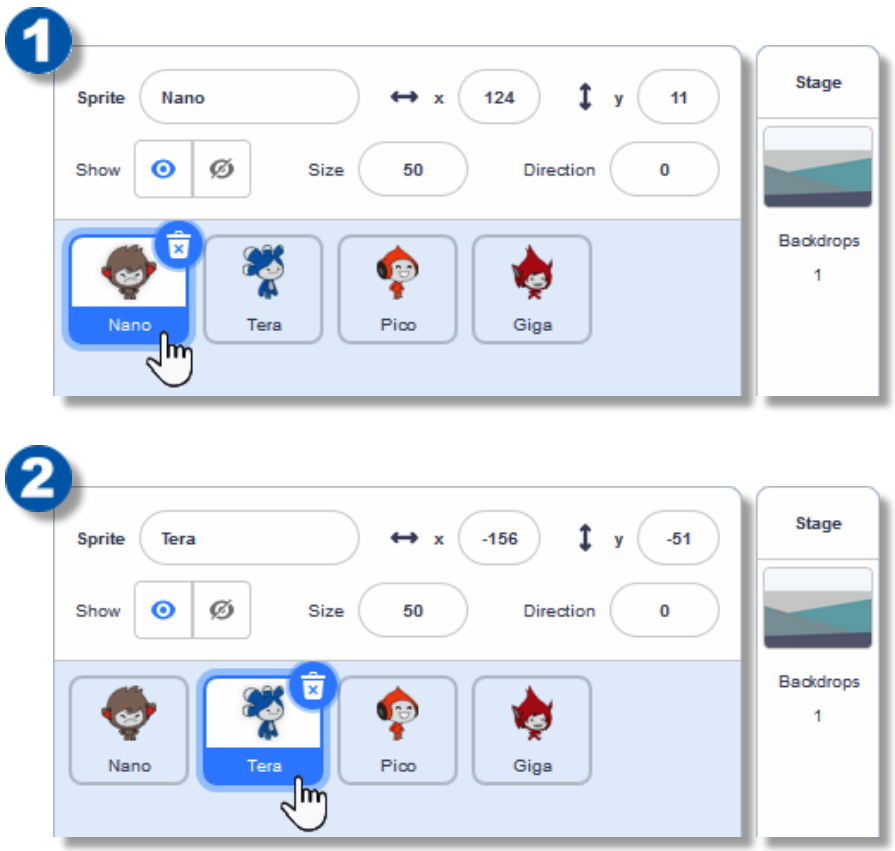
Discussion Points

- How many sprites are there in the project?
- What are the names of the sprites?
- What can each of the different sprites do?
- Why do the sprites not react when clicked? Where do the reactions come from?
- What are the two different ways of executing **when green flag clicked** script(s) that we have seen so far?

ADDITIONAL SUPPORT

Note that each sprite has a name, its own costume, and also **its own scripts area**. To build the script(s) or explore them, we first have to click that particular sprite's icon (see 1 and 2 bellow). When we click the icon, the scripts of that sprite are displayed in the scripts area.

Each script **belongs only to one sprite**.



Finding the coordinates of a sprite

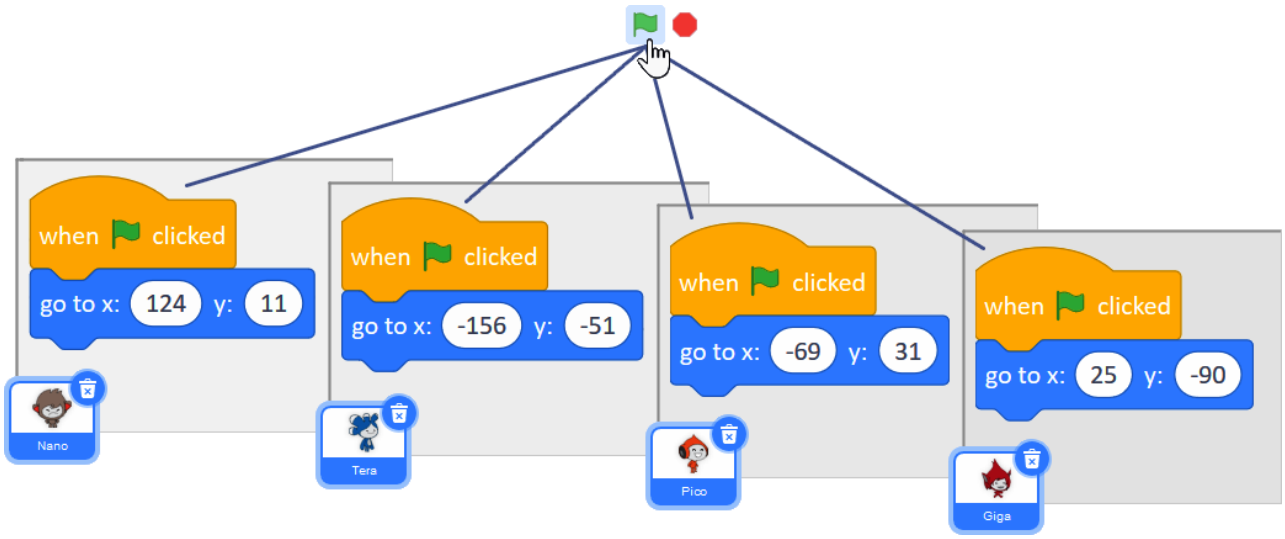
Within the **Motion** group, the **go to x: … y: …** block is automatically set with the current coordinates of that sprite. Drag the sprite within the stage and note that the coordinates in the **go to x: … y: …** block (and also in the **glide to …** block) are updated whenever you release the mouse. To create a *setup script* to return the sprite to its initial position, ensure the sprite is in the position you want it to return to, drag the **go to x: … y: …** block to the scripts area and snap it onto the *setup script*.

continued ▶

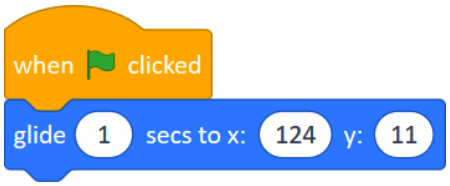
ADDITIONAL SUPPORT CONTINUED

Below are the example scripts to return each sprite to its initial position. Note that:

- If we click one of these scripts directly in the scripts area, only that script is run.
- However, if we click the green flag, all **when green flag clicked** scripts (four in this case) are run **simultaneously**, i.e. **in parallel**.



[Extension] Alternative with the **glide to x: … y: …** block.



MODULE 3 • INVESTIGATION 1 • ACTIVITY 3.1.2
Teleporting Nano

LEARNING O

Explore how to make a sprite jump to a random position.
Explain different strategies for how to hide and show a sprite.
bridgE to mathematical fluency (using number fact knwoledge/choosing appropriate strategy).

ACTIVITY INSTRUCTIONS

1. Pupils continue in their project **31-Multiple Sprites**.

The following activities develop different behaviours for each of the sprites. We start with Nano who has a special ability – **he can vanish and reappear somewhere else on the stage whenever he is clicked**.

2. Pupils select Nano, drag the **hide** and **show** blocks from the **Looks** group in the scripts area and explore them in isolation.
3. Pupils build a script to initiate Nano's special ability, starting with the **hide** block. They add the **go to x: … y: …** block and the **wait 1 secs** block (otherwise Nano would reappear instantly). They set the x and y positions to the location on the stage they want Nano to teleport to. Finally they add the **show** block to the bottom of the script.
4. Pupils run the script by clicking it. If it works correctly they drag the **when this sprite clicked** hat block into the scripts area and snap it to the top of the script (for basic teleporting script see 1 in additional support).

However, this teleporting script will work only when Nano is clicked for the first time. Each additional click will then make Nano reappear in the same place.

5. To make Nano always reappear **in a random position**, pupils will replace the values of both x and y in the **go to x: … y: …** block with the **pick random … to …** blocks. In each of them they set the furthest points of the stage (from left to right for x and from bottom to top for y) where they would want Nano to reappear (see 2 in additional support).
6. **[Extension]** Pupils make Nano disappear and reappear **slowly** by using the **change ghost effect by …** block. They build their own block **disappear** and use it instead of the **hide** block. Then they build their own block **reappear** for the **reverse process** and use it instead of the **show** block (see 3 and 4 in additional support).

THINGS TO NOTE

- The x axis ranges from -240 to 240 and the y axis from -180 to 180.
- Clicking the red stop sign will reset all graphic effects (including the ghost effect).

VOCABULARY

The **hide** block will hide the sprite on the stage if it is currently shown.
The **show** block will show the sprite on the stage if it is currently hidden.

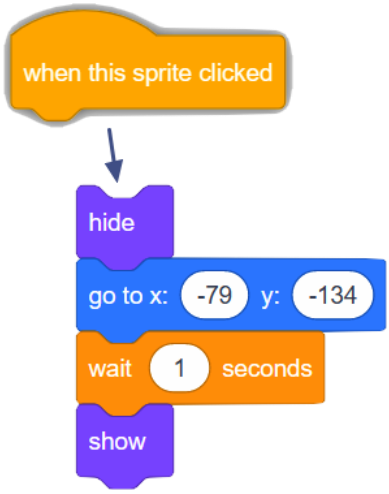
DISCUSSION POINTS

- What exactly does the **hide** block do? What would happen if the **show** block is removed from your script – where would Nano be?
- What other way could you 'hide' the sprite? (*setting its ghost effect to 100*)
- What values did you select for x and y? How did you select these values?
- What would happen if you selected only negative numbers for x and y?

ADDITIONAL SUPPORT

Encourage pupils to start by dragging the **hide** and **show** blocks into the scripts area and explore them first as isolated blocks, before using them in the script.

1 Basic teleporting script. This solution makes Nano always reappear in the **same** place on the stage.

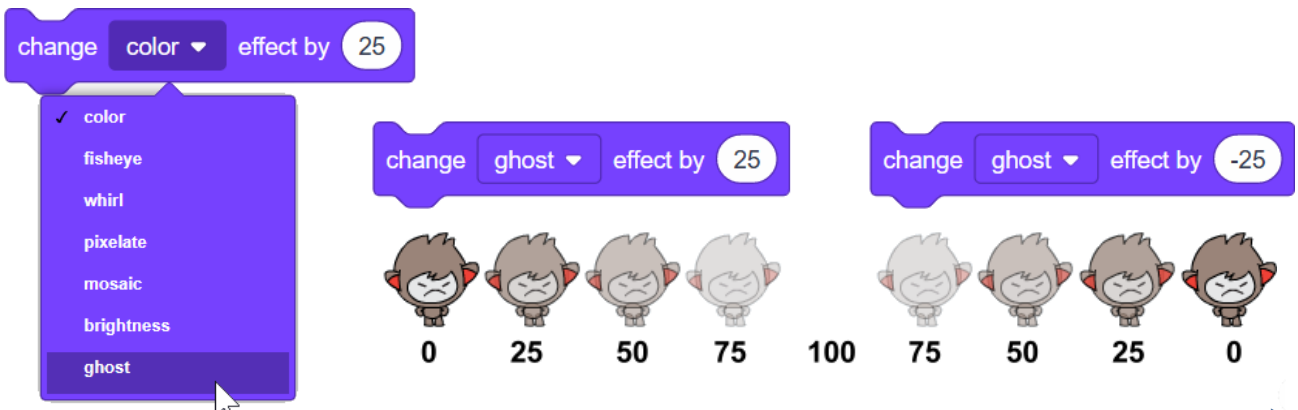


2 Improved teleporting with jumping to random x and y positions.

[Extension] First explore the **change ghost effect by …** block of the **Looks** group by itself. Note that the ghost effect varies between 0 (no ghost effect) and 100 (full ghost effect, costume invisible – which is different from being hidden). **3**

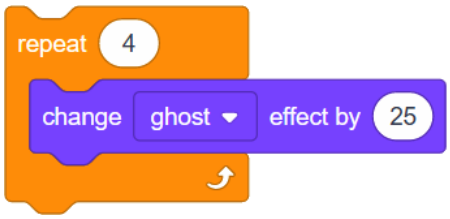
Click the **change ghost effect by …** block again and again. Then explore the **reverse process** by changing 25 into -25.

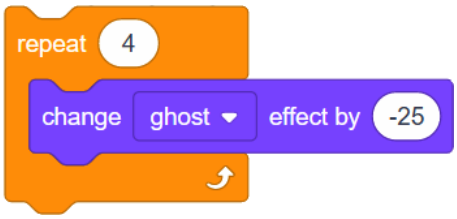
Put the **repeat** block around the **change ghost effect by …** block to enable these processes to happen in one click.



continued ▶

ADDITIONAL SUPPORT CONTINUED

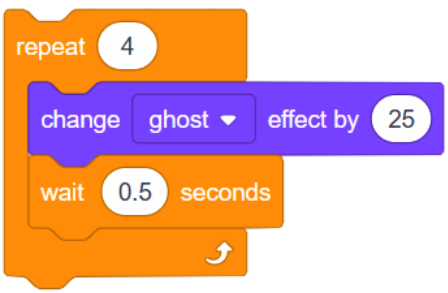


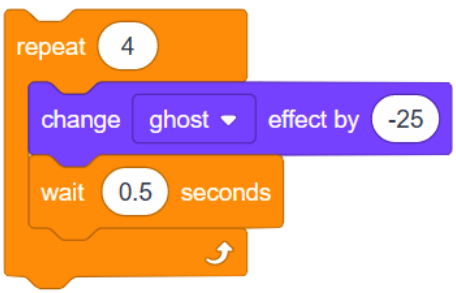


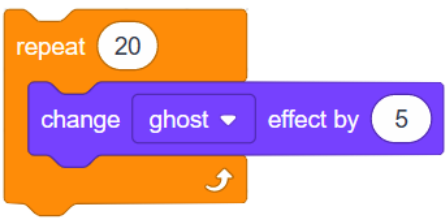
4

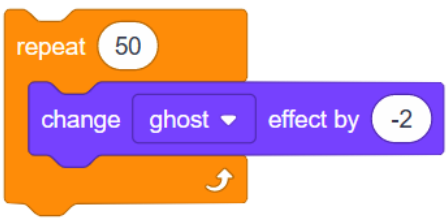
To make the process of disappearing and reappearing slower the **wait … secs** block with a small number can be added inside the **repeat**. However, the process is not smooth using this block.

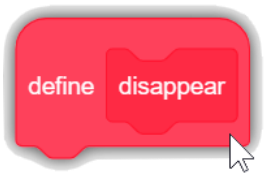
The speed of Nano disappearing and reappearing can be made smooth and slow (or even very slow) by modifying the number in the **repeat** block and the **change ghost effect by …** value.

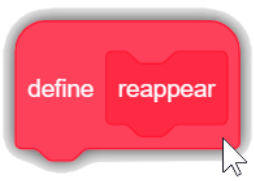


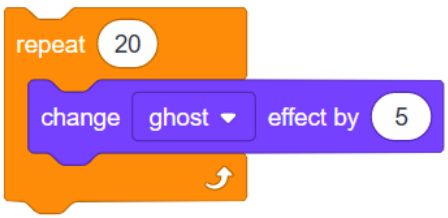


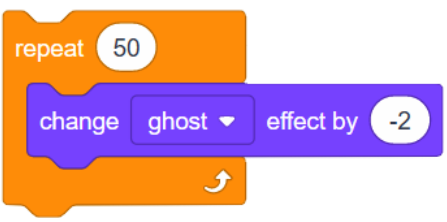












continued ▶

EXTENSION TO ACTIVITY 3.1.2

Pupils explore the difference between the **change ghost effect by …** and **set ghost effect to …** blocks. They drag them as isolated blocks in the scripts area, set both values to 10 and click each block repeatedly several times. Pupils then report back and discuss their findings.

Pupils explore the **different costumes** of Nano and use the **switch costume to …** blocks at the beginning and at the end of the teleporting process.

MODULE 3 • INVESTIGATION 1 • ACTIVITY 3.1.3
Jumping Tera

LEARNING OBJECTIVES

Explore how to change the y position to move the sprite up and down.
Explain how the sprite can be moved at different "speeds".
bridgE to mathematical fluency (using number fact knowledge) and reasoning (doing/undoing).

ACTIVITY INSTRUCTIONS

1. Pupils continue in their project **31-Multiple Sprites**.

Tera also has a special ability – **she can jump really high, then get back down to the same position whenever she is clicked**.

