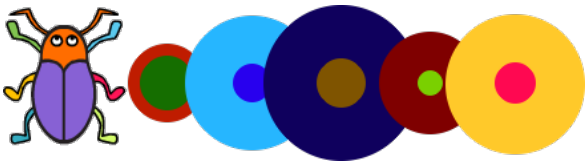
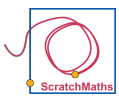
UCL ScratchMaths Curriculum

Module 2:
Beetle Geometry



Teacher Materials

Developed by the ScratchMaths team at the UCL Knowledge Lab, London, England



Module 2: Beetle Geometry

Investigation 1	Investigation 2	Investigation 3	Investigation 4
Exploring Pen	Drawing Polygons	Discovering Dots	Pen Project: Nature Scenes
			

Introduction to Module 2

Module 2 is focused around creating different drawings using the pen tool including numerals, patterns, polygons as well as entire scenes. This module could potentially be linked with several different areas of the Key Stage 2 curriculum including history and geography.

History: Romans and Morse Code

The first investigation looks at drawing Roman numerals which could link to history projects around the Roman empire.

The third investigation requires pupils to create Morse code messages, which could link to work on the history of communication.

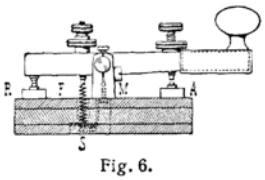


Fig. 6.

Geography: The Natural World

The final project in this module looks at creating different scenes found in nature including forests and beaches. This investigation could be linked in with geography topics exploring different elements of the natural world.

Key Vocabulary and Concepts Covered by Module 2

Scratch	Computing	Mathematics
▶ Pen down, pen up blocks ▶ Pen colour blocks ▶ Pen shade blocks ▶ Pen size blocks ▶ Backdrop ▶ Pre-defined blocks ▶ Pick random ▶ Repeat block ▶ Define block	▶ Initialisation ▶ Expressions ▶ Debugging ▶ Sequence ▶ Repetition ▶ Logical Reasoning ▶ Algorithm ▶ Definition	▶ Roman numerals ▶ Perimeter ▶ Regular and irregular polygons ▶ Multiplication and division ▶ Angles ▶ Rotation ▶ Positive and negative numbers ▶ Coordinates

MODULE 2: INVESTIGATION 1
Exploring Pen

KEY LEARNING OBJECTIVE: Explore the pen tool of a sprite and how to set its attributes. Build scripts to draw different figures and think about the corresponding algorithms.

This investigation introduces the pen tool and includes activities to explore the different pen features as well as the importance of block order within a script. It also provides opportunities for pupils to physically play out the movements of the sprite to help them identify and use different drawing algorithms. The investigation comprises of four activities.

- **Activity 2.1.1** – Drawing Numerals
- **Activity 2.1.2** – Swapping Blocks
- **Activity 2.1.3** – Unplugged: I am Beetle
- **Activity 2.1.4** – Different Drawing Algorithms

Scratch projects **2-Drawing Numerals**
2-Swapping Blocks

LINKS TO PRIMARY NATIONAL CURRICULUM

CURRICULUM OBJECTIVES	LINK WITH SCRATCHMATHS
Computing Design, write and debug simple programs that accomplish specific goals. Use sequence, selection, and repetition in programs. Use logical reasoning to explain how some simple algorithms work.	▶ Pupils are required to create a script that draws specific Roman numerals. ▶ Pupils are required to experiment with sequencing a fixed set of blocks. ▶ Pupils are required to play out and build scripts for different drawing algorithms.
Mathematics Read Roman numerals to 1000 (M) Identify angles at a point and one whole turn (total 360 degrees). Calculate the perimeter of a rectilinear figure (including squares).	▶ Pupils are required to build scripts to draw different Roman numerals to 1000 (M). ▶ Pupils are asked to calculate total degrees of angles within a square using knowledge of repeat and turn values. ▶ Pupils are asked to calculate perimeter of a square using knowledge of repeat and move values.

MODULE 2 • INVESTIGATION 1 • ACTIVITY 2.1.1

Drawing Numerals

LEARNING OBJECTIVES

Explain how the *setup script* works.
Explore how to draw roman numerals using different pen colours and pen sizes.
bridgE to existing knowledge of roman numerals.

ACTIVITY INSTRUCTIONS

- Pupils open project **2-Drawing Numerals**, save as a copy and add their name(s) to the title.
- Pupils are encouraged to read the *setup script* starting with the **when green flag clicked** hat block and explain in their own words what it does line by line (see 1 in additional support).
- Pupils spend a few minutes exploring the pen blocks by clicking each of the blocks in the scripts area (see 2 in additional support) but **do not snap them together** at this point.
- Pupils investigate how pen colours can be set and reset. They click on the colour square within the **set pen color** block and see how the mouse cursor changes into a **colour picker**. Then they click anywhere in the Scratch window (or for a specific colour on one of the colour circles on the left side of the stage) and see how the colour of the square changes.
- Pupils try changing the size of the pen of the Beetle in the **set pen size to …** block.
- Pupils choose one of the roman numerals below and build a script to draw it. They set the pen colour and pen size of their numeral within the script (see additional support).

For pupils who struggle

For higher attainers

THINGS TO NOTE

- Pupils should not edit the *setup script.*
- Be careful when looking for the **set pen colour** block as there are three very similar blocks (see additional support).
- I, L and C are the easiest numerals to draw.

VOCABULARY

Each sprite has a **pen tool**. The sprite can draw lines on the stage when its pen tool is down.

Pen down means that the sprite will continuously draw a trail wherever it moves (until **pen up** block is used).

The **set pen** blocks allow you to change the colour and width of the line that is drawn.

CLASS DISCUSSION POINTS

- What does **pen down** mean? What would happen if this block was not in the *setup script*?
- How can you set and reset the colour of the pen?
- How can you reset the pen size?
- How did you draw your numeral? Which blocks do you have in your script?
- Which roman numeral have you managed to draw? What number does it represent?

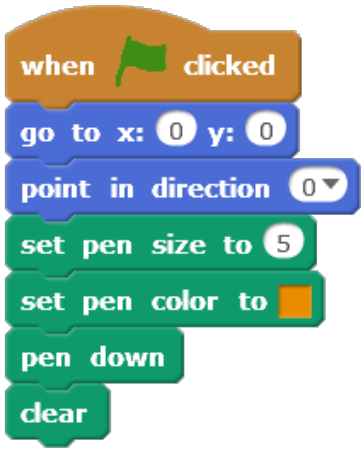
ADDITIONAL SUPPORT

In Module 1 we were working with the Tile sprite, i.e. with the sprite named Tile, with several costumes of different tiles. Throughout Module 2 we will be working with another sprite named Beetle (with only one costume). The Beetle will be helpful for all activities as it clearly depicts its *direction* and *position*.

The *setup script* on the right tells the Beetle sprite to go to the centre of the stage and point up. It then sets the pen size (i.e. width of the line) to 5 and its colour to orange, puts the pen down (so it will draw a line when the sprite moves) and lastly clears the screen.

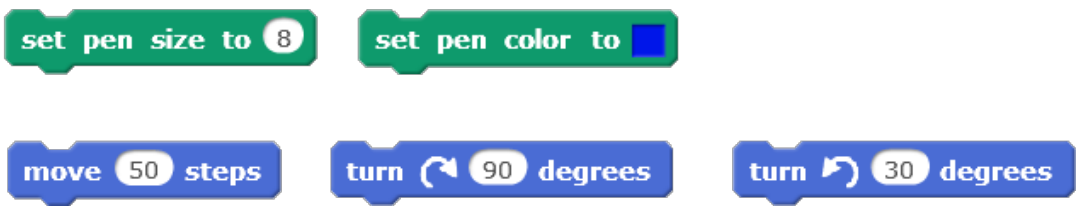
Important note: Pen down = pen on. Pupils frequently think the opposite of this so try to emphasize this point – for instance by using a retractable pen.

1



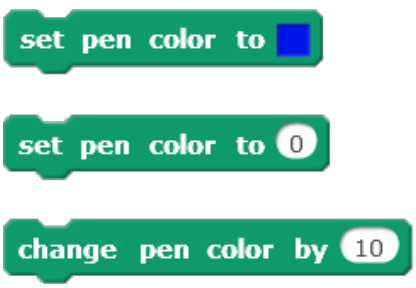
The blocks below will be initially available in the scripting area for this project.

2



Setting and changing pen colour

There are three very similar **set pen colour** blocks in the Pen group – see on the right.



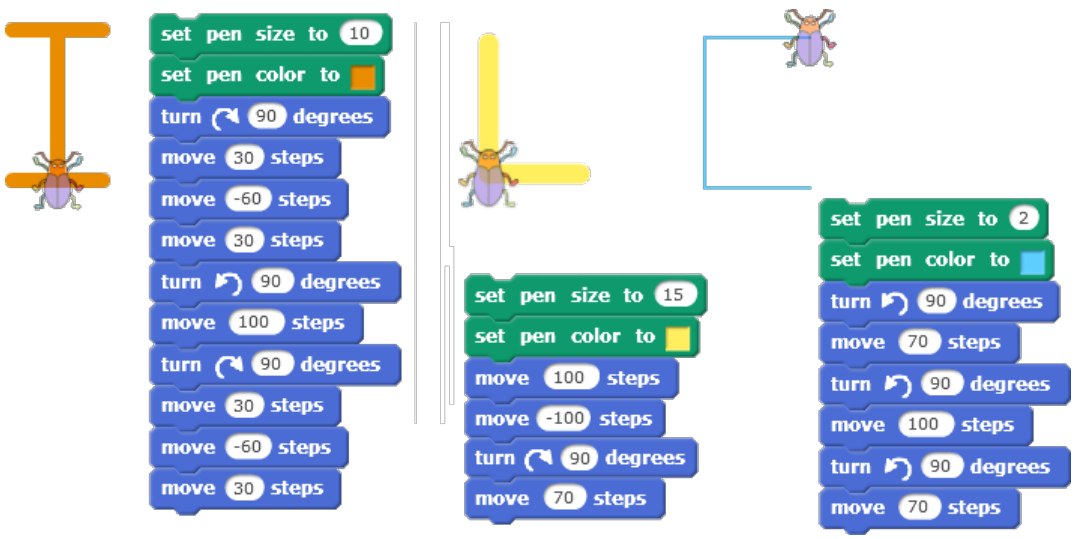
Usually we will use the top one with the colour picker. The other two refer to colours by numbers between 0 and 200. We will use them later, for programmers they are very important. However, the first block with the colour picker is the easiest to begin with.

ADDITIONAL SUPPORT CONTINUED

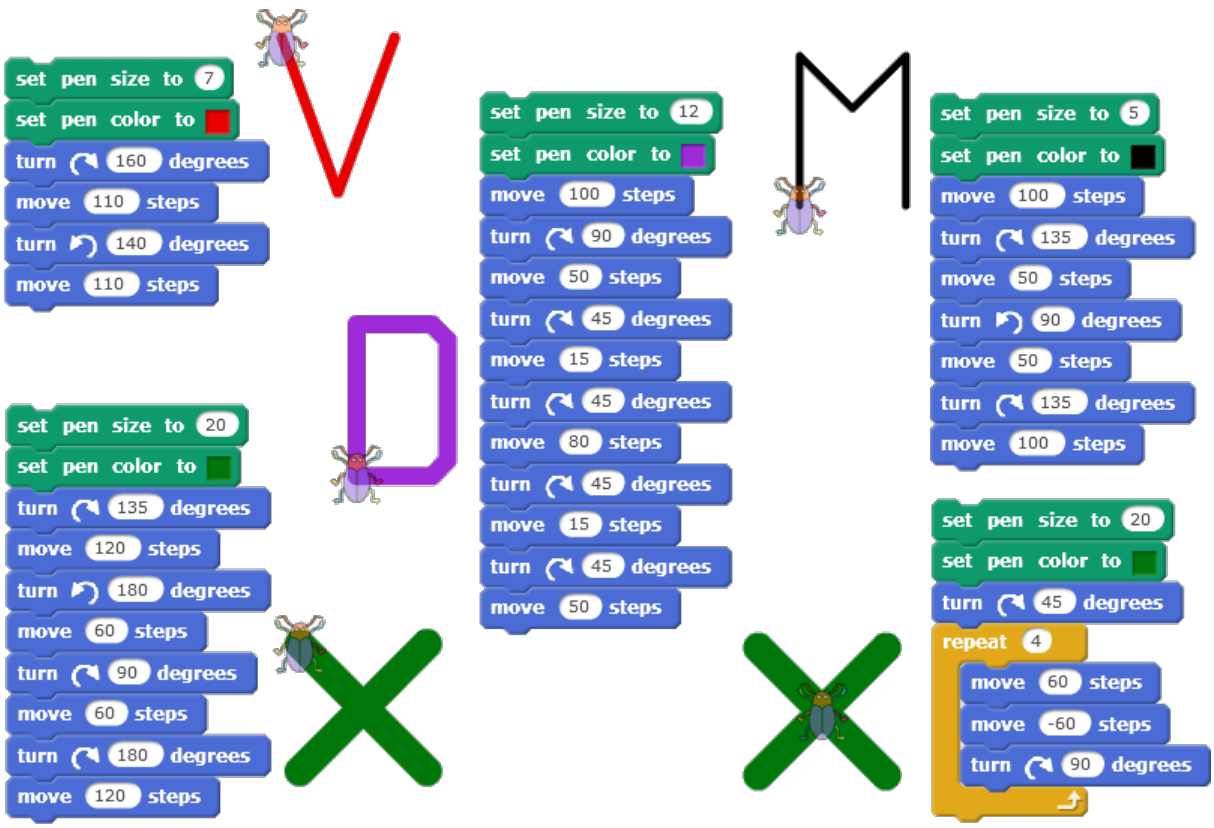
Encourage pupils to create scripts to draw their numerals so that they can recreate their drawing and explain to others how they have drawn them.

Below are the scripts for the roman numerals that are the most straightforward to draw (these are example scripts).

Note – the Beetle sprite (slightly ghosted) shows **the starting point of the drawing and the initial direction**. However, there are always several alternative approaches.



[Extension] Here are the scripts for the remaining roman numerals up to 1000.



Swapping Blocks

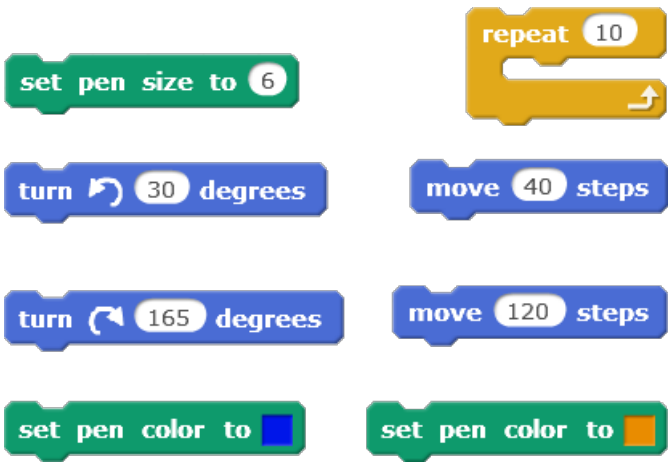
Learning Objectives

Explore how to create different line drawings by using a limited number of blocks.
Explain how the order of the blocks within a script changes what is drawn on the stage.

Activity Instructions

- Pupils open project **2-Swapping Blocks**, save as a copy and add their name(s) to the title.
- Pupils look at the eight individual blocks in the scripts area and discuss what they do.
- Pupils run the *setup script* (to initialize the drawing) and discuss the meaning of its blocks.
- Pupils combine the provided blocks in any way they want to make a script and create something. However, they must follow the three rules below:

  - **You cannot duplicate or drag in any new blocks – you should have no more than the original eight blocks in your script.**
  - **You don't have to use all the blocks.**
  - **You cannot change the values inside the blocks.**



Things to Note

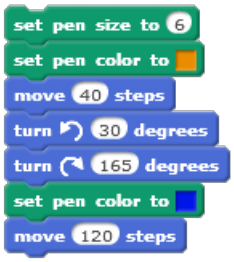
- It is very important to stick to the three rules for this activity.

