## SUBJECT MEDIUM TERM PLANNING - SUBJECT

**Year Group:** 1

**TERM:** **Spring 2**

**Theme:** **Programming Animations**

**National Curriculum:**

- Understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions
- Create and debug simple programs
- Use logical reasoning to predict the behaviour of simple programs
- Use technology purposefully to create, organise, store, manipulate and retrieve digital content
- Recognise common uses of information technology beyond school
- Use technology safely and respectfully, keeping personal information private; identify where to go for help and support when they have concerns about content or contact on the internet or other online technologies.

**Context:** - This unit introduces pupils to on-screen programming through ScratchJr. Pupils will explore the way a project looks by investigating sprites and backgrounds. They will use programming blocks to use, modify, and create programs. Pupils will also be introduced to the early stages of program design by developing their understanding of algorithms.

**Concepts:**

**Computer Science** – developing pupil's understanding of early programming by applying the concepts of commands, algorithms and program design to animations.

**Digital Literacy** – developing their knowledge of how people may behave online.

**Vocabulary:**

**program -** how the algorithm is run as a code on the computer

**command** – a single, specific instruction

**sprite** – the characters to be programmed in Scratch Jr.

**algorithm** – a precise set of instructions

**block –** the instructions we can use to program the sprites in Scratch Jr.

**predict –** say what will happen in the future

**background –** we can set different scenes for our animations in Scratch Jr.

**programming –** when we move the blocks into the position based on our algorithm design.

| | | |
|---|---|---|
| | | **run –** the term we use when we start our animation. |

### Prior Knowledge:

- Be able to run a command on a device (floor robot) **(Year 1, Spr 1 – Moving a Robot)**
- Be able to follow and give instructions **(Year 1, Spr 1 – Moving a Robot)**
- Be able to debug a program. **(Year 1, Spr 1 – Moving a Robot)**
- Be familiar with the terms command, sequence, algorithm, program **(Year 1, Spr 1 – Moving a Robot)**

### Future Knowledge:

- To create a program using my own design **(Year 2, Aut 2 – Robot Algorithms; Year 2, Sum 2 – Programming Quizzes)**
- To design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts **(KS2)**
- To use sequence, selection, and repetition in programs; work with variables and various forms of input and output **(KS2)**
- To use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs **(KS2)**
- To use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact. **(KS2)**

### End points /by the end of this unit pupils will…

- Be able to use commands to move a sprite.
- Be able to show that a series of commands can be joined together.
- Be able to explain that each sprite has its own set of instructions.
- Be able to delete a sprite.
- Be able to choose appropriate artwork for my project.
- Be able to create an algorithm for each sprite.
- Be able to use my algorithm to create a program.
- Be able to test the programs that I have created.
- Be able to recognise that there are some people online who can make others feel upset.
- Be able to identify an example of how I can speak to a trusted adult if I feel upset / worried about something online.

### Crucial Knowledge:

Pupils will need to be able to create algorithms and use them to create a program for an animation.

Pupils will need to be able to design parts of a project.

Pupils will need to be able to run and test the program that they have created in order to identify whether it matches their design.

Pupils will need to be able to identify when they should ask for support from a trusted adult if something worries or upsets them online.

**Lesson Number - 1**

**Key learning:**
To join a series of commands together.

**Concepts:**

Computer Science

**Success Criteria:**

- I can find the commands to move a sprite
- I can use commands to move a sprite
- I can compare different programming tools
- I can use more than one block by joining them together
- I can use a **Start** block in a program
- I can run my program

**Suggested resources:**

Ipads / Chromebooks

Scratch Jr app

Flipchart

Worksheet of algorithms (available to print from Year 1 Programming Animations – Lesson 2 – A2 Worksheet)

**Lesson structure: Introduction, direct teaching, activities, key questions**

**Engage:**

Begin by reminding pupils of activities in the previous unit.

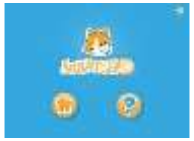
Q – what device did we use last half term in computing? (Bee-bots)

Q – how did you program the Bee-bot? (Pressed buttons on the top of the Bee-bot to give instructions for it to move.)

**Introduce (1):**

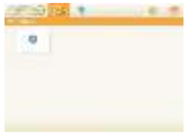
Tell pupils that this half term they are going to be practising more of those skills but with an application called Scratch Jr. (Show a picture of the app symbol).

Tell pupils that when they click to open the app for the first time, it will open on the welcome page.



***NOTE:*** *If ScratchJr has been used on the tablet previously, it may not open on the welcome page, and will start on the home page.*

Tell pupils that if they reach this page, they should click on the **house**.



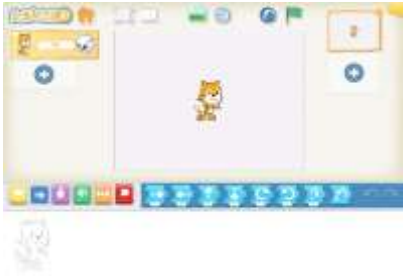
Tell pupils that this is the home page. Inform them that if they ever need to get back to the home page, they can click on the **house** at the top of the page.

Tell pupils that to start a new project they should click on the **blue circle** with the **+** button inside.

**Allow all pupils time to open up the Scratch Jr app and start a new project.**

**P&C**

Show pupils the programming blocks. Explain that the programming blocks are used to tell the computer what you want it to do, just as the buttons are used to give instructions to a Bee-Bot.



Explain that the white space at the bottom of the page is the programming area. This is where they can drag the blocks of code.