2. Pupils select Tera and explore one isolated **change y by …** block, changing the values in it and clicking it. Some pupils may try negative inputs as well.
3. Pupils then start building a script to initiate Tera's special ability. In the **change y by …** block they set the value they want Tera to jump (e.g. 100). Next they drag a second **change y by …** block into the scripts area, keeping it isolated from the first one, type in the value to allow Tera to **return to her starting position**, then explore both blocks by clicking them alternately.
4. Pupils snap together both **change y by …** blocks and click this short script to see what happens (it appears that nothing happens as Tera jumps up and down almost instantly). Therefore they add in the **wait 1 secs** block between two **change y by …** blocks to see Tera jump (see additional support). Pupils add the hat block **when this sprite clicked** on top of their script and test it by clicking Tera.

We now want to make Tera jump back more smoothly – as if she *floats in the air*.

5. Pupils add a **repeat** block around the second **change y by …** They edit their script to move Tera down more slowly but still the same total distance, for example by changing y by -5 each time and repeating this 20 times, or changing y by -2, repeating it 50 times (see additional support).
6. **[Extension]** Pupils make Tera jump up smoothly as well, but not as slowly as when floating down. They add one more **repeat**, this time around the first **change y by …** block and set the correct values both in **repeat** and **change y by …** .
7. **[Extension]** Pupils use different costumes within her jump (see additional support).

THINGS TO NOTE

- If Tera starts her jump too high on the stage her jump will be cut short by hitting the top edge of the stage.
- Be careful to not mix up the blocks **change y by …** with **set y to …**

VOCABULARY

The **change y by …** block changes the sprite's y position by the specified amount.

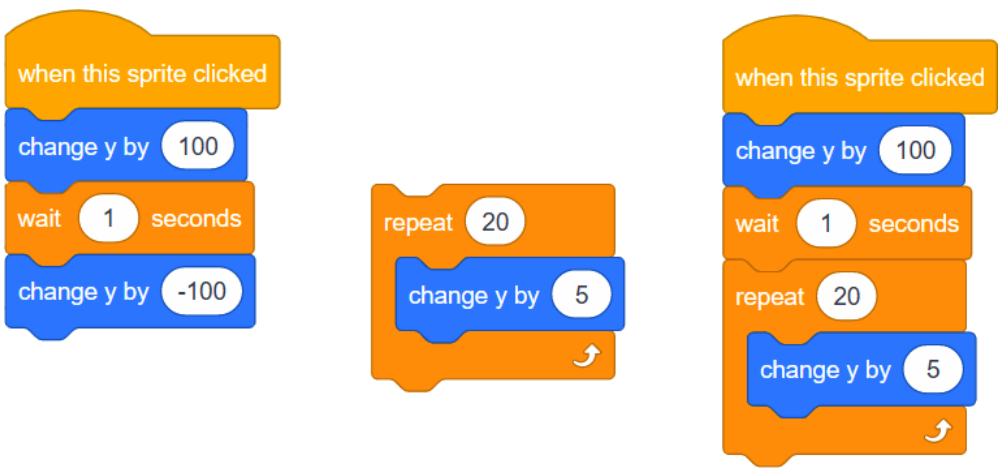
DISCUSSION POINTS

- How did you ensure Tera returned to the same position that she started her jump from? What inputs did you try?
- What happens when you drag Tera somewhere else on the stage and click her?
- Does Tera jump and float back at different speeds?
- How did you calculate the **repeat** and **change y by …** values to float Tera back?

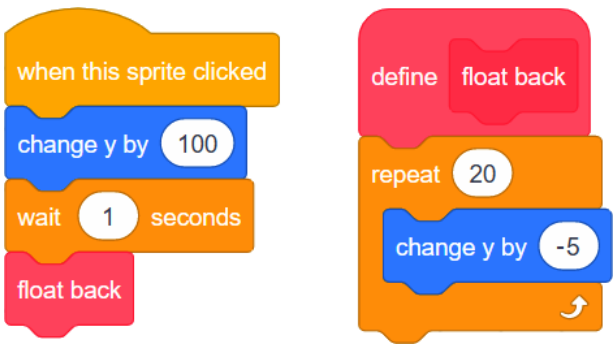
ADDITIONAL SUPPORT

Encourage pupils to start by exploring the two **change y by …** blocks **as isolated blocks**, before snapping them into a script.
Below is the basic jumping script.

The **change y by -100** block can be replaced by repeating smaller steps several times and thus making the movement of Tera smoother and slower:



Alternatively, pupils can build a new block **float back** and use it in the script:



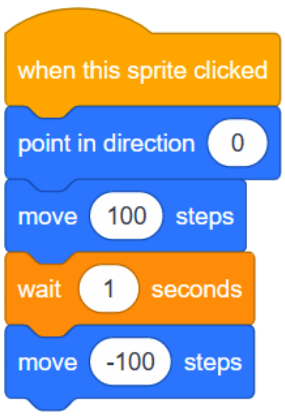
continued ▶

ADDITIONAL SUPPORT CONTINUED

3 – Extended solution with both jumping up and floating back in a smooth way, with different 'speeds of movement'. Note that floating back is now so slow that the **wait** block can be removed altogether. Both repeats are turned into new blocks **jump high** and **float back**.

4 – Extended version using several costumes of Tera. The whole jumping is now turned into a single new block **jump**.

Be careful not to click Tera if you have two or more experimental/alternative **when this sprite clicked** scripts. They will react (i.e. be run) in parallel and may interact in a confusing way. To avoid this, always click a particular script to run it (instead of the sprite). When your explorations are finished, delete all of the scripts that you no longer want to use.



Discussing alternative solutions

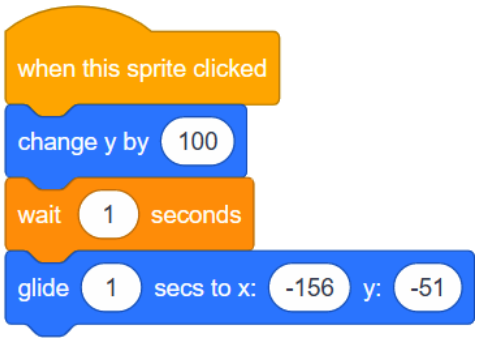
The solution on the left has been used in the previous modules when we were moving the Tile and Beetle sprites. Although it is perfectly correct, with Nano and Tera we want to employ an alternative approach using coordinates: instead of moving by the **move** block, we want these sprites to *jump* by applying the **change y by …** and **change x by …** blocks where:

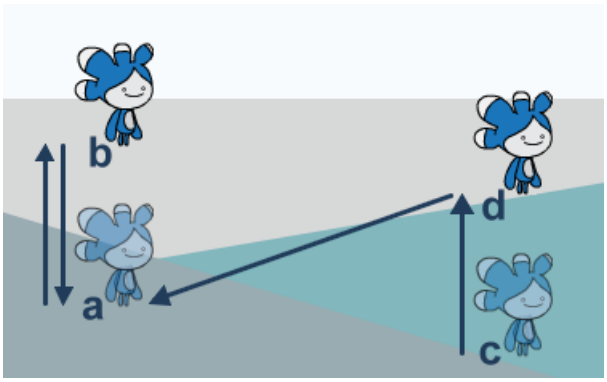
change y by … means *jumping up or down*
change x by … means *jumping left or right*

continued ▶

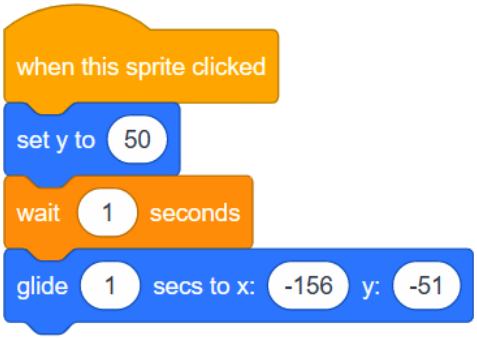
ADDITIONAL SUPPORT CONTINUED

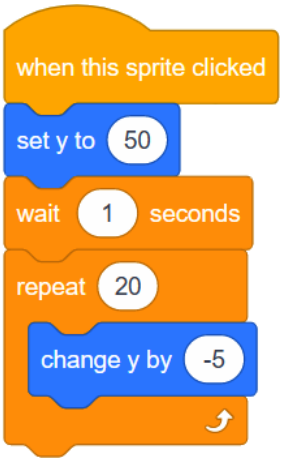
Using a coordinates approach does not necessarily mean jumping to any particular position specified by fixed coordinates. For example, the two solutions on the left both make Tera jump from position **a** to **b** then back to **a**.





However this will not work if Tera is dragged somewhere else, e.g. to **c**. When Tera is clicked there, she will jump to **d** by changing her y coordinate by 100. But then, instead of floating back to **c**, she will jump or glide back to **a**.





In the solutions above, instead of using **change y by …** the height of the jump, the exact y coordinate is specified by **set y to …** . Pupils can experiment with these scripts and find out why Tera would not float back to her starting position – if she has been dragged somewhere else in the stage before she is clicked.

MODULE 3 • INVESTIGATION 1 • ACTIVITY 3.1.4
Walking Pico

LEARNING OBJECTIVES

Explore how to switch between multiple costumes to animate a sprite.
Explore how to run a script forever.
Explain how to ensure a sprite stays within the stage area.
bridgE to mathematical reasoning (always, sometimes, never).

ACTIVITY INSTRUCTIONS

1. Pupils continue in their project **31-Multiple Sprites**.

Pico has a special ability – **he can walk around the stage**.

2. Pupils select Pico and explore his costumes in the Costumes tab by clicking on them one by one to see how he can be animated i.e. how he can be made to walk.
3. Pupils drag the **next costume** block into the scripts area (keeping it isolated), then click it repeatedly and observe Pico.
4. They add a **repeat** block around this block and set its value to e.g. 100.
5. Pupils make Pico walk by adding a **move** block inside **repeat** and set its value to a small number, e.g. 1 or 2. They click the script and observe the behaviour. They may decide to add a **wait** block with very short wait time, e.g. 0.1 or 0.2 seconds. They then add the **when this sprite clicked** hat block on top of their script (see additional support).
6. Pupils click on Pico and observe him walking. They click Pico again to see what happens when he touches the edge of the stage. They add the **if on edge, bounce** block within the **repeat** block and then change the **repeat** block to the **forever** block (see additional support).
6. **[Extension]** Pupils explore different values in the **move** and **wait** blocks to make Pico walk faster or slower (walk, stride, march, run… see 3 in additional support for scripts).

THINGS TO NOTE

- To stop Pico walking click on the red stop sign above the stage.

VOCABULARY

The **if on edge, bounce** block checks if the sprite is touching the edge of the stage. If true it reverses the direction the sprite is pointing.

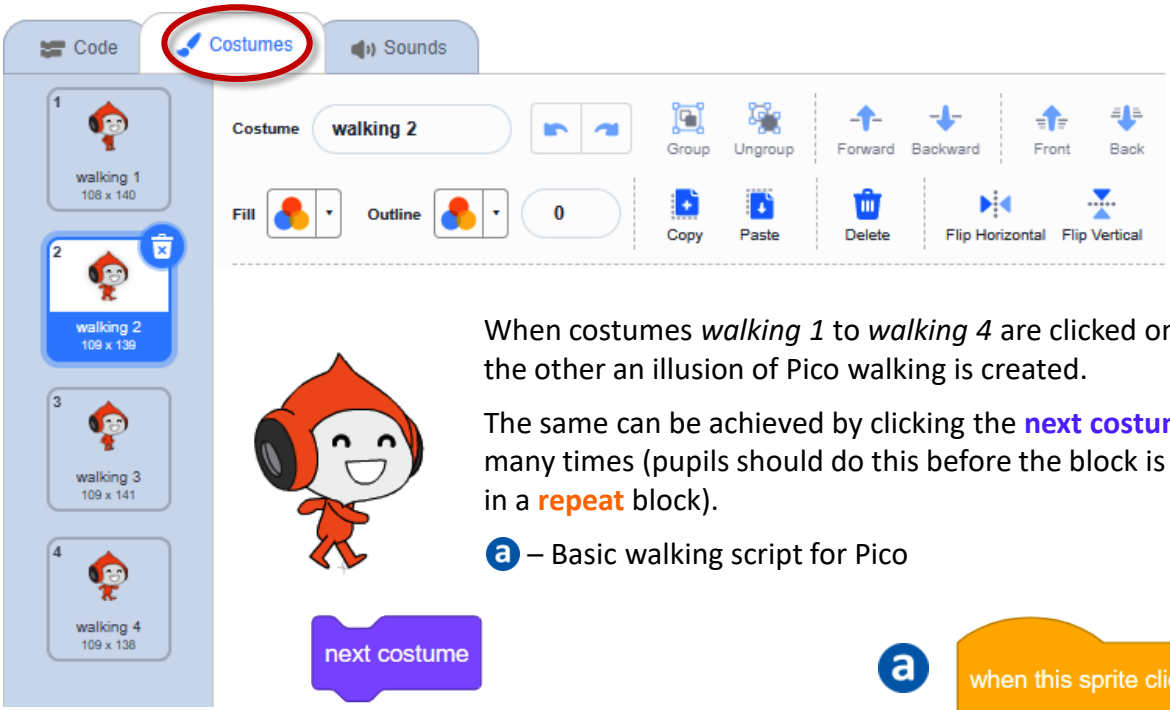
Animation is an illusion created by showing different costumes in sequence.

The **forever** block loops through the blocks within it (unless the stop sign is clicked).

DISCUSSION POINTS

- How is the illusion of walking created?
- What happens when the sprite touches the edge? What does bounce mean?
- What happens if the **if on edge, bounce** block is placed outside of the **repeat** block, before or after?
- What is the difference between the **repeat** and **forever** blocks?
- Why can you not connect any blocks to the bottom of the **forever** block?
- How did you make Pico walk faster or slower?
- What direction does Pico walk in when he bounces off the edge of the stage? How does the bouncing work?

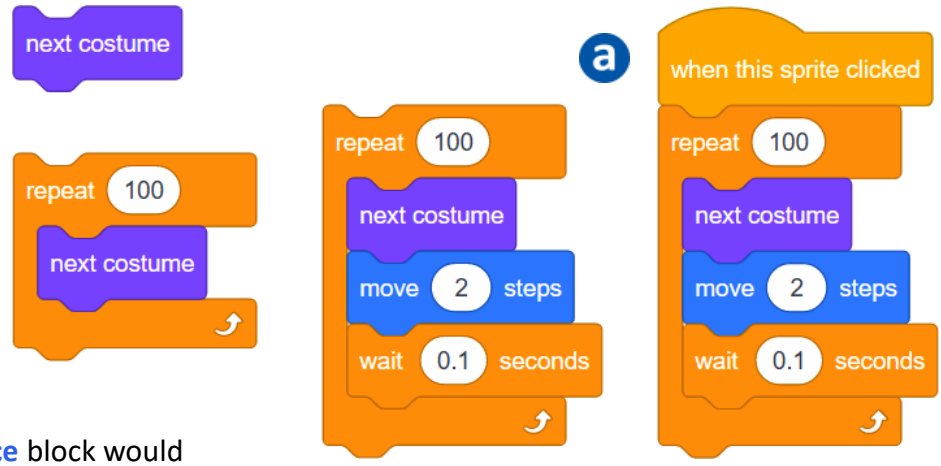
ADDITIONAL SUPPORT



When costumes *walking 1* to *walking 4* are clicked one after the other an illusion of Pico walking is created.

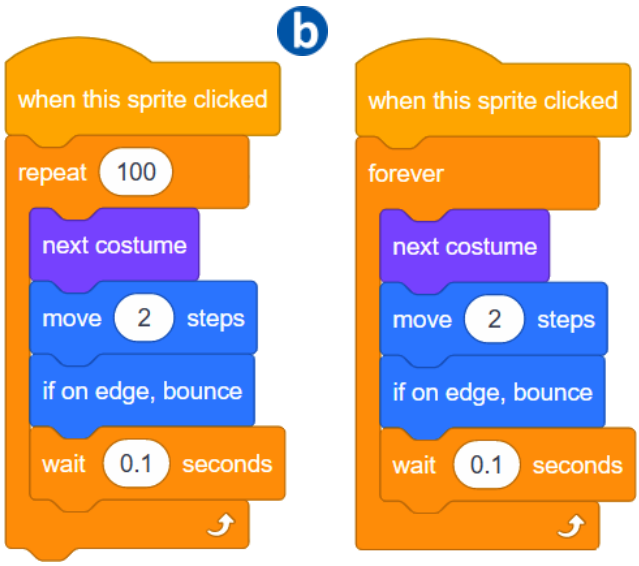
The same can be achieved by clicking the **next costume** block many times (pupils should do this before the block is inserted in a **repeat** block).

a – Basic walking script for Pico



The **if on edge, bounce** block would be run only once if we add it on top or below the **repeat** block. If we want Pico to bounce whenever he touches the edge, this special block must be added inside **repeat**. Think of the **forever** block as a version of **repeat** with a "very big" value of repetition.

b – Walking and bouncing forever

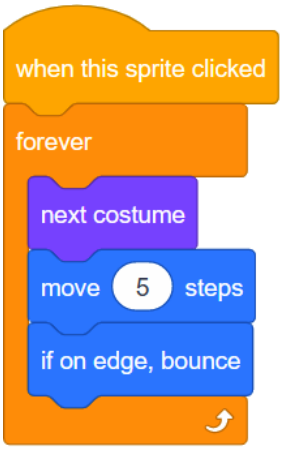


continued ▶

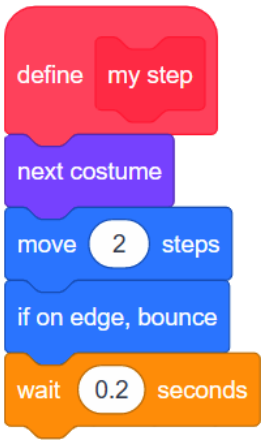
ADDITIONAL SUPPORT CONTINUED

c – Running and marching scripts
d – Solution with a new block

c



d



MODULE 3: INVESTIGATION 2
Encountering Conditions

OVERALL LEARNING OBJECTIVE: Explore conditions and the ways they are used to control the flow of the sprites' behaviour. Discover how to sense the current situation of a particular sprite.