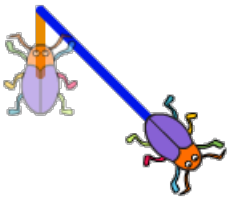
Class Discussion

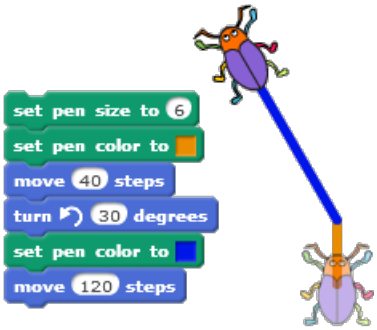
- What drawing have you created? Which blocks did you use and in what order?
- Did you try putting the **turn** and **move** blocks in front of and inside the **repeat** block – what was the difference?
- What happened if you put the two **set color** blocks next to one another?
- What is the total number of steps your Beetle moved to create your drawing?

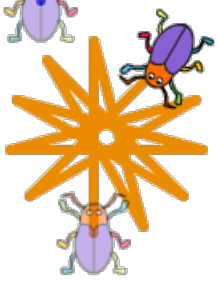
ADDITIONAL SUPPORT

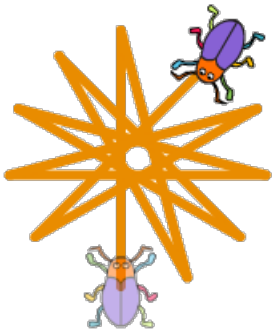
Below are some example drawings that could be created.

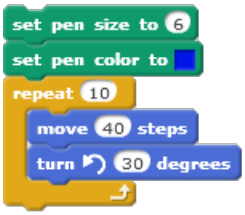


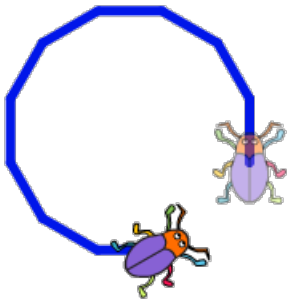


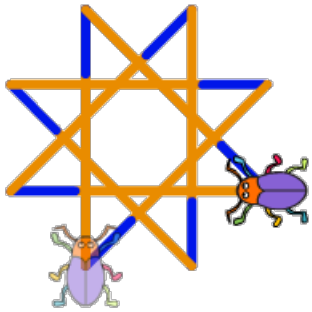


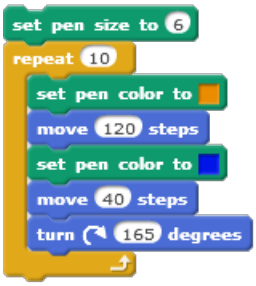


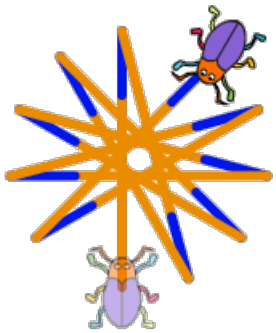


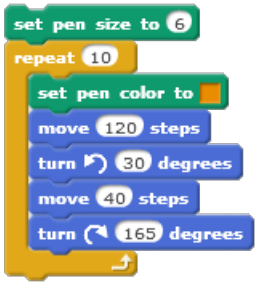


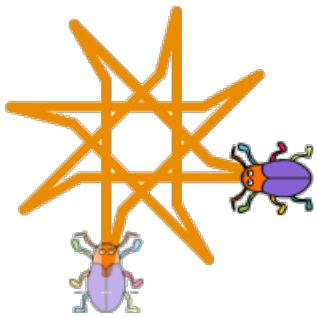


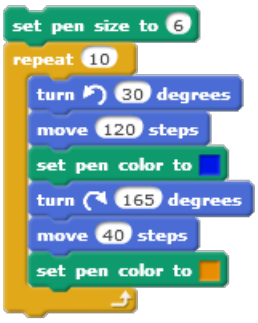


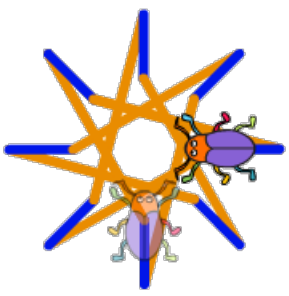


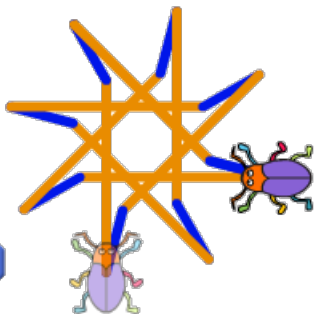










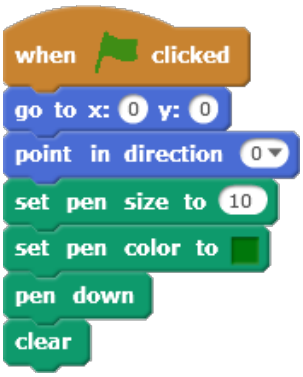


ADDITIONAL SUPPORT CONTINUED

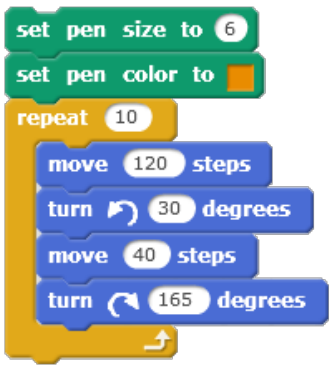
Discuss with pupils the role of setting pen size and/or pen colour before the **repeat** block, inside it or after it:

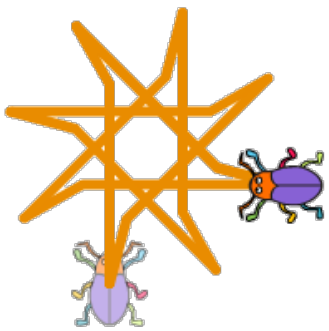
- If put in front of the **repeat**, the **set pen size …** and/or **set pen colour** blocks will reset the pen size and colour from the *setup script* (see 1).
- If two **set pen colour** blocks are attached together the first one has no role (see 2 and 3) and the second one will set the colour.
- If **set pen size …** block is moved inside **repeat** (see 4), the output drawing will be the same, however, the same setting is inefficiently rerun 10 times.
- Setting pen size, or pen colour (see 5) or turning the Beetle (see 6) after the **repeat** block will not effect the drawing.

setup script

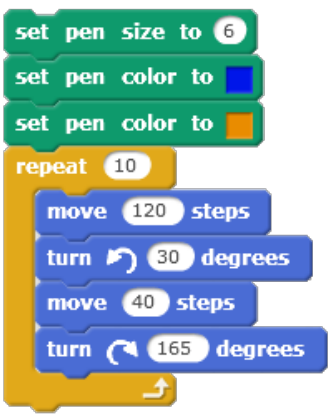


1

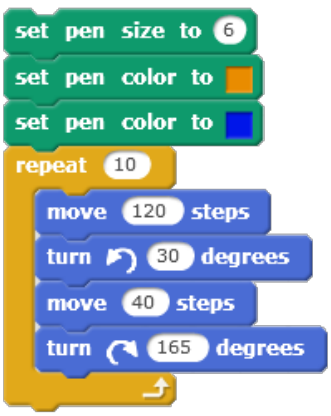


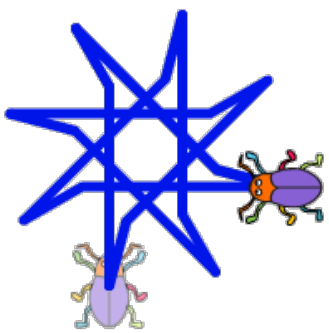


2

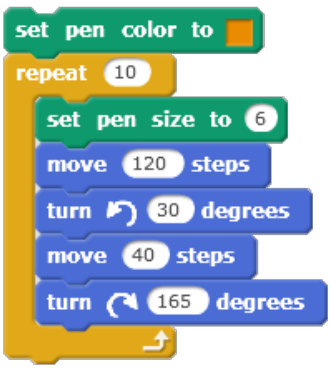


3

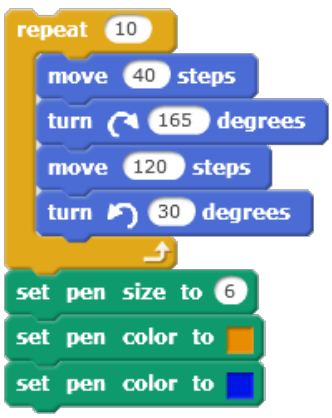




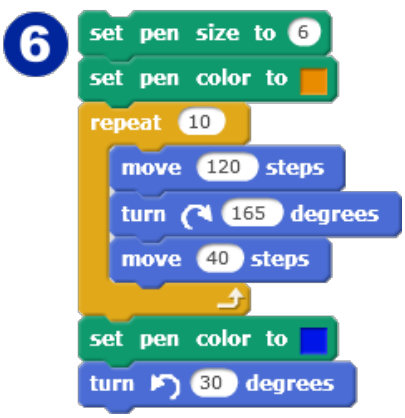
4



5



6



MODULE 2 • INVESTIGATION 1 • ACTIVITY 2.1.3

Unplugged: I am Beetle

LEARNING OBJECTIVES

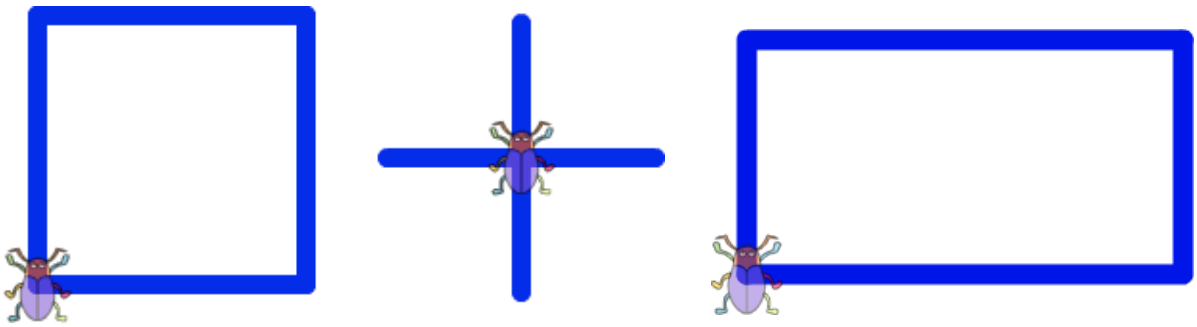
Explore different algorithms for drawing using our bodies.
Exchange drawing instructions to enable someone else to physically trace the drawing.

ACTIVITY INSTRUCTIONS

This activity can be done in pairs or as a class.

- One pupil should be shown one of the drawings (see additional support) and imagine this drawing on the floor.
- They must then instruct another pupil how to move so that they trace out this drawing on the floor with their feet.
- The second pupil should then try to imagine what the drawing the first pupil is looking at is and draw this on the SmartBoard or another piece of paper.
- Compare this drawing with the drawing on the card.
- Now switch roles and trace out a different drawing.

Alternatively you could use masking tape to draw the shapes on the floor and blind fold the pupil who is being instructed how to move.

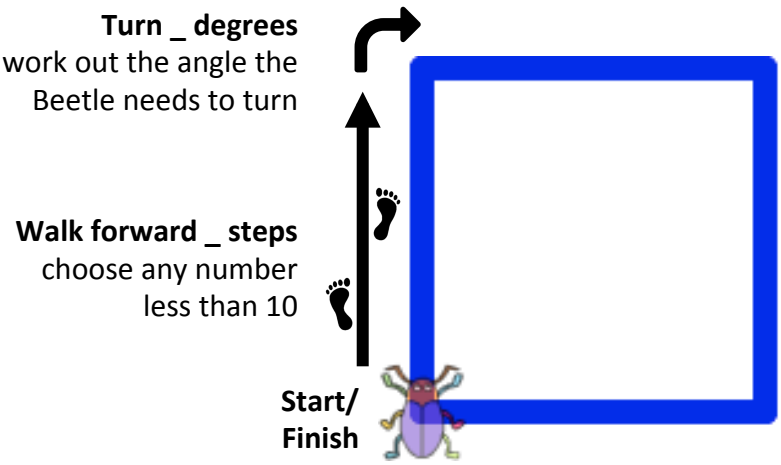


CLASS DISCUSSION POINTS

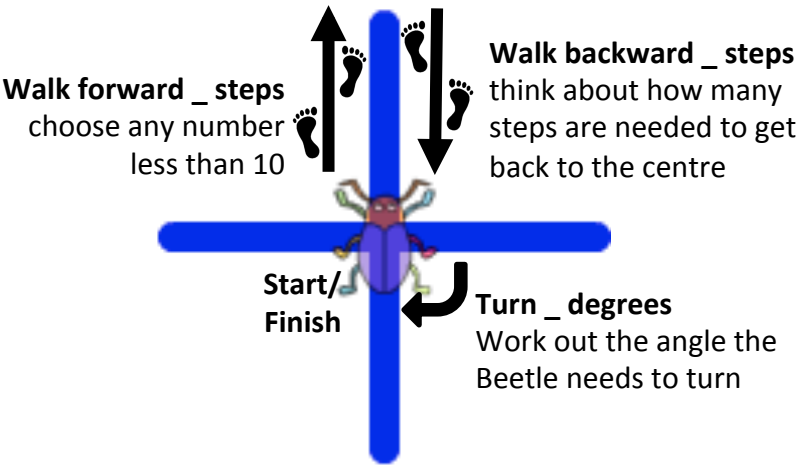
- Did your partner always move where you wanted them to? If not why not?
- What was important for you to make clear when instructing them what to do?
- What information did you remember to help you recreate the drawing on paper?

Print and cut out the cards below (two pages):

- Imagine the drawing below is drawn on the floor and your partner is the Beetle.
- Tell your partner where to walk so that they trace out this drawing with their feet.
- Your partner (the Beetle) should finish at the same starting position and be pointing in the same direction.



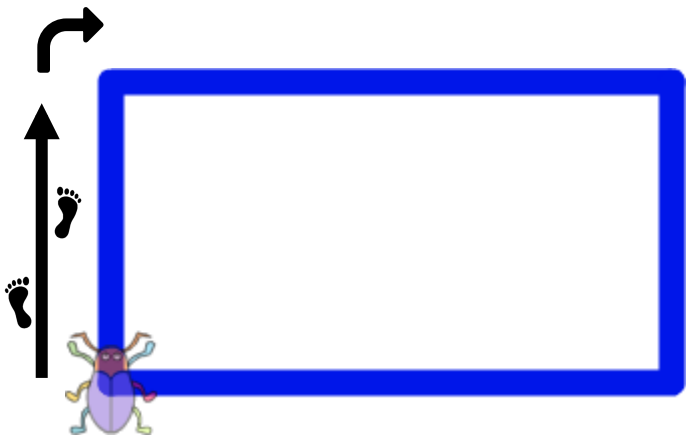
- Imagine the drawing below is drawn on the floor and your partner is the Beetle.
- Tell your partner where to walk so that they trace out this drawing with their feet.
- Your partner (the Beetle) should finish at the same starting position and be pointing in the same direction.



- Imagine the drawing below is drawn on the floor and your partner is the Beetle.
- Tell your partner where to walk so that they trace out this drawing with their feet.
- Your partner (the Beetle) should finish at the same starting position and be pointing in the same direction.