Tell pupils that they can choose a blue block, drag it into the programming area, and tap it to make the cat move.

1. Q - 'Can you make the cat move to the right?'
   Allow time for pupils to practise making the cat move on screen.

Tell pupils that they have just programmed the cat. In ScratchJr the cat is called a sprite. Sprites are objects that can be programmed to do different things. There are lots of different sprites that can be used in ScratchJr, and they will look at these in more detail in future lessons.

2. Q - 'Can you make your sprite move to the left, up and down?'
   Allow pupils a short amount of time to have a look at the different movement blocks and try to make the cat move left, move up and move down.
   **Assessment opportunity** Ask pupils how they achieved this. Allow pupils to describe what they did e.g. I dragged the **Move left** block down into the programming area and tapped on it to make the cat go left. Encourage appropriate vocabulary covered so far.

*CHECKPOINT*

*Q - 'What has happened to my sprite?'*
*Discuss that the cat is now on its side as if it is going to sleep.*
*Q - 'Which block do you think was used to do this?'*
*Allow pupils time to think, pair, share with a partner, and feedback to class.*
*If there is time, pupils could be given the opportunity to make their sprite turn to the right or left.*

**Introduce (2)**

Tell pupils that they are now going to learn how to add a background behind the sprite.
Model pressing on the background image at the top of the screen and choosing the classroom background.

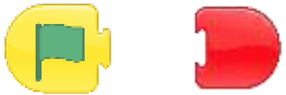
Q – Can you add the classroom background?
Allow pupils time to add the classroom background to their project.

***NOTE*** *– if pupils are confident with using Scratch Jr, you may want to ask them to change their background to one of their own choice.*

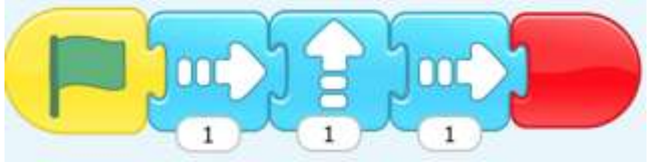
Talk pupils through selecting the different block categories. Tell pupils that there are a number of block categories. When you click on one of the block categories it changes the programming blocks shown. Explain that each of the block categories has a different colour, and that each colour allows the user to program different things. For example, all of the **Move** blocks are blue.

Tell pupils that **Start** blocks in ScratchJr are yellow. These are used to start or 'run' your program.
Tell pupils that **End** blocks in ScratchJr are red. These are used to show what will happen at the end of your program.

Q - 'Can you choose this start and end block and drag them down into the programming area?'



Explain that when they are writing programs in ScratchJr, they can join the blocks together like a jigsaw. Programs should start with a **Start** block and end with an **End** block.
They can read the program from left to right, just like reading the pages of a book.



Q - 'Can you make a program with this **Start** block, this **End** block, and five **Move** blocks of your choice?'
Tell pupils they can delete blocks if they need to. Model draging blocks back onto the blocks palette to be deleted. Tell pupils they can snap off blocks on the right-hand side and drag them back onto the blocks palette to delete them. Alternatively, they can drag joined-up blocks from the left-hand side to delete multiple blocks at once.

Allow pupils time to create their programs.

Tell pupils it is time to run their program. Model how to play back in full-screen mode. Allow pupils time to run their programs and share with their partners.

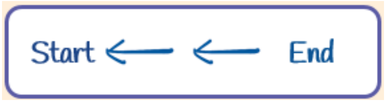
***NOTE:*** *It is important that pupils use full-screen mode to run their programs. This stops them making any changes (such as dragging the sprite) while the code is being run. Only actions that have been programmed will be possible. For example, if you click on the* ***Green flag*** *at the start of the program in the programming area, this will run the program. If you click the* ***Green flag*** *again, the sprite won't return to its original place, but the code will run from where it finished in the previous program. Encouraging pupils to use full-screen mode will eradicate this issue.*

**<u>Independent</u>**
Tell pupils that there are lots of objects in the classroom background in ScratchJr.

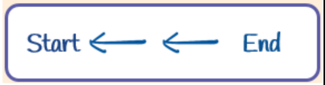
Tell pupils that during this activity they will be using 'algorithms' to help them create programs to direct the cat to different objects in the classroom background.

Q - 'What is an algorithm?'

An algorithm is part of the design of the program. It is a set of precise instructions showing what you want the program to do.



Show pupils an example algorithm for the cat sprite.
Q - 'How many programming blocks would you need to use to make this happen on screen?'
Answer: 4 (Including a **Start** and **End** block)



Tell pupils that you would like them to choose the classroom background, and then create the program shown on the slide to see which object the cat lands on (cake, planet or bus).
Allow pupils time to copy the program using the appropriate **Start**, **End**, and **Move** blocks and run in full screen mode.

Q - 'Where did the sprite stop?' Answer: Bus

Tell pupils they can now delete their blocks and click on the **Reset** button next to the **Flag** ready to start their next program.

Tell pupils they are going to use the worksheet to program the cat to land on different objects in the classroom background by following the given algorithms.

***NOTE:*** *Suggest that pupils mark on the sheet once they have programmed each instruction so they can track their progress.*

Allow pupils time to complete the worksheet. (Year 1 Programming Animations – Lesson 2 – A2 Worksheet)

This could be stuck into discovery books.

**Deepen**

Remind pupils that they have now used both ScratchJr and Bee-Bots.

Q - 'Can you find things that are the same about Bee-Bots and ScratchJr?'

Allow pupils to think, pair, and share their ideas with the class.