This investigation introduces conditions and two new control structures: the first control structure runs a block or several blocks only if a certain condition is *true*, and the second repeats several blocks only until certain condition is *true*. These structures allow the building of scripts which sense the current situation of the sprites and then react accordingly.

- **Activity 3.2.1** – Repeat until…
- **Activity 3.2.2** – Touching Colour?
- **Activity 3.2.3** – Walking in the Air
- **Activity 3.2.4** – Unplugged: True or False?

Activity 3.2.1	Activity 3.2.2	Activity 3.2.3	Activity 3.2.4
10-20 mins	10-20 mins	15-20 mins	10-15 mins

We recommend allowing **55 to 85 minutes** for this investigation.

Scratch project Pupils continue in their **31-Multiple Sprites** project or use **32-Multiple Sprites**

LINKS TO PRIMARY NATIONAL CURRICULUM

CURRICULUM OBJECTIVES	LINK WITH SCRATCHMATHS
Computing Design, write and debug programs that accomplish specific goals. Solve problems by decomposing them into smaller parts. Use sequences, selection, and repetition in programs.	▶ Pupils are required to explore conditions. ▶ Pupils are required to explore different forms of repetition: repeat, repeat until condition, and forever. ▶ Pupils are required to build scripts to ensure their sprites react in certain way to specific events. ▶ Pupils are required to build simple expressions using the reporter blocks which inform about the x and y positions of a sprite.
Mathematics Solve problems by applying their mathematics to a variety of routine and non-routine problem with **increasing sophistication**, including **breaking down problems into a series of simpler steps** and **persevering in seeking solutions**;	▶ Pupils are required to apply their logical thinking skills to control the behaviour of sprites (i.e. using repeat until…, if…then…) as well as apply their understanding of reflection and ability to order positive and negative numbers to determine the area within which a sprite can move on the stage.

MODULE 3 • INVESTIGATION 2 • ACTIVITY 3.2.1

Repeat Until…

LEARNING OBJECTIVES

Explain how to point one sprite towards another sprite.
Explore how to execute a script until a specified condition is true.

ACTIVITY INSTRUCTIONS

1. Pupils continue in their project **31-Multiple Sprites**.

First we want to teach Giga to walk in the same way as Pico.

2. Pupils select Giga and build the same behaviour for her as for Pico in activity 3.1.4. (Alternatively, they may select Pico and directly **copy his walking script** to Giga by dragging his script and dropping it on the picture of Giga within the sprites area.)

We want to adapt Giga's walking script to walk **towards Tera**, so she can deliver a message.

3. Pupils click Giga and keep the walking script running. Then they drag the **point towards …** block from the Motion group into the scripts area and keep it isolated from other scripts until they explore how it works.
4. They open the drop down menu of the block, select **Tera** and click the block (see additional support). While Giga is walking, encourage pupils to explore the isolated **point towards …** block by opening the drop down menu, selecting **Nano** or **Pico** as a new destination for Giga to walk towards and then clicking the block.
5. Pupils then snap the **point towards …** block into the walking script (ensuring it is now set back to **Tera**), just in front of **forever** – so that whenever Giga starts walking, she first sets her direction towards Tera. Pupils observe what happens when Giga reaches Tera (see additional support).

However, we want Giga **to stop when she reaches Tera** – not to continue walking through her.

6. Pupils replace **forever** of the walking script with the **repeat until …** block. Now they need a block that checks whether Giga is touching Tera or not **– the condition**.
7. They drag the **touching … ?** block from **Sensing**, keeping it isolated, select **Tera** from its drop down menu and click the **touching Tera ?** block – it always says *true* or *false* (see additional support).
8. Pupils add the **touching Tera ?** block into **repeat until …** and test the script. Giga should now stop when she touches Tera (see additional support).

VOCABULARY

The **point towards …** block points the current sprite towards the sprite selected in the drop down menu.

When we click a **condition block**, it always reports *true* or *false*.

The **repeat until …** block loops through the blocks inside it until its condition is *true*.

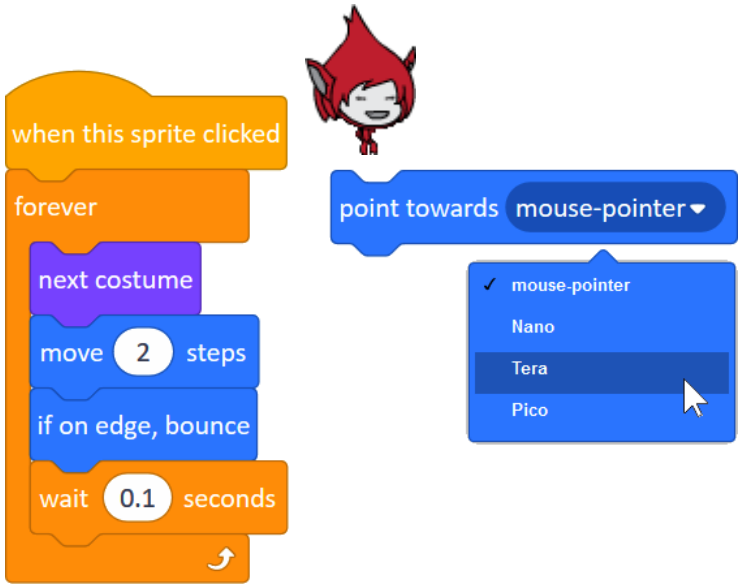
The **touching … ?** condition reports *true* only if its sprite is touching the sprite chosen from its drop down menu, otherwise it reports *false.*

DISCUSSION POINTS

- What does the **point towards Tera** block do? What would happen if you changed it to Pico?
- How does the **repeat until…** block work?
- What if **point towards Tera** block is moved inside **repeat until…** ?
- What are the three different types of **repeat** blocks we have used?

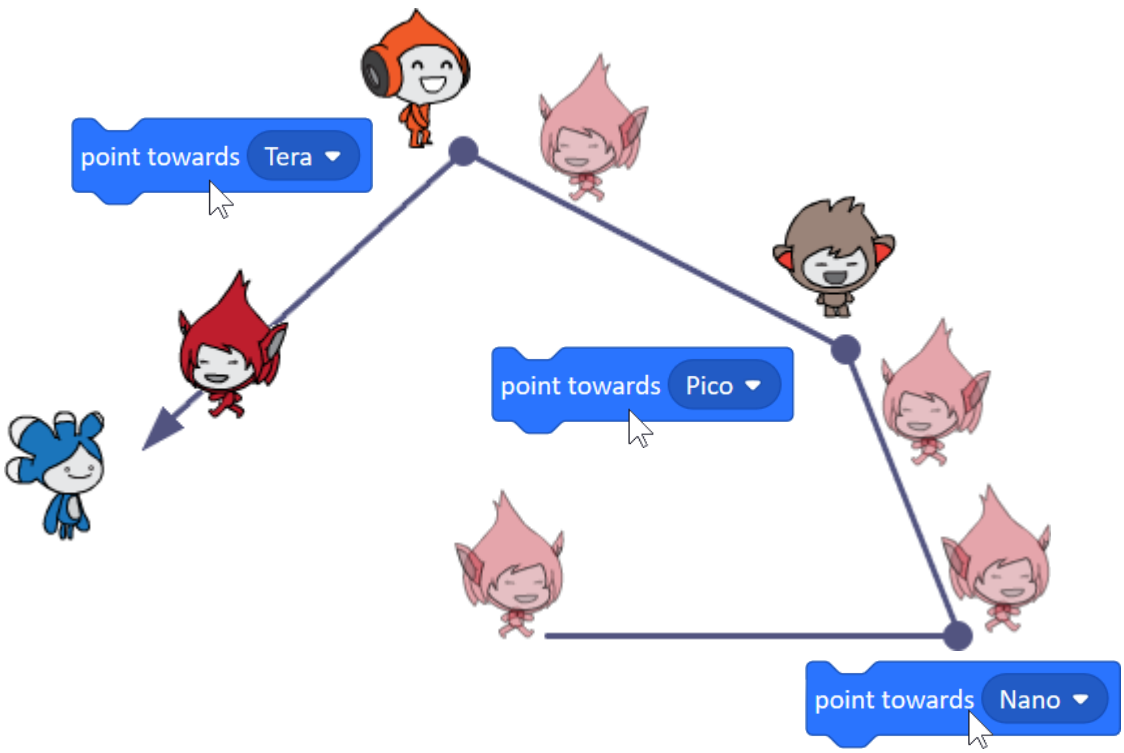
ADDITIONAL SUPPORT

Giga initially keeps walking horizontally, bouncing from the edges. However, when we select a sprite from the drop down menu of an isolated **point towards ...** block and click it, she will reset her direction towards that sprite. She will reset her direction each time the **point towards ...** block is clicked on.

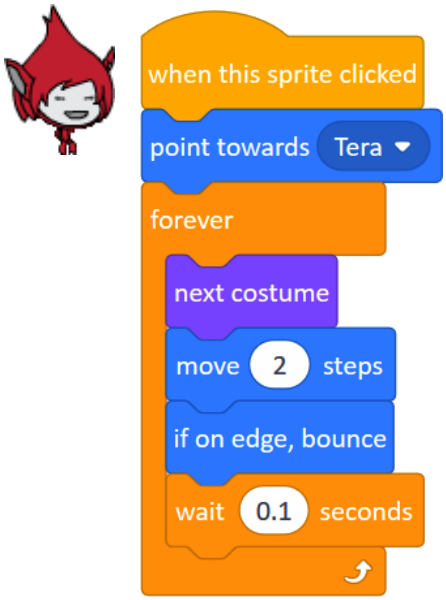


Pupils can explore this block by keeping Giga walking and repeatedly reselecting the destination sprite and clicking the **point towards ...** block several times.

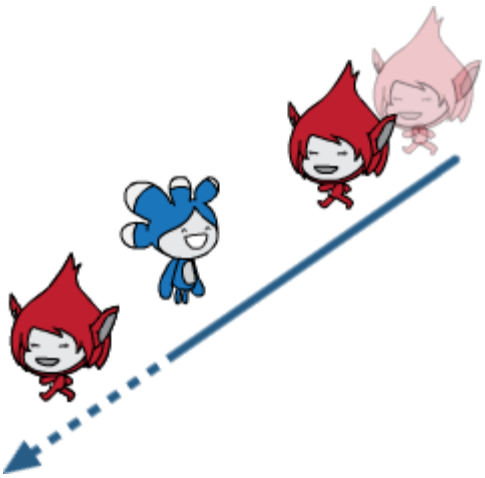
It may happen that pupils select a sprite in the drop down menu of the **point towards ...** block but forget to run it afterwards by **clicking the block**.

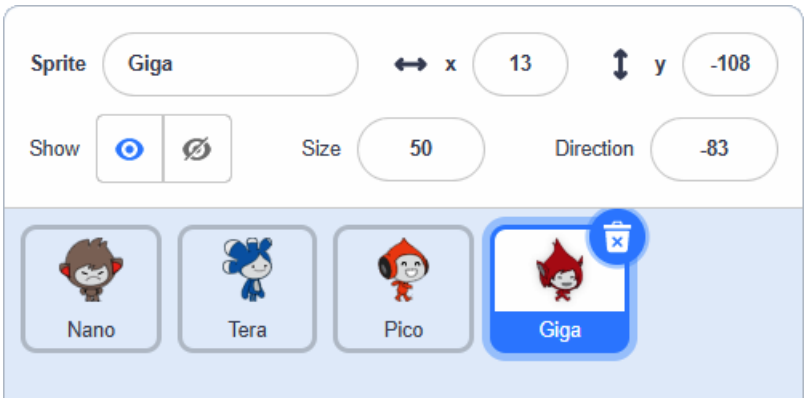


continued ▶



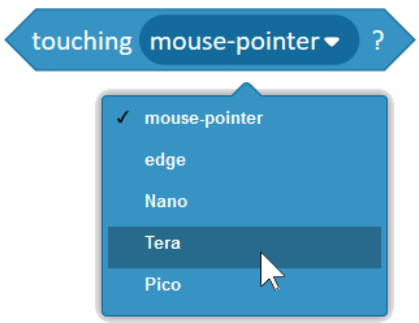
When Giga is clicked, she sets her direction towards Tera and starts walking in that direction **forever**, **passing through Tera**.





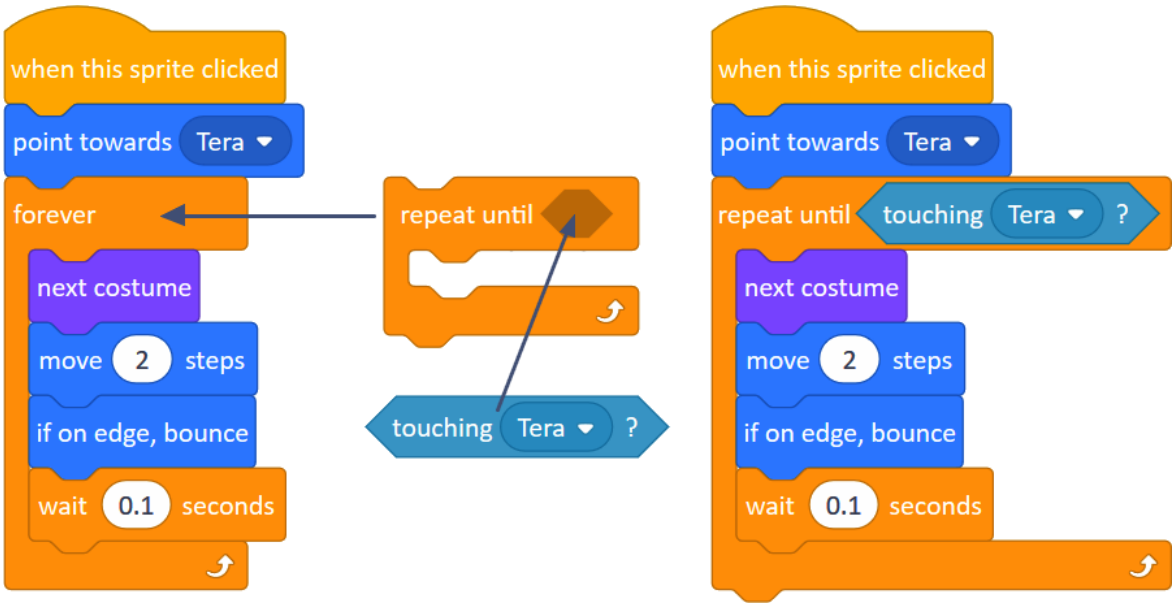
Remember that all scripts and blocks in the scripts area belong to **the sprite which is currently selected** in the sprites area. That is, each sprite has its own scripts area.

As Giga is currently selected, the condition block below checks whether Giga is currently touching Tera.