Turn _ degrees
work out the angle the Beetle needs to turn

Walk forward _ steps
choose any number less than 5



Start/Finish

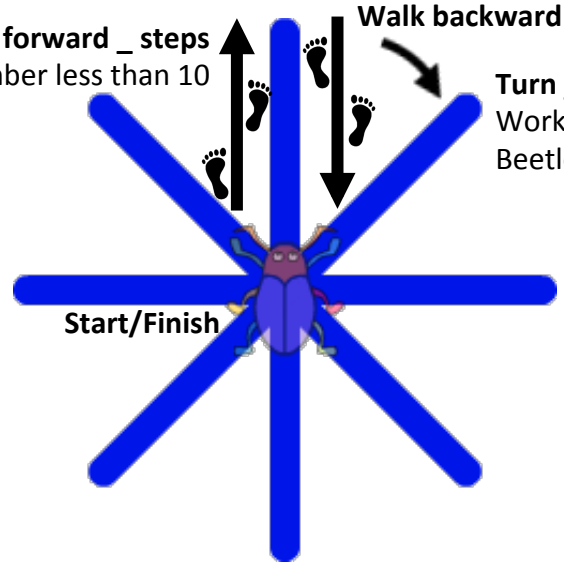
[Extension]:

- Imagine the drawing below is drawn on the floor and your partner is the Beetle.
- Tell your partner where to walk so that they trace out this drawing with their feet.
- Your partner (the Beetle) should finish at the same starting position and be pointing in the same direction.

Walk forward _ steps
choose any number less than 10

Walk backward _ steps

Turn _ degrees
Work out the angle the Beetle needs to turn



Start/Finish

MODULE 2 • INVESTIGATION 1 • ACTIVITY 2.1.4

Different Drawing Algorithms

LEARNING OBJECTIVES

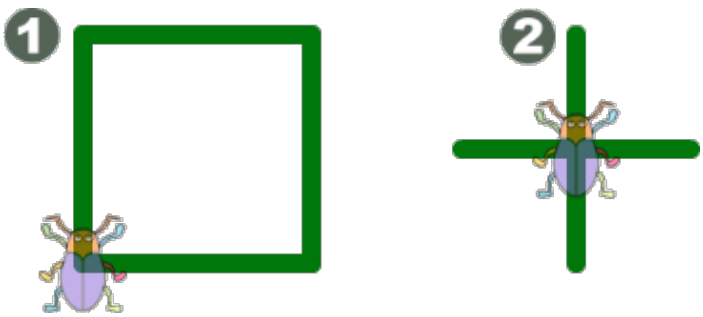
Explore the physically traced out algorithms within the Scratch environment.
Explain how to build a script based on a specific algorithm.
Exchange ideas about how to create a script for drawings based on different algorithms.
bridgE to mathematical fluency (identifying multiple approaches to solving a problem).

ACTIVITY INSTRUCTIONS

- Pupils continue in project **2-Swapping Blocks**.

In the previous activity we followed two different algorithms to trace out shapes with our bodies. Now we want to use the same algorithms in Scratch.

- Pupils work in pairs. Within the pairs each pupil chooses a different drawing (below) and then individually builds the script in Scratch to recreate that drawing (Beetle indicates where to start).
- Once both pupils have built their script, each person will explain to their partner what they have done and then help their partner to build the script themselves, so both pupils end up with a script for each drawing.



- **[Extension]** Imagine Beetle can only move backwards – recreate Drawing 1 only moving the Beetle sprite backwards.
- **[Extension]** Imagine Beetle can only more forwards – recreate Drawing 2 only moving the Beetle sprite forwards.

THINGS TO NOTE

- Note that in both algorithms (and in both of the extensions above) the Beetle sprite starts and finishes at exactly the same position, pointing in the same direction.

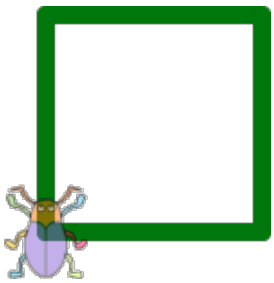
CLASS DISCUSSION POINTS

- How did you explain your script to your partner? Did you have any difficulties doing this?
- What were the differences between the two scripts?
- In Drawing 1 how could you calculate the total number of steps your Beetle moved? What is this distance known as in mathematics? *(Perimeter)*
- In Drawing 1 how could you calculate the total number of degrees your Beetle turned?

Algorithm 1 is to repeat the following steps:

- **move**
- **turn**

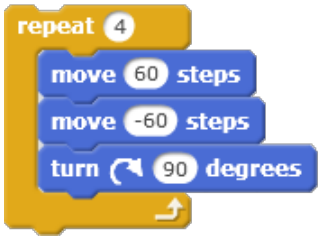
This algorithm is used by Drawing 1 (script below).

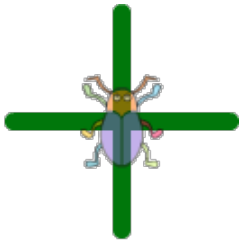


Algorithm 2 is to repeat the following steps:

- **move**
- **move backwards**
- **turn**

This algorithm is used by Drawing 2 (script below).



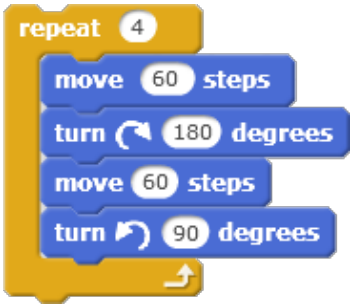


We have already used these two algorithms in Module 1 when drawing the symmetrical patterns. Discuss this with your pupils.

[Extension]

The script for Drawing 1 only moving backwards is:

The script for Drawing 2 only moving forwards is:



MODULE 2: INVESTIGATION 2

Drawing Polygons

KEY LEARNING OBJECTIVE: Build on previous knowledge of the pen tool, drawing polygons and defining new blocks. Use defined blocks to create more complex drawings.

This investigation continues to focus on the pen tool and includes activities which explore drawing different regular polygons. It also returns to the concept of definition, providing opportunities to use and modify pre-defined blocks as well as defining and combining new blocks to create a more complex drawing. The investigation comprises of 4 activities.

- **Activity 2.2.1** – Drawing Regular Polygons
- **Activity 2.2.2** – Unplugged: Polygon Scripts
- **Activity 2.2.3** – Using and Defining More Blocks
- **Activity 2.2.4** – Combining New Blocks

Scratch project **2-Drawing Polygons**

LINKS TO PRIMARY NATIONAL CURRICULUM

CURRICULUM OBJECTIVES	LINK WITH SCRATCHMATHS
Computing Design, write and debug simple programs that accomplish specific goals. Solve problems by decomposing them into smaller parts. Use logical reasoning to explain how some simple algorithms work.	▶ Pupils are required to create a script that draws specific regular polygons. ▶ Pupils are required to draw a house by defining blocks for the different polygons within it. ▶ Pupils are required to identify the polygons a given script will draw.
Mathematics Compare and classify geometric shapes. Plot specific points and draw sides to complete a given polygon. Use properties of a rectangle to deduce related facts. Distinguish between regular and irregular polygons based on reasoning about equal sides and angles. Identify angles at a point and one whole turn (360°). Identify angles on a straight line and ½ a turn (180°).	▶ Pupils are required to identify and draw polygons including squares and triangles. ▶ Pupils are required to use reasoning about polygons to match scripts and drawings of a square, triangle, hexagon and rectangle. ▶ Pupils are required to use knowledge of angles to draw squares and triangles.

Drawing Regular Polygons

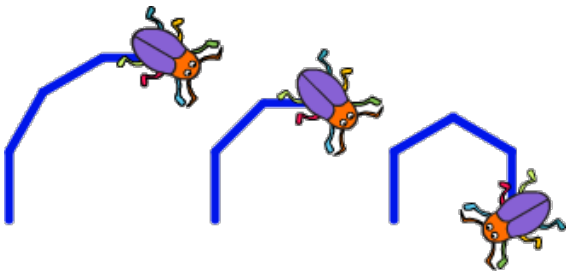
Learning Objectives

Explore how to use the pen tool to draw regular polygons.
bridgE to knowledge about properties of different regular polygons.

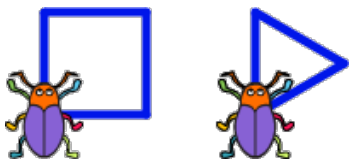
Activity Instructions

- Pupils open project **2-Drawing Polygons**, save as a copy and add their name(s) to the title.
- Pupils run the *setup script* – this will put the pen down so the Beetle is ready to draw.
- Pupils snap together one **move** block and one **turn** block in the scripts area.
- Pupils choose and set the values in each block and then click their short script several times (without using **repeat**).

They will come up with different polygons – discuss which polygons they drew.



- Pupils add the **repeat** block around their script and set it to the smallest number they can to complete their polygon in one click (i.e. to make it a closed shape).



- Pupils create a script to draw a **square** and then an **equilateral triangle** (see additional support).

Things to Note

- Ensure pupils pick a value larger than 10 within the **move** block otherwise the Beetle will cover up the whole drawing.
- Drawing a very long line with one **move** instruction may make the Beetle sprite disappear from the stage – you can use the **go to x: 0 y: 0** block to get it back to the centre of the stage.
- It is recommended to pick a value not bigger than 12 in the **repeat** block (so angles are not smaller than 30 degrees in the turn block) otherwise the shape will *look like* a circle.
- Drawing the triangle requires knowledge of the interior and exterior angles, which is quite challenging – physically acting out the drawing process can help understanding of this concept (see additional support).

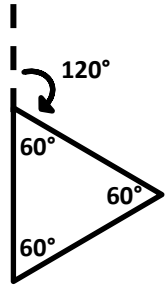
Class Discussion Points

- How did you work out how to draw your square or triangle?
- How many sides did your other polygons have? What polygons did you draw?
- How many degrees did the Beetle turn in total to make it a closed shape? How many steps did the Beetle move in total?
- What is the link between the move and repeat blocks in your polygon script?
- Did you manage to draw an equilateral triangle? How did you build your script to ensure it was equilateral?
- Why was it more difficult to draw a triangle than it was to draw a square?

ADDITIONAL SUPPORT

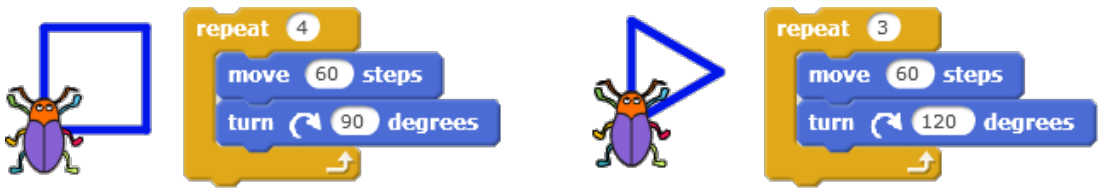
To help with understanding of how to draw polygons, get your pupils to physically act out the drawing process:

- Ask a child to pretend they are the Beetle.
- Instruct them (or ask another child to instruct them) to step forward a certain number of steps and then turn 90 degrees repeatedly to draw a square.
- Do the same for the triangle.
- Discuss the different angles that they had to turn in each case.



As with the previous investigation it could be helpful to draw out the shapes using masking tape on the floor. You could also use masking tape to draw a triangle on a straight line and point out the difference between the interior and exterior angles (see picture above).

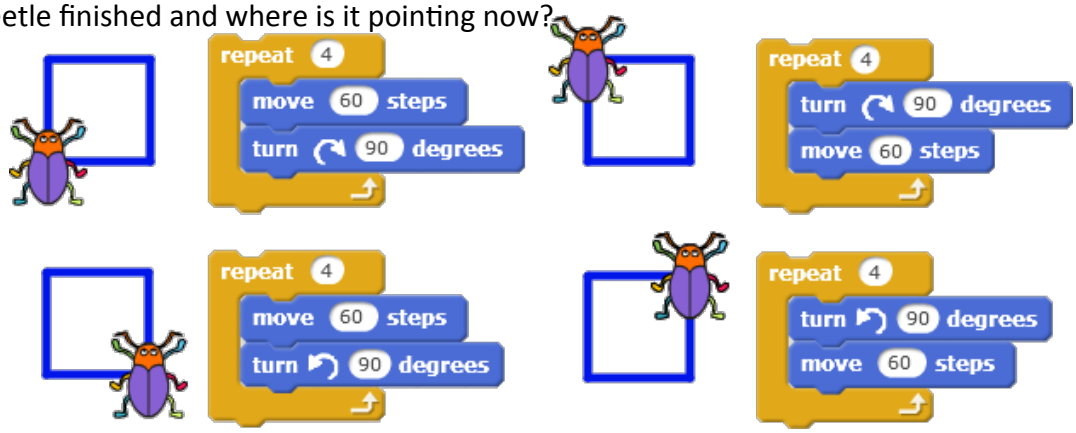
The scripts below can be used to draw a square and an equilateral triangle.



The picture below illustrates there are several ways how to draw a square (or any polygon).

However, if the first block in the **repeat** is **turn**, it is much more difficult and confusing to use such script in more complex drawings like towers of squares, houses etc. which pupils will start drawing later in this investigation.

Encourage pupils to start inside **repeat** with the **move** block (see the left column). Also, always discuss: Where did the Beetle start and where was it pointing? Where has the Beetle finished and where is it pointing now?



Unplugged: Polygon Scripts

Learning Objectives

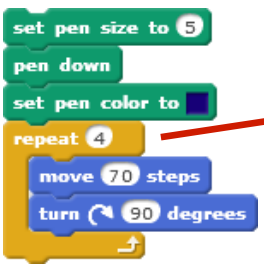
bridgE to knowledge about polygons (using mathematical fluency) and problem solving (logical reasoning/visualising).
Envisage what polygon a given script will draw.

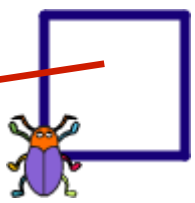
Activity Instructions

- Print and distribute the unplugged pupil worksheet 2.2.2 or do the activity as a class (for higher attainers use the extension version on page 19).
- Ask the pupils to match the script with the polygon it creates.
- Discuss the answers as a class. Choose 4 students to go to the front and follow the instructions for each of the the 4 scripts. Which blocks helped to identify the shape?

Worksheet Solution

1

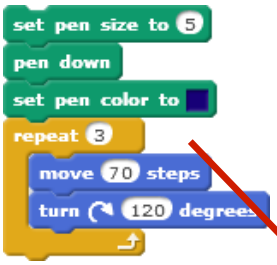


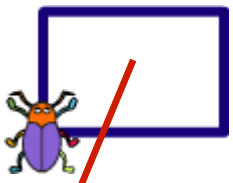


This is the correct polygon because:

- it has 4 equal sides and 4 right angles
- the blocks in the **repeat** are run 4 times
- the **move** block is 70 steps
- the Beetle always turns right by 90 degrees

2

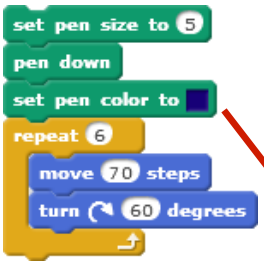


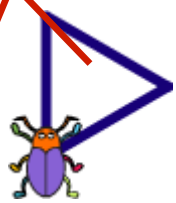


This is correct polygon because:

- it has 2 pairs of equal sides opposite each other, a shorter pair and a longer pair
- it has 4 right angles
- the blocks in the **repeat** are run 2 times
- there are 2 **move** blocks in the **repeat**
- there are 2 **turn** blocks in the **repeat** and the Beetle always turns right by 90 degrees

3

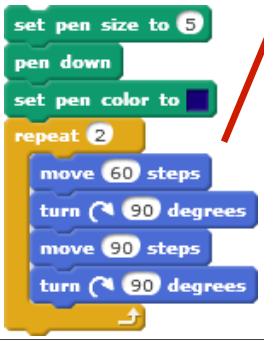


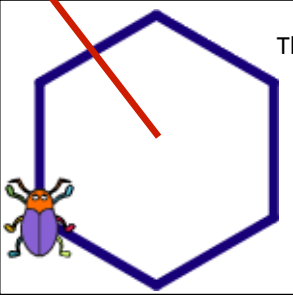


This is correct polygon because:

- it has 3 equal sides and 3 equal angles
- the blocks in the **repeat** are run 3 times
- the **move** block is 70 steps
- the Beetle always turns right by 120 degrees

4





This is correct polygon because:

- it has 6 equal sides and 6 equal angles
- the blocks in the **repeat** are run 6 times
- the **move** block is 70 steps
- The Beetle always turns right by 60 degrees

PRINT

INVESTIGATION 2
Activity 2.2.2

NAME

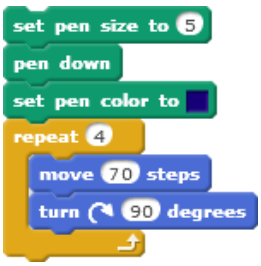
WHAT TO DO

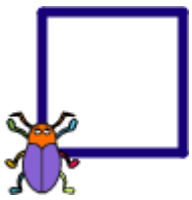
Match the script with the polygon that it would draw when you click on it by **drawing a line** between the polygon and the script.

