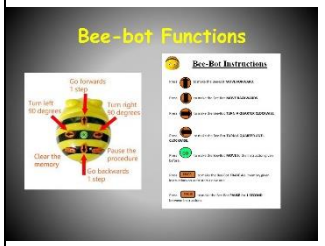
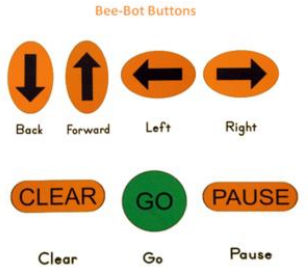
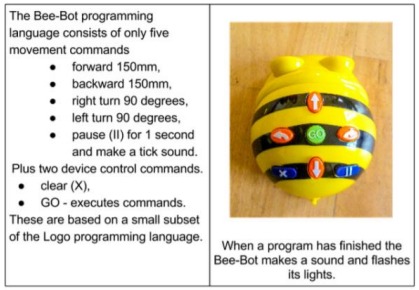
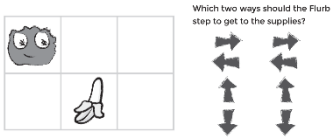
Key Concept(s)	Key Question	Activities	Resources	Assessment
PROGRAMMABLE ROBOTS				
KC1	To understand and use algrorithms	Be 'bossy' and instruct a child to do something e.g. stand up, go to door, open it, come back to carpet place and sit down. Say that sequences of instructions are important as they help us to know what to do and how to make things happen. Explain the lesson is going to be about programming and how **algorithms** help us write code (that algorithms are steps to make something happen and are for people to understand but that programs are for computers) Show children a Bee-Bot. Ask children how we program it ('what do the different buttons do?'). Sit your group in a circle and ask them how we could get the Bee-Bot to write the numeral 1. Write the numeral 1 on a whiteboard to show them. They might suggest forward, forward. Some might start at the top of the number, others at the bottom. Both options are valid. In groups children use small laminated BeeBot button pictures to make their instructions (algorithm). Explain it is often a good idea to walk through an algorithm before we program it. Stand up and say, 'I start here, facing this way, I take a step forward, and another step forward – yes that makes a number 1'. Choose a child to be a coder and input the algorithm into the BeeBot. Repeat with a different pattern for the BeeBot to make. Add BeeBot mazes to the provision for children to access independently	BBC Bitesize video BeeBots BeeBot instruction cards  BeeBot buttons (cut and laminated)  	

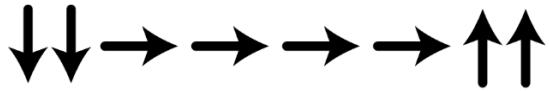
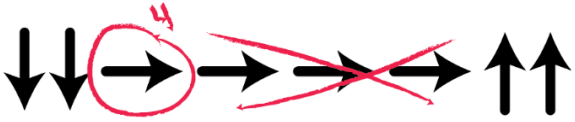
KC1	To understand bugs and begin to fix them	Explain bugs and debugging (debugging is the process of detecting and correcting the errors in a program. Bugs happen in programs all the time and therefore debugging is important knowledge to have. Show children the BeeBot moving through a maze/pattern but with an instruction that is wrong. Can children spot the bug and correct it? Place Beebots and mazes into the provision for children to access independently	BeeBots BeeBot mazes/pattern cards	

Key Concept(s)	Key Question	Activities	Resources	Assessment
CODING				
KC1	To know how to drag and drop	Code.org Course A, Lesson 2: Drag and Drop This stage was designed to give students the opportunity to practice hand-eye coordination, clicking, and drag & drop techniques. Students will also play with sequence. Take some time to explicitly teach how to click, drag, and drop. Take time to introduce the language around the devices students will be using when they work on the puzzles. If you have tablets, students will be using a touch screen. If you have laptops, they will likely be using a trackpad. Desktop computers like you might find in a lab will rely on the use of the mouse.	Laptops/iPads/desktop computers (individual or pairs)	