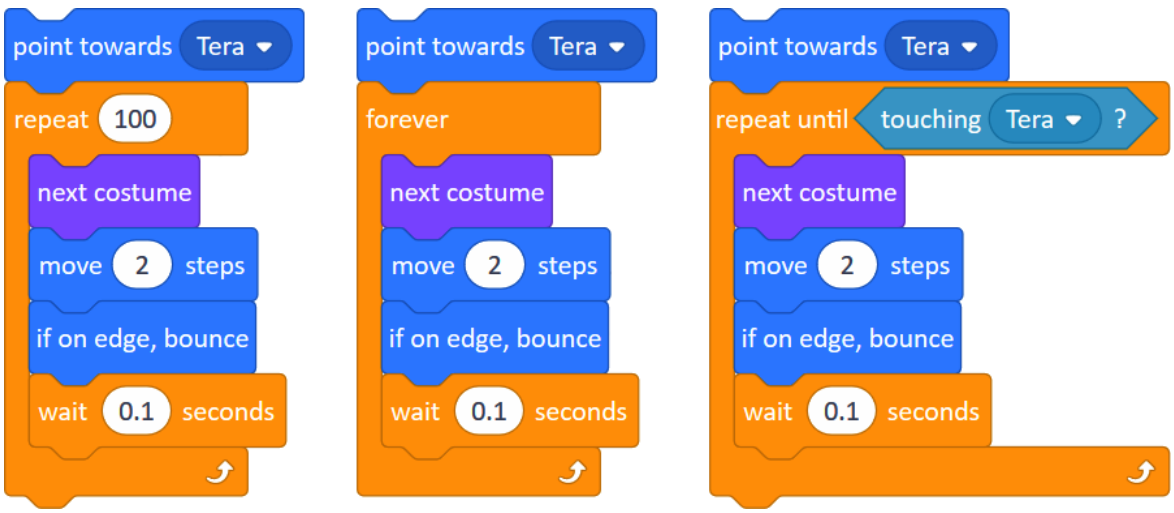


continued ▶

Note that whenever Giga is touching Tera, Tera is also touching Giga. This is true for any pair of sprites.



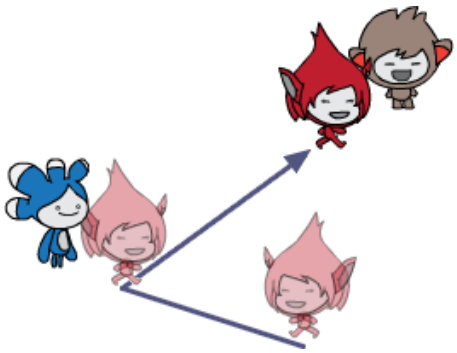
Below are the three different ways for *repeating several blocks*. Discuss with pupils the differences between the following three scripts.

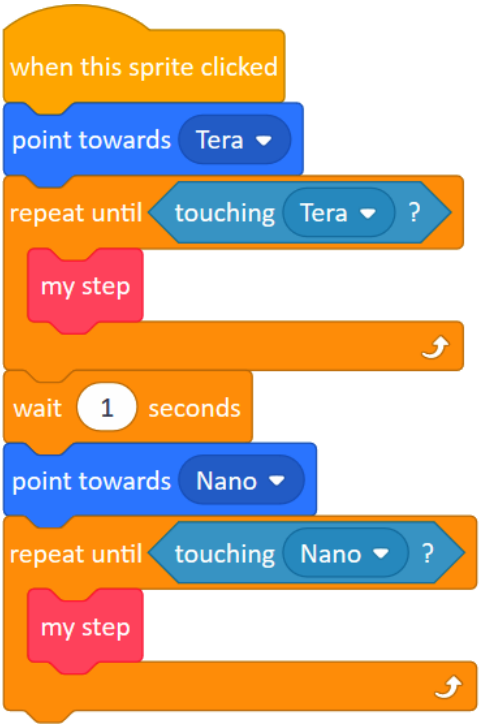


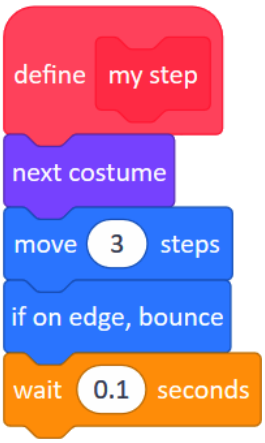
continued ▶

EXTENSION TO ACTIVITY 3.2.1

Pupils modify their script so that Giga first goes to Tera, then to Nano and finally stops walking there.

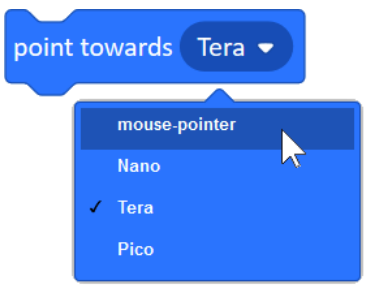




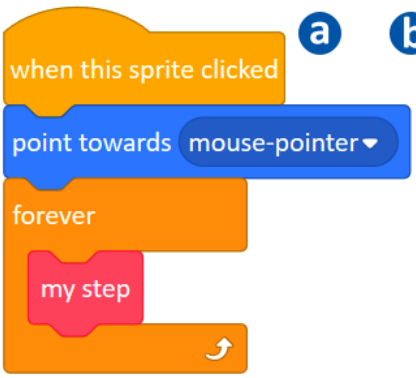


Some pupils may have noticed that it is also possible to select **point towards mouse-pointer** in the drop down menu. Pupils may experiment with this option, explore and discuss two alternative approaches (see below):

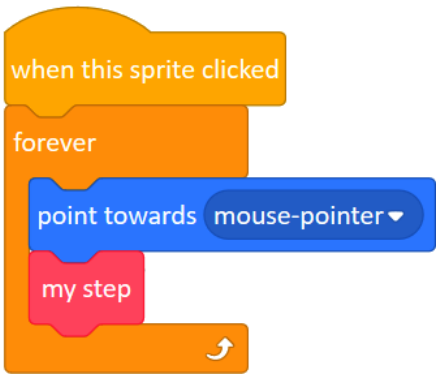
- If the **point towards mouse-pointer** block is placed before **forever**, the direction of Giga is set towards the mouse-pointer only once (see **a**).
- If the **point towards mouse-pointer** block is placed inside **forever**, the direction of Giga is re-set again and again (see **b**). Pupils explore what happens when this script is running while the mouse pointer is moved around the screen.



a



b



Note that when experimenting and exploring as above, it is important to be careful not to keep multiple alternative scripts (with the same hat block) in the scripting area because if Giga was clicked both alternative scripts would run – which might sometimes result in a confusing outcome.

MODULE 3 • INVESTIGATION 2 • ACTIVITY 3.2.2
Touching Colour?

LEARNING OBJECTIVES

Explore how a sprite can react to touching a specific colour.
Explore how to run a script only if a certain condition is true.
bridgE to mathematical problem-solving (generalising). **[Extension]**

ACTIVITY INSTRUCTIONS

1. Pupils continue in their project **31-Multiple Sprites** and select Pico.
2. Pupils add the block **point in direction …** under the **when this sprite clicked** hat block and set this to a random value (e.g. between 45 and 135) using the **pick random … to …** block – this means Pico will start walking in a random direction, not just horizontally, bouncing from the edges.

We want Pico to walk forever, but **turn back** when he reaches certain colour.

3. Pupils drag the new **Sensing** block **touching color … ?** into the scripts area, keeping it isolated and click on the square of colour in the block. The mouse pointer should turn to a hand – a 'colour picker'.
4. They can choose a colour of the backdrop at which Pico should turn back (e.g. dark grey). The block will now act as a condition checking whether Pico is touching that particular colour.
5. Pupils check the block by dragging Pico onto different colours on the backdrop and clicking the **touching color … ?** block to see if it says *true* or *false.* They can try changing the colour in their block and then dragging Pico around and clicking the block again.

We now have a condition to recognize a particular colour. Next we need to add an action to **react in the case the condition is true**.

6. Pupils drag the **if *<condition>* then …** block from the **Control** group, keeping it isolated and adding the **touching color … ?** condition in it. They add the **turn right 180 degrees** inside it.
7. While Pico is still walking pupils click the **if *<condition>* then …** block and observe what happens – when Pico is or is not touching that particular colour (clicking it once means checking the condition once). If they now insert it inside the **forever** block of the walking script the condition will be checked again and again.
8. **[Extension]** Pupils try changing Tera's jumping behaviour so that she first jumps high then floats down until she reaches certain colour.

THINGS TO NOTE

- Sometimes Pico may get stuck turning back and forth – if this happens simply drag him to another location on the stage.

VOCABULARY

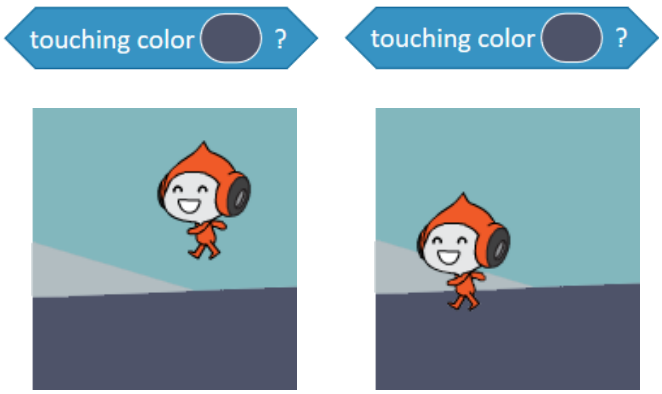
The **touching color … ?** block checks whether a sprite is touching a specified colour.
The **if *<condition>* then …** block checks its specified condition and if this is *true* it runs the blocks inside it.

DISCUSSION POINTS

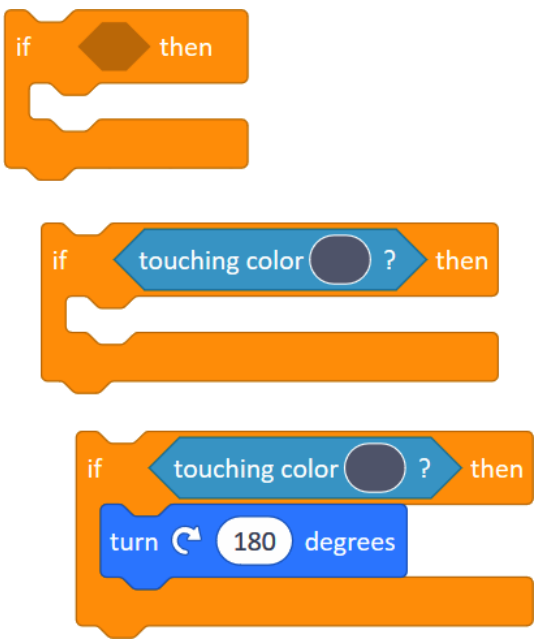
- Why did we change the direction the sprite is pointing from 90?
- When exactly does Pico turn around?
- If we tell Pico to **turn right 180** when touching green what does this mean he will do?
- What direction does Pico walk when he bounces off the stage?

ADDITIONAL SUPPORT

Below is the script to animate Pico to walk around forever, starting in a random direction. While he keeps walking the **touching color … ?** condition can be explored as an isolated block, to find out whether Pico is currently touching a particular colour or not i.e. *true* or *false*…



The condition block may then be snapped into the **if <condition> then …** block – still isolated from the script – to check the condition and react if it is *true*. Finally, after exploring it, this block can be snapped into the **forever** script.

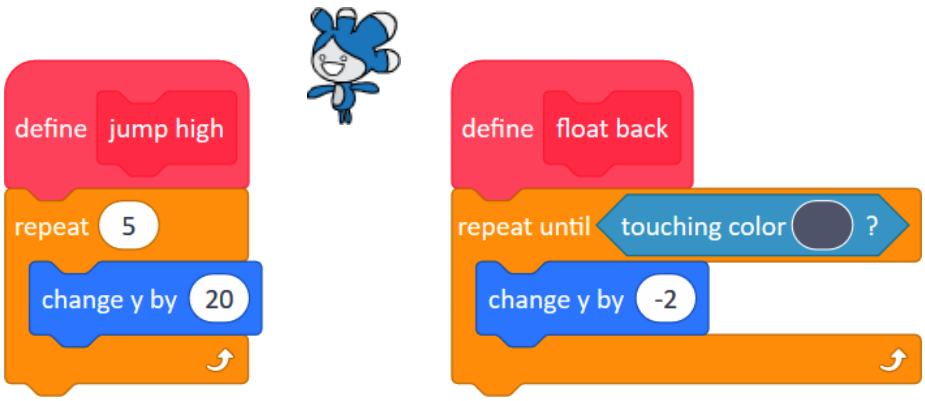


continued ▶

ADDITIONAL SUPPORT CONTINUED

Sometimes the sprite 'gets stuck' in the colour and keeps turning back and forth. If this happens add the **move** block inside **if…then…** block – so that it turns back and 'jumps out' of this colour.

[Extension] Below is one possible solution of Tera jumping high then floating down until she touches a particular colour of the backdrop:



MODULE 3 • INVESTIGATION 2 • ACTIVITY 3.2.3
Walking in the Air

LEARNING OBJECTIVES

Explore how to make a sprite react to reaching a specific vertical position on the stage.
Explain how to check whether the position of a sprite is more than a specified y value.

ACTIVITY INSTRUCTIONS

1. Pupils continue in their project **31-Multiple Sprites**, still working with Pico.

We want to stop Pico from walking 'in the air', i.e. in the upper white part of the stage, using a similar approach to Activity 3.2.2.

2. Pupils duplicate the whole **if touching color … ? then …** block and insert it in the existing **forever** walking script of Pico, replacing the dark grey colour by white colour of the sky in the stage (see additional support).
3. Pupils run the script, observe Pico and discuss his behaviour, noticing that he turns back as soon as he touches the sky colour by the top of his head instead of when his feet reach the same point.

To modify this behaviour it is possible to check Pico's y position instead of checking the colour .

4. Pupils find out the highest y position Pico should walk to. To do so they use the **y position** block from the Motion group, a **reporter block.**
5. Pupils drag the **y position** block into the scripts area and keep it isolated. They drag Pico vertically then click the **y position** block to find out his current y position. They repeat this to confirm his y position should never get bigger than 75.
6. To check whether Pico's y position is bigger than 75, pupils drag the **… > …** condition block from the **Operators** group in the scripts area, keeping it isolated, insert the **y position** block in its left hole and typing 75 in its right hole. They drag Pico up and down repeatedly, clicking this new compound condition **y position > 75** to find out whether it is *true* or *false*.
7. Pupils replace the **touching color … ?** block from the second **if … then …** block with their new condition to check Pico's **y position** and test it out (see additional support).

THINGS TO NOTE

The **y position** block reports the current y position of the sprite.

The **… > …** (is greater than) block checks to see if the first value is greater than the second one and reports *true* if it is, otherwise it reports *false* (we can find out by clicking it as an isolated block).

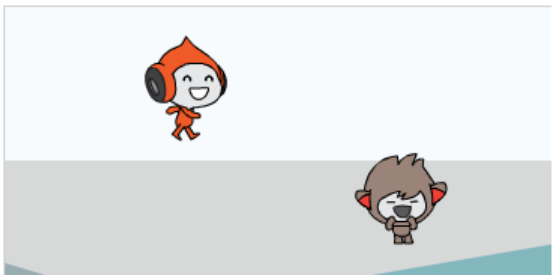
VOCABULARY

- It is easy to get the **… < …** and **… > …** blocks confused.

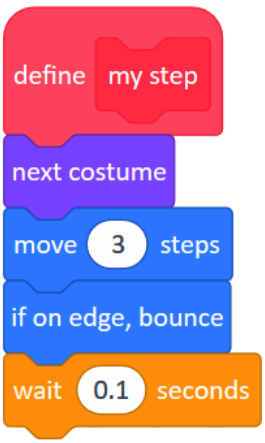
DISCUSSION POINTS

- Why does Pico turn back earlier than he needs to in the first solution (i.e. when checking whether touching colour of the sky)?
- What does the **… > …** (greater than) block check for? What would happen if the > sign was switched around?
- What does the **y position** block report? When would the reported value change?

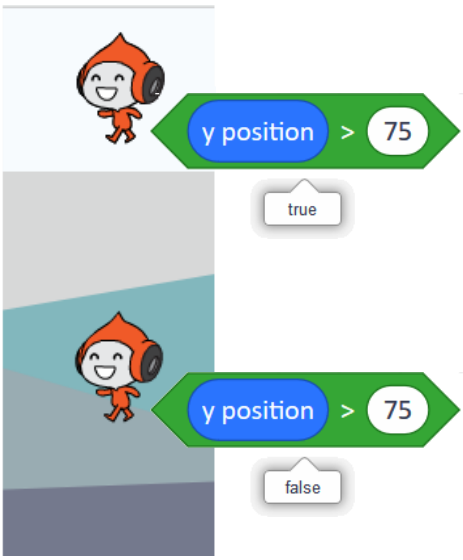
ADDITIONAL SUPPORT



While Pico walks forever around the stage, we want him avoid walking 'in the air' or in the sky, as if above the planet. Applying the strategy of checking whether he touches the colour of the sky does not work very well – Pico turns back as soon as he touches the sky by the tip of his cap.



Pupils should always explore new blocks in isolation. When they understand the **y position** reporter block, they insert it into another new block **... > ...** and again explore this compound block in isolation before using it inside **if ... then ...** block.



