

Year 1. Route Explorers

This unit builds on children's understanding that an algorithm is a precise, step by step set of instructions to complete a task (building on knowledge gained in Unit: Creating and Following Instructions) and continue to develop the understanding that this is at the core of coding. They are introduced to a screen sprite in the 2Go app and through this unit learn how to control the movement of the sprite through a number of challenges. They begin each challenge by verbally planning the route that they wish the sprite to take in order to arrive at a specific place (i.e. to complete a task).

In Challenge 1 they use a single command (direction – Left, Right, Up, Down) to move the sprite

In Challenge 2 they use both direction and the number of steps (distance) to move the sprite. They begin to understand that if the sprite does not move in the intended place, their instruction must have been incorrect and they should go back and correct the instruction they gave. They are reminded of terms 'bug' and 'debug' (revisiting knowledge from Unit: Creating and Following Instructions).

In Challenge 3, pupils are introduced to diagonal directions and increasingly complex mazes with multiple tasks that need to be completed. Their understanding that 'if the sprite does not move in the intended place, their instruction must have been incorrect and they should go back and correct the instruction they gave,' is reinforced, as are the terms 'bug' and 'debug'.

Pupils are reminded of what a 'program' and an 'algorithm' are (recapping on knowledge gained in Unit: Creating and Following Instructions) and recap algorithms that they have come across so far in Computing lessons. They then complete Challenge 4 but by writing an algorithm (as a form of a very simple program) to control it. Through the use of directional arrows and numbers when writing their algorithm, they are introduced to the beginnings of using code when writing programs. Pupils are encouraged to test their algorithm regularly and reinforce their understanding that if the sprite does not move where they intended it to move, then their algorithm must be incorrect and in this case they should try to 'fix' their algorithm. They recap terms 'bug' and 'debugging' to ensuring algorithms run as intended and continue to reinforce their understanding that when writing programs, it is important to test and debug.

Challenges 5 and 6 allow for longer algorithms to be written and for knowledge of testing and debugging to be embedded. Children's learning is extended for some as they are encouraged to predict where the sprite might go when their algorithms are followed, before testing.

Knowledge covered:

NC	Ladders Objectives
Understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions.	Y1- CS1 I understand that an algorithm is a precise, step-by-step set of instructions to complete a task Y1- CS2 I know that an algorithm written for a computer is called a program
To create and debug simple code	Y1- CS3 (KPI) I can design a simple program to control the movements of a screen sprite Y1-CS4 I can say that if something does not work how it should, it is because my code is incorrect Y1-CS5 I can try and fix my code if it isn't working properly
Use logical reasoning to predict the behaviour of simple programs.	Y1 – CS6 I can make good guesses of what is going to happen in a program. For example, where the sprite might go (Extended learning)

Medium Term Lesson Plan:

Lesson 1	L.I: I can control a screen sprite *Carry out pre-assessment learning* Challenge 1 using direction Challenge 2 using direction and distance	S.C: Plan the route with your finger Control the direction (Right, Left, Up, Down) Count the number of steps	<u>Key Question (from KWL)</u> Why might a sprite not move where you wanted it to? (Y1-CS4) What should you do if a sprite does not move where you wanted it to? (Y1-CS5)
Lesson 2	L.I: I can control a screen sprite Starter challenge 2 recap Challenge 3	S.C: Plan the path with your finger Control the direction (Right, Left, Up, Down, Diagonal) Count the number of steps	<u>Key Question (from KWL)</u> Why might a sprite not move where you wanted it to? (Y1-CS4) What should you do if a sprite does not move where you wanted it to? (Y1-CS5)
Lesson 3	L.I: I can write a program to control the movements of a screen sprite (Y1-CS3 KPI) Challenge 4 with direction and distance. Challenge 4 as algorithm 'lesson 3 EXT' – in My Work folde)	S.C: plan the route with your finger Control the direction Control the number of steps Test the program Debug the program (Y1-CS4; Y1-CS5)	*Prior knowledge* What is a program? (Y1-CS2) *Prior knowledge* What is an algorithm? (Y1-CS1) *Prior knowledge* What algorithms do you follow in your daily life? Why might a sprite not move where you wanted it to? (Y1-CS4) What should you do if a sprite does not move where you wanted it to? (Y1-CS5)
Lesson 4 (Part 1)	L.I: I can write a program to control the movements of a screen sprite (Y1-CS3 KPI) challenge 5 (with the extended tasks)	SC: Plan the route with your finger Direction and number of steps Test the program Debug the program (Y1-CS4; Y1-CS5) EXT: Begin to predict where the sprite might go before the code is run	<u>Key Question (from KWL)</u> *Prior knowledge* What is a program? (Y1-CS2) *Prior knowledge* What is an algorithm? (Y1-CS1) Why might a sprite not move where you wanted it to? (Y1-CS4) What should you do if a sprite does not move where you wanted it to? (Y1-CS5) EXT: Can you predict where the sprite will go? (Y1-CS6)

Year 1 Route Explorers Medium Term Plan

Lesson 4 (Part 2)	L.I: I can write a program to control the movements of a screen sprite (Y1-CS3 KPI) Challenge 6 and 7 *Carry out end of unit assessment*	SC: Plan the route with your finger Direction and number of steps Test the program Debug the program (Y1-CS4; Y1-CS5) EXT: Begin to predict where the sprite might go before the code is run	*Prior knowledge* What is a program? (Y1-CS2) *Prior knowledge* What is an algorithm? (Y1-CS1) Why might a sprite not move where you wanted it to? (Y1-CS4) What should you do if a sprite does not move where you wanted it to? (Y1-CS5) EXT: Can you predict where the sprite will go? (Y1-CS6)
------------------------------	--	--	--