Year Group:	4	Strand: 2	Programming
	How do I use repetition and loops to create games?		
Key NC Objectives	• Design, write and debug programs that accomplish specific goals; solve problems by decomposing them into smaller parts • Use sequence, selection, and repetition in programs • Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs • Use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour		
Unit Objectives	• Develop the use of count-controlled loops in a different programming environment. • Predict the outcome of a snippet of code and modify a snippet of code to create a given outcome. • Explain that in programming there are infinite loops (forever) and count controlled (repeat) loops. • Choose when to use a count-controlled and an infinite loop. • Recognise that some programming languages enable more than one process to be run at once. • Develop a design which includes two or more loops which run at the same time. • Modify an infinite loop in a given program by identifying which parts of a loop can be changed and explaining these. • Design and create a project that includes repetition (independently). • Refine and debug the algorithm in design as program is built.		
Suggested learning activities	• Pupils look at real life examples of repetition, and identify which parts of instructions are repeated. Pupils write examples of algorithms for these repetitions (unplugged). Pupils use Scratch to create shapes using count-controlled loops. They consider what the different values in the loop signify, use existing code to modify and create new code, and work on reading code and predicting what the output will be once the code is run. See Teach Computing materials: https://teachcomputing.org/resources Year 4 Unit –Programming B Repetition in games, Lesson 1 • Pupils look at different types of loops - both infinite loops and count-controlled loops. They practise using these within Scratch and think about which might be more suitable for different purposes. See Teach Computing materials: https://teachcomputing.org/resources Year 4 Unit –Programming B Repetition in games, Lesson 2 and 3 • Pupils look at an existing game and match parts of the game with the design. They make changes to a sprite in the existing game to match the design. They then look at a completed design, and implement the remaining changes in the Scratch game. They add a sprite, and reuse and modify code blocks within loops, and explain the changes made. See Teach Computing materials: https://teachcomputing.org/resources Year 4 Unit –Programming B Repetition in games, Lesson 4 • Pupils explore a model project using repetition. From this they then design their own game based on the model project, producing a design and algorithm for sprites in the game. Pupils share these designs with a partner and have time to make any changes to their design as required. Pupils then build their games using Scratch. They follow their algorithms, fix mistakes and refine designs in their work as they build. They evaluate their work once it is completed, and showcase games at the end. See Teach Computing materials: https://teachcomputing.org/resources Year 4 Unit – Programming B Repetition in games, Lesson 5 and 6		
Extra Links and Planning Resources	Teach Computing Unit: Year 4 – Programming B Repetition in Games https://teachcomputing.org/curriculum/key-stage-2/programming-b-repetition-in-games		

Online Safety	**Sharing Online** Pupils understand that what personal information is and how we must keep our personal information safe online. Pupils understand what to and not to be sharing through websites and games. See Sheffield Online Safety Curriculum 2019 – Protecting Ourselves L2	
Previously Taught Vocabulary	Algorithm, Program, Sequence, Blocks, Sprites, Commands, Debug	
New Key Vocabulary	**Count Controlled Loop:** a set of actions repeated for a number of set times. **Infinite Loop:** a set of actions repeatedly indefinitely.	**Loop:** The action of doing something over and over again. **Repetition:** where actions or commands in programming are repeated also referred to as a loop.
Core Substantive Knowledge	Teachers need to have a good understanding of the pedagogy behind teaching computer programming and computational thinking. They need to have a good understanding of how the software Scratch works. In this project, pupils develop their understanding of repetition within Scratch. 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 and are known as infinite loops, or for a set number of times, known as ‘count controlled loops’. There is also a focus on developing pupil’s understanding of how to create designs in programming through four levels (abstraction – see below in computational thinking). 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. (Teach Computing) Developing your subject knowledge of Scratch: **Teacher subject knowledge development online courses at:** https://teachcomputing.org/primary-teachers **Scratch online courses:** https://www.futurelearn.com/courses/teaching-programming-primary-school **Computational Thinking:** Computational thinking is about looking at a problem in a way in which a computer can help us to solve it. This is a two-step process: 1. First, we think about the steps needed to solve a problem. 2. Then, we use our technical skills to get the computer working on the problem. Computational Thinking is all about solving problems effectively – with or without a computer Computational thinking involves six different concepts and five approaches to working.	

	Computation Concepts Logic Algorithms Decomposition Patterns Abstraction Evaluation Computational Approaches: Tinkering Creating Debugging Persevering Collaborating For more information on computational thinking: **Computational thinking and concepts:** https://www.barefootcomputing.org/concept-approaches/computational-thinking-concepts-and-approaches **Introduction to Computational Thinking:** https://www.bbc.co.uk/bitesize/guides/zp92mp3/revision/1 **Guidance on Computational Thinking:** https://community.computingatschool.org.uk/resources/2324/single
Prior Knowledge	In previous units, pupil will have developed their knowledge of scratch at a basic level. In year three, pupils will have been introduced to repetition and loops in the software logo.
Assessment	Pupils that are secure in this unit can: Pupils will show a good understanding of the difference between a count-controlled loop and an infinite loop. The will have planned, designed, created, made, modified and evaluated a game that involves loops. They will have had the opportunity to develop and practise computational thinking sills, with a key focus on abstraction through the planning stages. This will be assessed through outcomes, observations and questioning in lessons.