Fill in the gaps in the text below to show how you worked out the answer.

SHAPE SCRIPTS

1

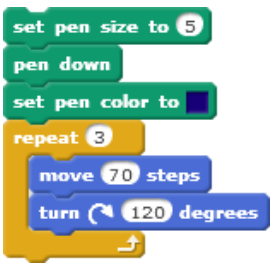


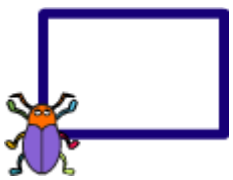


This is the correct polygon because:

- it has ____ equal sides and _____ right angles
- the blocks in the **repeat** are run _____ times
- the **move** block is _____ steps
- the Beetle always turns right by _____ degrees

2

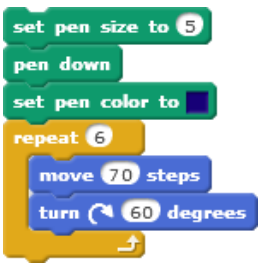




This is correct polygon because:

- it has _____ pairs of equal sides opposite each other, a shorter pair and a longer pair
- it has _____ right angles
- the blocks in the **repeat** are run _____ times
- there are _____ **move** blocks in the **repeat**
- there are _____ **turn** blocks in the **repeat** and the Beetle always turns right by _____ degrees

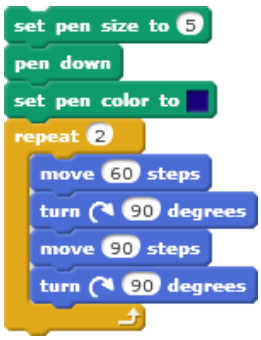
3

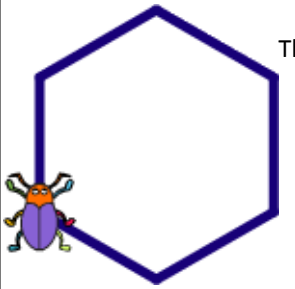


This is correct polygon because:

- it has _____ equal sides and _____ equal angles
- the blocks in the **repeat** are run _____ times
- the **move** block is _____ steps
- the Beetle always turns right by _____ degrees

4





This is correct polygon because:

- it has _____ equal sides and _____ equal angles
- the blocks in the **repeat** are run _____ times
- the **move** block is _____ steps
- the beetle always turns right by _____ degrees

PRINT

INVESTIGATION 2
Ext. Activity 2.2.2

NAME

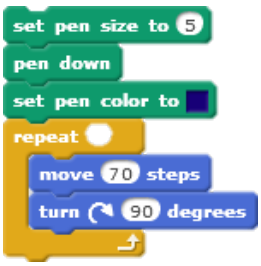
WHAT TO DO

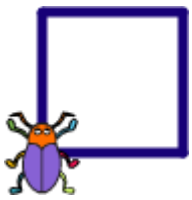
Match the script with the polygon that it would draw when you click on it by **drawing a line** between the polygon and the script.

Fill in the gaps in the text below to show how you worked out the answer.

SHAPE SCRIPTS

1

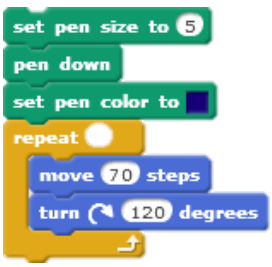


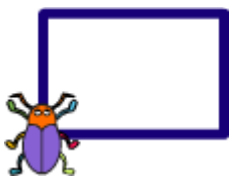


This is the correct polygon because:

- it has ____ equal sides and _____ right angles
- the blocks in the **repeat** are run _____ times
- the **move** block is _____ steps
- the Beetle always turns right by _____ degrees

2

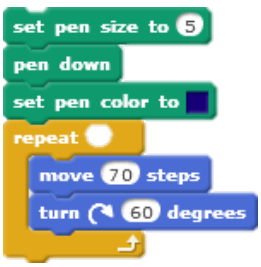




This is correct polygon because:

- it has _____ pairs of equal sides opposite each other, a shorter pair and a longer pair
- it has _____ right angles
- the blocks in the **repeat** are run _____ times
- there are _____ **move** blocks in the **repeat**
- there are _____ **turn** blocks in the **repeat** and the Beetle always turns right by _____ degrees

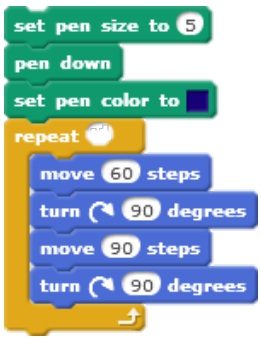
3

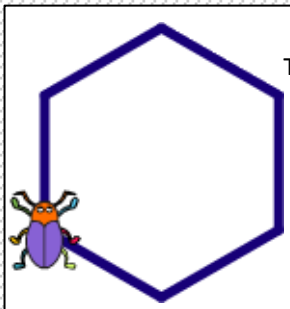


This is correct polygon because:

- it has _____ equal sides and _____ equal angles
- the blocks in the **repeat** are run _____ times
- the **move** block is _____ steps
- the Beetle always turns right by _____ degrees

4





This is correct polygon because:

- it has _____ equal sides and _____ equal angles
- the blocks in the **repeat** are run _____ times
- the **move** block is _____ steps
- the beetle always turns right by _____ degrees

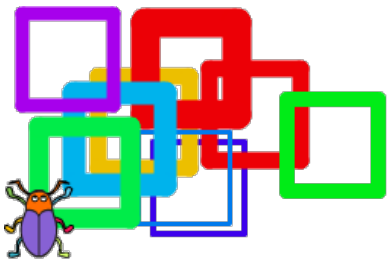
Using and Defining More Blocks

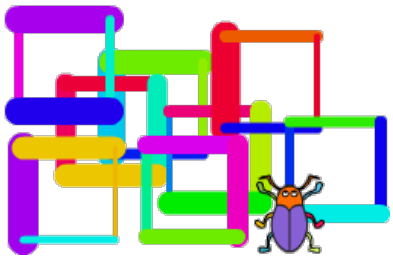
Learning Objectives

Explore how to use pre-defined new blocks.
Explore how to define a new block that will draw a square.
Explain why a new block should be given a meaningful name.

Activity Instructions

- Pupils continue in their **2-Drawing Polygons** project.
- Pupils are encouraged to go to the **More Blocks** group where they will find several new useful blocks (see additional support).
- Pupils add some of these new blocks to the top of their square script or inside it and experiment with them. They drag their Beetle around the stage to draw squares with different coloured and sized lines.





- In the **More Blocks** group pupils click the **Make a Block** button (see 1 in additional support) and give a new block a meaningful name e.g. *square* (see 2).
- A new **define square** hat block will appear in the scripting area (see 3). Pupils will drag it and put it as a hat on top of their script (see 4).

This is like **teaching a sprite a new word and telling it *"if you want to draw a square follow this script"***. From now on, a new block **square** will appear in the list of **More Blocks** (see 5) and it can used in scripts like any other block.

Things to Note

- Pupils should not edit the scripts that define the three new blocks **set random pen size**, **set random pen colour** and **set random pen shade** as this will change what they do when used in their scripts. (The definitions are 'hidden' in the far right part of the scripts area.) We will explore and modify them in the next investigation.

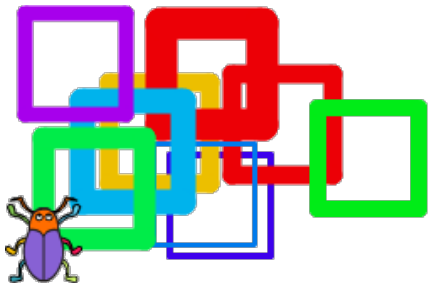
Class Discussion Points

- What is the difference between *pen colour* and *pen shade*?
- Where did you try placing the **set random** blocks in your script – how did this change your drawing?
- Why would it be a good idea to define a new block for a script you use many times in your project (e.g. drawing a square)?
- Why is it important to give a new block a meaningful name?

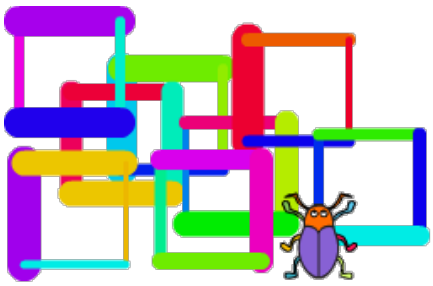
ADDITIONAL SUPPORT

These are the new blocks that you will find in the **More Blocks** group.

Below are the scripts using these new blocks.



```
set random pen size
set random pen colour
repeat 4
    move 40 steps
    turn 90 degrees
```



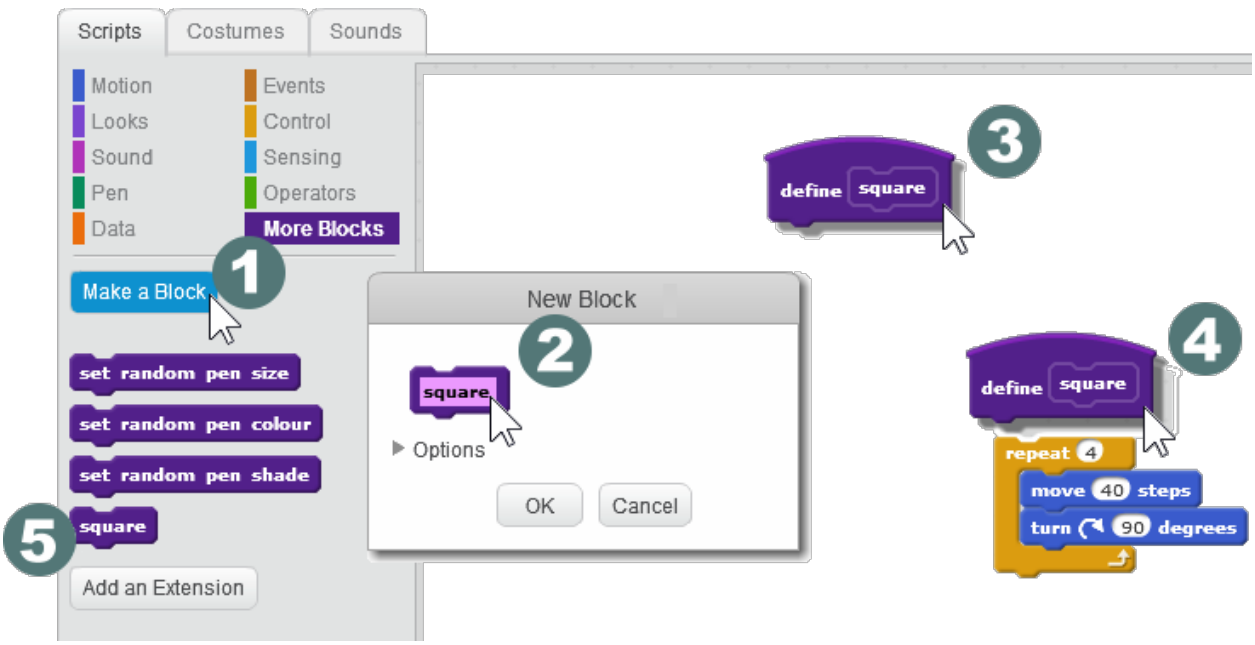
```
repeat 4
    set random pen size
    set random pen colour
    move 40 steps
    turn 90 degrees
```

These scripts are also worth exploring:

```
set random pen size
repeat 4
    set random pen colour
    move 40 steps
    turn 90 degrees
```

```
set random pen colour
repeat 4
    set random pen size
    move 40 steps
    turn 90 degrees
```

Below are the steps for creating your new **square** block.



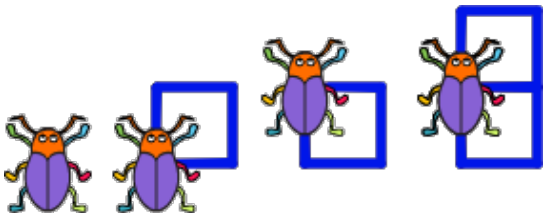
Combining New Blocks

Learning Objective

Explore how to use new blocks within a larger drawing.

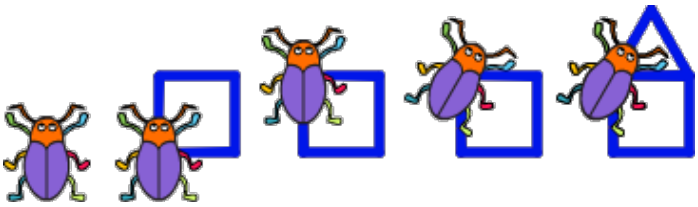
Activity Instructions

- Pupils continue in their project **2-Drawing Polygons**. They will use their new block **square** created earlier.
- Pupils build a script, using their **square** block, to **draw a tower of two squares**.

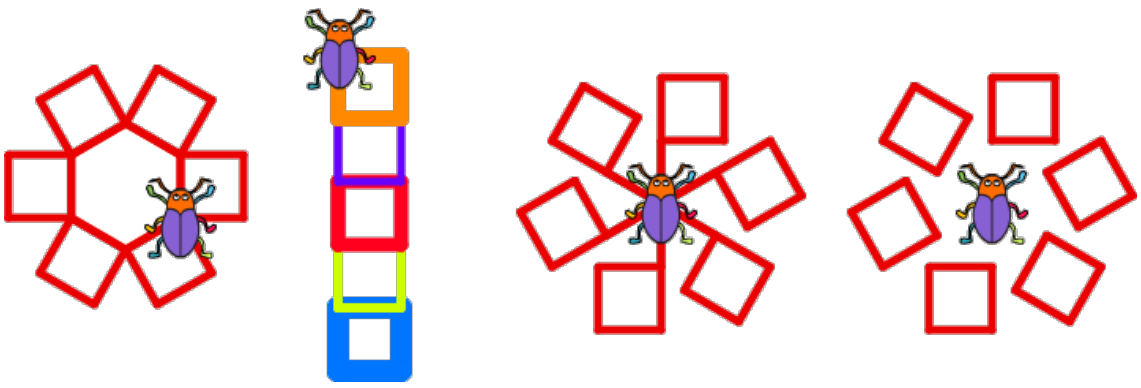


- Pupils make another new block **triangle** with sides that are the same length as their square.

- Pupils try combining their **square** and **triangle** blocks in one script to **draw a house**.



- **[Extension]** Pupils try building scripts to draw some of the example pictures below using just their **square** block.