KC1	To understand and use algorithms	Code.org Course A, Lesson 3: Happy Maps The bridge from algorithms to programming can be a short one if students understand the difference between planning out a sequence and encoding that sequence into the appropriate language. This activity will help students gain experience reading and writing in shorthand code. Recap 'algorithms' (instructions for computers) from the BeeBot lessons. In this exercise, the class will get map cards that have a pre-defined start space (Flurb) and end space (fruit). Students will need to get the Flurbs to the fruit on each card, using the arrows provided. Select one of the maps from the Happy Maps Cards worksheet. Display it for the class and work through the puzzle together. Have students look at the puzzle, then think-pair-share their solution for how they would get the Flurb to the fruit (the 'code' they will use) Children continue own sheets.	Happy Maps sheets	
KC1	To understand and use sequencing	Code.org Course A, Lesson 4: Sequencing In this lesson, students will develop programming and debugging on a computer platform. The block-based format of these puzzles help students learn about sequence and concepts, without having to worry about perfecting syntax. Project a puzzle from the lesson. Show the class how to click on a block and place it in the correct spot by dragging and	Laptops/iPads/desktop computers (individual or pairs)	

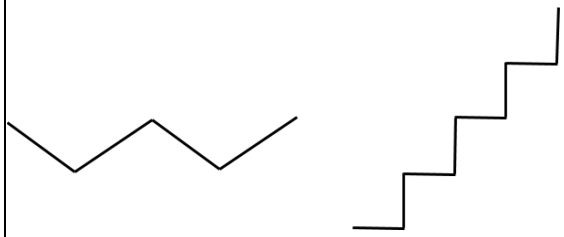
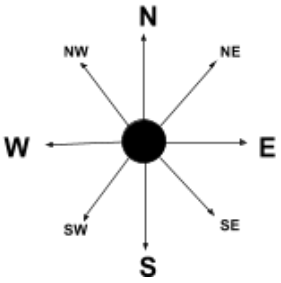
		dropping. Purposely make mistakes such as clicking the background or dropping the image before it's at the right spot. Ask for help from volunteers in the class when you run into these problems, and help them use the techniques that they developed in the last unplugged lesson to make things right. Children then complete the level (pairs or individually)		
KC1	To understand programming	Code.org Course A, Lesson 5: Programming In this lesson, students will develop programming and debugging on a computer platform. The block-based format of these puzzles help students learn about sequence and concepts, without having to worry about perfecting syntax. Show Lesson 5, Puzzle 5. Point out the "Play Area" with Scrat, as well as the "Work Space" with the Blockly code. Explain that this Blockly code is now the language that students will be using to get Scrat to the acorn. Do they see any similarities to the exercise that they just did? What are the big differences? Work with your class to drag code into the workspace in such a way that Scrat (eventually) gets to the acorn. Children then complete the level (pairs or individually)	Laptops/iPads/desktop computers (individual or pairs)	
KC1	To know how to improve programming	Code.org Course A, Lesson 6: Programming In this lesson, students will use their newfound programming knowledge in more complicated ways to navigate a tricky	Laptops/iPads/desktop computers (individual or pairs)	

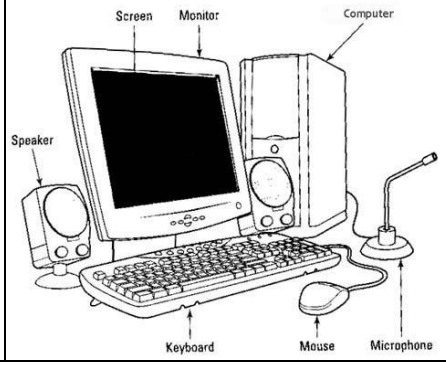
		course. With transfer of knowledge in mind, this lesson gives students a new environment to practice the techniques that they have been cultivating. Each puzzle in this series has been added to provide a deeper understanding of the basic concepts that they will be using throughout the rest of this course. Model first, then children complete the level (pairs or individually)		
KC1	To understand loops	Code.org Course A, Lesson 7: Happy Loops This lesson serves as an introduction to loops. Loops allow for students to simplify their code by grouping commands that need to be repeated. Students will develop critical thinking by noticing repetition in movements of their classmates and determining how many times to repeat commands. Discuss (ctrl+click to view bigger images):  		

		The resulting code might look something like this:  It's a bit long, isn't it? Model how to say it in a shorter way – 'east 4 times' is quicker than 'east east east east'.  To  To  Once students have put together the idea of "repeating" code, give them the vocabulary around it. Make sure to share with them that often the terms "repeat something" and "loop something" will be used interchangeably in Code Studio.		