***NOTE:*** *It may be helpful to give pupils access to ScratchJr and Bee-Bots at this point, so that they can make comparisons.*

Discuss the answers with the learners. The main three things you want the learners to understand during this discussion are:

- Both Bee-Bots and ScratchJr can be programmed
- They can give instructions to both Bee-Bots and ScratchJr
- Both Bee-Bots and ScratchJr will follow the instructions as they are told to

Bring the pupils back together.

Q – 'What are the differences between ScratchJr and Bee-Bots?'
Allow them to think, pair, and share their answers with the class.

Answers could include differences, or things that appear different, as the learners do not have extensive experience of using the app. These could include:

- ScratchJr works on-screen
- The Bee-Bot moves physically on the floor/table
- You press **Go** to run a program on the Bee-Bot
- The Bee-Bot makes a noise (ScratchJr can make noises, but learners may not be aware of this yet)

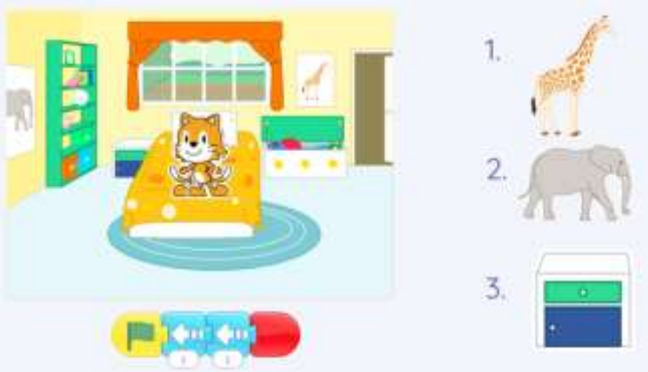
**Reflection**

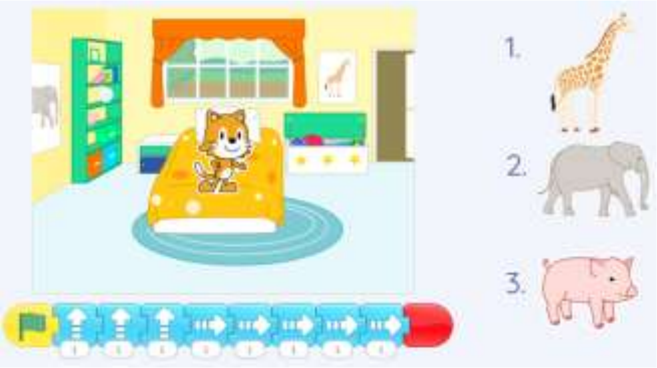
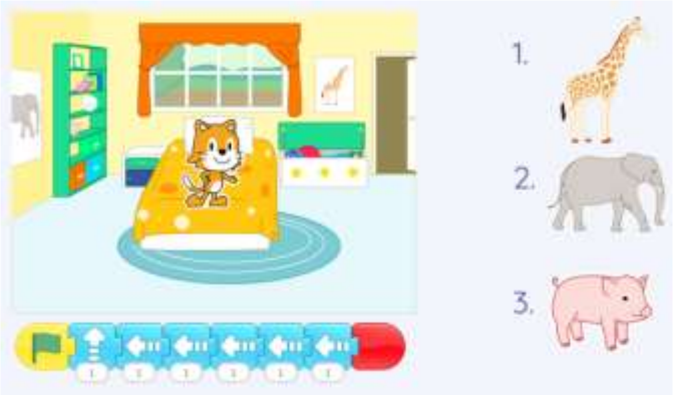
Use thumbs up (3 – confident), thumbs middle (2 – unsure), thumbs down (1 – not confident) to reflect on the three statements.

- I can use more than one block by joining them together
- I can use a **Start** block in a program
- I can run my program

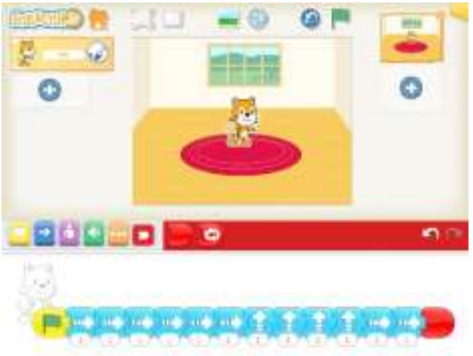
**Vocabulary:**

**command, sprite, compare, program, programming area, block, run, delete, reset, algorithm**

<table>
<tr><th colspan="3">Lesson Number - 2</th></tr>
<tr><td><b>Key learning:</b><br>To identify the effect of changing a value</td><td><b>Concepts:</b><br>Computer Science</td><td rowspan="2"><b>Lesson structure: Introduction, direct teaching, activities, key questions</b><br><b>Engage:</b><br>Q – what does ‘predict mean?’<br>It is like a guess but that they will use some of the knowledge they already have to help them make their decisions.<br>Tell pupils you are going to show them some programs, and they are going to predict where the sprite would land, based on the programs shown.<br>For each picture, ask the pupils ‘Where do you think the sprite would land if it followed this program?’ (e.g. The giraffe, the elephant, the chest of drawers, etc.) Allow pupils to describe their thinking.<br><br> <br>Answer: Chest of drawers.</td></tr>
<tr><td><b>Success Criteria:</b><br>• I can find blocks that have numbers<br>• I can change the value<br>• I can say what happens when I change a value</td><td><b>Suggested resources:</b><br>Ipads / Chromebooks<br>Scratch Jr app<br>Flipchart<br>Hunt the Block Worksheet (Available to print from Year 1 Programming Animations – Lesson 3 – A1 Worksheet)</td></tr>
</table>

| | | |
|---|---|---|
| | | <br> <br>Answer: Giraffe.<br><br> <br>Answer: Pig.<br>**Introduce:**<br>LET'S RECAP<br>Remind pupils that during the last lesson they created programs in ScratchJr by joining blocks together. (Show an example on the flipchart). |

Tell pupils that sometimes programs can become long and can be tricky to work with. Some programs are so long they don't even fit on one page!



