

National Curriculum objectives

- Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts;
- Use sequence, selection, and repetition in programs; work with variables and various forms of input and output;
- Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs;
- Select, use and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including collecting, analysing, evaluating and presenting data and information.

To begin this unit, the children should have already learnt:Year 1 – Moving a Robot (Spring 1)

Floor robots have buttons which help us to direct them. We can use algorithms (a set of guidelines to perform a task) to program floor robots along routes.

Year 1 – Introduction to Animation (Summer 2)

Programming is when we make a set of instructions for computers to follow. *ScratchJr* is a program that we can use in order to code our own stories and animations.

Year 2

We can create simple quizzes in *ScratchJr* where the user can select an answer by clicking on a sprite. An outcome occurs when the sprite is clicked.

Year 3

We can use event and action command blocks in order to make sprites carry out actions when certain prompts take place. Algorithms (a set of instructions to perform a task) allow us to sequence movements, actions and sounds in order to program effective animations.

Year 4

Count-controlled and infinite loops can be used to create different examples of repetition in games: using repeat and loop operator blocks in *ScratchJr* can make our programs more logical and efficient.

The learning in this unit will prepare the children to learn these things in the future:Year 6

Micro:bits are small computers that perform different actions based on programs written on computer software. Programs are then downloaded to the micro:bit. Micro:bits have a range of input sensors that can be used as input triggers for different codes to run. Output devices on Micro:bits (e.g. LED displays) can be programmed to display words, pictures and numbers.

<p>Key Enquiry Question</p> <p>How are conditions used in selection? Why are infinite loops vital to programming? When might you use the 'if... then... else...' structure in programming? Why are 'yes/no' questions important to outcomes when programming? Can you explain the task as concisely as possible? Why is sharing your finished program with other programmers helpful? What bugs have you identified and how did you improve them?</p>	<p>The Big Idea:</p> <p>'Conditions' can be used in programming: the 'if... then... else...' structure can be used to select different outcomes depending on whether a condition is 'true' or 'false'. Issues with programs can arise when answers similar to those in the condition are given as inputs. We must predict such issues and identify ways to avoid such problems.</p>
<p>To achieve ARE, pupils will need to be secure in the following knowledge:</p>	
<p>By the end of this unit, children will know:</p> <ul style="list-style-type: none"> • A condition can only be true or false; • A count-controlled loop contains a condition; • Similarities and differences between a count-controlled and condition-controlled loop; • A condition-controlled loop will stop when a condition is met; • When a condition is met, a loop will complete a cycle before it stops; • Selection can be used to branch the flow of a program; • A loop can be used to repeatedly check whether a condition has been met; • The importance of instruction order in 'if... then... else...' statements. 	<p>Vocabulary:</p> <p>Programming; <i>Scratch Jr.</i>; command; algorithm; sprite; home; block; stage; background; app (introduced in Y1).</p> <p>Sequence; quiz; debugging (introduced in Y2).</p> <p>Code; events; motion; trialling (introduced in Y3).</p> <p>Logical; condition; selection.</p>
<p>By the end of this unit, children will be able to do:</p> <ul style="list-style-type: none"> • Choose a condition to use in a program; • Create a condition-controlled loop; • Use a condition in an 'if... then...' statement to start an action; • Use selection to switch program flow; • Use 'if... then... else...' to switch program flow in one of two ways. 	<p>Useful Resources:</p> <p>Online training courses</p> <p>Raspberry Pi online training courses</p> <p><i>ScratchJr</i> for iPads and/or computers.</p>



COMPUTING: PROGRAMMING

KNOWLEDGE ORGANISER

Y5



Overview

Quizzes in Scratch



- Programming is when we make a set of instructions for computers to follow.
- Scratch is a program that we can use in order to code our own quizzes, stories, animations and games. We can input questions using the 'ask' command blocks. We can use selections and conditions in order to ensure that there are different outcomes depending upon a user's response.
- We use algorithms (a set of instructions to perform a task) to sequence movements, actions and sounds in order to program effective animations.

Selections and Conditions

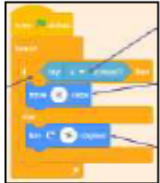
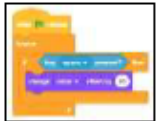
- **Creating Conditions:** The 'If-then' command block helps us to create conditions. It is one of the darker orange control blocks. Other blocks are placed inside the 'If-then' blocks to create conditions.

The 'senses' blocks (light blue) create the 'trigger' (e.g. when a certain key is pressed). We can change the trigger by pressing the downward arrow and selecting from the range of keys/ actions. The 'actions' blocks (e.g. motions, sounds, etc.) are then used to program what will happen when the 'senses' command is triggered.

- **Different Outcomes:** The 'If-then-else' command block helps us to write programs that have selections with two outcomes.

- Actions to be carried out if the condition is 'true' (if the conditions of the 'sense' command are met) are placed below 'then.' Actions to be carried out if the condition is 'false' (e.g. if any other key is pressed) go below 'else.'

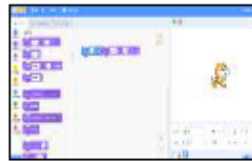
- The 'forever' block means that the command will happen continually.



The Basics of Scratch

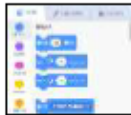
- **What is Scratch?** Scratch is a website/ app that lets us code our own quizzes, stories, games and animations.

- Scratch helps us to learn how to use programming language, whilst also being creative and using problem-solving skills.



There are three main areas in Scratch:

- The Blocks Palette (on the left) contain all of the different blocks: puzzle piece commands which control the animation.
- Code Area (in the middle) is where the blocks are placed to create a program.
- Stage with Sprite (right) is where the output of the program is presented. The sprite is the character.



Attributes: There are three attributes of the sprite which we can change to make our animation: Code, Costumes, Sounds.

-Event Blocks:

Event blocks are coloured yellow and are used to sense different events that happen e.g., the green flag being clicked.



-Action Blocks: Action blocks include 'Motion' blocks, 'Sound' blocks and 'Looks' blocks. They make the sprite move, make sounds and change appearance.



Asking Questions

- Questions can be included by using the 'ask' command blocks.

- If specific answers are needed (e.g. yes or no), these can be typed in when using the 'answer' sensing block within the = 'Operators' block - drag it into the first white space. In the second white space, we can then type in the desired answer.



- The 'say' command block (in looks) is used to inform the user if the response was correct.

Algorithms, Trialling, Debugging

- Designing an **algorithm** (set of instructions for performing a task) will help you to program the sequence that you require.
- Programmers do not put their computer programs straight to work. They **trial** them first to find any errors:
- Sequence errors: An instruction in the sequence is wrong or in the wrong place.
- Keying errors: Typing in the wrong code.
- Logical errors: Mistakes in plan/thinking.
- If your algorithm does not work correctly the first time, remember to **debug** it.



Important Vocabulary

Programming

Scratch

Logical

Commands

Algorithm

Condition

Selection

Sequence

Trialling

Debugging