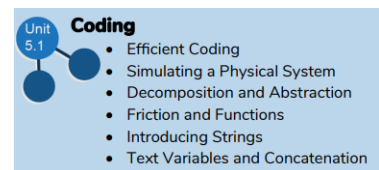
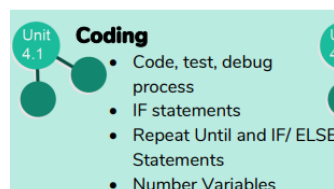
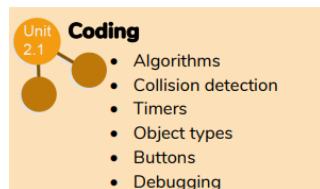
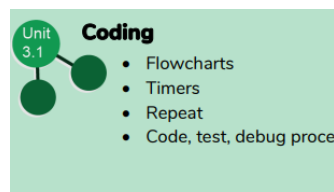
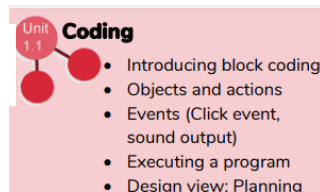


Year 6. Coding

Previous learning:



Technical knowledge from the previous years' learning is required from the beginning of this unit, as the children are immediately challenged to include at least two different variables in their first stage of programming in order to create a functional game. They are introduced to functions for the first time; why they can be useful and time-saving, as well as understanding that they are a type of variable. Their previous work in planning and predicting code will be called upon in lesson 4, when they will be required to use flowcharts in order to test and debug a program. This will test their debugging skills to a far greater extent than in previous lessons, as the simulation they are provided with contains a lot of bugs and missing code, and will require them to examine it in depth. They will also be expected to show a good understanding of coding from a user's perspective and evaluate what a user will want, showing an ability to predict all possible forms of input for their programs in lesson 5.

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	Y6 - CS2 - I can use IF, THEN, ELSE commands to control what happens. Y6 - CS3 - I can create and call FUNCTIONS to better organise my coding. Y6 - CS6 (KPI) - I can control a physical system using a range of commands (e.g. LEGO)

Year 6 Coding

(2.2) Use sequence, selection, and repetition in programs; work with variables and various forms of input and output	Y6 - CS4 - I understand what variables can be used to make a game more realistic (e.g. a timer or score). Y6 - CS5 - I understand that any system requires input devices (e.g. keyboards) and output devices (character on screen/ printer/ model) and that the system processes the data.
(2.3) Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs	Y6 - CS1 (KPI) - I can write programs to do what I want them to do. I can test and debug these.
(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.	Y6 - CS7 - I understand that my work can be saved in a range of places (the device itself, USB stick or 'in the cloud') and can talk about this.
(2.5) Use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content.	Y6 - CS8 - I can identify how to use a search engine Y6 - CS9 - I can describe how search engines select results
(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 & Lesson 2	Designing and Making a more Complex Program L.I I can design a playable game with a timer and a score. L.I I can plan and use selection and variables. L.I I can understand how the launch command works. Y6 – CS1 (KPI) Y6 - CS2 Y6 - CS3 Y6 - CS5 Y6 - CS6 (KPI) Y6 - CS7	S.C: I can plan a program which includes a timer and a score. I can follow my plans to create a program. I can debug when things do not run as expected.	<u>Key Questions</u> *Prior knowledge* What do these coding terms mean? Which of these terms do you need a refresher with? How does using a variable help us with our coding? Which of these coding operations have you used before? Could you use them again?	Children have a go at their final coding vocabulary quiz – it should give their best score yet. Children look at the design of a game and then predict how it will work, then they examine the code. Children add a timer variable. They will need to add co-ordinates into their code blocks in order for their programs to work correctly. Demonstrate how variables can be used to simplify code to save time. Children plan and design a game of their own that uses a similar coding mechanic. Children create their game using Free Code Gorilla. Possible extensions: Include characters that respond to being clicked, characters which respond to collisions, using timers to repeat actions, use 'repeat until' etc (Outlined in short term lesson plans). Children likely to struggle may modify the existing 'Splatty Bug' game in their own way – challenge to add to it and improve it.
Lesson 3	Using Functions L.I I can use functions and understand why they are useful. L.I I can understand how functions are created and called. Y6 – CS1 (KPI) Y6 - CS2 Y6 - CS3	S.C: I can create a program that makes use of functions. I can create a program that uses multiple functions with the code arranged in tabs. I can explain how my code	<u>Key Questions</u> *Prior knowledge* Which of these terms do you remember? Do you remember learning about functions in your year 5 coding lessons? What angle will the turtle need to turn?	Go over key vocabulary that will be used in this lesson. Children open a tutorial which will re-acquaint them with functions. Are shown, step by step, how to create a function and use it to program a turtle to draw a shape. Children are challenged to find two different ways to incorporate 'pen up' and 'pen down' into their code (two different ways possible). Discuss. Children are then challenged to create a range of shapes using the turtle.