Explain that some of the blocks in ScratchJr have numbers underneath, and that these can help us solve the problem of having very long programs.

**P&C**

Tell pupils that before they start to use these blocks to help them improve their programs, they are going to have a hunt to see which blocks have numbers underneath.
Q - 'Can you find the blocks that have numbers underneath?'

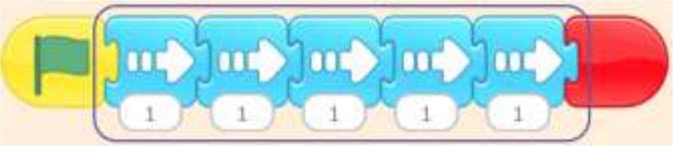
1. Hunt for any blocks with numbers underneath.
2. Draw the picture in the block and colour it in the correct colour?

Explain that they need to be detectives and go into each block category in ScratchJr to hunt for blocks with numbers underneath. They will record the results on their worksheets.
This can be stuck into their discovery books.

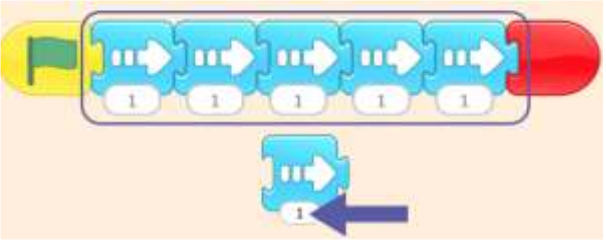
***NOTE*** *- Tell pupils that some of the blocks may not look like the blocks on the worksheet (i.e. a different shape), but may still have numbers underneath. Ask pupils to record what they find.*

**Independent**

Explain to pupils that sometimes when they write programs in ScratchJr they will use the same block lots of times. Tell pupils that this program has a **Start on green flag** block, then five **Move right** blocks, and an **End** block.

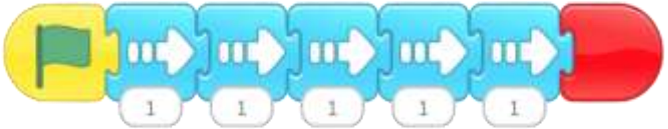


Tell pupils that instead of using all these blocks, they can use just one block and change the number underneath.



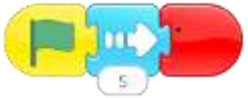
Model on scratch how to change the value underneath a block.

1. Can you write this program with fewer blocks?

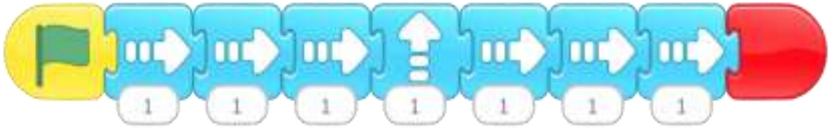
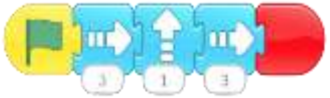
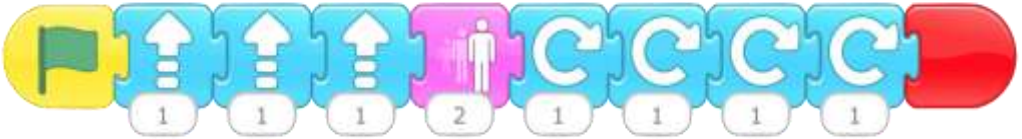
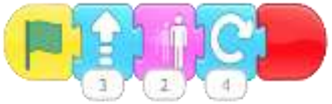


Allow time for pupils to try and create this program on their devices with fewer blocks. **Assessment opportunity** - Pupils to respond with thumbs up/down and discuss their ideas.

ANSWER: -

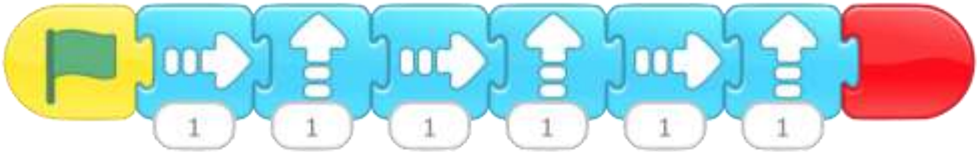


2. Can you write this program with fewer blocks?

| | | |
|---|---|---|
| | | <br> <br>Allow time for pupils to try and create this program on their devices with fewer blocks.<br>**Assessment opportunity** - Pupils to respond with thumbs up/down and discuss their ideas.<br>ANSWER: <br> <br>3. Can you write this program with fewer blocks?<br><br> <br>Allow time for pupils to try and create this program on their devices with fewer blocks.<br>**Assessment opportunity** - Pupils to respond with thumbs up/down and discuss their ideas.<br>ANSWER: <br> |

**Q – What did the pink block do?** (This is the **Grow** block.  The bigger the number, the larger the sprite will grow).

