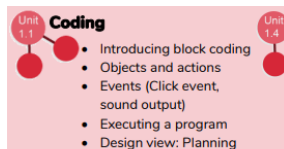
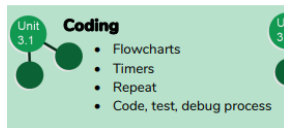
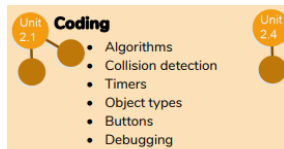


Year 4. Coding



Previous learning:



This unit reinforces the code, test, debug process or the children after having formally discussed this in year 3. It also introduces IF statements to the children for the first time, so it is important that teachers ensure they practise creating every day 'IF' sentences with the children. This then evolves into IF/ ELSE statements, which take the concept further and add an alternative consequence in the event that the 'IF' statement does not occur. Although the children did encounter repeat commands in their coding module last year, they will refine that further this year by making use of the Repeat Until command, which adds an event at the end of a predefined repetition cycle. Number Variables are introduced for the first time this year; children will very likely need lots of reinforcement around what a variable is, what they do, and with lots of examples. By the end of the unit, we would hope the majority of the class would be able to explain what each of these new terms mean and can use them in their programming.

Adaptive teaching – Recommended tried and tested practise in these lessons:

Due to the nature of Computing lessons and the common 'firefighting' required in lessons with technical issues or children who are individually confused over what may be considered small stumbling blocks, it is recommended that teachers put time and effort into reinforcing the use of school-wide learning ethos and systems (for example, the use of the 4B's – brain, board, buddy, boss) to encourage children to try to solve their problems in the right way rather than going straight to the teacher, or lessons will quickly become unmanageable. It is also recommended to follow our school's use of the Rosenshine model of mastery teaching strategy and teach lessons in a series of stages, with short, simple tasks being set, and children who finish early can be given encouragement to support and 'lifeguard' other students. This has twofold benefits; the process of teaching others cements and secures pupils' own mastery as well as ensuring the teacher isn't overwhelmed with personal requests for help. If one-to-one support is unavailable for high need children, it is a good idea to make sure these children are sat next to or near a higher ability student who may act as peer support for them, and these children are monitored carefully throughout lessons to ensure they stay on task and understand each objective given. Ensure that any hardware support that is required for individual's SEND requirements are adjusted appropriately and are tied to those students' individual log ins (e.g. Zoom tools or screen tints for students with visual impairments etc). Teachers should have a knowledge of Pupil Premium children, particularly those who have been seen to struggle in Computing lessons, and these children should be monitored alongside those with SEND during lesson time to help ensure they access lessons along with the rest of the class.

Knowledge covered:

NC	Ladders Objectives
(2.1) Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts	Y4 - CS3 - I recognise the need to test and retest whilst a program is being developed (using my prediction skills of what MIGHT work. Y4 – CS5 - I can understand that many programs can follow more than one route (more than one thing can happen) (e.g. IF THEN statements).

Year 4 Coding

(2.2) Use sequence, selection, and repetition in programs; work with variables and various forms of input and output	Y4 - CS6 - I understand that repeating instructions can save time and make a program more efficient. Y4 - CS7 - I can explain what a variable is and include them in my programming
(2.3) Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs	Y4 - CS1 - (KPI) - I can recognise an error in a program and debug it so that it works. Y4 - CS2 - I understand that algorithms (step by step instructions) will help the user to solve problems. Y4 - CS4 - I understand that a program is built up of sequences of instructions that are in order.
(2.4) Understand computer networks including the Internet; how they can provide multiple services, such as the world wide web; and the opportunities they offer for communication and collaboration.	Y4 - CS8 - I can describe how networks physically connect to other networks Y4 - CS9 - I can recognise how networked devices make up the Internet
(2.5) Use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content.	
(2.6) 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.	
(2.7) Use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact.	

Medium Term Lesson Plan:

Lesson 1	Design, Code, Test and Debug L.I I can review my previous coding knowledge. L.I I can create a simple computer program. Y4 - CS1 - (KPI) Y4 - CS2 Y4 - CS3 Y4 - CS4 Y4 - CS5 Y4 - CS6 Y4 - CS7 Y4 - CS8 Y4 - CS9	S.C: I can explore different object types in 2Code. I can use a background and objects to create a scene. I can plan an algorithm for my scene and use 2Code to program it.	<u>Key Questions</u> *Prior knowledge* Which words do you remember from coding done in the past? Which coding blocks in 2code have you used in the past? What do you want to happen in your scene?	Vocabulary knowledge review, summative assessment. Review how to place objects in 2code. Look carefully at properties, adjust to make an object/vehicle move in the desired way. Children add their own objects to backgrounds, code. Create a plan and implement it. Review vocabulary and success criteria.
Lesson 2	IF Statements L.I I can begin to understand selection in computer programming. L.I I can understand how an IF statement works. Y4 - CS1 - (KPI) Y4 - CS2 Y4 - CS3 Y4 - CS4 Y4 - CS5 Y4 - CS6 Y4 - CS7 Y4 - CS8 Y4 - CS9	S.C: I can understand how to use co-ordinates in computer programming. I understand how an IF statement works.	<u>Key Questions</u> *Prior knowledge* How can we test how this program works? What do you predict will happen if we type 'yes'? No?	Recap recent vocabulary. Introduce the term 'selection'. (Show video) Show 'Is it raining?' – an example of an 'IF' statement at work. Show code to children and get them to predict how it will work. Talk though how it works with the class. Test different inputs. Show 'Lost'. Explain how this program works. Children to re-skin sprites and background to