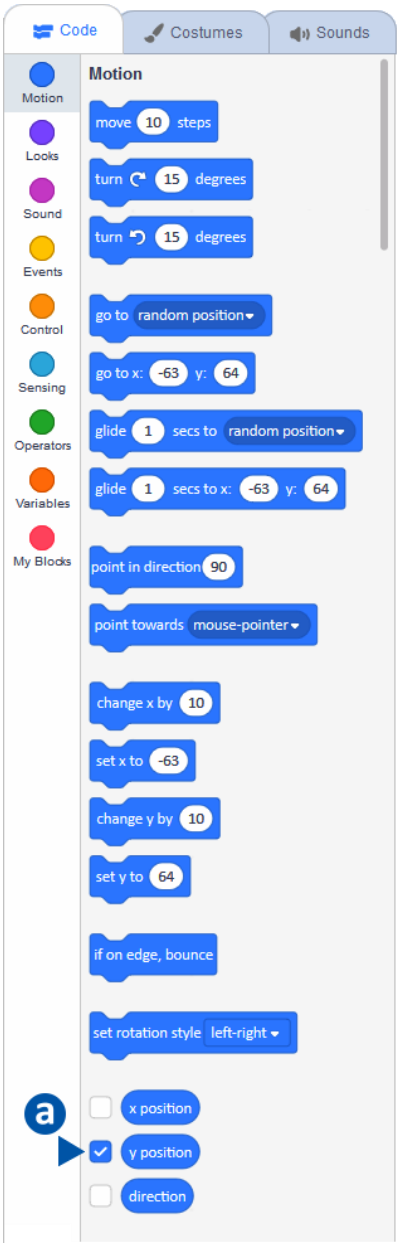
continued ▶

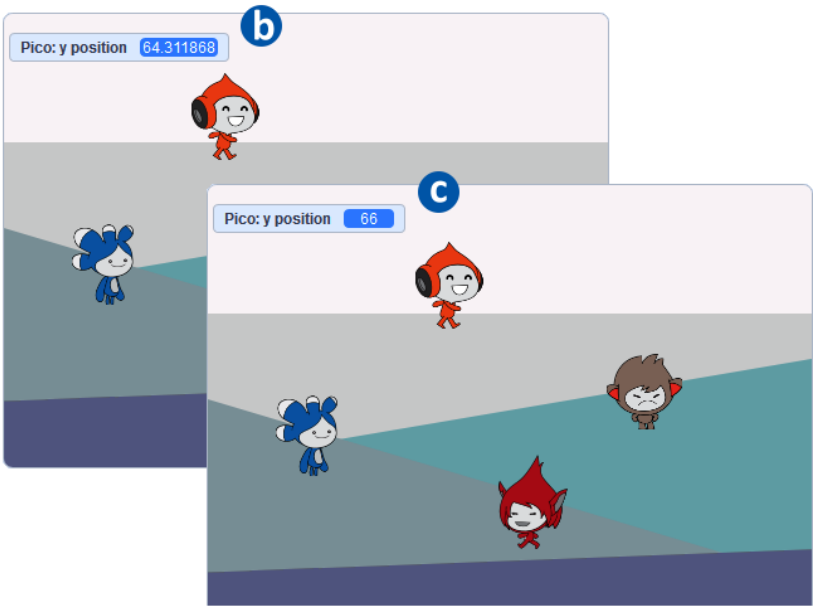
ADDITIONAL SUPPORT CONTINUED

Note that it is possible to click the checkbox to the left of the **y position** block in the Motion group (when Pico is selected), see (a).

A small reporter window with the actual value of his y position is displayed. When Pico's basic walking script (without checking the sky colour or his y position so far) is run it can be observed how his y position changes. However, the actual y position is displayed here as a **number with several decimal digits**, see (b).

Stop the walking script and simply drag Pico around the stage – in such case the value is rounded and displayed as integer, see (c).





If you decide to make use of this feature, in the following step pupils should still drag the **y position** reporter block in the scripts area and click it for different positions of Pico – **to explore it as an isolated block**.

The **final version of the script** (right) that checks Pico's actual **y position** and compares it to the boundary value of 75.

MODULE 3 • INVESTIGATION 2 • ACTIVITY 3.2.4

Unplugged: True or False?

LEARNING OBJECTIVES

Envisage if a condition is true or false based on the current state of the sprites on the stage.
bridgE to knowledge of coordinates.

ACTIVITY INSTRUCTIONS

- Print and distribute the unplugged pupil worksheet 3.2.4 or do the activity as a class.
- Ask the pupils to use the pictures to identify whether a condition is true or false.
- Discuss the answers as a class.

WORKSHEET SOLUTIONS

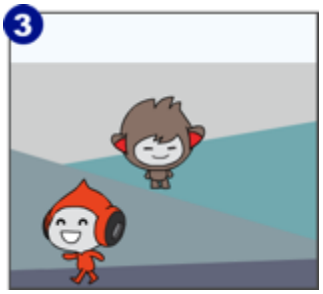
1

Is Giga touching Pico?

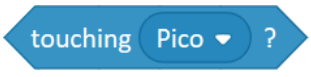
2

Is Pico touching Giga?

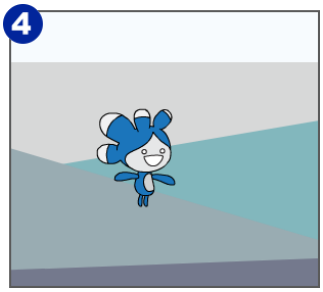
3



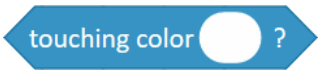
Is Nano touching Pico?



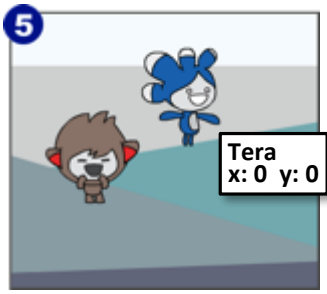
4



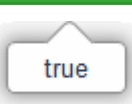
Is Tera touching the white colour?



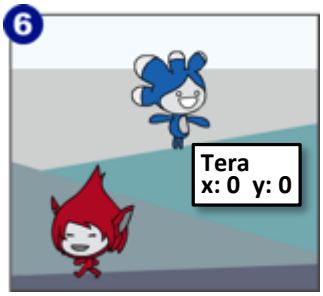
5



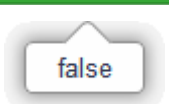
Tera stands in the centre of the stage. Is Nano's **x position** smaller than 0?



6



Tera stands in the centre of the stage. Is Giga's **y position** bigger than 0?



PRINT

INVESTIGATION 2
Activity 3.2.4

NAME

WHAT TO DO

Look at the pictures. Write down if the condition block would say **true** or **false** when clicked.

WORKSHEET TASKS

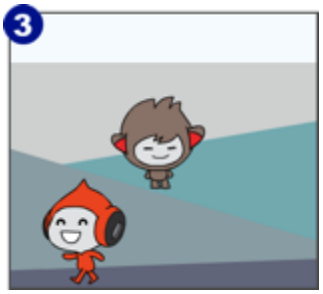
1

Is Giga **touching** Pico?

2

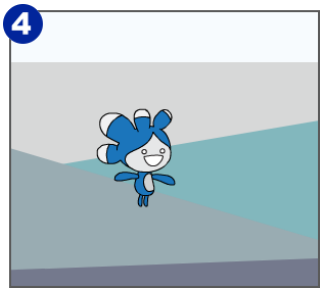
Is Pico **touching** Giga?

3

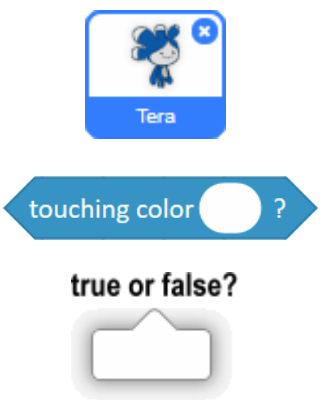


Is Nano **touching** Pico?

4

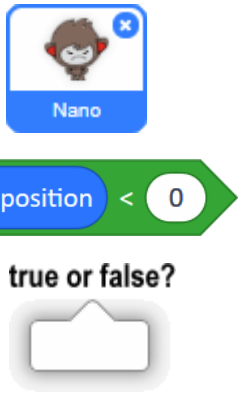


Is Tera **touching** the white colour?

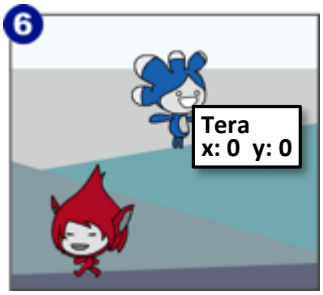


5

Tera stands in the centre of the stage. Is Nano's **x position smaller** than 0?



6



Tera stands in the centre of the stage. Is Giga's **y position bigger** than 0?

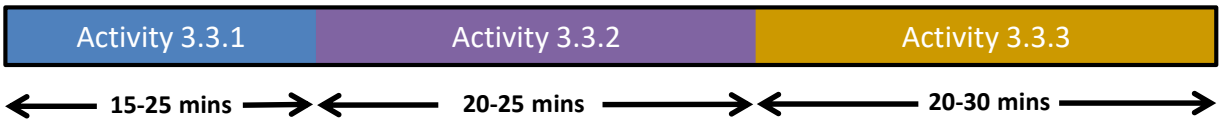
MODULE 3: INVESTIGATION 3

Broadcasting Messages

OVERALL LEARNING OBJECTIVE: Explore the concept of broadcasting messages between sprites, i.e., how to broadcast a message, how to receive it, and how to react to it. Build one or several scripts for different sprites to react to the same message in parallel.

This investigation introduces the concept of broadcasting as a means for sprites to communi-cate and collaborate. It also develops the concept of events and parallel reactions: whenever a message is broadcast, several sprites may react to it in parallel. This investigation comprises of three activities:

- **Activity 3.3.1** – Unplugged: Broadcast & Receive
- **Activity 3.3.2** – Introductions: One to One
- **Activity 3.3.3** – Come to Tera: One to Many



We recommend allowing **65 to 90 minutes** for this investigation.

Scratch project Pupils continue in their **31-Multiple Sprites** project or use **33-Multiple Sprites**

LINKS TO PRIMARY NATIONAL CURRICULUM

CURRICULUM OBJECTIVES	LINK WITH SCRATCHMATHS
Computing Design, write and debug programs that accomplish specific goals. Solve problems by decomposing them into smaller parts. Use repetition in programs. Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs.	▶ Pupils are required to build scripts that incorporate broadcasting and receiving messages to initiate actions; ▶ Pupils are required to build the interaction between the sprites in multiple small parts; ▶ Pupils are required to use repetition to continuously listen for messages being broadcast; ▶ Pupils are required to use logical reasoning to build a program for each sprite react to a specific message;
Mathematics Solve problems (KS3): • develop their mathematical knowledge, in part through solving problems and evaluating the outcomes, including multi-step problems; • begin to model situations mathematically (computationally) and express the results using a range of formal mathematical (computational) representations;	▶ Pupils are required to solve multi-step problems which involve broadcasting messages to initiate a series of events; ▶ Pupils are required to model more complex scenarios and apply their logical thinking skills in order to build parallel behaviours using multiple sprites;

MODULE 3 • INVESTIGATION 3 • ACTIVITY 3.3.1
Unplugged: Broadcast & Receive

LEARNING OBJECTIVE

Explore the concept of broadcasting and receiving messages.
Explain who reacts to a broadcast message and when.
Envisage how to broadcast messages between sprites in Scratch.

ACTIVITY INSTRUCTIONS

This unplugged activity is designed to introduce the concept of broadcasting in a playful way.

Print out copies of the cards from the additional support.

The purple cards should only be printed out once and each card should be given to **one child only** (i.e. 5 children should receive a purple card). You can print as many of the orange cards as you like to distribute to the **rest of the class**. The blue card is for the **teacher (or a chosen pupil) and should be read out to start the rhyme**.

1. Pupils think of themselves as sprites and interpret the instructions on their cards as scripts.
2. Pupils should only react when they hear the specific line (message) stated at the top of their card. In such case they follow the instructions on their card – to stand up, read aloud their line when they hear the previous line, then sit down.
3. Repeat this several times to ensure all pupils understand the process. Encourage them to 'perform' their line in some way.
4. Discuss as a class what happened.

It is important to understand the concept of **broadcasting** different messages to all the sprites but the sprites only reacting to a specific message otherwise the rhyme does not make sense.

5. **[Extension]** Remove one of the cards and discuss what happens (if it is a purple card this will result in a breakdown, if it is an orange card it may not – why is this?)
6. **[Extension]** Give some pupils two different cards.

THINGS TO NOTE

- Some of the lines are quite similar – pupils must listen very carefully to make sure they react at the right time.
- Note and discuss that there are some lines (*messages*) that make several pupils (*sprites*) react in parallel – as they all have instructions (*scripts*) to react to that particular line (*message*).
- If nobody had a card with instructions to react to certain line of the rhyme, the whole flow of reactions (*the script*) would get stuck.

DISCUSSION POINTS

- How did you know when to say your line (i.e. react to an event)? What was it important for you to do in order to do this? *(Listen)*
- Who could hear the lines of the poem (i.e. the messages)? *(Everyone that was listening)*
- What happened when multiple children had the same card (i.e. same script)? What do you think would happen in Scratch if multiple sprites had the same script?

ADDITIONAL SUPPORT

Print **one** copy of the cards below.

STARTER CARD

Teacher reads the following:

The Grand old Duke of York he had ten thousand men

When I hear the line

The Grand old Duke of York he had ten thousand men

Stand up and say:

He marched them up to the top of the hill

When I hear the line

He marched them up to the top of the hill

Stand up and say:

And he marched them down again

When I hear the line

And he marched them down again

Stand up and say:

When they were up

continued ▶

ADDITIONAL SUPPORT

Print **one** copy of the cards below.

When I hear the line

They were up

Stand up and say:

And when they were down

When I hear the line

They were down

Stand up and say:

And when they were only halfway up

continued ▶

ADDITIONAL SUPPORT

Print **multiple** copies of the cards below.

When I hear the line

When they were up

Stand up and say:

They were up

When I hear the line

And when they were down

Stand up and say:

They were down

When I hear the line

And when they were only halfway up

Stand up and say:

They were neither up nor down

continued ▶

MODULE 3 • INVESTIGATION 3 • ACTIVITY 3.3.2
Introductions: One to One

LEARNING OBJECTIVES

Explore how sprites can collaborate by using broadcasts.
Explain how to broadcast and receive a message between two sprites.

ACTIVITY INSTRUCTIONS

Continue in your project **31-Multiple Sprites** and select Nano. Build the following behaviour for him: when Nano is clicked, Tera will react by jumping high and floating slowly back.

1. Pupils drag a new block **broadcast** *message* from the **Events** group into the scripts area and keep it isolated from other scripts. In the drop down menu they choose *new message…* and type in *Jump!* – the message to be broadcast. Pupils click the block, making Nano broadcast his message – but nothing happens (discuss why this is).
2. Pupils then select Tera and make her "listen" to that message: from the **Events** group they drag in the **when I receive** *Jump!* hat block and add Tera's jumping reaction to it (see additional support).
3. Pupils **extend** Nano's behavior **when this sprite clicked:** he will first **teleport** himself and only then broadcast the *Jump!* message.

Now pupils select Giga. She wants to make friends with other sprites: when clicked, she will broadcast a message *make friends*. Currently only Tera will react.

4. Pupils drag the **broadcast** *message* block from the **Events** group, change the *message* to *make friends*, and add the **when this sprite clicked** hat block.
5. Pupils then select Tera and make her "listen" to that message: from the **Events** group they drag in the **when I receive** *make friends* hat block to build a script. They build a reaction e.g. the **say** *Hello, I am Tera!* **for 2 secs** block from the **Looks** group (see additional support).

Pupils will extend this to become **more complex behaviours** for Giga and Tera.

6. When Giga is clicked, she will walk over to Tera, **say** *Hello! I am Giga. And you?* **for 2 secs**, then broadcast her message. Pupils change Tera's reaction so she will also jump high and slowly float back, then say *Hello! I am Tera!*
7. **[Extension]** If pupils completed the extension from activity 3.2.1 then Giga will also walk to Nano. Repeat the same process and introduce Giga and Nano as well.

VOCABULARY

A **broadcast** is a message that is sent and activates receiving blocks. This is the way to make other sprite(s) react.

The script attached to a **when I receive …** block will be run after the specified broadcast message has been sent.

