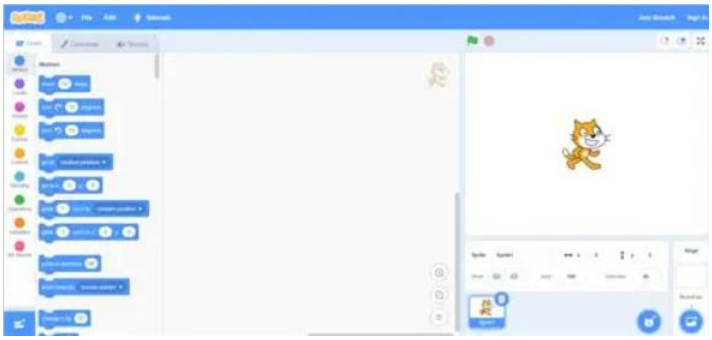
Jeavons Wood
Primary School

Knowledge Organiser for Year 4

Big question: How can we use loops to make programs more efficient and creative?

KS2 National curriculum specification
• 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



In this unit, the children will:
Develop the use of count-controlled loops in a different programming environment.
Explain that in programming there are infinite loops and count-controlled loops.
Develop a design that includes two or more loops which run at the same time.
Modify an infinite loop in a given program.
Design a project that includes repetition.
Create a project that includes repetition.

Key vocabulary:	
Programming	When we make a set of instructions for computers to follow.
Scratch	Scratch is a website/ app that lets us code our own stories, games and animations.
Blocks	We use repeat and loop operator blocks in order to make out programs more logical and efficient. These help to run code continuously or a set number of times.
Commands	Tell a computer or robot what to do.
Code	A set of commands written in a programming language that a computer can understand.
Events	Something that happens and makes a program run a certain part of the code.
Motion	Movement in a program or by a robot.
Sequence	The order in which commands happen in a program.
Trialling	Testing something to see if it works as planned.

Debugging	Finding and fixing mistakes in your code so it works properly.

The basics of scratch

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

-The Blocks Palette contain all of the different blocks: puzzle piece commands which control the animation.

-Code Area is where the blocks are placed to create a program.

-Stage with Sprite 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:

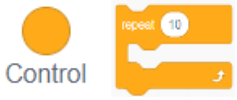
Event blocks are coloured yellow and are used to sense different events that happen e.g., the green flag being clicked.

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

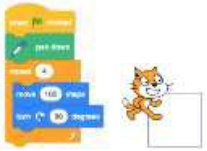
Loops and Repetition

-Pen Drawing in Scratch: Select the 'add extension' icon in the bottom left corner. Then select 'pen.' This allows you to draw with your sprites.

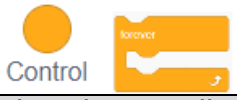
-The Repeat Block: Select 'code' and then the 'control' blocks (orange). Here you will find the repeat block. It should be placed around the command blocks that you want to repeat. The number of times something is repeated can be typed into the white area.



-Creating Shapes: Selecting 'pen down' (in the 'operators' blocks) can be followed by use of the motion blocks to determine the line that will be drawn (e.g. 'move 10 steps'). Turning a number of degrees changes the direction of the pen. Placing the repeat block around this motion code can allow more complex shapes to be drawn.



-Count-Controlled/Infinite Loops: We can control the number of 'loops' of a command with the number typed into the 'repeat' block. The 'forever' block makes a command continue infinitely (forever).



Events Managing and Efficiency

-We should ensure that programs are coded and labelled in easy-to-understand, user-friendly ways.

-Using the 'events' blocks logically can help to make your programming easy to use. E.g. when 's' key pressed a square is drawn, when 'h' key is pressed a hexagon is drawn.

-Efficiency is about getting the right result in the easiest way possible, wasting little time or effort. Our use of the repeat and loop tools should help to create efficient programs.

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.

Teacher Subject Knowledge:	
Prior Knowledge & progression	This unit assumes that learners will have some prior experience of programming. The KS1 NCCE units cover floor robots and ScratchJr, and Scratch is introduced in the Year 3 programming units. However, experience of other languages or environments may also be useful.
Teacher knowledge	This unit focuses on developing learners' understanding of repetition within the Scratch programming environment. Repetition is where actions or commands in programming are repeated. The repeating commands can also be referred to as a 'loop'. Loops can be repeated indefinitely (known as 'infinite loops'), or for a set number of times (known as 'count-controlled loops'). This unit also develops learners' understanding of design in programming, using the approach outlined below. When programming, there are four levels which can help describe a project (known as 'Levels of abstraction'). Research suggests that this structure can support learners in understanding how to create a program and how it works: • Task - what is needed • Design - what it should do • Code - how it is done • Running the code - what it does Spending time at the 'task' and 'design' levels before engaging in code-writing can aid learners in assessing the 'do-ability' of their programs. It also reduces a learner's cognitive load during programming. Learners will move between the different levels throughout the unit, and this is highlighted within each lesson plan.
Resources	It is recommended that learners use desktop or laptop computers to access Scratch (scratch.mit.edu). We recommend the use of teacher accounts in Scratch to make it easier to manage student accounts. For guidance on setting up teacher accounts, please visit the Scratch website. (https://scratch.mit.edu/educators/faq) Throughout this unit, there are opportunities to model within Scratch or to demonstrate a concept through a video. Pedagogically, it is more beneficial to model the concepts to the learners, which allows for easier questioning and understanding. We recommend that you use the videos to see what needs to be modelled, but give a live demonstration within the lesson. However, the videos are provided on the slides if you wish to use them instead.