Class Discussion Points

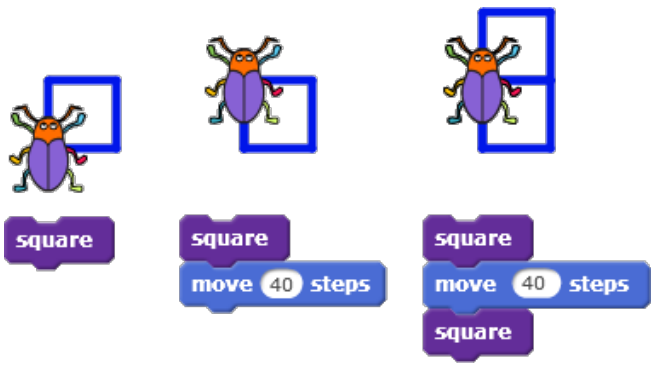
- What problems did you encounter when building a script for drawing a tower of two squares and how did you solve these?
- How did you use your new blocks to create a house?
- Did defining a new block for **square** and **triangle** make it easier to draw the house? How?
- What problems did you encounter when drawing your house?
- How did you discover the angle that you need to turn by in order to draw the roof on your house correctly?

ADDITIONAL SUPPORT

The script below will create a tower of two squares:

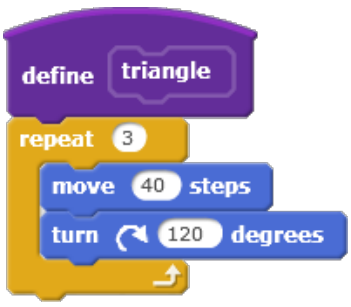
Encourage your pupils to build it and run it step by step thinking about the questions below.

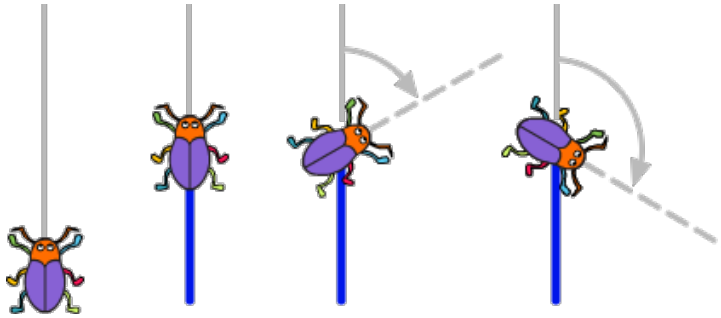
- ► Where will my Beetle finish after drawing the first **square**?
- ► Which direction will it point in?
- ► Where exactly do I want it to draw the second **square**? Which block will make the Beetle get there?
- ► Will it then point in the correct direction?
- ► Where will it finish after drawing the second **square**?



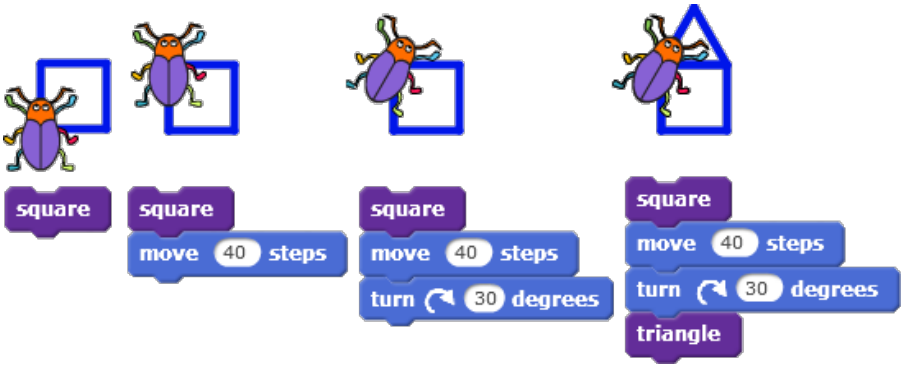
Note also that if the definition of our **square** block started with the **turn** block, it would be much more difficult to build the tower and answer the suggested questions.

The script below defines the **triangle**. Your pupils should discover proper turning by physically stepping through the drawing and/or by playing the Beetle and drawing on the paper. Discuss the thought process they went through when solving this problem.





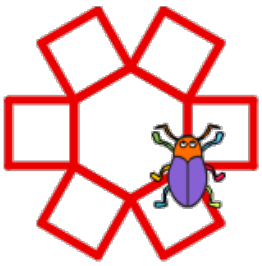
When building the script of a house, encourage your pupils to apply the same strategy and questions as above when building the tower of two squares. The most challenging issue is to discover the correct angle for the **turn 30** block.



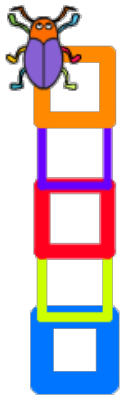
Below is an alternative solution for the case when **square** and **triangle** are defined with **turn left**...

ADDITIONAL SUPPORT CONTINUED

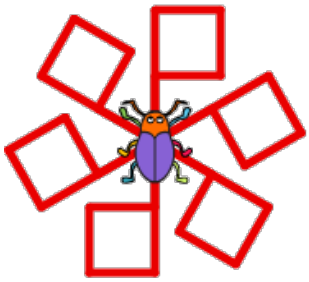
[Extension] The scripts below will create the extension pictures.



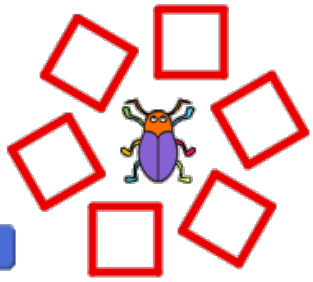
```
set pen color to
repeat 6
    square
    move 40 steps
    turn 60 degrees
```



```
repeat 5
    set random pen size
    set random pen colour
    square
    move 40 steps
```

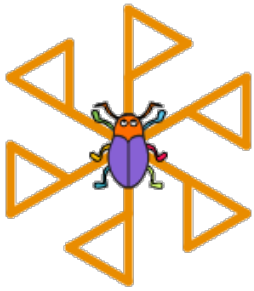


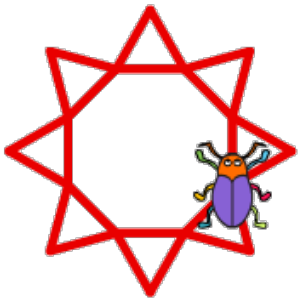
```
set pen color to
repeat 6
    move 40 steps
    square
    move -40 steps
    turn 60 degrees
```

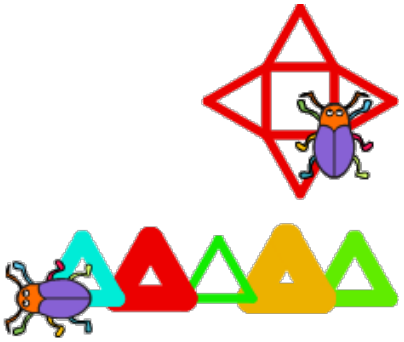


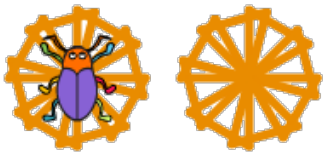
```
pen up
set pen color to
repeat 6
    move 40 steps
    pen down
    square
    pen up
    move -40 steps
    turn 60 degrees
```

[Extension] Similar pictures can be created by using the **triangle** block.









```
set pen color to
repeat 6
    move 40 steps
    triangle
    move -40 steps
    turn 60 degrees
```

```
set pen color to
repeat 4
    triangle
    turn 90 degrees
```

```
set pen color to
repeat 8
    triangle
    turn 45 degrees
```

MODULE 2: INVESTIGATION 3
Discovering Dots

KEY LEARNING OBJECTIVE: Build on previous knowledge of using the pen tool to draw dotted and dashed lines or dots of any size and colour. Explore how to exploit randomness in different desired ways.

investigation explores the use of the pen tool to draw dots in lines, circles and randomly across the stage. It provides the opportunity for pupils to use multiple new blocks within the same script, predict the visual outcome of different scripts by looking at the positioning of the pen block within a script as well as introducing the concept of randomness. The investigation comprises of 4 activities.

- **Activity 2.3.1** – Dots and Dashes
- **Activity 2.3.2** – Unplugged: Picture Predictions
- **Activity 2.3.3** – Swarming Dots
- **Activity 2.3.4** – A Sky Full of Stars

Scratch projects **2-Dots and Dashes**
2-Swarming Dots

LINKS TO PRIMARY NATIONAL CURRICULUM

CURRICULUM OBJECTIVES	LINK WITH SCRATCHMATHS
Computing Design, write and debug simple programs that accomplish specific goals. Use sequence, selection, and repetition in programs. Use logical reasoning to explain how some simple algorithms work.	▶ Pupils are required to create scripts to draw dotted and dashed lines. ▶ Pupils are required to use logical reasoning skills to predict how changing the sequence of blocks would affect the outcome of a script.
Mathematics Multiply numbers up to 4 digits by one or two digit numbers. (KS3) Simple probability experiments involving randomness. Interpret negative numbers in context. Describe positions on the full coordinate grid.	▶ Pupils are required to multiple two digit numbers together to calculate the total turn within a given script. ▶ Pupils are required to use randomness within their scripts and to experiment with the range from which values can be randomly selected. ▶ **[Ext]** Pupils may use knowledge of negative numbers and coordinates to define the area of the stage that can be drawn on.

Module 2 • Investigation 3 • Activity 2.3.1

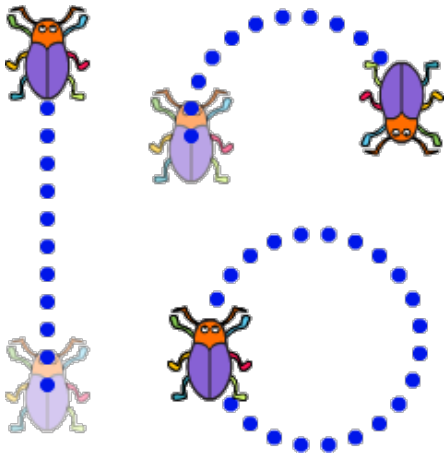
Dots and Dashes

Learning Objectives

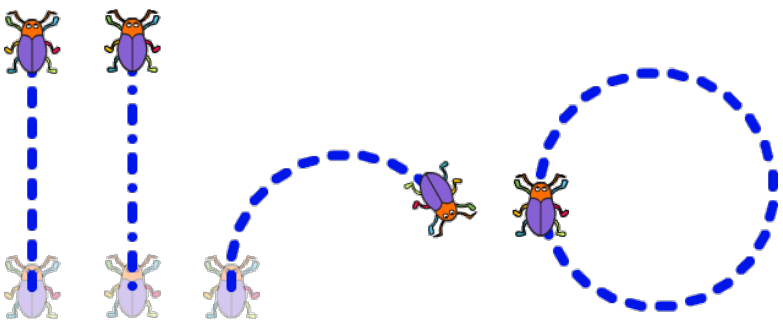
Explore how to use the pen tool to draw dotted and dashed lines.
Explore how to combine multiple new blocks together within the same script.

Activity Instructions

- Pupils open the **2-Dots and Dashes** project, save as a copy and add their name(s) to the title.
- Pupils run the *setup script* – this will initialize the pen colour and pen size of the Beetle.
- Pupils experiment with the **pen down** and **pen up** blocks to find out how the Beetle can draw a dot.
- As we will be drawing dots many times, encourage pupils to make their own new block **dot** to draw a dot.
- Pupils explore using their new **dot** block in scripts that draw a dotted line and a dotted circle.



- Now pupils create a new block called **dash** and use this to create a dashed line.
- Pupils try combining their two new blocks **dot** and **dash** together to draw a line and circle with both dots and dashes.



Things to Note

- If we increase the pen size of the Beetle, the space between the dots and dashes should be adjusted as well.

Vocabulary

Pen up block stops the sprite from continuously drawing a trail wherever it moves (if the **pen down** block has previously been used).

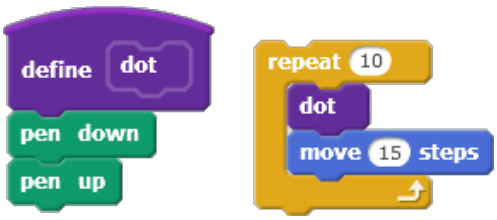
Class Discussion Points

- How did you draw a dot?
- What was the difference between drawing a dot and a dash?
- How did you ensure you had a space between your dots and dashes?
- Where did you place the block to create this space? Could you place it in the definitions of the dot and dash blocks?

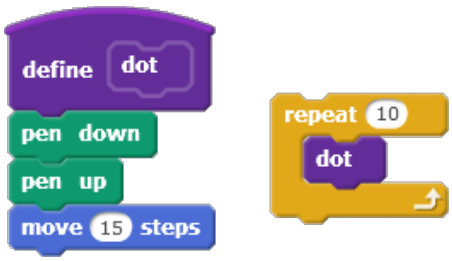
ADDITIONAL SUPPORT

Whenever the Beetle sprite puts its **pen down** and then **pen up** it draws a dot. The definition of the **dot** block is short, but giving a name to that short script is meaningful and will increase the readability of our scripts.

To change the size of the **dot** simply change the size of the pen (before the **dot** block is run or directly in the *setup script*). The space between the dots can either be added in the script to draw a line of dots (see example 1) or directly to the definition (see example 2).

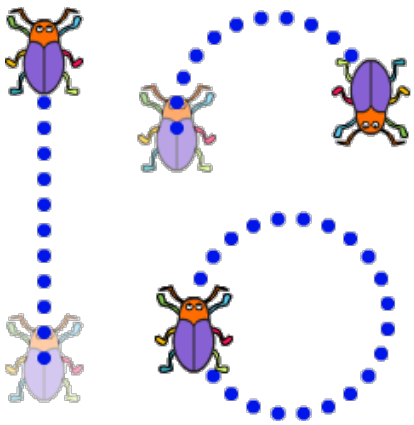


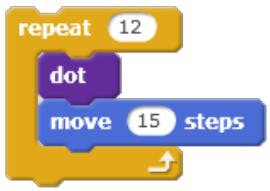
Example 1

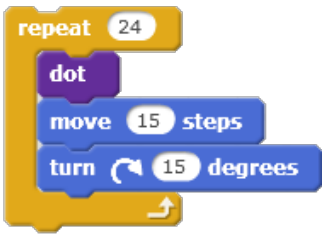


Example 2

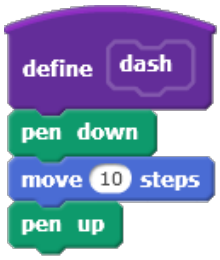
Add (for example) **turn right 15 degrees** inside the script which draws a line of dots. Then work out the correct number for **repeat** to draw a full circle of dots (i.e. 360/**turn** *value*).

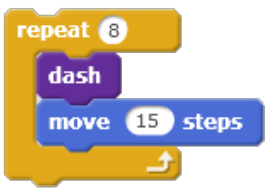


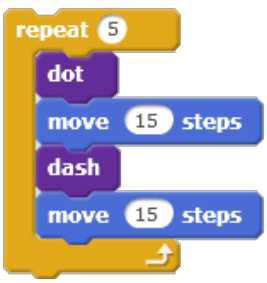




To define a dash the Beetle sprite needs to put its **pen down**, **move** forward a bit then take the **pen up**. As with the **dot** block, the space between the dashes can either be added to the definition or to the script which uses the **dash** block.



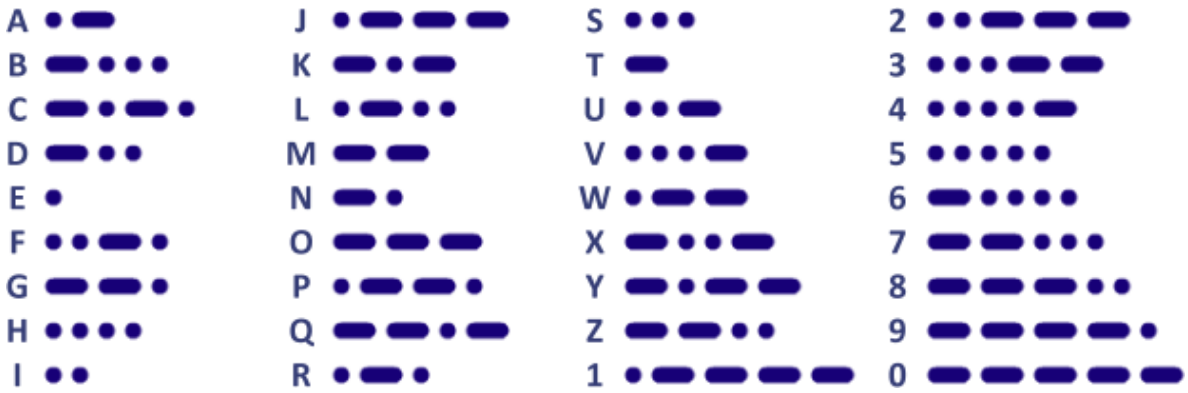




EXTENSION TO ACTIVITY 2.3.1

Dots and dashes can be used to represent Morse code.