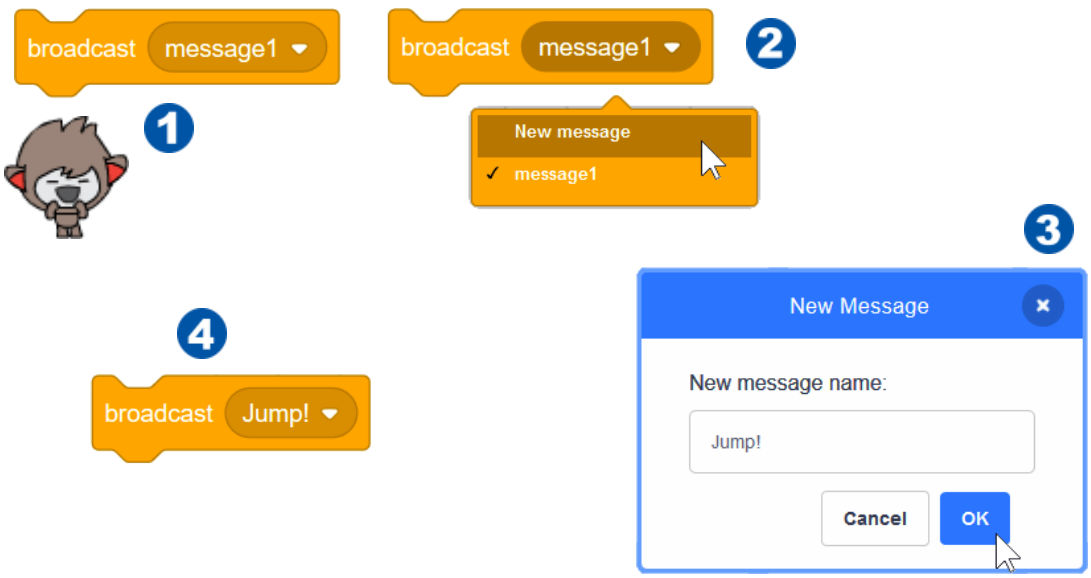
DISCUSSION POINTS

- What would happen if you broadcast a message but no sprite has a corresponding **when I receive …** block?
- What would be the difference if you used the **say …** block instead of **say … for 2 secs**? (*click the red Stop sign to get rid of the **say** bubble*)

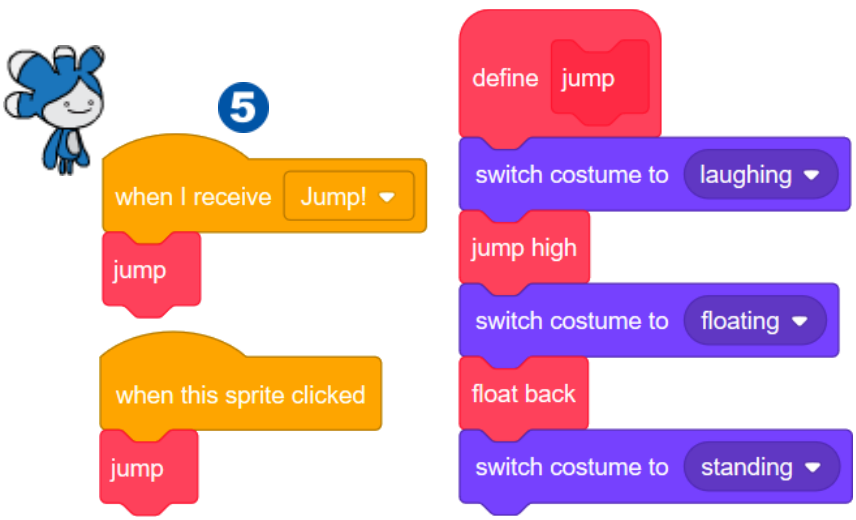
ADDITIONAL SUPPORT

Sprites broadcast messages using the **broadcast** block. It is important to note that:

- broadcasting has no 'addressee', the *message* is just being broadcast in a similar way to being 'shouted' out,
- whoever 'listens' to that particular *message* – only Tera in this case – that sprite must have a script with the hat block **when I receive …** with that particular *message* and other attached blocks to specify the reaction,
- it is a good idea to use clear and informative names for messages as they help make scripts more readable.



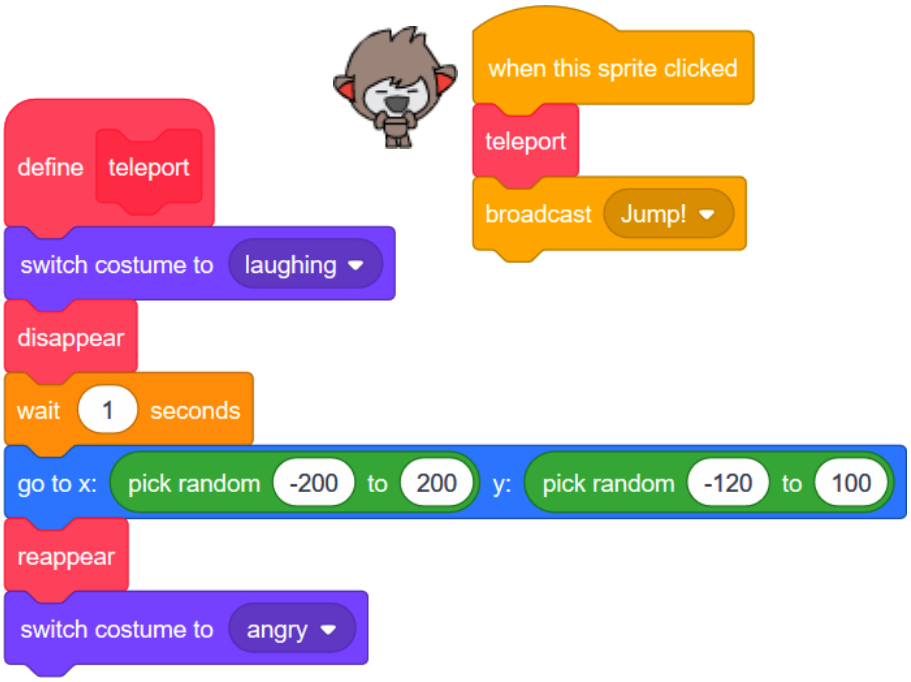
Note that Tera now has two scripts with the same reaction: when she receives the *Jump!* message and when she is clicked, she reacts by jumping and floating back.



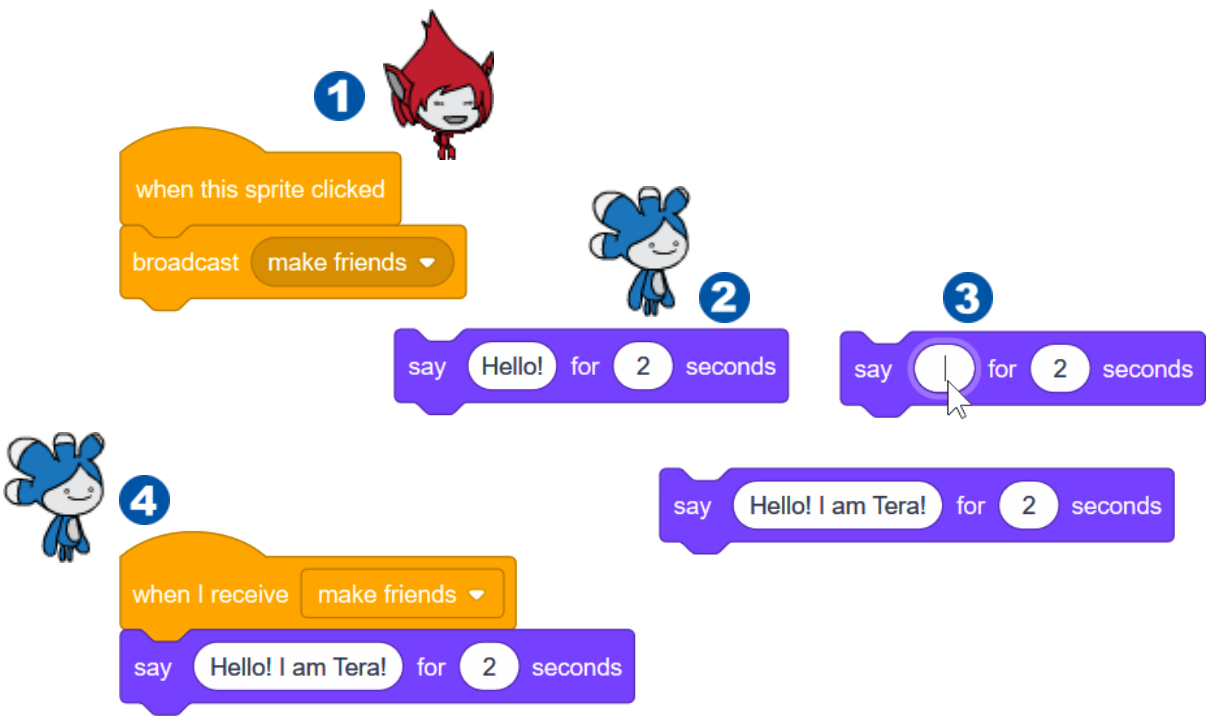
continued ▶

ADDITIONAL SUPPORT CONTINUED

Extended script for Nano: when clicked he will first teleport himself and only then broadcast the *Jump!* message.



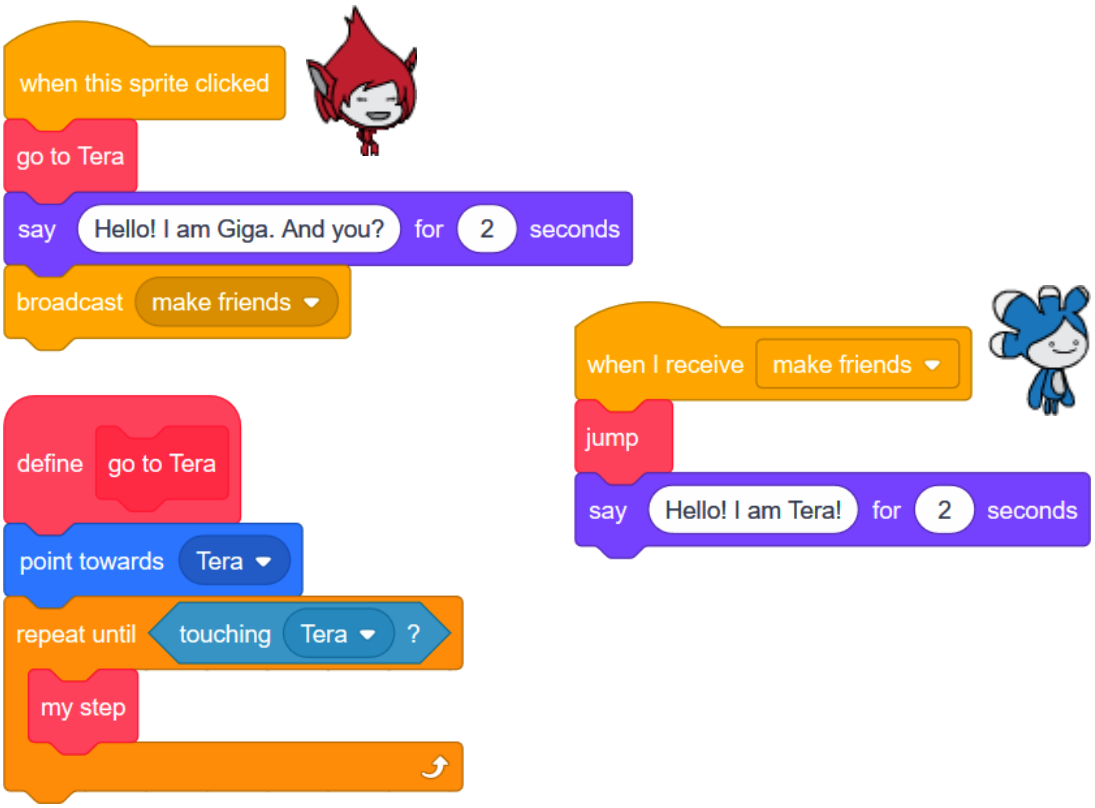
Giga wants to make friends with other sprites: when clicked, she will broadcast a message *make friends*. Below is an example script to make Tera react to this message.



continued ▶

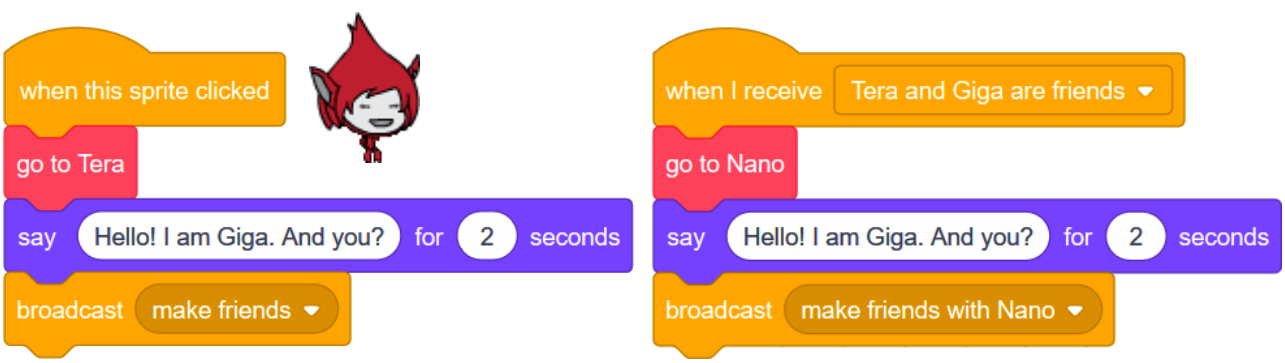
ADDITIONAL SUPPORT CONTINUED

Now Giga – when clicked – will walk over to Tera, **say** *Hello! I am Giga. And you?* **for 2 secs**, then broadcast her message *make friends*. The script below (right) will make Tera jump high and slowly float back, then say *Hello! I am Tera!* You may also wish to define Giga's reaction as a new block **go to Tera**.



[Extension] When Giga is clicked, she will walk over to Tera, **say** *Hello! I am Giga. And you?* **for 2 secs**, then broadcast her message. Tera will react by jumping high, slowly float back and then say *Hello! I am Tera!* Giga will then walk to Nano (can be defined as a new block **go to Nano**) and introduce herself. Nano will reply then teleport somewhere else.

Note that when Tera jumps high and floats back and introduces herself to Giga, she will broadcast a message, thus **giving a signal** to Giga **that she can continue** and walk towards Nano.



continued ▶

ADDITIONAL SUPPORT CONTINUED

define go to Tera
point towards Tera
repeat until touching Tera ?
my step

define go to Nano
point towards Nano
repeat until touching Nano ?
my step

when I receive make friends
jump
say Hello! I am Tera! for 2 seconds
broadcast Tera and Giga are friends

when I receive make friends with Nano
switch costume to laughing
say Hello! I am Nano. for 2 seconds
teleport

MODULE 3 • INVESTIGATION 3 • ACTIVITY 3.3.3
Come to Tera: One to Many

LEARNING OBJECTIVES

Explore how to use broadcasts to initiate actions in multiple sprites at the same time.
Explain the difference between saying a message and broadcasting a message.

ACTIVITY INSTRUCTIONS

1. Pupils continue in their **31-Multiple Sprites** project and select Tera.

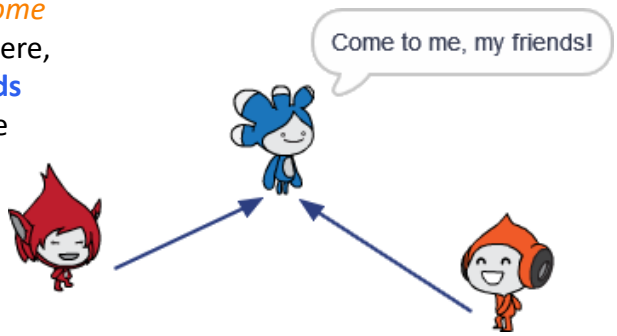
We want to extend Tera's reaction: when she is clicked, Tera will jump high and float back, then **say** *Come to me, my friends!* **for 2 secs**, then broadcast the invitation message *come to Tera*, thus inviting **whoever is listening to her message**.

Note that Tera already jumps and floats back in more than one situation: when reacting to Giga's message *make friends* and also when she is clicked. This requires her *jumping and floating* reaction to be defined as a new block to avoid repeating the same script.

2. Pupils build new block **jump** to combine Tera's jumping/floating behaviours.
3. Pupils add the **say** *Come to me, my friends!* **for 2 secs** block to the **when this sprite clicked** script of Tera, and also the **broadcast** *Come to Tera* block.

Note and discuss the difference between **say** and **broadcast** blocks (the **say** block has only visual effect on the stage: other sprites will not "see" that Tera said anything. The only way to let them know is to **broadcast** a message).