4. Can you write this program with fewer blocks?



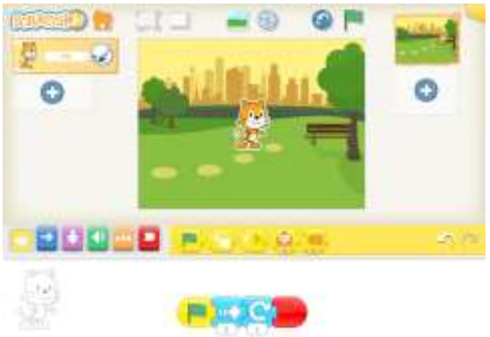
Allow time for pupils to try and create this program on their devices with fewer blocks.
**Assessment opportunity** - Pupils to respond with thumbs up/down and discuss their ideas.

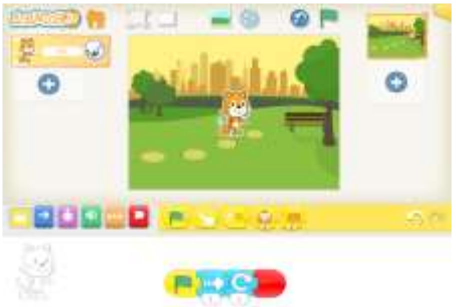
ANSWER: No (There are no two blocks next to each other that are the same so it cannot be made with fewer blocks.

**Deepening**

Q – Can you spot the difference?
Show them 2 programs side by side and ask them to spot the difference.
Tell pupils that you want them to explain what the difference is, and how this affects the program.

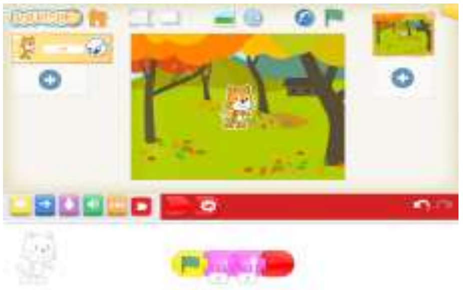
1.

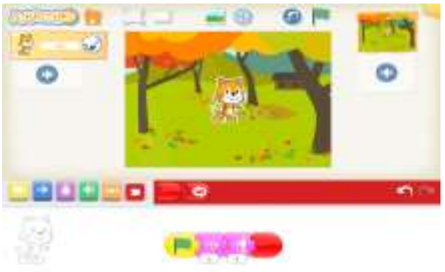




ANSWER - The first program has a ‘5’ in the **Move right** block. This means the cat will move further to the right in this program than in the second program, which has a ‘1’ in the **Move right** block.

2. Q – What do you think these blocks do?
   Remind them that they saw the Grow block earlier, and explain that the other one is a Shrink block. The Shrink block makes the sprite get smaller.





ANSWER - Both programs will make the sprite grow and shrink, but as the numbers are larger in the second program, the sprite will grow and shrink by a greater amount.

| | | |
|---|---|---|
| | | 3. <br><br>ANSWER - The first program has a '12' in the **Turn right** block, whereas the second program only has a '1'. This means that in the first program, the cat will turn to the right further than in the second.<br><br>***NOTE:*** *Turning either left or right by '12' is a full rotation in ScratchJr..*<br><br>**Reflection**<br><br>Use thumbs up (3 – confident), thumbs middle (2 – unsure), thumbs down (1 – not confident) to reflect on the three statements.<br><br>● I can find blocks that have numbers<br>● I can change the value<br>● I can say what happens when I change a value |
| **Vocabulary:**<br>**Predict, effect, change, value, block** | | |

| **Lesson Number - 3** | | |
|---|---|---|
| **Key learning:**<br>To explain that each sprite has its own instructions | **Concepts:**<br>Computer Science | **Lesson structure: Introduction, direct teaching, activities, key questions** |

**Success Criteria:**

- I can show that a project can include more than one sprite
- I can delete a sprite
- I can add blocks to each of my sprites

**Suggested resources:**

Ipads / Chromebooks

Scratch Jr app

Flipchart

**Engage:**

Recap what different blocks do.

Show pictures of the blocks that pupils have been exposed to so far (start block, end block, blue move blocks, pink grow and shrink).

Q – Can you explain what each of these blocks do?

Show picture of the cat.

Q – What do we call the characters we are programming in Scratch Jr? (ANSWER – a sprite)

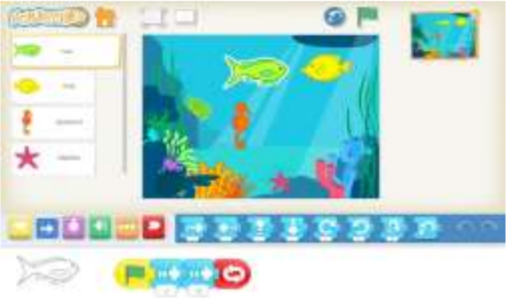
**Introduce:**

Show 3 different backgrounds with the cat sprite in the middle.
Q – What is the problem will all of these pictures? (ANSWER – the cat doesn't belong in any of these settings).

Q – How could you solve these problems?
ANSWER – (Some pupils may say change the background to somewhere the cat would be, or some may say change the sprite. Either answer is correct).

Explain that using different sprites can help to make our projects a little more interesting.

Q – how many sprites can you spot in this project?



Q – How did you know this? (Encourage pupils to look down the left hand side of the screen which lists the 4 sprites instead of the actual picture).

Tell pupils that they can have more than one sprite in their projects, each with its own program.