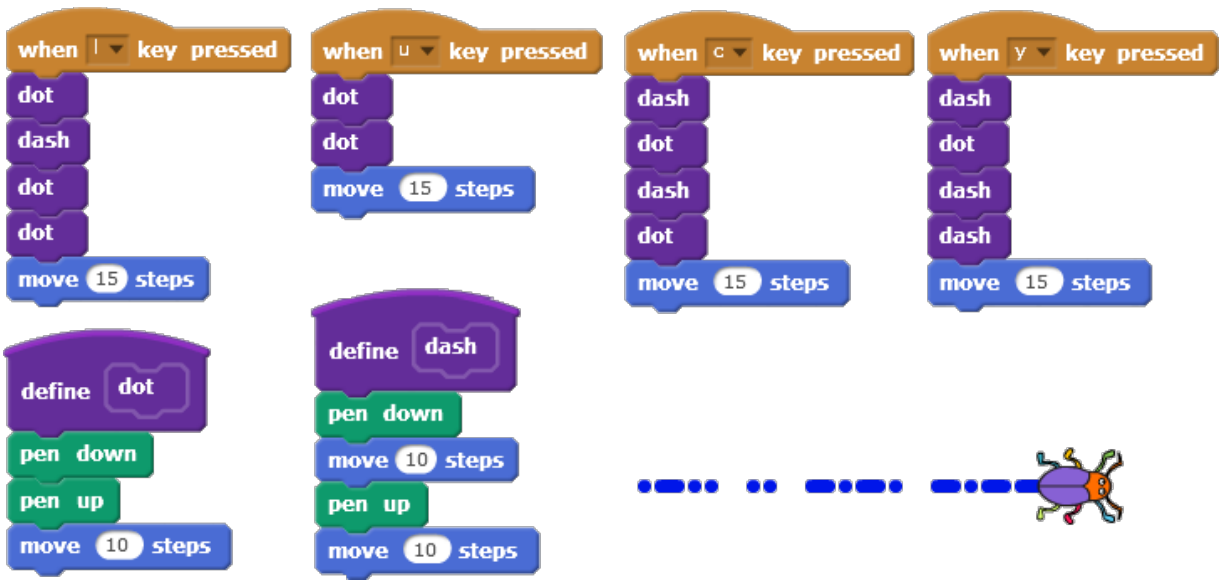
- Pupils modify the *setup script* so that the Beetle starts drawing closer to the left edge of the stage, pointing it direction 90.
- Using the chart below pupils build short separate scripts for the letters of their names. Each script will contain only a combination of the **dot** and **dashes** blocks and will end up with a slightly bigger **move** (for example **move 15** for a space between letters). As an example, the script on the right will draw the Morse code for L.



Challenge pupils to program their keyboard to type a secret word in Morse code…

- Pupils use the **when … key pressed** hat block and create a script for each letter of their chosen word. Use the chart above to guide the sequence of dots and dashes for each letter.
- Pupils type their secret word in Morse code and challenge a friend to decode it.

ADDITIONAL SUPPORT CONTINUED



MODULE 2 • INVESTIGATION 3 • ACTIVITY 2.3.2

Unplugged: Picture Predictions

LEARNING OBJECTIVES

Envisage what the scripts will draw as well as the colour and size of the dots.
bridgE to mathematical reasoning (i.e. does order matter; always, sometimes, never; what's changing and staying the same) and problem-solving (i.e. logical reasoning/visualising) skills.

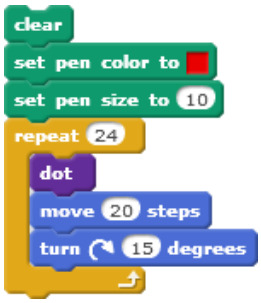
ACTIVITY INSTRUCTIONS

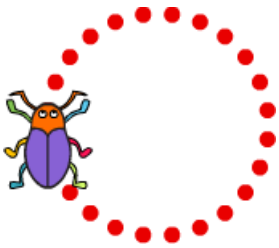
- Print and distribute the unplugged pupil worksheet 2.3.2 or do the activity as a class.
- Ask pupils to draw predicted pictures for each script and/or explain in words.
- Discuss as a class what the differences between the pictures are and why.

Note – for pupils who struggle you may firstly wish to discuss what shape the scripts create.

WORKSHEET SOLUTION

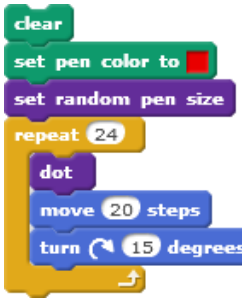
1

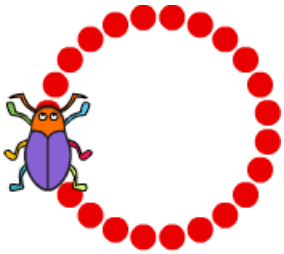




A circle of identical red dots, all of pen size 10

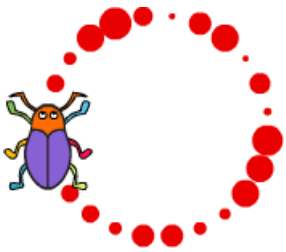
2





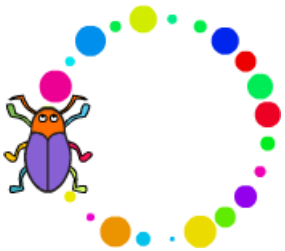
A circle of red dots of random but unique size (note **set random pen size** outside of **repeat**)

3



A circle of randomly sized red dots (note **set random pen size** inside **repeat**)

4



A circle of randomly sized and randomly coloured dots (note **set random pen size** and **set random pen colour** inside **repeat.** Note also that setting pen colour to red could be omitted in this case)

NAME

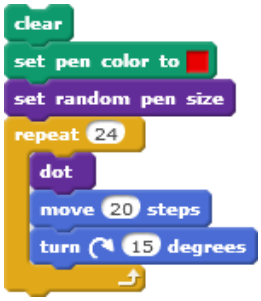
WHAT TO DO

Read each of the scripts. Draw and/or explain in words the picture that each script would create in the box on the right.

DOT PICTURE SCRIPTS

1

2



3

4

Swarming Dots

Learning Objectives

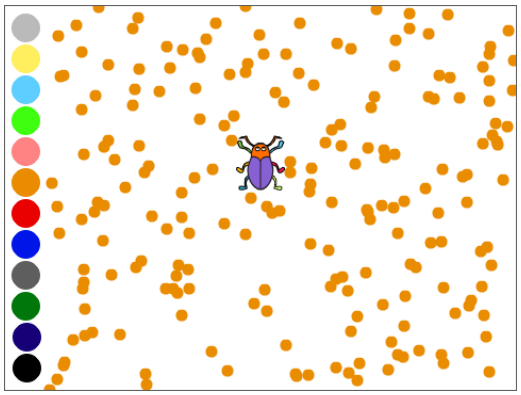
Explore how to instruct the Beetle sprite to jump to random positions on the stage.
Explain how to switch the backdrop of the stage.

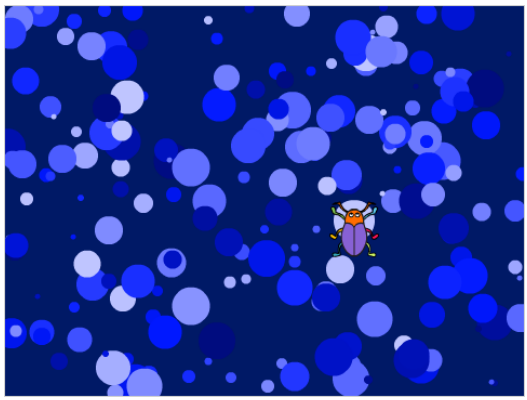
Activity Instructions

- Pupils open project **2-Swarming Dots**, save as a copy and add their name(s) to the title.
- Pupils run the *setup script* – this will initialize the pen colour and pen size of the Beetle.
- In the previous unplugged activity (2.3.2) the Beetle sprite moved around in a circle to draw the dots. Now pupils re-create the **dot** script and build the script for the Beetle so that it draws a circle of dots.

Pupils will change this behaviour so that the Beetle sprite repeatedly **jumps to a random position on the stage** and draws a dot there. (See additional support for ways to demonstrate the difference between the two behaviours.)

- Pupils replace the **move** and **turn** blocks in their script with the **jump to random position** block from the **More Blocks** group and run this script.
- Pupils try switching the backdrop to *night* or *day* by using the **switch backdrop to …** block from the Looks group..
- **[Extension]** Add the blocks **set random pen size** and **set random pen colour** or **set random pen shade** to your script to modify the dots.





Vocabulary

A **backdrop** is a background of the stage. There can be multiple backdrops and the stage can change its look to display any of its backdrops.

Class Discussion Points

- Where did you place the **jump to random position** block within the **repeat** block?
- How can you change the backdrop of the stage?
- What number did you put in the **repeat** block? What happened when you decreased or increased this number?
- What does jumping to a random position mean? Do you know the position the dot will be drawn beforehand?

ADDITIONAL SUPPORT

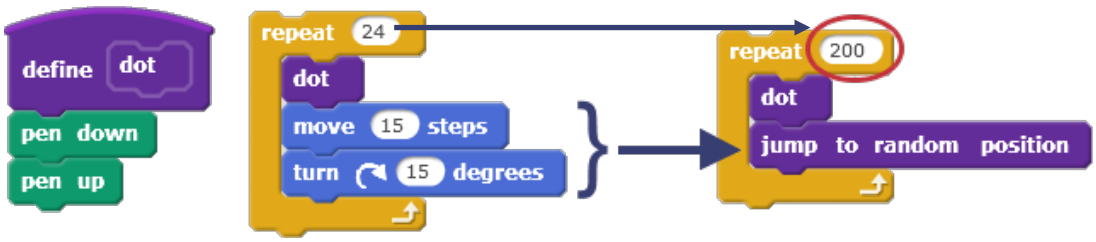
A **physical activity** for showing the difference between a circle of dots and random dots:

- First ask pupils to arrange themselves in a circle. Give a tennis ball to one pupil and tell the pupils to pass the ball to the next pupil in the circle until the ball is reaches the first pupil.

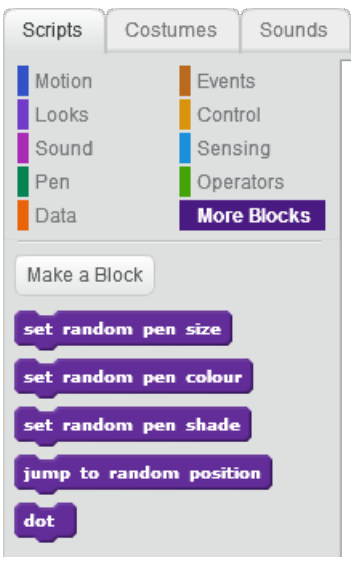
- Now ask the pupils to arrange themselves randomly in a set space. Give the tennis ball to one pupil, set the number of dots to be drawn and tell the pupils they can choose to pass the ball to any pupil (suggest they say the name of the pupil they are passing to before throwing the ball). Count the number of passes until it reaches the set number of dots and then stop the activity.

Ask pupils about the differences between the two situations. Did they all pass/receive the ball the same number of times? Did they know who would receive the ball next? How? Would the same thing happen if they repeated the activity again with the same rules?

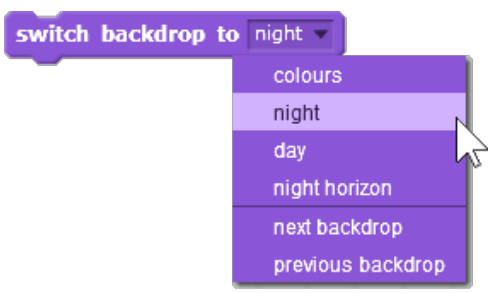
Here are the basic scripts to draw 200 randomly positioned dots.



New blocks in the More Blocks group:

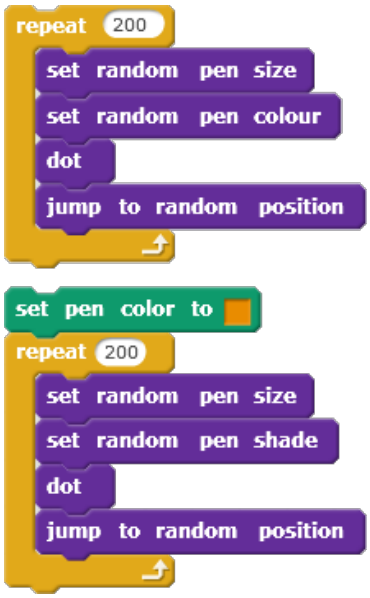


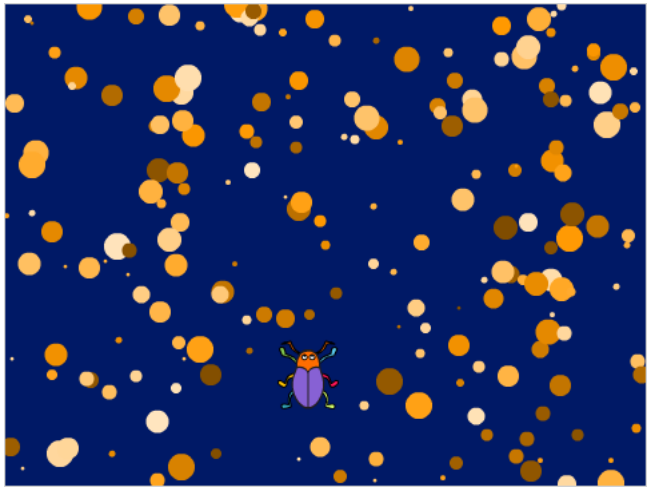
Similar to costumes of a sprite, there could be multiple pictures or **backdrops** of the stage. To change the backdrop, drag the **switch backdrop to …** block from the **Looks** group in the scripts area, click its drop down menu and choose the name of a backdrop you want to switch to. Then run the block by clicking it.



ADDITIONAL SUPPORT CONTINUED

[Extension] Example scripts which incorporate **set random pen size** and **set random pen colour** or **set random pen shade**.





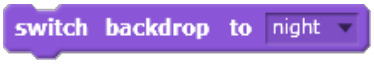
A Sky Full of Stars

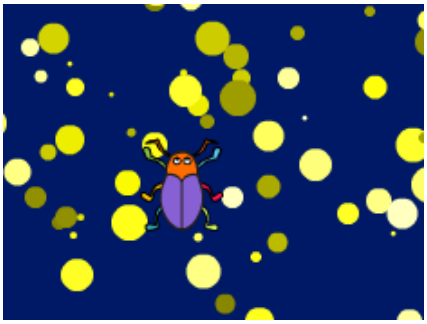
Learning Objectives

Explore how to edit the definition of a block to change the sizes of random dots.
Explain what is meant by minimum and maximum pen size.

Activity Instructions

- Pupils continue in their copy of the **2-Swarming Dots** project.
- Pupils run the *setup script* to initialize the stage and the pen colour and pen size of the Beetle.
- Pupils change the backdrop to the *night* backdrop.