4. Pupils select Pico and build his **when I receive** *Come to Tera* reaction: he will come to Tera and stay there, applying a similar strategy, with the **point towards** *Tera* and **repeat until touching** *Tera* blocks, as we did in Activity 3.2.1.
5. Pupils build or duplicate identical **when I receive** *Come to Tera* reaction for Giga and test all scripts by clicking on Tera.



6. **[Extension]** Pupils create another script for Nano to make him teleport to Tera when she broadcasts her *Come to Tera* message.
7. **[Extension]** Pupils modify Nano's reaction from the previous Extension so that he teleports somewhere close to Tera, but not exactly there.

THINGS TO NOTE

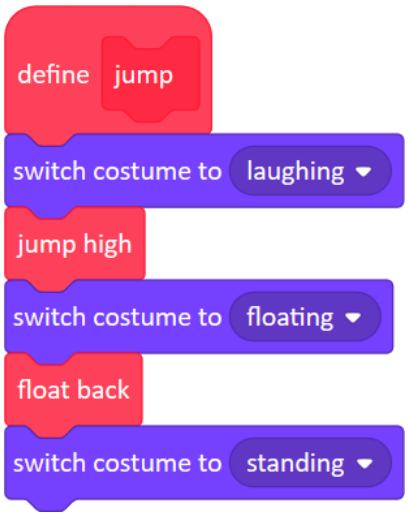
- Note that scripts duplicated between sprites always appear in the upper left corner of the scripts area and thus can visually cover other scripts.

DISCUSSION POINTS

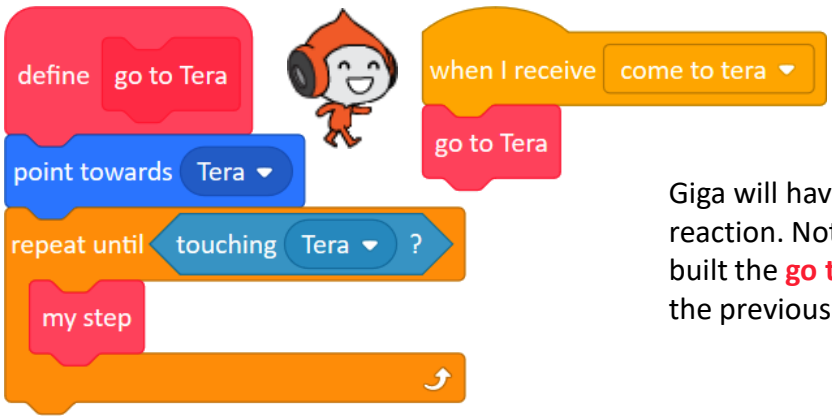
- Why is it important to build a new block to combine Tera's jumping and floating?
- What is the difference between the **say** and **broadcast** blocks?
- What did you need to do to ensure both sprites reacted at the same time?

ADDITIONAL SUPPORT

Below is the script for Tera jumping and floating back, **saying** the invitation and **broadcasting** the message to initiate reactions of other sprites.

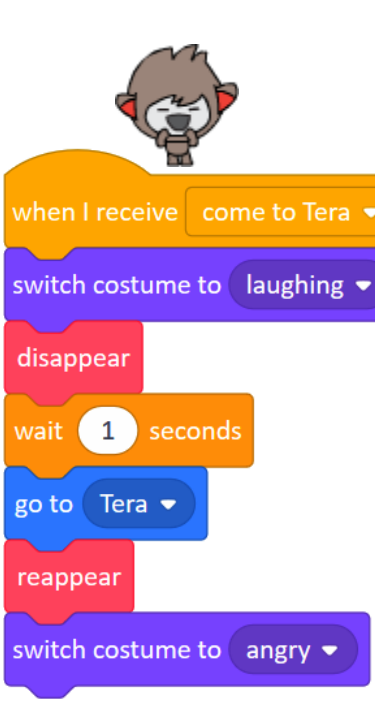


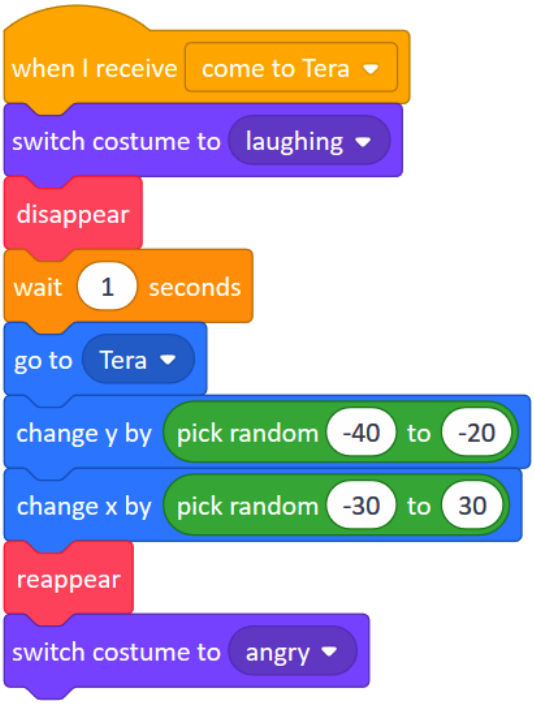
Pico will react by walking towards Tera and staying there. Envisage, explore and discuss whether each block in this **go to Tera** script is necessary.



Giga will have exactly the same reaction. Note that we have already built the **go to Tera** script for her in the previous Activity.

Nano will react by disappearing, jumping to the actual position of Tera then reappearing again. In the second alternative solution (on the right) Nano jumps to Tera but slightly changes his y and x positions before reappearing back.





MODULE 3: INVESTIGATION 4

Interactive Stories

OVERALL LEARNING OBJECTIVE: Building on previous knowledge of broadcasting messages and other programming concepts and procedures, plan out and implement a story of your own in which the Module 4 characters will interact, react in parallel and experience things together.

This investigation further develops the concept of events, messages, conditions and parallel reactions. Pupils will plan out a story of their own where characters will talk, move, jump and interact in different ways. The investigation also includes an unplugged assessment activity requiring pupils to apply what they have learned throughout the whole module. The investigation comprises of one assessment activity and one extension activity.

- **Activity 3.4.1** – Unplugged: Reading Scripts
- **Extension Activity 3.4.2** – The Story of the Sprites

We recommend allowing **65 to 110 minutes** for this investigation.

Scratch project Pupils continue in their **31-Multiple Sprites** project or use **34-Multiple Sprites**

LINKS TO PRIMARY NATIONAL CURRICULUM

CURRICULUM OBJECTIVES

Computing

Solve problems by decomposing them into smaller parts.

Design, write and debug programs that accomplish specific goals.

Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs.

Mathematics

Solve problems (KS3)

- work with experiments that involve **randomness**;
- select appropriate concepts, methods and techniques to apply to unfamiliar and non-routine problems.

LINK WITH SCRATCHMATHS

- Pupils are required to build several interactions between the sprites, i.e. build scripts that incorporate broadcasting and receiving messages to initiate and run actions, sometimes by several sprites in parallel;
- Pupils are required to present and explain their solutions to others;
- Pupils are required to share their solutions and modify solutions obtained from others to fit into their stories;
- Pupils are required to model more complex scenarios and apply their logical thinking skills in order to build parallel behaviours using multiple sprites;
- Pupils are required to apply their logical thinking skills to control the behaviour of sprites by broadcasting and receiving messages;
- Pupils are required to use what the have learned throughout the module to select appropriate methods to solve the problem(s) they have chosen.

MODULE 3 • INVESTIGATION 4 • ACTIVITY 3.4.1
Unplugged Assessment: Reading Scripts

LEARNING OBJECTIVES

bridgE to knowledge of coordinates in all four quadrants, factors as well as positive and negative numbers.
Envisage the outcome of different scripts, behaviours, reactions and interactions between sprites.
Explain why a script would implement an expected behaviour and how to modify it to achieve an expected interactions and outcomes.

ACTIVITY INSTRUCTIONS

1. Print and distribute the unplugged pupil worksheet 3.4.1.
2. Ask pupils to work individually to check what they have learned during Module 3.
3. The answers to the worksheet are below:

    1. The **third script** is correct. The first one always teleports Nano to the same position, while in the second script the **wait** block is not correctly positioned.
    2. The repeat number must be **10** so that Tera floats back by 100 altogether. Floating down is implemented in the script by decreasing the **y position** by 10 repeatedly.
    3. Tera will move **down**, as the value to move by can be any random number between -50 and -10.
    4. The missing block is **next costume**. Without this block the sprite does not change its costumes while moving, thus producing no animation effect.
    5. First value will be **False** as Giga is not touching Nano. Second value will be **True** because Giga is touching that colour (beige/yellow). The third value will be **False** because Giga is not touching that colour (dark blue).
    6. [top row] **Up** (y axis is changing by a positive value)**, Right** (x axis is changing by a positive value) [bottom row] **Left** (x axis is changing by a negative value)**, Down** (y axis is changing by a positive value).
    7. The **second script** is correct– making Pico stop walking when his **x position** gets smaller than 0, i.e. very close to the centre of the stage.
    8. When Pico is clicked, **he will broadcast go!** Nano will not react at all as he has no script reacting to the go! message. However, **Tera will run her go! script**, so she will **move down 100 steps**. Her reaction to the now! message will not be run.
    9. Top script: **Pico will hide** (nothing else will happen as the condition of the **if** block is False). Middle script: **Pico will go to Tera then hide** (as long as the condition of the **if** block is True). Bottom script: **Pico will say Hello! for 2 secs** (the **if** block will not run as its condition is False)
    10. **[Extension]** When Giga is clicked she will **say Hello! for 1 second and broadcast** Bye! Nano will react by running his Bye! script – **he will hide for 1 second then show again**.

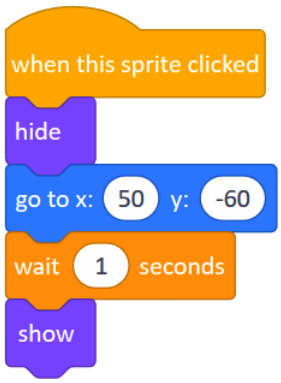
NAME

WHAT TO DO

Read the scripts below and answer the question next to it. The questions refer to characters from the **31-Multiple Sprites** project – *Nano, Tera, Pico and Giga*.

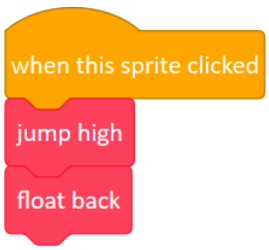
ASSESSMENT TASKS

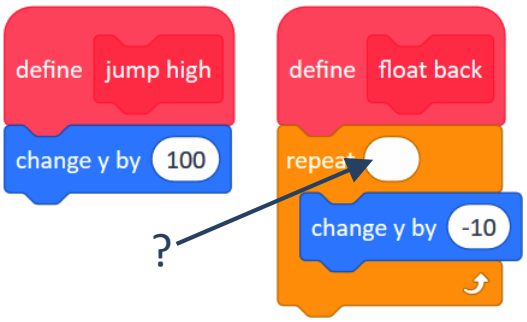
1 **Circle the script** that will make Nano disappear, teleport to a random position and then reappear after **1 second**.



---

2 What **number** should go in the **repeat** block so that Tera floats back to the same position when she is clicked? Explain why.

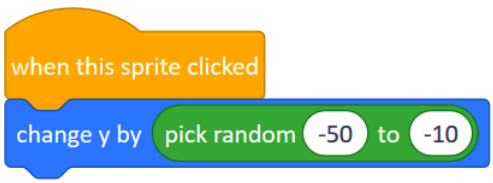




Repeat number = _____ Explain why = _____

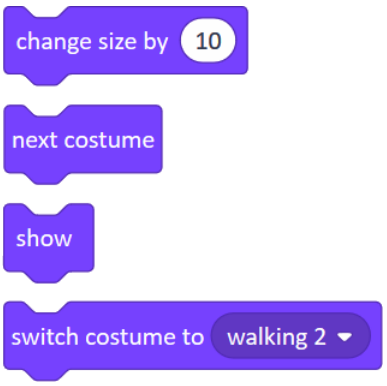
---

3 Will Tera move **up or down** when the script on the right is clicked? Explain your thinking.

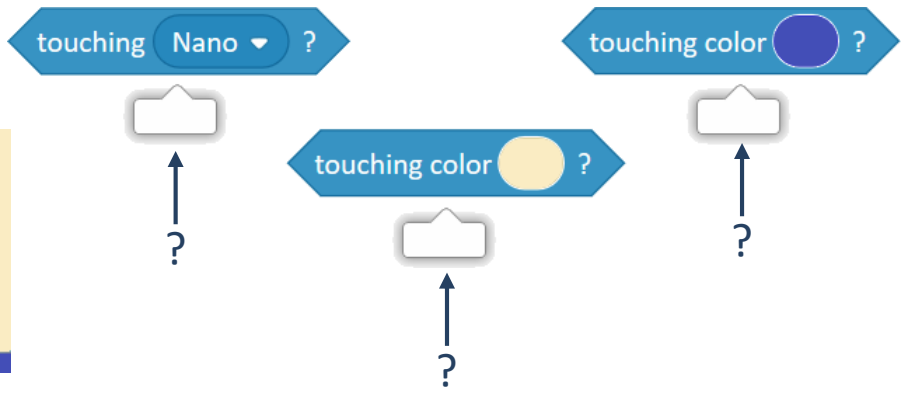


Tera will move _____ because = _____

4 **Circle the purple block** that should be added in the script below to make Pico walk like the picture below.

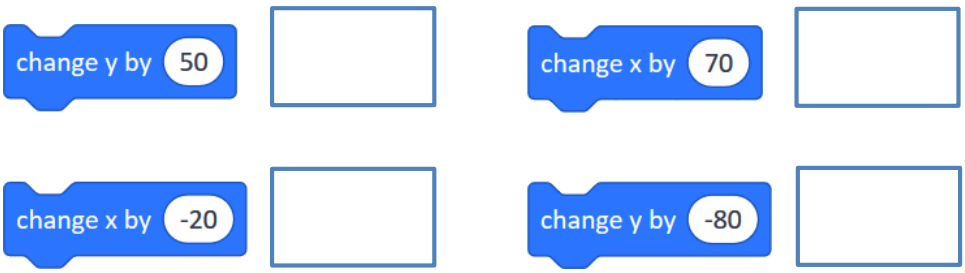


5 Giga is walking towards Nano (see picture). For each of the blocks below **fill in the value** that will be shown when it is clicked.



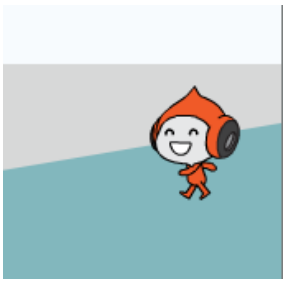
6 Next to each of the blocks **fill in the direction** that Tera would move when that block is clicked.

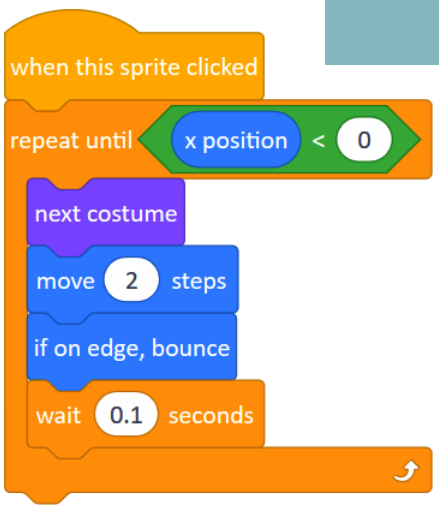
Will Tera move **Up**, **Down**, **Left** or **Right**?



ASSESSMENT TASKS CONTINUED

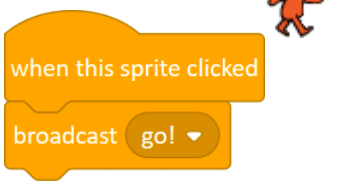
7 Pico is near the right edge of the stage. **Mark the script** that will make him walk and stop when he reaches the middle of the stage.





8 Look at Pico's script below. What will Nano and Tera do after Pico is clicked? **Explain what will happen** on the screen in the box below.

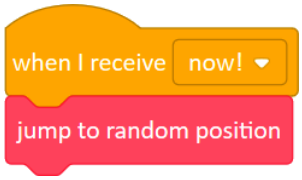
Pico



Tera

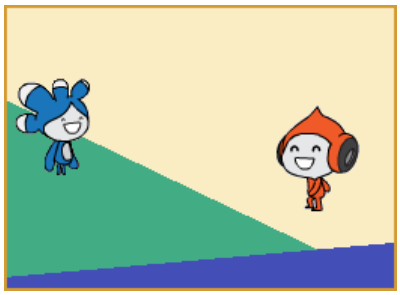
When Pico is clicked...