***NOTE:*** *For this activity, pupils are going to use a sample project on Scractch Jr. Pupils may not be familiar with some of the blocks used as they have not been covered in the lessons at this point. However, this does not matter as the main learning point is that they can move between different sprites and understand that each sprite has its own program. Any blocks they don't understand can be deleted.*

Show pupils how to find the sample projects. Click on the **House** icon to go to the home page, then select the **Question mark** and choose 'Under the sea'. This reduces the time they need to spend finding backgrounds and sprites, so they can focus on programming.

| | | |
|---|---|---|
| | | Tell pupils that each of these sprites is programmed, but they don't do what you want them to do. Talk the learners through the animation showing how to delete the program for each sprite.<br><br>***NOTE:*** *In previous lessons pupils discovered that moving the blocks from the left-hand side of the sequence moves all the connected blocks, and moving blocks from the right-hand side of the sequence deletes block by block.*<br><br>Allow pupils time to find the sample project and delete the programs for each of the sprites.<br><br>*CHECKPOINT*<br>*Q – what is an algorithm? (See definition in previous lessons)*<br><br>**P&C**<br><br>This is where pupils will transition from the 'design' level (provided in the form of an algorithm) to the 'code' level. It is important that pupils are able to undertake the process of using an algorithm and implementing it as code, so that they can learn to create their own algorithms.<br><br>Tell pupils they are going to add some programs to the sprites, based on the algorithms shown on the worksheet (A2 Worksheet – Lesson 4). Talk pupils through the algorithms on the worksheet, pointing out the sprites they apply to. Recap the **Grow** and **Shrink** blocks.<br><br>Model how to add programs to each of the sprites.<br><br>Allow pupils time to add their programs to the correct sprites.<br><br>**Independent**<br><br>Pupils are going to create their own projects, choosing one background and two sprites.<br><br>Explain that to do this, they need to know how to delete the cat sprite. Model how to delete the cat sprite. Either tap and hold on the cat in the picture, or on the label on the left hand side until it starts to wobble.  Click on the red X. |

| | | |
|---|---|---|
| | | Q -  ‘Can you delete the cat sprite?’<br>Ask pupils to open a new project in ScratchJr, and give them time to delete the cat sprite on their own devices.<br><br>Model how to add new backgrounds and sprites.<br><br>Allow pupils time to create their project.<br>Q - ‘Can you add a background?<br>Q - Can you add a program to each sprite?’<br><br>**Deepening**<br><br>Give pupils time to share their programs with a partner.<br><br>Q – Can you explain how you have created your program to each other?<br><br>**Assessment opportunity – teachers to move round and check pupil’s understanding by their explanations.**<br><br>**Reflection**<br><br>Use thumbs up (3 – confident), thumbs middle (2 – unsure), thumbs down (1 – not confident) to reflect on the three statements.<br><br>● I can show that a project can include more than one sprite<br>● I can delete a sprite<br>● I can add blocks to each of my sprites |
| **Vocabulary:**<br>**instructions, sprite, delete, program, algorithm** | | |

## Lesson Number - 4

**Key learning:**
To design the parts of a project

**Success Criteria:**

- I can choose appropriate artwork for my project
- I can decide how each sprite will move
- I can create an algorithm for each sprite

**Concepts:**

Computer Science

**Suggested resources:**

Flipchart

Worksheets (available to print from Year 1 Programming Animations – Lesson 5)

Scissors

Blue tac

**Lesson structure: Introduction, direct teaching, activities, key questions**

**Engage:**

Recap learning from the prior lesson (have an image of a blank project page on the screen):

Q – What button do I press to change the background?
Q – How do I delete a sprite?
Q – What do I press to add a new sprite?
Q – What do I press to change the character of a sprite?

**Introduce:**

Tell pupils that today they will be designing a 'Space race' project, which they will create in ScratchJr next lesson. Reinforce that this lesson is purely the 'design' stage of the project, and that all work on ScratchJr will be done next lesson, using the designs as a guide.

***NOTE:*** *The theme of the project can be altered to match a class topic – however teachers will need to make sure that there is a choice of backgrounds and sprites that would fit with the theme. The idea is pupils will be creating a race between three sprites.*

Tell pupils they will need to pick an appropriate background in ScratchJr for their space race. Give pupils a choice of 2 and allow them time to quickly sketch their choice onto their sheet.

***NOTE:*** *This could be a timed challenge.*

Tell the pupils that this is a quick sketch. It doesn't need to have too much detail. It just needs to have enough detail so they can remember which background to use next lesson. Sprites do not need to be included.

Bring pupils back together, and tell them they will be racing space rockets in their project. Show the pupils the worksheet, and tell them they can colour the rockets and cut them out ready to add to their project in the next activity.

Allow pupils time to design the rockets, colour them in, and cut them out (DO NOT STICK THEM DOWN YET)

***NOTE:*** *Pupils will use the* ***Fill*** *tool in the next lesson to change the colour of the rocket sprite. Therefore, they need to colour the sections of their rockets in solid colours and not patterns, in order to be able to recreate them in ScratchJr.*

**P&C**

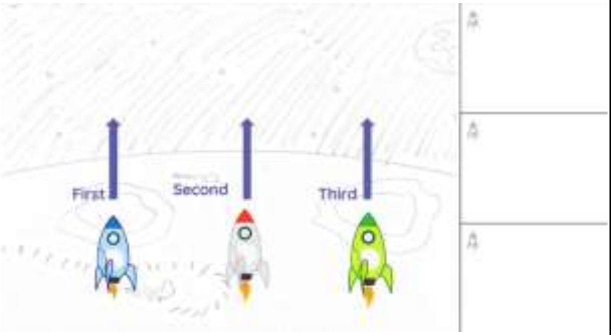
Tell pupils they are going to decide how their race will work.

