Year 2 Coding (Unit 2.1) Medium Term Plan

Note: In 2025-2026 this unit is taught directly after the 'Coding 1.7 for Year 2 Catch-up' Medium Term Plan. To ensure learning from the Catch-up module is supportive of a smooth build up and progression into Coding Unit 2.1, lesson 1 has been amended (as some of the learning will have been carried out in the 'Coding 1.7 for Year 2 Catch-up' lessons directly before this unit).

Year 2. Coding (Unit 2.1)

This unit builds on knowledge gained in Year 1 (initially gained in Unit 1.4 Lego Builders, and then built upon though Unit 1.5 Maze Explorers and Unit 1.7 Coding). Pupils begin the unit by revisiting what an algorithm is. They recap various activities they carried out in year 1 (following instructions to create a Lego model, following instructions to complete a Paint Project), relating these precise, step by step instructions that were given, to an algorithm in computing and revisit knowledge that an algorithm for a computer is called a 'program' - Year 1. They continue to develop the knowledge that algorithms are at the core of coding as they are referred to throughout the unit. When planning their algorithms in this unit, pupils are encouraged to be precise to help ensure success when their codes are executed.

Pupils' programming in this unit builds upon previous knowledge of using block coding to design simple programs that control an object in 2Code (Year 1 Unit 1.7 Coding) to programming increasingly complex programs which incorporate more varied blocks with a wider range of outputs than in Year 1. Pupils begin by revisiting a familiar coding scene (Air Traffic Control Activity from Unit 1.7 Coding) to recap previous coding knowledge gained in Year 1 [recapping the terms 'backgrounds, events, objects and actions']. Building on from Year 1, rather than completing this challenge through guided steps, they use the plan of an algorithm and previous knowledge to code this in Free Mode (from scratch). The use of a plan recaps and flows on from Coding Unit 1.7 where pupils used an image of the program design and their own words to write a plan for their program before writing the code. The success of the coded program is related to the precision that was evident in the plan.

Following this recap lesson, in subsequent lessons pupils learn to design simple programs that achieve various objectives: a program that uses collision detection (an event block); a program that follows a timed sequence (using the control block 'timer'); a program that includes different object types (which result in different actions being available); and a program that includes a button object (to make other objects perform actions when used with the 'when clicked' event block). Pupils are exposed to new event blocks and actions throughout the unit, with opportunity being given to: spot these; be able to say the cause and effect of what will happen in a program as a result of their use; and use these both through guided challenges, and independently when in Free Mode. Pupils edit digital content by editing the background used in the design view, and editing properties of objects.

In each lesson, pupils are given the opportunity to predict what will happen when a plan of an algorithm or exemplar coding is executed before it is run. This builds on from Year 1 knowledge of being able to read coding line by line to make good guesses as to what might happen in a program. Ample opportunity is provided in this unit to continue developing this skill so that pupils can predict with increasing accuracy what will happen when a code is run. They are encouraged to also read their own coding, as a part of the debugging process. Children draw upon previous knowledge that bugs are errors in coding which stop the program from working the way that it was designed and that debugging is the fixing of those errors (Year 1 Unit 1.7). Challenges set towards the end of some lessons specifically focus on developing practice in the debugging process. This ties in all previous learning - the need to be precise in algorithms for desired outputs; the practice of reading code line by line to make predictions as to the output of those lines; and the understanding of the various blocks and how they work, all contributing to having programs that contain no bugs.

Vocabulary: Algorithm, Precise, Bug, Debug, Sprite, Object, Object name, Program, Plan, Purpose/Function, Test, Code, Run, Execute, Save, Code blocks, Action blocks, Object Blocks: including Button object,

Event Blocks: when clicked, when key event, when swiped event, collision detection event,

Collison detection action, Collision detection event, Timer, Sequence, Interval, Scene, Background, Properties, Scale, Text, Command, Design view, Code View, Image, Predict

Year 2 Coding (Unit 2.1) Medium Term Plan

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.	Y2-CS1: I can explain an algorithm is a precise, step by step set of instructions to complete a task Y2-CS2: I know I need to be precise when I plan my algorithm so it will work when I make it into a code
To create and debug simple code	Y2-CS3: I can design a simple program that achieves a purpose Y2-CS4: I can find and correct some errors in my program (debugging)
Use logical reasoning to predict the behaviour of simple programs.	**Y2-CS5 (KPI): I can say what will happen in a program** Y2-CS6: I can spot something in a program that has an action or effect (does something)
Use technology purposefully to create, organise, store, manipulate and retrieve digital content	Y2- CS3: I can design a simple program that achieves a purpose **Y2-IT1 (KPI): I can use technology to create a range of content (image and text based)** Y2-IT2: I can name, save and load my work Y2-IT3: I can edit digital content

Medium Term Lesson Plan:

Lesson 1 (**split over 2 lessons if needed)**	L.I: I can design a simple program that achieves a purpose **(Y2-CS3)** *Carry out pre-assessment learning* (Follow unit plan lesson 1) **Amended for 2025-2026: LI and SC remain the same.** **Pre-assessment learning will have already been carried out in amended lesson 1 of 'Coding 1.7 Catch up for Year 2'** **Follow unit plan 1 Coding 2.1, but amend to recap the use of the 'steps to make a computer program'**	S.C: Use the plan to be precise (Y2-CS2) Use collision detection (Y2-CS6) Save work (Y2-IT2)	<u>Key Question (from KWL)</u> *Prior learning* What is an algorithm? (Y2-CS1) *Prior learning* What is a program? What is meant by the function/purpose of a program? (previously referred to as 'the task') *Prior learning* Why might a program not achieve its planned purpose/function? *Prior learning* What should you do if your program is not achieving its planned purpose/function? What is this called? (debugging) *Prior learning* What is it called when there is an error in your code? (bug) What does precise mean? For the lesson: How does the plan help us to be precise when we code? (Y2-CS2)

	when writing their program (as completed in Coding Unit 1.7 for Year 2 amended Lesson 3) and link to using the steps to code Air Traffic Control.		What is the event in the collision detection event? What do we want the action to be in the collision detection event? (Y2-CS6) Can you predict what will happen when this code is run? (Y2-CS5)
Lesson 2 (part 1) Princess and the Frog guided steps and debugging	L.I: I can design a simple program that uses collision detection **(Y2-CS3)** (1st lesson: Follow unit plan lesson 2 challenges 1-4)	S.C: Collision event, collision action (Y2-CS6) Be precise (Y2-CS2) Predict what will happen (Y2-CS5) Debug in challenge 4 (Y2-CS4; Y2-IT3) Save Work (Y2-IT2)	<u>Key Question (from KWL)</u> *Prior learning* What is a program? *Prior learning* What is an algorithm? (Y2-CS1) What is the event in the collision detection event? What do we want the action to be in the collision detection event? (Y2-CS6) Can you predict what will happen when this code is run? (Y2-CS5) *Prior learning* What is it called when there is an error in your code? (bug) *Prior learning* What should you do if there is a bug? What is this called? (debugging)
Lesson 2 (part 2) Princess and the Frog using own plan and coding ending	L.I: I can design a simple program that uses collision detection **(Y2-CS3)** (2nd lesson: Follow unit plan lesson 2 challenge 5)	S.C: Write the algorithm – be precise (Y2-CS2) Code the algorithm - be precise (Y2-CS2) Collision event, collision action (Y2-CS6) Debug (Y2-CS4; Y2-IT3)	<u>Key Question (from KWL)</u> *Prior learning* What is an algorithm? (Y2-CS1) What is the event in the collision detection event? What do we want the action to be in the collision detection event? (Y2-CS6) How does the plan help us to be precise when we code? (Y2-CS2) *Prior learning* Why might a program not achieve its planned purpose/function? *Prior learning* What should you do if your program is not achieving its planned purpose/function? What is this called? Can you predict what will happen when this code is run? (Y2-CS5)

Year 2 Coding (Unit 2.1) Medium Term Plan

Lesson 3 Magician	L.I: I can design a simple program that follows a timed sequence **(Y2-CS3)** (Follow unit plan - lesson 3)	S.C: Use timer after the command Be Precise (Y2-CS2) Predict what will happen (Y2-CS5) Debug in challenge 4 (Y2-CS4; Y2-IT3)	<u>Key Question</u> What does sequence mean in coding? How can we code an action to happen after a certain time? What will happen when you run this code? (Y2-CS5) *Prior learning* Why might a program not achieve its planned purpose/function? *Prior learning* What should you do if your program is not achieving its planned purpose/function? What is this called? (debugging)
Lesson 4 (possibly over 2 lessons: part 1 – snail	L.I: I can design a simple program that includes a different object type **(Y2-CS3)** (follow unit plan – lesson 4 stopping after snail race and predicting program outcome)	SC: Explore actions for snail object Use event and action commands (Y2-CS6) Make predictions (Y2-CS5)	<u>Key Question (from KWL)</u> What objects have we programmed before? *Prior knowledge* What actions have we used? (Y2-CS6) Can you predict what will happen when this code is run? (Y2-CS5) *Prior learning* Why might a program not achieve its planned purpose/function? *Prior learning* What should you do if your program is not achieving its planned purpose/function? What is this called? (debugging)
Part 2 – make own)	L.I: I can design a simple program that includes different object types **(Y2-CS3)** (follow unit plan – lesson 4 having predicted outcome)	SC: Edit properties of objects and background (Y2-IT3) Use event and action commands (Y2-CS6) Make predictions (Y2-CS5)	<u>Key Question (from KWL)</u> Can you predict what will happen when this code is run? (Y2-CS5) *Prior knowledge* How do you change the image attributes of a sprite? How do you change the size of the sprite? Can you predict what will happen when this event block is used? [when clicked, when key is pressed, when swiped, collision detection] (Y2-CS5) What do we want the action to be (for various objects)? (Y2-CS6) What event will cause the code with the actions in to be carried out? (Y2-CS6) *Prior learning* What should you do if your program is not achieving its planned purpose/function? What is this called?

Year 2 Coding (Unit 2.1) Medium Term Plan

Lesson 5 Using buttons	L.I: I can design a simple program that includes a button object **(Y2-CS3)** **(Y2-IT1 KPI: text based and image based)** (follow unit plan – lesson 5) *Carry out end of unit assessment*	SC: Edit properties of buttons, objects and background (Y2-IT3) Use event and action commands (Y2-CS6) Predict what will happen (Y2-CS5)	<u>Key Question (from KWL)</u> *Prior learning* What is a program? What is a button? What actions can your chosen objects do? What action will you choose when the 'when clicked' event is used? (Y2-CS6) What will happen when you run this code? (Y2-CS5) What event will cause the code with the actions in to be carried out? (Y2-CS6) *Prior learning* What is it called when there is an error in your code? (bug) *Prior learning* What should you do if your program is not achieving its planned purpose/function? What is this called? (debugging) What does precise mean? For the lesson: How does the plan help us to be precise when we code? (Y2-CS2)