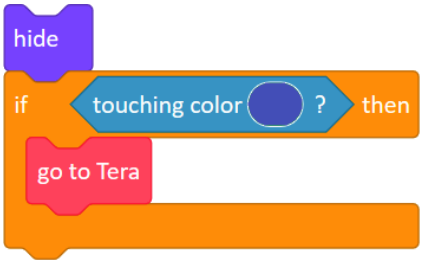
Nano

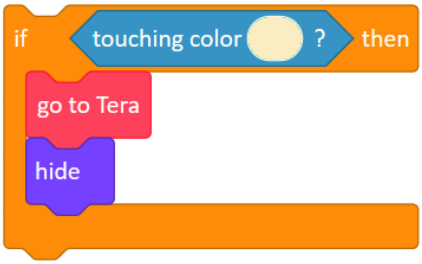


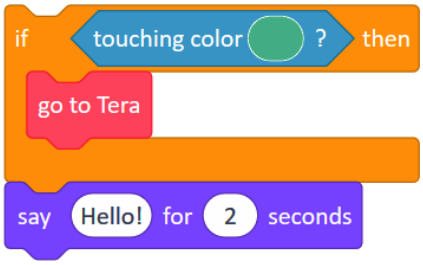
ASSESSMENT TASKS CONTINUED

9 Look at the current position of Pico on the stage. The scripts on the right are in Pico's scripts area – **explain what will happen to Pico** if each of these scripts is clicked.





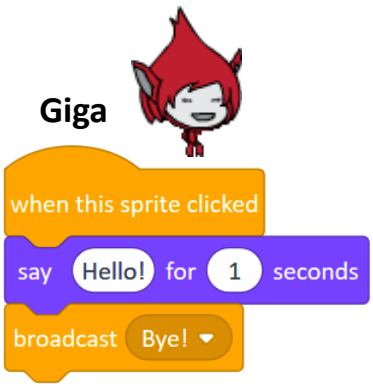


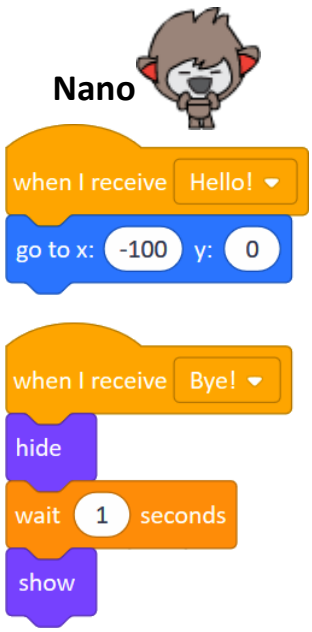


When this script is clicked… what will happen?	When this script is clicked… what will happen?	When this script is clicked… what will happen?

[Extension]

10 When Giga is clicked what will Nano do? **Write your answer** in the box on the right.





When Giga is clicked Nano will…

Module 3 • Investigation 4 • Activity 3.4.2
[Extension] The Story of the Sprites

Learning Objective

Explore concepts learned earlier within the module to build an interactive story between two or more sprites.
Exchange scripts with others in order to expand the interaction.

Activity Instructions

It is recommended for pupils to work on this activity in small groups.

1. Pupils continue in one of their projects **Multiple Sprites**.

Assign each group two or more characters (Nano, Tera, Pico or Giga) to tell their story about.

2. Pupils plan out the story:
   - How do these characters know one another?
   - How could one character visit or greet the others?
   - What could they say to each other? What will they do together?
   - What could happen when one character touches another character?

3. Pupils build the interactions between their characters using the concepts and procedures they have learned over the module.

Now groups will share what they have done with other groups.

4. Each group take turns to describe to the rest of the class what they have built.

Note that if you are using the online version of Scratch, it is possible to exchange scripts within the class (within one Scratch account) using the **Backpack feature**:

- To open the Backpack click on the bar at the bottom of the screen (see additional support).
- Drag the script you want to share with the rest of the class into the Backpack.
- To use a script from the Backpack in another project, open the Backpack and drag the script from there into the scripts area.

It is also possible to save a sprite together with all its costumes and scripts into a local file (with the file extension **.sprite3**) using both online and offline Scratch then upload it to another computer running Scratch (see additional support). If the scripts are short and readable, it is also possible to easily reconstruct them in other computers by displaying them on the interactive whiteboard at the front of the class.

Things to Note

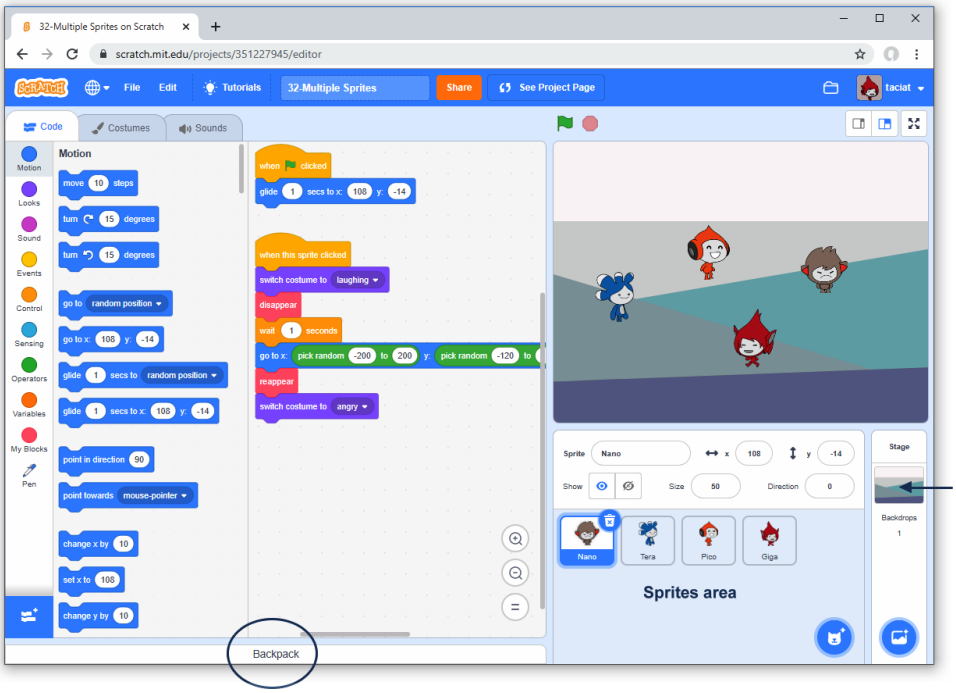
- It is not possible to name or label scripts within the Backpack and therefore it can quickly become confusing if there are lots of scripts placed here – this process needs to be managed carefully.
- Ensure pupils have the correct sprite selected when dragging a script from the backpack to the scripts area.

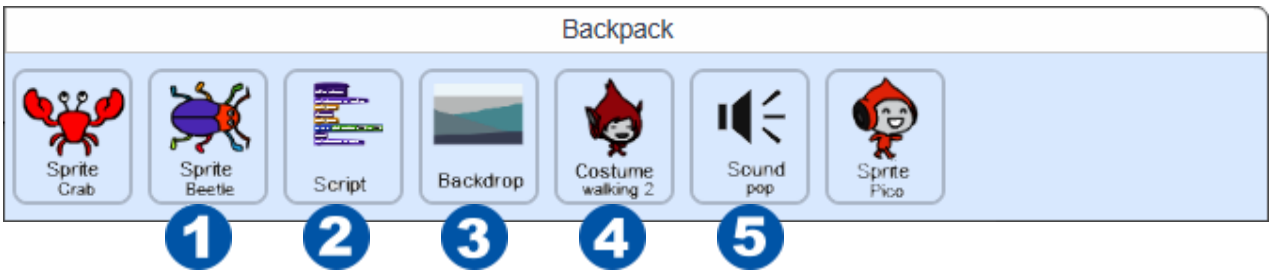
Discussion Points

- What interactions did you build?
- Did you incorporate a script from another group? How did you do this?
- Did you learn any new ideas from the other groups? Did they learn anything new from your group?

ADDITIONAL SUPPORT

The backpack can be used only with the **online version of Scratch 3.0**. Its actual content is associated with that particular Scratch account. Therefore if pupils are sharing one common class Scratch account the backpack is a way how to exchange elements of projects between them (similar to the concept of the Clipboard being a "space" shared between several applications in a computer). However, the shared content of the backpack must be managed carefully as elements cannot be labelled. To open or close the backpack click the **Backpack bar** on the bottom of the scripts area (see on the left).



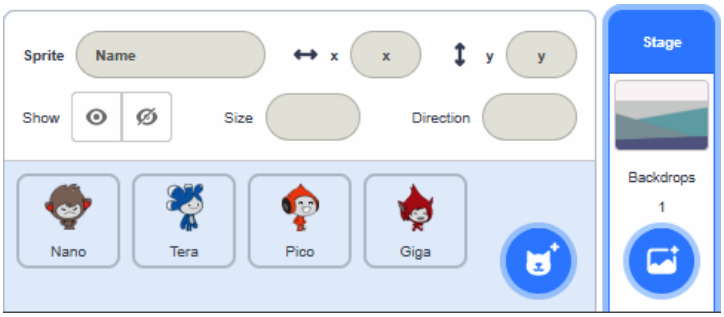


- A sprite **(see 1 above)** can be dragged in the backpack from the sprites area. Note that it will be added to the backpack together with all its scripts and costumes.
- To get a sprite with all its scripts and costumes from the backpack, drag it from there into the sprites area.
- A script **(see 2 above)** can be dragged in the backpack from the scripts area. Note that it always has an the same name in the backpack 'Script' and is shown as a miniature of the original script. Thus this feature must be used carefully as it is almost impossible to visually distinguish one script from another in the backpack.
- To get a script from the backpack, first select the sprite which you want the script to belong to, then drag the script into its scripts area.
- To share a backdrop **(see 3 above)**, select the Stage icon (to the left from the sprites area), then click the Backdrops tab and drag any of the backdrops from there into the backpack.

continued ▶

ADDITIONAL SUPPORT CONTINUED

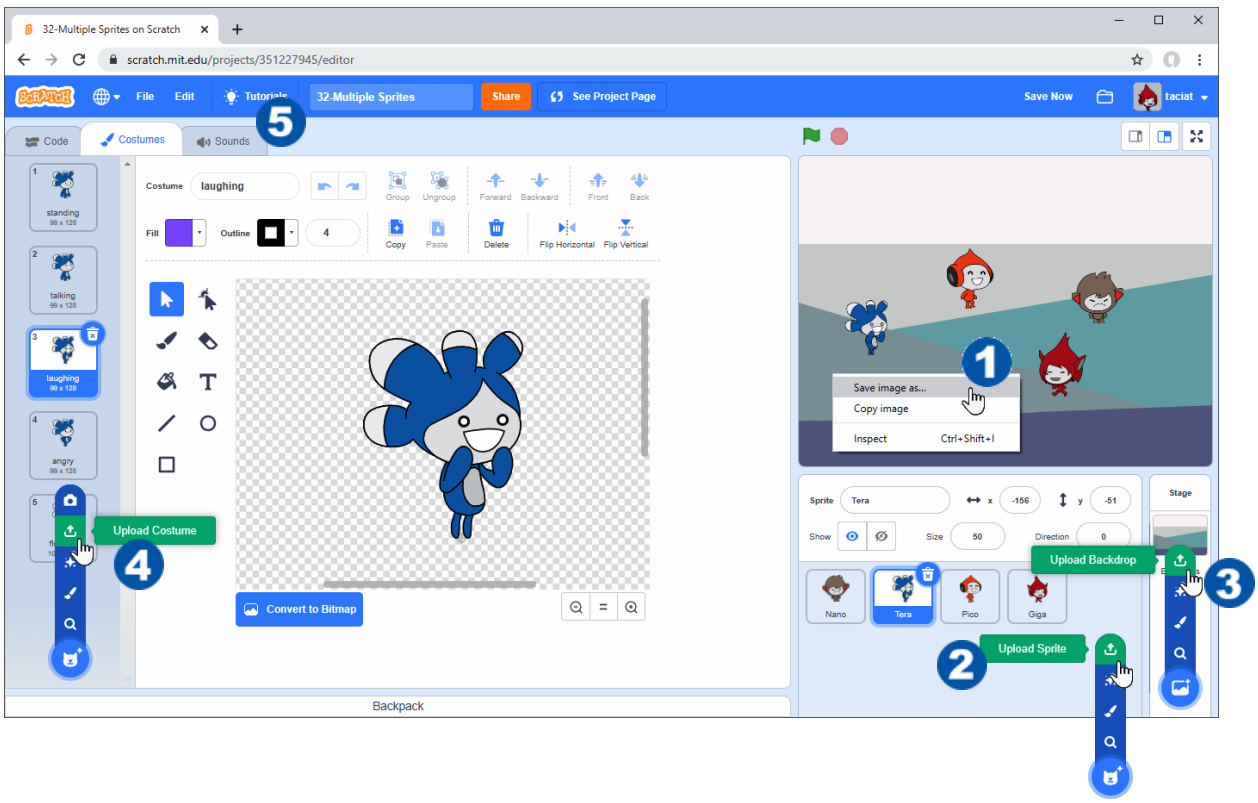
- To get a backdrop from the backpack, select the Stage icon, click the Backdrops tab and drag the backdrop from there into the Backdrops tab. Be careful not to drag it into the sprites area as this would create a new sprite with the backdrop as its costume.



- A costume of a sprite **(see 4 – previous page)** can be dragged in the backpack from the Costumes tab, or the other way round.
- A sound **(see 5 – previous page)** can be dragged into the backpack from the Sounds tab or dragged from there back into the Sounds tab.

If you are using the **desktop** or **online version of Scratch 3.0**, you can exchange sprites (including all of their scripts), individual scripts, individual costumes, backdrops or sounds by saving them to and uploading them from the local shared files:

- Right click the **stage** and choose **Save image as…** to save its actual content (including the pictures of all sprites) into a **.PNG** file **(see 1 below**, this option is not available in desktop version**)**.
- Right click a **sprite** and choose **export** to save it (including all its scripts) in a local **.sprite3** file.
- Right click a **costume** in the Costumes tab and choose **export** to save it in a local **.SVG** file or **.PNG** file (depending on whether it is a vector or a bitmap graphic).
- Upload a sprite from a local file by clicking **(2)**, upload a backdrop by clicking **(3)**, upload a costume by clicking **(4)**, or upload a sound by clicking the **Upload Sound** icon in **(5)**.



Module 3: Success Criteria

By the end of Module 3 pupils should be able to:

Blocks

- Use the **show** and **hide** blocks.
- Use the **say** block, set its text input.
- Use **set _ position** and **change _ position** blocks, use go to … blocks.
- Use **point towards …** block.
- Use **if on edge, bounce** block.
- Use conditions, conditional, **forever** and **repeat until** blocks.
- Use **broadcast** message and **when I receive** message blocks.

Sprite

- Control when the sprite will be shown or hidden.
- Set and change the x and y coordinates of a sprite.
- Set the direction of a sprite towards another sprite.
- Use costumes of a sprite to animate it.
- Make a sprite jump or glide to certain position, make it jump to another sprite, make it bounce from an edge.
- Make a sprite say a text.
- Make a sprite communicate and collaborate with other sprites.

Scripts

- Use repeat to control the speed of a graphic effect or a movement of a sprite.
- Build scripts to control ghost graphical effect.
- Build scripts for animated walking movement of a sprite using the next costume block. Make a walking sprite bounce from the edge.
- Use forever to build an infinite loop.
- Use conditions for repeating a sequence of blocks only until a condition is true.
- Use if… then conditional block in a forever loop to monitor certain coordinates, an interaction with a sprite, or touching a colour of the backdrop.
- Build scripts with different event hat blocks.

Problem Solving

- Exploit randomness.
- Build parallel reactions of different sprites.
- Build reactions which monitor certain conditions, build conditional processes.
- Build conditional and infinite loops.
- Coordinate sequences of reactions by broadcasting and receiving messages.
- Share scripts with other pupils and modify and exploit them to extend the project.
- **[Extension]** Combine several algorithms to create an interactive story with multiple sprites.

Mathematical Understanding

- Use the coordinates to control where a sprite can appear on the stage.
- Use random numbers to tell a sprite to move to a random position.
- Order positive and negative numbers to determine the area within which a sprite can move on the stage.
- Use different factors of a number to control the speed of a graphic effect and movements of sprites.
- Use negative numbers to build a reverse effect and movement.
- Explore the concept of a moving object which bounces from the edge of the stage.
- Apply logical thinking skills in order to build parallel behaviours using multiple sprites.