Year 4 Coding

				create their own version of 'lost', then see what they can add to it using other coding blocks whilst maintaining the integrity of the existing program. Review new vocabulary.
Lesson 3	Co-ordinates L.I I understand how to use co-ordinates in computer programming. L.I I understand how an IF statement works. Y4 - CS1 - (KPI) Y4 – CS2 Y4 – CS3 Y4 – CS4 Y4 – CS5 Y4 – CS6 Y4 – CS7 Y4 – CS8 Y4 - CS9	S.C: I can make use of the X and Y attributes of objects in my coding. I can create a program that includes an IF statement.	<u>Key Questions</u> *Prior knowledge* What is the 'x' axis? (Refer back to co-ordinates lesson) Why is the knight disappearing off the screen? Why do we need to use an 'every' timer in this code? What instructions do we need to give each knight?	Review vocabulary. Discuss co-ordinates and ways to remember whether we say the X or Y axis first. Move an object around on blank Free Code design screen and show how co-ordinates change. Give children time to explore and find out where 0,) is on the grid. Complete 'Guard the Castle' 2Do activity.
Lesson 4	Repeat Until/IF/Else Statements L.I I can use coding knowledge to create a range of programs. L.I I understand the importance of nesting. Y4 - CS1 - (KPI) Y4 – CS2 Y4 – CS3 Y4 – CS4 Y4 – CS5 Y4 – CS6 Y4 – CS7 Y4 – CS8 Y4 - CS9	S.C: I can read code that includes repeat until and IF/ ELSE and explain how it works. I can create a program that includes an IF/ ELSE statement. I can interpret a flowchart that depicts an IF/ ELSE statement.	<u>Key Questions</u> *Prior knowledge* What type of selection criteria did we use in 'Is it Raining?' ('IF' statement) How do 'IF' statements work? What does an 'IF/ELSE statement allow us to do?	Start the lesson by recapping 'Is it raining'. Complete a class 'IF' statement. Look at the 'Is it raining' IF/ELSE flowchart. Challenge children to re-code the program to include the extra functions (may need to be modelled in stages). Look at 'Reginald Rocket' example program. Ask children to predict how the program will run.' Explore
Lesson 5	Number Variables L.I I can understand what a variable is in programming. I can use a number variable. Y4 - CS1 - (KPI) Y4 – CS2 Y4 – CS3 Y4 – CS4 Y4 – CS5 Y4 – CS6 Y4 – CS7 Y4 – CS8 Y4 - CS9	S.C: I can explain what a variable is in programming. I can create and use variables when programming	<u>Key Questions</u> *Prior knowledge* What does the word 'vary' mean? Why has this variable been created? What is it doing? Why is nesting our instruction correctly important?	Recap vocabulary. Introduce 'Variables' as a term. Play a quiz with the children, repeatedly talking about variables when the score increases (scores and timers are good examples of variables). Play 'Genie' 2Do. Create a number variable, show the 'variable count' window. Children must make sure their 'IF' statement is nested correctly within their 'When clicked' event, or it won't be checked when the lamp is clicked. Then complete 'Night and Day'. Opportunities to extend and add more to build up the scene they create.

Year 4 Coding

				Review S.C
Lesson 6	Making a playable game L.I I can review vocabulary and concepts learnt in Year 4 Coding. L.I I can create a playable game.	S.C I can use the correct code to make their game work. I can explain how my code makes my game work.	<u>Key Questions</u> <i>*Prior knowledge*</i> How were variables used in our last game? How are they being used in this one? What is the number object doing?	Review vocabulary. Display 'Turtle Race' game on board. Children predict then explore and test to see how it works. Can they make any changes to this game or improve how it works? Children create own game with a score variable. Review learnt skills so far – can they include them? (Number variables to keep score, selection, IF and IF/ELSE statements, co-ordinates (using X and Y) , Timers – after and every (could be handy to set a time limit), repeat and repeat until, alert boxes (these could be good for instructions)