

Mid-term plan to teach Computing using the micro:bit and Scratch

Lesson	What will pupils do?	Why should they do this?	National Curriculum links
1: Familiarising pupils with Scratch and the micro:bit	<p>Allow pupils to explore the micro:bit, discovering the features that it has and what they may be used for.</p> <p>Children may decide to label the micro:bit or test each other on what they have discovered - the resources for this can be found here.</p>	<p>It can be useful for children to have an understanding of the micro:bit so that they know what functions it has and what it can be used for.</p> <p>Allowing them to explore and discover independently will be engaging and motivating for the children.</p>	<ul style="list-style-type: none"> Gain an understanding of how physical systems work .
2: Exploring the micro:bit through programming with the BBC micro:bit Javascript Blocks Editor	<p>Allow pupils to use the micro: bit Javascript Blocks Editor to practice writing code for the micro: bit.</p> <p>Children will be able to drag and drop blocks and then test out the program on the simulator. From here, children will be able to download the program that they have created, causing it to be sent to the micro:bit, which will run the code.</p>	<p>Children should be given the opportunity to test out the features that they discovered in the previous lesson.</p> <p>Using the Javascript Blocks Editor will support children in putting code onto their micro:bit, as well as supporting them in understanding the way that it works and the algorithms that it requires to work.</p>	<ul style="list-style-type: none"> Use sequence, selection and repetition in programs; Work with various forms of input and output; Design programs that accomplish specific goals, including controlling physical systems.
3/part 1: Play Scratch games that are designed for the use of a keyboard and mouse	<p>Allow pupils to play games on Scratch and explore the code of these programs.</p>	<p>Children should play different games on Scratch and manipulate their codes to build on their ability to reason logically, where they are required to explain how the algorithms work.</p>	<ul style="list-style-type: none"> Understand what algorithms are and how they are implemented as programs on digital devices;
3/part 2: Play Scratch games that are designed to be used with the Micro:Bit	<p>Encourage the children to remix these games, manipulate some codes and predict what will happen when certain codes are changed.</p>		<ul style="list-style-type: none"> Appreciate that programs need to follow precise instructions in order to work effectively; Use logical reasoning to predict the behaviour of simple programs.

4: Design a simple game that uses computer inputs such as: the space bar, mouse clicks and arrow keys	Allow the children to design a simple game on paper, taking into consideration key features of a game and how they may be implemented using code: characters, backdrops, lives, score.	Children should design their game on paper first, enabling them to have a clear idea of the outcome that they will produce in the following lesson.	<ul style="list-style-type: none"> Design programs that accomplish specific goals; Cross-curricular link to design and technology and art where children are required to design sprites and backdrops for their game.
5: Begin to create a game using standard scratch blocks	Allow the children to begin creating their sprites and code their game. Children should use normal scratch blocks to build their game as opposed to Scratch's micro:bit extension blocks.	Children should already be familiar with Scratch programming language therefore creating a simple game using blocks is something that should come easy to them. This will make it easier for the children to manipulate their code for the micro:bit at a later date. For example, a left arrow key algorithm can be changed to be a 'if tilted left' algorithm.	<ul style="list-style-type: none"> Use sequence, repetition and selection in programs, work with variables and various forms of input; Use logical reasoning to predict the behaviour of simple programs; Cross-curricular link to design and technology and art where children are required to design sprites and backdrops for their game.
6/part 1: Finish creating game and test its functions to ensure that it works correctly 6/part 2: Change input codes required on the computer to input blocks that are from Scratch's micro:bit block extensions.	Children should finish creating their games and test them using the different algorithms that they have written. They should be given the opportunity to debug any problems that occur.	Children should test their game using computer input controls - this allows them to see if it is working. If it is, great, they can move onto the next part of the lesson. If not, children can practice the process of debugging, where they are then required to pinpoint the code that is not working and rectify it. When games are working correctly, children should change their algorithms to use Scratch's micro:bit extension blocks, allowing the game to be controlled by the micro:bit.	<ul style="list-style-type: none"> Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; Decompose problems into smaller parts and then solve them; Detect and correct errors in algorithms and programs; Use various forms of input and output.
7: Exhibit games in the classroom	Allow the children to share their games with each other. Children should test each others games with the micro:bit and look for ways that games and codes could be improved.	Supports children in working together to find solutions to games that perhaps don't work. Will help children to recognise that there is always more than one way to write code.	<ul style="list-style-type: none"> Create and debug simple programs Control and simulate a physical system; Work with various forms of input and output; Detect and correct errors in algorithms and programs.