KC1	To use loops when programming	Code.org Course A, Lesson 8: Loops Building on the concept of repeating instructions from "Happy Loops," this stage will have students using loops to get to the acorn more efficiently on Code.org. In this lesson, students will be learning more about loops and how to implement them in Blockly code. Using loops is important in programming because manually repeating commands is tedious and inefficient. With these Code.org puzzles, students will learn to add instructions to existing loops, gather repeated code into loops, and recognize patterns that need to be repeated. Show Lesson 8, Puzzle 5. Point out the "Play Area" with Scrat and the acorn, as well as the "Work Space" with the Blockly code. Explain that this Blockly code is now the language that the class will be using to help Scrat get to the acorn. Do students see any similarities to the exercise that they just did? What are the big differences? Work with your class to drag code into the workspace in such a way that Scrat (eventually) gets to the acorn. Children then complete the level (pairs or individually).	Laptops/iPads/desktop computers (individual or pairs)	
KC1	To use loops when programming	Code.org Course A, Lesson 9: Loops In this lesson, students continue learning the concept of loops. In the previous lesson, students were introduced to loops by		

		moving through a maze and picking corn. Here, loops are used to collect treasure in open cave spaces. Quickly review the definition of a loop, the action of doing something over and over again. • What are loops? • Why do we use them? Model first, then children complete the level (pairs or individually)		
KC1	To understand why loops are important	Code.org Course A, Lesson 10: Loops Returning to loops, students learn to draw images by looping simple sequences of instructions. In the previous plugged lesson, loops were used to traverse a maze and collect treasure. Here, loops are creating patterns. At the end of this stage, students will be given the opportunity to create their own images using loops. This lesson gives a different perspective on how loops can create things in programming. Students can also reflect on the inefficiency of programming without loops here because of how many blocks the program would require without the help of repeat loops. Quickly review the definition of a loop, the action of doing something over and over again. Discuss different patterns like zigzags and stairsteps: *How would you explain to someone how to draw that pattern? How could you draw this using a loop?*	Laptops/iPads/desktop computers (individual or pairs)	

		In the artist levels students will be using 45 degree angles described as northwest, northeast, southwest, southeast. We recommend briefly discussing these directions with the class and drawing an image for students to refer back to. (ctrl+click to view bigger images)   Model first, then children complete the level (pairs or individually)		

Key Concept(s)	Learning Intention	Activities	Resources	Assessment
CREATING DIGITAL CONTENT				
KC3	To understand what software is	Watch video, pausing for clarification and to check understanding	**Vocab** Software, application (app), hardware, coding, programs	
KC3	To understand what hardware is	Explore the physical parts of a computer/laptop – mouse, keyboard, track pad, screen etc Explore the desktop of a PC – icons, start menu etc Explain the functions of these Label large diagram as class  	Laptops/PC, large diagram to be labelled, parts of a computer labels **Vocab** Desktop, icon, start menu, keyboard, monitor	

	To know how to open a computer program	Introduce purpose (link to a curriculum subject). Model how to open the relevant software (word or powerpoint, whichever is relevant to the project). Explain the purpose of word processing software (e.g. Word) is for inputting, editing and formatting text, often with some additional features e.g. inserting pictures, diagrams or tables. Presentation software (e.g. PowerPoint) is designed to help people present their information to others and focuses on images, diagrams, flow charts with some text to support Children use laptops practise finding the desktop shortcut and opening the program they have chosen to use	Laptops **Vocab** Software, word processing, presentation, formatting, input, editing, typing, tables, desktop shortcut	
KC3	To know how to add type to text to word processing software	Open Word or PowerPoint and model using it to: - add text - delete text - press enter to start a new line - use the space bar for spacing - use caps lock for uppercase Children practise the above on laptops	Laptops **Vocab** enter, delete, space bar, caps lock, software, word processing, presentation, formatting, input, editing, typing, tables, desktop shortcut,	
KC3	To know how to edit text in a word processing program	Open Word or PowerPoint and model using it to edit the text: - font/tpyeface - colour - size	Laptops **Vocab** Font/typeface, colour, size, enter, delete, space bar, caps lock, software, word	

		Children practise the above on laptops	processing, presentation, formatting, input, editing, typing, tables, desktop shortcut	
KC3	To save a document	Open Word or PowerPoint and show children the steps needed to save a document (save on Global Area in class folder) Children practise saving a document on laptops	**Vocab** Save, save icon/button, save as, file, folder	
KC3	To open a saved document	Open Word or PowerPoint and show children the steps needed to open a saved document (from Global Area in class folder) Children practise opening a document on laptops	**Vocab** Open, open button/icon, save, save icon, save as, file, folder, retrieve, find	

KC1	**Computer programming and perseverance**	Understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation Analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems
KC2	**Using technology to solve problems**	Evaluate and apply information technology, including new or unfamiliar technologies, analytically to solve problems
KC3	**Creating digital content**	Be responsible, competent, confident and creative users of information and communication technology