Year 6 Coding

	Y6 - CS5 Y6 - CS6 (KPI) Y6 - CS7	executes when my program is run.	Is that angle the inside angle or the outside angle of the shape?	Extensions: Add new buttons for new shapes, new turtles, different coloured lines, get your turtle to start and return at a certain position, use random co-ordinates. Children who may struggle should be guided as to angles needed for their shapes so any maths-based issues don't interfere with their coding.
Lesson 4	Flowcharts and Control Simulations L.I I can use flowcharts to test and debug a program. L.I I can create a simulation of a room in which devices can be controlled. Y6 – CS1 (KPI) Y6 - CS2 Y6 - CS3 Y6 - CS5 Y6 - CS6 (KPI) Y6 - CS7	S.C: I can follow flowcharts to create and debug code. I can create flowcharts for procedures. I can be creative with the way they code to generate novel visual effects.	<u>Key Questions</u> *Prior knowledge* What does the word 'simulation' mean? Can you remember learning about simulation from your year 5 coding lessons? How should this simulation work? Do you spot any areas in the simulation that don't work in the way they are supposed to?	Review technical vocabulary which will be used in this lesson. Discuss what simulations are and why computer simulations are used in industry. Children look over a flowchart of a simulation. Examine it in detail and explain what they think will happen. Children run the simulation. Encourage children to look at the chart, then predict what they think will happen. Then test the program to see if it works in the correct way. Children add missing code and debug the simulation. Create flowcharts for and program missing elements of the simulation. Children who need support can be guided through coding the movement of the fairies.
Lesson 5	User input L.I I can understand the different options of generating user input in 2Code. L.I I can understand how user input can be used in a program. Y6 – CS1 (KPI) Y6 - CS2 Y6 - CS3 Y6 - CS5 Y6 - CS6 (KPI) Y6 - CS7	S.C: I can code programs that take text input from the user and use this in the program. I can attribute variables to user input. I am aware of the need to code for all possibilities when using user input.	<u>Key Questions</u> *Prior knowledge* What is user input? Which of these commands from Free Code Gorilla can be used to get user input? Do you remember learning about concatenation in year 5? What happens if you give a word answer in the prompt when this code is run? How could the code be developed so the user could pick any of these 6 alien cards to play it?	Review vocabulary for this lesson. Children examine the term user input and differentiate between different commands about user input in Purple Mash. Children are given some example code to link text strings together to complete. Predict how it will work, test and debug. Children look at another example to study and predict. Predict, test, debug. Children create a user input command of their own. Run 'Guess the alien' program (works like 'Guess Who?' – asking questions and eliminating possibilities). Look at a sample of code from the program. Children add code so that the program works for any of the 6 aliens chosen. Support children having difficulty by writing code with them one step at a time.
Lesson 6	Text-based adventures L.I I can understand how 2Code can be used to make a text-based adventure game. Y6 – CS1 (KPI) Y6 - CS2 Y6 - CS3 Y6 - CS5 Y6 - CS6 (KPI) Y6 - CS7	S.C: I can follow through the code of how a text adventure can be programmed in 2Code. I can design my own text-based adventure game based on one I have played. I can adapt an existing text	<u>Key Questions</u> *Prior knowledge* What is a text adventure? Has anyone here ever played one before? What is the aim of this game?	Vocabulary review. Discuss and explain what a text-based adventure is. Children play a simple text-based adventure game and reflect on how and why it works. Children draw a map of the rooms as they navigate (may need modelling for any students struggling with how to do this). Encourage them to draw a compass. Number the rooms. 5 variables from the code are explained. Then a function with an 'IF' statement. Discuss. Encourage children to work out and understand the code for themselves. Children look at some code than runs when the game is finished. Discuss, predict, test.

Year 6 Coding

		adventure so it reflects my own ideas.		Children write out their own plan for their own text adventure. Encourage a different theme e.g. Alice in Wonderland. Children then create their own adventure. Extension: Add more elements/more rooms. Support for struggling students: Begin a plan with a group for them to follow and provide example code for them and thematic ideas.
--	--	--	--	---