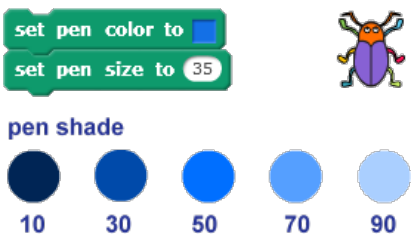


- Pupils duplicate one of their swarming dots scripts with **set random pen shade**, set the initial pen colour to *yellow* and then run the script to create a sky full of stars.
- Pupils try changing the sizes of the stars – they should go to the definition of the **set random pen size** block (on the far right side of the scripts area) and look at how the **minimum and maximum size** is defined.
- Pupils change these values to ensure the stars are an appropriate size for their picture.

- **[Extension]** If pupils find some of the stars are too dark and don't show up very well they can go to the definition for **set random pen shade** and try changing one of the numbers .

Things to Note

- Pen shades with small numbers are dark and shades with numbers close to 100 are bright.

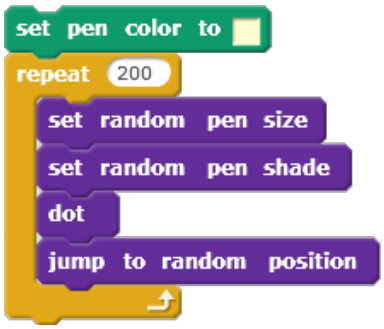


Class Discussion Points

- How did you know there must be a definition of the **set random pen size**? (*because it belongs to the **More Blocks** group*)
- What did you choose your minimum and maximum pen sizes to be? Why?
- What do we mean by a minimum and maximum size?
- If our minimum size is 2 and our maximum size is 7, what would the possible sizes of our pen be?

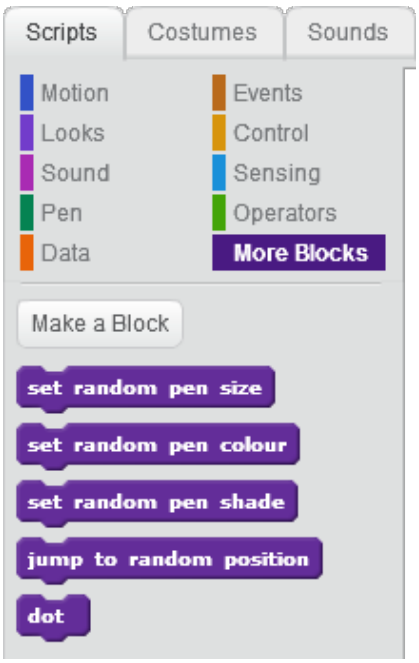
ADDITIONAL SUPPORT

On the right is an example script for creating a sky full of stars. Run the setup script first to clear the stage and set the Beetle's pen up.

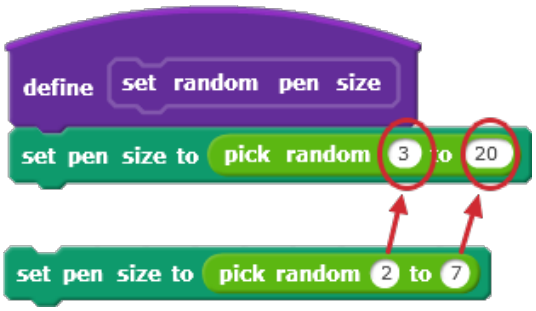


Remember that each block in the More Blocks must have been defined before. To explore what exactly it does and to modify it, look for the definitions.

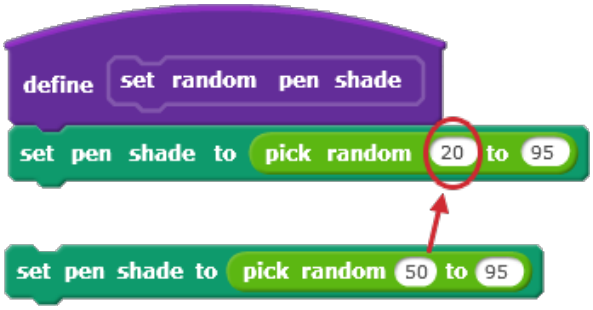
If they are not displayed, just go to the far right side of the scripts area using the scroll bar at the bottom.



You can change the values in the definitions to adjust their behaviour, e.g. the minimum and maximum values of the interval from which random values for the pen size are picked:

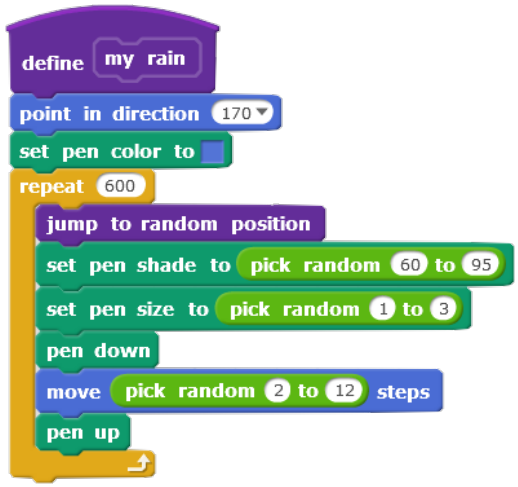


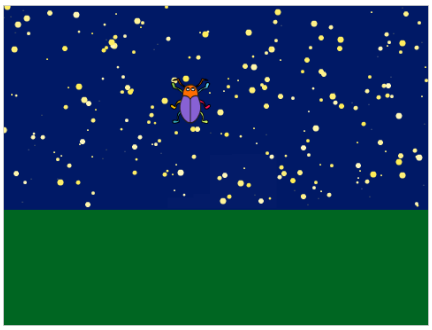
Similarly, you can change the interval for the random shades: lower numbers will result in a darker pen shade, higher numbers in a lighter pen shades.

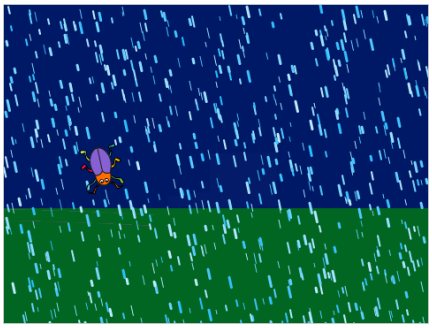


EXTENSION TO ACTIVITY 2.3.4

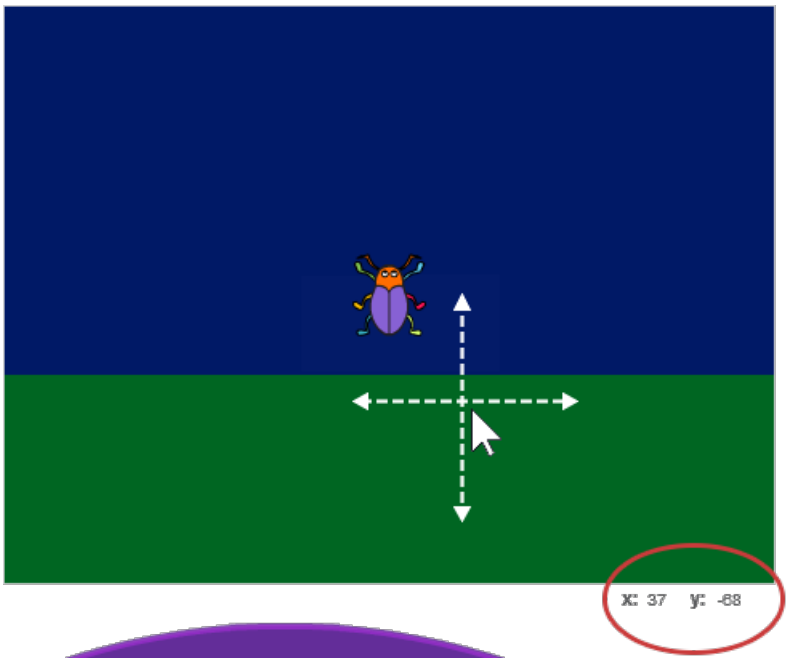
- Pupils change their backdrop to the backdrop called **night horizon**.
- Pupils edit their script so the stars only appear in the sky.
- Instead of stars pupils can now try to make it rain.

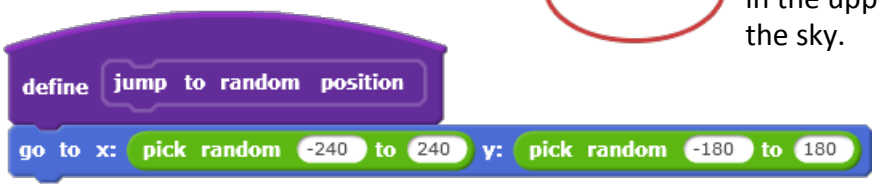






ADDITIONAL SUPPORT CONTINUED

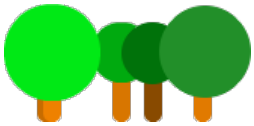




To solve the first challenge – drawing stars only in the sky – requires pupils to explore and modify the **jump to random position** block. Encourage pupils to drag the mouse pointer horizontally and vertically and observe its actual x position and y position (coordinates) below the lower right corner of the stage. Pupils should try to understand what are the intervals for both x and y positions set in the **jump to random position** block and restrict the y position interval to be **pick random from -45 to 180** – so that the random positions for the stars are generated only in the upper (dark blue) part of the sky.

MODULE 2: INVESTIGATION 4
Pen Project: Nature Scenes

KEY LEARNING OBJECTIVE: Build on previous knowledge of exploiting randomness to draw different features from nature scenes – trees, beach houses, flowers, birds etc. Combine multiple drawings within a single nature scene.



This investigation provides further opportunities to build on the use of randomness within drawing as well as drawing regular polygons to create more complex pictures. It also includes an unplugged assessment activity requiring pupils to apply what they have learned throughout this module. The investigation comprises of two core activities and two extension activities.

- **Activity 2.4.1** – Drawing Trees
- **Activity 2.4.2** – Unplugged: Reading Scripts 2
- **Extension Activity 2.4.3** – A Walk in the Woods
- **Extension Activity 2.4.4** – Life's a Beach

Scratch project **2-Pen Project**

LINKS TO PRIMARY NATIONAL CURRICULUM

CURRICULUM OBJECTIVES	LINK WITH SCRATCHMATHS
Computing Design, write and debug simple programs that accomplish specific goals. Solve problems by decomposing them into smaller parts. Use logical reasoning to explain how some simple algorithms work.	▶ Pupils are required to build scripts that create specific features within a nature scene. ▶ Pupils are required to plan how to draw their nature scenes by decomposing them into individual features. ▶ Pupils are required to use logical reasoning to predict the outcomes of scripts that include concepts learned across the whole module.
Mathematics (KS3) Simple probability experiments involving randomness.	▶ Pupils are required to use randomness within their scripts and to experiment with the range from which values can be randomly selected.

Drawing Trees

Learning Objective

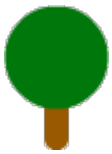
Explore how to create a drawing that includes multiple random elements.

Activity Instructions

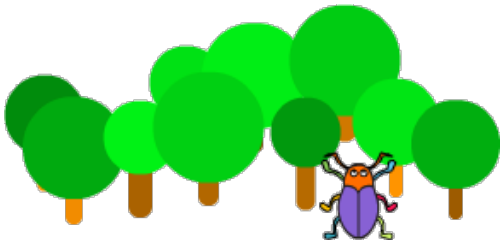
- Pupils open the project **2-Pen Project**, save as a copy and add their name(s) to the title.
- Pupils run the *setup script* and discuss what is the initial direction and the pen state (**pen down** or **pen up**) of the Beetle sprite.

We want to draw a tree as a short line (a trunk of the length 40) and randomly sized dot (as a tree top of random size between 35 and 65).

- Pupils build a script to draw a tree. To set the random pen size to draw the tree top as a dot, use the **set random pen size** block. Check whether its interval of values is **35 to 65**, if not modify the definition.



- Pupils define a new **my tree** block with their script to draw a tree as its definition.
- Pupils build a script to draw many trees randomly positioned around the stage.
- Pupils make their trees more random by adding **set random pen shade** for both the trunk and the tree top. The interval of minimum and maximum values of the **set random pen shade** may need modification.



- **[Extension]** Pupils make their tree with a random trunk size and random trunk length.
- **[Extension]** Pupils switch the backdrop to *night horizon* and build a script to draw many trees only in the lower dark green area.

Things to Note

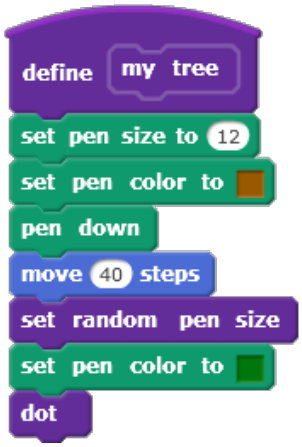
- When using the new **my tree** block in the **repeat** with jumping to random positions, pupils should think about where to put **pen down** (before drawing a tree) and **pen up** (before jumping).
- If the **pick random … to …** block is needed, it can be dragged from the **Operators** group, or duplicated (by right click or Shift + left click) from any script which already uses it.

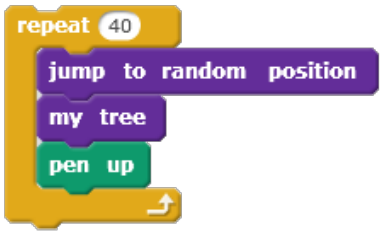
Class Discussion Points

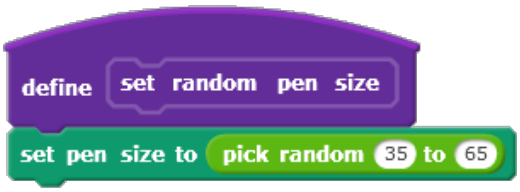
- In the first solution we used constant **move 40** to draw a trunk then added a randomly sized tree top between 35 and 65. When running this script several times the trunk lengths seemed to vary - why? *(partly covered by randomly sized tree tops)*
- Where did you put your **pen down** and **pen up** blocks when drawing many trees?
- Does the Beetle change direction when drawing your tree?

ADDITIONAL SUPPORT

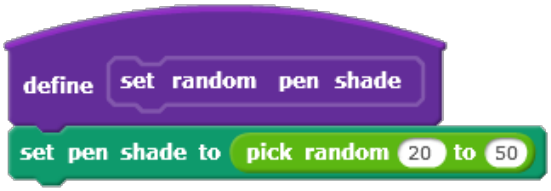
Below is the initial version of the **my tree** block together with the script to draw many trees and the modified **set random pen size**.

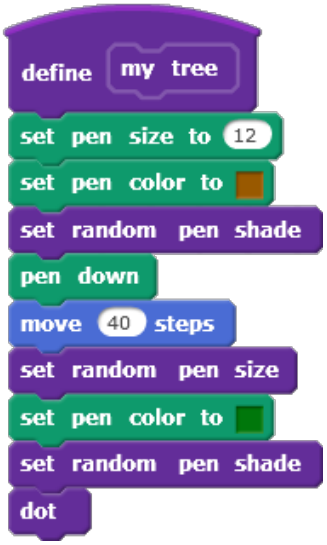






This version includes random pen shades





This version of the **my tree** block requires two different intervals of the random pen size values: one for setting the trunk width (see 1a) and another one for setting the tree top size (see 1b). It is possible to either set the pen to a random size directly in the **my tree** script (as in the script on the left) or to create multiple **set random pen size** blocks of different random intervals.

To make the trunk of a random length between 15 and 50 (see 2) it is necessary to build the **pick random 15 to 50** block, insert it into the **move** block hole (see 3 and 4) and then insert it into the **my tree** definition script (at 2).

Unplugged Assessment: Reading Scripts 2

Learning Objectives

bridgE to knowledge of rotation, angles, regular polygons and multiplication.
Envisage the outcome of different scripts.
Explain why a script would have a particular outcome and how to complete a script to generate a specified outcome.