Q - Will the rockets start on the planet and race up to the sky?

Q - Will the rockets start in the sky and race to be the first to land on the moon?

Q - Will the rockets start on one planet and race to the other?

Tell pupils that you would like them to use blue tac to place the rockets in their starting places so that they know where they will begin at the start of the program. Ask them to draw arrows on the background to show where they will finish. They should also label their rockets 'First', 'Second', and 'Third' as shown in the example, to show the order in which the rockets will finish the race.

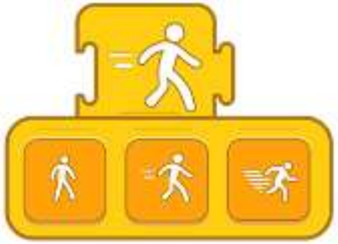


**Independent**

Tell pupils that they need to think about how the sprites will move on the screen.
Q - 'How can you plan and record their movements?' (Answer: By writing an algorithm – A precise set of instructions which can be **turned into a code**)
Remind pupils it is important that they think about what is possible when using the ScratchJr app. There is no point writing an algorithm that cannot be achieved in the ScratchJr app.

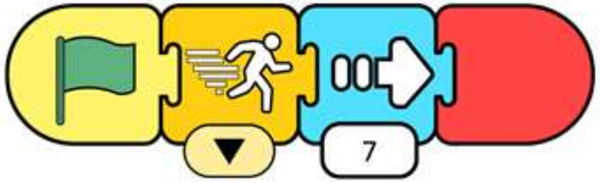
Q – How do we make the blue rocket come first?
We can make our program more exciting by using the 'Speed block'.

The speed block has 3 levels of speed (Slow, medium fast).
This is indicated by the number of lines on the block.



Show how you would put the orange speed block in front of a blue block to speed up the action and press the arrow to dictate the speed level.

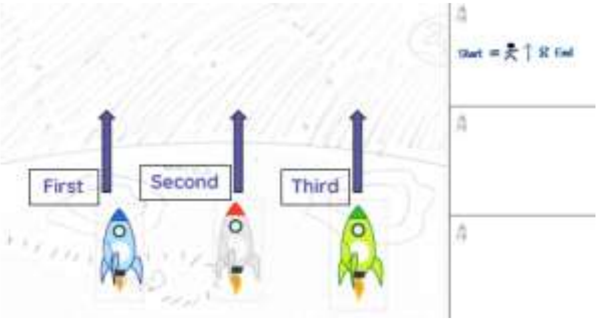
In this code, the character would move fast to the right.



The speed block is going to be very useful when designing a race between 3 sprites – it will allow all 3 sprites to start at the same time, travel in the same direction, arrive at the same point but at different times.

Q – What algorithm would you use to move the blue rocket from the moon up to the sky?
Record their ideas as an algorithm on the board (either in words or symbols)
EXAMPLE - **Start**, **Speed fast**, **Move up 6**, **Stop**

Pupils to create an algorithm for each sprite in the boxes to the right of their background drawing. They can either write words or draw symbols. Pupils will need to indicate which box is for which sprite (e.g. by their colour).



These sheets will need to be named and saved for next week's lesson. Can be stuck into discovery books.

## Deepening

Use slides 20 – 22 on the lesson 5 presentation to play animations to the pupils.
Ask pupils which of the algorithms doesn't match the programs being run. Allow time for the pupils to offer and explain their answers.

Slide 20 Answer: The green rocket. It goes up instead of down.

Slide 21 Answer: The green rocket. It moves at the same speed as the silver rocket, but this is not reflected in the **Speed** block used.

Slide 22 Answer: The silver rocket. Its program turns by '8', whereas the other two both turn by '3'. When the program runs, they all turn by the same amount.

## Reflection

Use thumbs up (3 – confident), thumbs middle (2 – unsure), thumbs down (1 – not confident) to reflect on the three statements.

- I can choose appropriate artwork for my project
- I can decide how each sprite will move
- I can create an algorithm for each sprite

**Vocabulary:**
**sprite, background, appropriate, algorithm**

| Lesson Number - 5 | | |
|---|---|---|
| **Key learning:**<br>To use my algorithm to create a program | **Concepts:**<br>Computer Science | **Lesson structure: Introduction, direct teaching, activities, key questions**<br>**Engage:**<br>Recap their designs from last week.<br>Give pupils time to look over the designs and check that they are happy with them. If they need to make any amendments / alterations they can do this now.<br>**Introduce:**<br>Remind pupils that last lesson they chose 3 different colour for their sprites.<br>Show pupils how to block-fill parts of their sprites to match their designs.<br>Tell pupils that they shouldn't worry if they make a mistake, as they can use the 'undo' and 'redo' buttons to help.<br>Q - 'Can you choose your background and design your three rockets?'<br>Allow pupils time to add and design their artwork. Remind pupils to follow the plans they completed during the last lesson.<br>**P&C**<br>Remind pupils that during the last lesson they wrote some algorithms to help them to program their rockets to move on-screen. Show some examples to the class.<br>Tell pupils they now need to add the programs to the correct rockets. Explain that they just need to add the programming blocks to the correct rockets using their design to help them. They will run and test the programs in the next activity.<br>Allow pupils time to create their programs. |
| **Success Criteria:**<br>● I can use sprites that match my design<br>● I can add programming blocks based on my algorithm<br>● I can test the programs I have created | **Suggested resources:**<br>Ipads / Chromebooks<br>Scratch Jr app<br>Flipchart<br>Work from the previous lesson | |