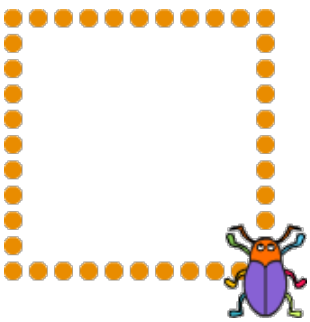
Activity Instructions

- Print and distribute the unplugged pupil worksheet 2.4.2.
- Ask pupils to work individually to check what they have learned during Module 2.
- The answers to the worksheet are below:

1. Down
2. Left
3. Pen size 20 (set pen size rather than change)
4. 8 (360 / 45)
5. 60 (360 / 6)
6. 200 dots
7. 5,6,7,8,9,10 (5 to 10 inclusive)
8. The middle script (dots in the first script would all be the same random size and dots in the third script would be different sizes and colours)
9. 600 ((4 x 50) x 3)

[Extension]

10. A square made of 40 dots



PRINT

INVESTIGATION 4
Activity 2.4.2

NAME

WHAT TO DO

Read the scripts below and answer the question next to it.
The pen tool is always down at the beginning of each task.

ASSESSMENT 2

1. Which **direction** would my Beetle point if I clicked on the block below? (circle the correct picture)

 Up

 Right

 Down

 Left

2. If the Beetle starts pointing in direction 0 (up) and I click the script on the right which **direction** will it end up pointing? (circle correct picture)

 **Starting position**

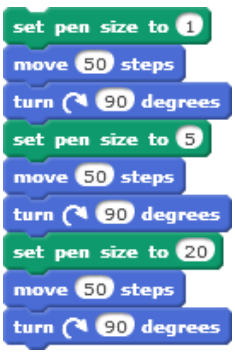
 Up

 Right

 Down

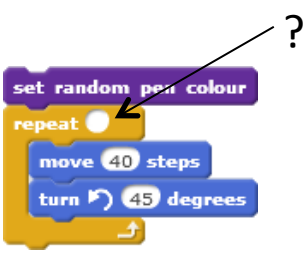
 Left

3. What will the **pen size** of the Beetle be when it completes this script?



Pen size = ☐

4. What is the **lowest number** that I need to put in the **repeat** block to draw a regular polygon?



Repeat number = ☐

5. What **number** do I need to put into the **turn** block to create a hexagon?

Number of degrees =

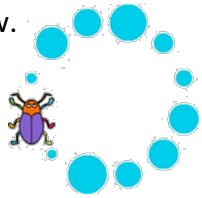
6. The **dot** block draws a dot on the stage each time it is run. If I click on this script **how many dots** will be drawn?

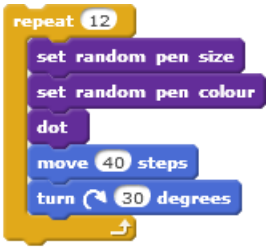
Number of dots =

7. What are the possible **pen sizes** the Beetle could have if I click on this block?

Possible pen sizes =

8. Circle **the script** that produced the drawing below.





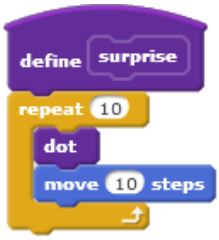
9. I have made a new block called **square**. **How many steps** will the Beetle move in total if I click on the script on the right?

If I click on this script ↓

Total number of steps =

[Extension]

10. I have made another new block called **surprise**. In the box **draw** what the Beetle would draw if I clicked on the script on the right?



If I click on this script ↓

Start

MODULE 2 • INVESTIGATION 4 • ACTIVITY 2.4.3

[Extension] A Walk in the Woods

LEARNING OBJECTIVE

Explore how to use all of the techniques learned during Module 2 to plan and build a forest scene.

ACTIVITY INSTRUCTIONS

- Pupils continue in their project **2-Pen Project**.

We want to combine what we have learned so far in Module 2 to create a forest scene.

- Pupils start by changing the backdrop to the *forest scene*.
- Pupils add some stars to the sky as they did in Activity 2.3.4.
- Pupils define a block called **cabin** and create a script that draws a house in the same way they did in Activity 2.2.4 by combining the **square** and **triangle** blocks in one definition.
- Pupils use their **cabin** block to draw some wood cabins by dragging the Beetle sprite.
- Pupils add in some trees using the **tree** block they defined during the earlier activity.
- Pupils build another script by copying their **tree** definition and changing the **set pen size**, **set pen colour** and **move** blocks to create smaller mushrooms.

- Encourage pupils to save their stage as a picture to keep a copy of their forest creation.

THINGS TO NOTE

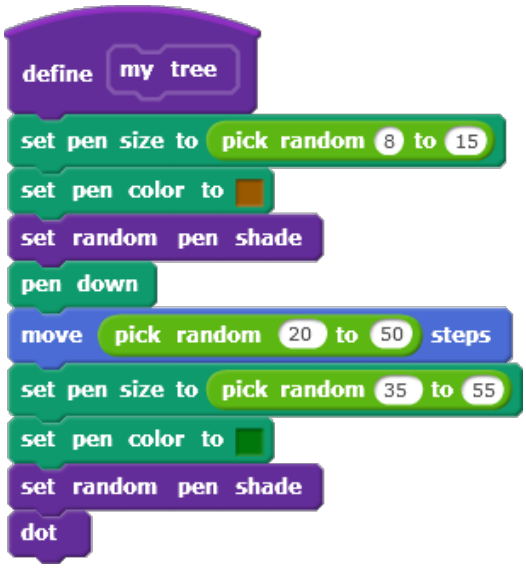
- Currently only the outline of the squares and triangles is drawn – it will not be filled with colour (and there is not a straightforward way to do this in Scratch). It is possible to experiment with large pen sizes or using many lines drawn next to each other to achieve a similar effect.

CLASS DISCUSSION POINTS

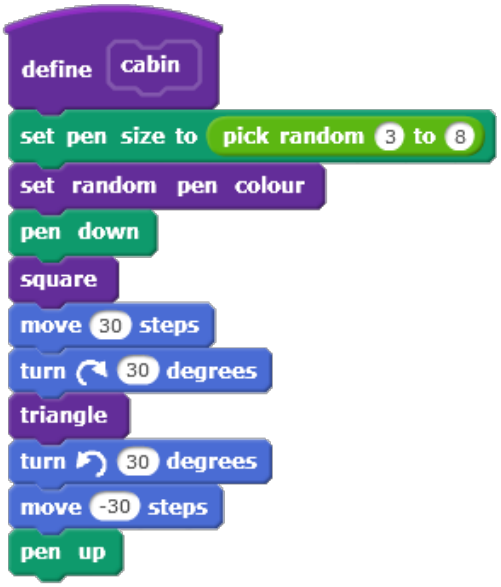
- How did you draw a forest cabin? Is each cabin drawn with random pen colour?
- Did you build a script which will draw stars, then cabins and trees (i.e. the whole scene created by a single click)?
- How did you ensure that all the stars were only drawn in the sky? In which direction did you have to restrict/change the values?

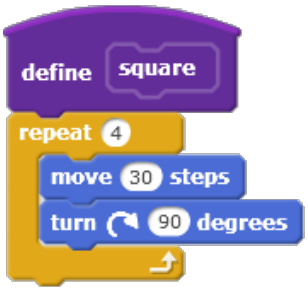
The scripts for creating the forest scene below.

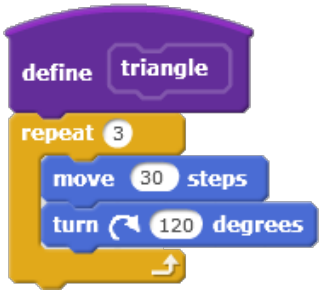
Tree script



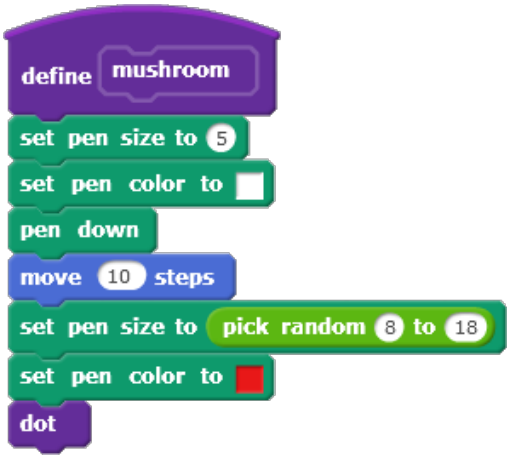
Cabin scripts



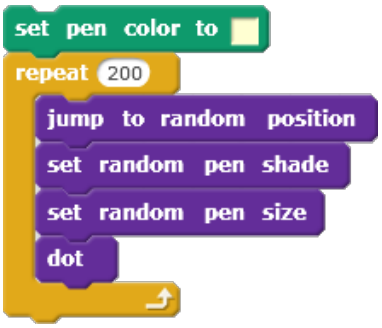




Mushroom script



Star scripts



MODULE 2 • INVESTIGATION 4 • ACTIVITY 2.4.4
[Extension] Life's a Beach

LEARNING OBJECTIVE

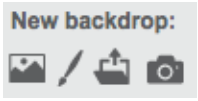
Explore how to use all of the techniques learned during Module 2 to plan and build a beach scene.

ACTIVITY INSTRUCTIONS

- Pupils continue in their project **2-Pen Project**. They start by running the setup script to clear the stage and initialise the Beetle's pen tool.

We will now imagine we are spending the day at the beach, what might you see there?

- Pupils can change the backdrop of their stage by picking a new backdrop from the Scratch library (e.g. *beach Malibu*) or by painting their own new backdrop.



- Pupils try creating some blocks to draw the sun, seagulls or palm trees like in the example on the right – or think of their own drawings to add.

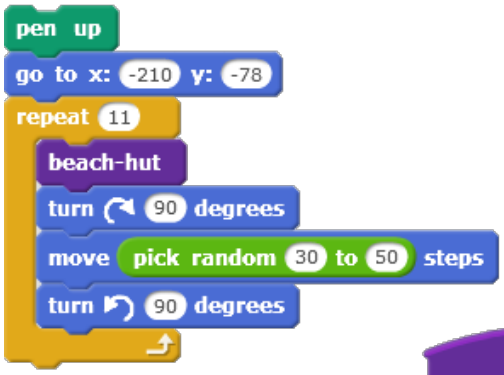
CLASS DISCUSSION POINTS

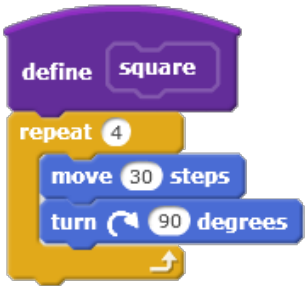
- What drawings did you add to your beach scene?
- Can you describe how you created these?
- Which of the techniques that you learned during Module 2 have you managed to use in your beach scene?

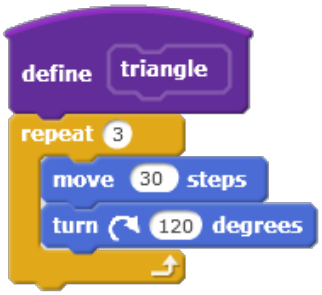
ADDITIONAL SUPPORT

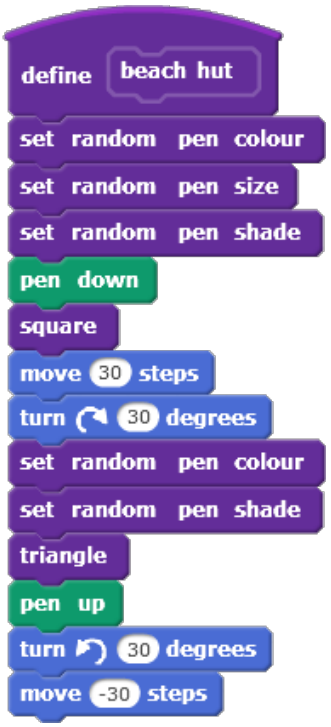
The scripts for creating the beach scene on the previous page.

Row of beach huts scripts

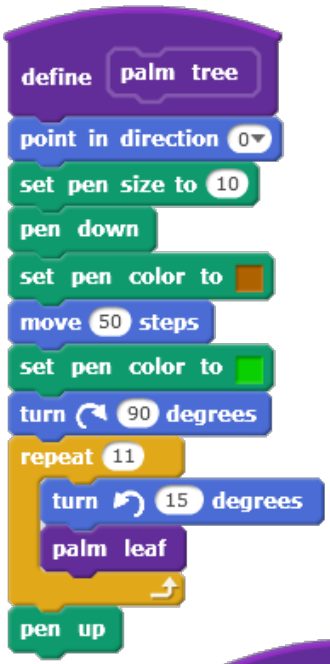


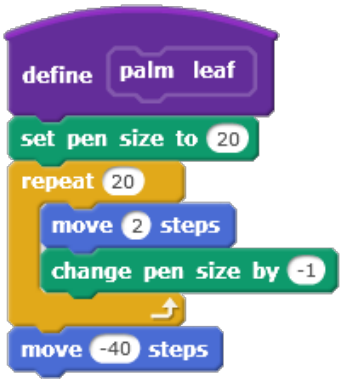




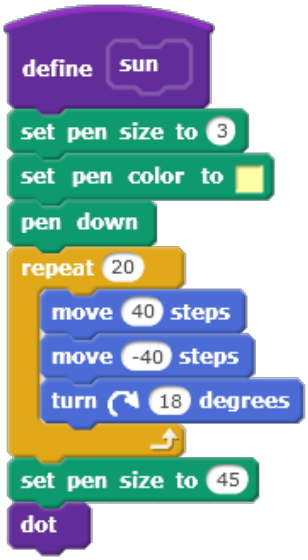


Palm tree scripts

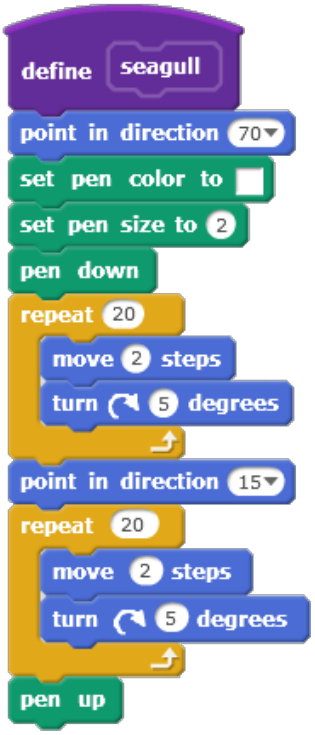




Sun scripts



Seagull script



Module 2 Success Criteria

By the end of Module 2 pupils should be able to:

Blocks

- Use the blocks to set the pen tool
- Use the colour picker to set the set pen colour input
- Explore, use and modify new pre-defined blocks from the More Blocks group
- Modify values in the blocks with compound inputs

Sprite

- Set its pen tool (colour, size, shade, pen up/down)

Stage

- Switch the backdrop by using the **switch backdrop to …** block

Problem Solving

- Predict the outcome of a script when some blocks are moved inside or outside the **repeat**
- Exploit randomness for random positioning and randomly varied polygons and compositions
- Plan an algorithm by decomposing a problem into smaller parts
- Give exact commands and receive/ interpret them, identifying oneself with a sprite
- **[Extension]** Combine several algorithms to produce a nature scene

Scripts

- Understand and modify the definitions of the pre-defined blocks
- Modify the values in the *setup script*
- Understand the importance of how the blocks are ordered and structured in a script
- Build a script to draw basic regular polygons (triangle, square, hexagon…)
- Define new blocks to draw basic regular polygons and use them in scripts to create complex drawings
- Design and debug more complex scripts

Mathematical Understanding

- Calculate the perimeter of rectilinear figures using knowledge of **repeat** and **move** values
- Identify and draw several different regular polygons using mathematical understanding of angles
- Envisage what specific polygon a given script will draw
- Multiply two digit numbers to calculate the total turn within a given script
- **[Extension]** Position a sprite on the stage using the x and y positions