**Independent**

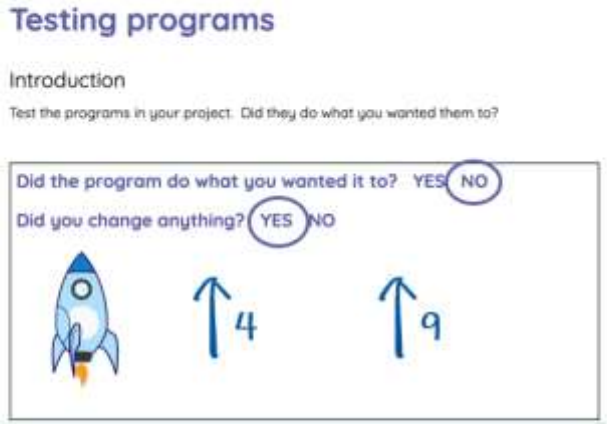
Tell pupils that they are now going to test their programs to see whether they did what they planned when they were creating their algorithms.

Watch the video (on slide 14 lesson 6) and ask pupils
Q - 'Does this video match the plan?'.
Allow pupils to feedback to class.
Tell pupils that although it matches some parts of the plan, the rockets do not reach the top of the screen as shown by the arrows on the plan on the background.

Tell pupils that they are going to look at their own programs and complete a worksheet to show whether their programs worked as expected, or whether they need to be changed.

Show the learners the worksheet, and an example of a completed worksheet.
Explain that they should indicate which rocket is which using coloured pencils.

These worksheets can be stuck in their discovery books.



Testing programs

Introduction

Test the programs in your project. Did they do what you wanted them to?

Did the program do what you wanted it to? YES [✓ NO]

Did you change anything? [✓ YES] NO

**Deepening**

Tell pupils that they can share their finished projects with another learner or with their teacher.
Show questions on the board for the pupils to discuss.
Q - What did you find easy?
Q - What did you find hard?
Q - What was your favourite part?

|  |  |  |
|---|---|---|
|  |  | Q - Did you enjoy making your space race project?<br>Q - What would you do differently next time?<br>**Assessment opportunity – teachers to engage in discussions with the pupils**<br>**<u>Reflection</u>**<br>Use thumbs up (3 – confident), thumbs middle (2 – unsure), thumbs down (1 – not confident) to reflect on the three statements.<br>● I can use sprites that match my design<br>● I can add code blocks based on my algorithm<br>● I can test the code I have created<br>At the end of the lesson (or at another time before next week) complete the Project Evolve Pre-assessment Knowledge Map for Year 1 – Self Image and Identity. |
| **Vocabulary:**<br>**sprite, design, programming blocks, algorithm, programs** |  |  |

| **Lesson Number - 6** |  |  |
|---|---|---|
| **Key learning:**<br>I can recognise that there may be people online who could make someone feel sad, embarrassed or upset. | **Concepts:**<br>Digital Literacy | **<u>Engage</u>**<br>Q – what games do you like to play online?<br>Discuss with pupils they types of games they like to play. |

**Success Criteria:**

- I can recognise that there may be people online who could make me feel sad, embarrassed or upset.
- I know when I should ask an adult for help with things online that upset me.
- I can give examples of different adults I can ask for help.

**Suggested resources:**

Project Evolve knowledge map

Flipchart

Project Evolve resources for "Year 1 – Self Image and Identity"

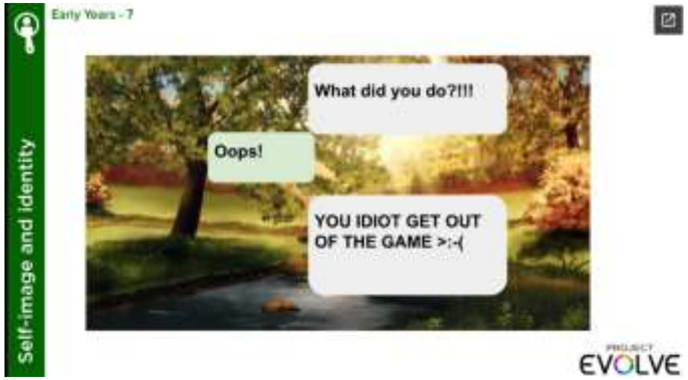
***NOTE –** some children may mention games that are not age appropriate for them. This may need to be raised with parents / added to CPOMS.*

**Introduce**

Introduce the scenario of a little boy called Alex who was playing an online game with his friend.

Alex makes a mistake in the game.

Show an image of a chat room with his friend in the game.



Early Years - 7

Self-image and identity

What did you do?!!!

Oops!

YOU IDIOT GET OUT OF THE GAME >:-(

PROJECT EVOLVE

**Practise & Consider**

Q – What could Alex's friend Lily say to him?

1. "Try to forget about it"
2. "Don't play that game"
3. "Tell an adult"

| | | |
|---|---|---|
| | | Discuss each response in turn. Discuss how Alex might feel after each response. How would they feel if it was them?<br>Q – Which is the best piece of advice?<br>**Independent**<br>Q – Who could you tell if someone has upset you in a game?<br>Q – How could they help you?<br>Pupils to identify 5 trusted adults that they could go to if someone has upset them in a game. Pupils could draw around their own hand print (or be given a print out of one) and write / draw on each finger a trusted adult.<br>To go into their discovery books.<br>**Deepening**<br>Pupils to write a sentence explaining how a trusted adult could help you.<br>**Reflection**<br>Complete knowledge map "assess impact" on the lesson outcome that has been taught to identify impact of learning. |
| **Vocabulary:**<br>**online, upset, advice, trusted adult** | | |