

Subject: Computing	Year: 1 – Summer 2 – <b>Programming B – Introduction to Animation</b>
<p><b>National Curriculum objectives</b></p> <ul style="list-style-type: none"> <li>• Understand what algorithms are, how they are implemented as programs on digital devices, and that programs execute by following precise and unambiguous instructions;</li> <li>• Create and debug simple programs;</li> <li>• Use logical reasoning to predict the behaviour of simple programs.</li> </ul>	
<p><b>To begin this unit, the children should have already learnt:</b>  This unit builds on the children’s knowledge and understanding from ‘Programming A – Moving a robot’ (Spring 1) in which children will have learned to program a floor robot using instructions.</p>	<p><b>The learning in this unit will prepare the children to learn these things in the future:</b></p> <p><u>Year 2</u>  We can create simple quizzes in <i>ScratchJr</i> where the user can select an answer by clicking on a sprite. An outcome occurs when the sprite is clicked.</p> <p><u>Year 3</u>  We can use event and action command blocks in order to make sprites carry out actions when certain prompts take place. Algorithms (a set of instructions to perform a task) allow us to sequence movements, actions and sounds in order to program effective animations.</p> <p><u>Year 4</u>  Count-controlled and infinite loops can be used to create different examples of repetition in games: using <u>repeat and loop operator</u> blocks in <i>ScratchJr</i> can make our programs more logical and efficient by running code continuously or for a set number of times.</p> <p><u>Year 5</u>  ‘Conditions’ can be used in programming: the ‘if... then... else...’ structure can be used to select different outcomes depending on whether a condition is ‘true’ or ‘false’. Issues with programs can arise when answers similar to those in the condition are given as inputs. We must predict such issues and identify ways to avoid such problems.</p> <p><u>Year 6</u>  Micro:bits are small computers that perform different actions based on programs written on computer software. Programs are then downloaded to the micro:bit. Micro:bits have a range of input sensors that can be used as input triggers for different codes to run. Output devices on Micro:bits (e.g. LED displays) can be programmed to display words, pictures and numbers.</p>
<p><b><u>Key Enquiry Question</u></b>  What do you call a character in <i>ScratchJr</i> and how can you make it move? How do you know your algorithms have been successful? What happens when you change a block’s value? What can you do if you</p>	<p><b><u>The Big Idea:</u></b>  Programming is when we make a set of instructions for computers to follow. <i>ScratchJr</i> is a program that we can use in order to code our own stories and animations. It involves sprites (characters on</p>

<p>want to change your sprite? Why did you choose to move your sprite in that way? What do you like about your rocket sprite?</p>	<p>the screen), which can be programmed to perform actions based on individual values and series of commands (blocks) linked together.</p>
<p><b>To achieve ARE, pupils will need to be secure in the following knowledge:</b></p>	
<p><b>By the end of this unit, children will know:</b></p> <ul style="list-style-type: none"> <li>• How to enact a given word;</li> <li>• Words that can be enacted;</li> <li>• Predictions of the outcome of a command on a device;</li> <li>• To choose a command for a given purpose;</li> <li>• To match a command to an outcome;</li> <li>• How to run a command (press a button);</li> <li>• Building a sequence of commands in steps;</li> <li>• What a given command does;</li> <li>• Commands can be used on a given device;</li> <li>• A series of instructions can be issued before they are enacted;</li> <li>• To choose a series of commands that can be run as a program;</li> <li>• How to combine commands in a program.</li> </ul>	<p><b>Vocabulary:</b></p> <p><b>Programming; <i>Scratch Jr.</i>; command; algorithm; sprite; home; block; stage; background; app.</b></p>
<p><b>By the end of this unit, children will be able to do:</b></p> <ul style="list-style-type: none"> <li>• Choose a series of words that can be enacted as a program;</li> <li>• Choose a series of commands that can be run as a program;</li> <li>• Run a program on a device.</li> </ul>	<p><b>Useful Resources:</b></p> <p><b>Online training courses</b>  <a href="#">Raspberry Pi online training courses</a></p> <p><i>ScratchJr</i> for iPads and/or computers.</p>



# COMPUTING: PROGRAMMING

## KNOWLEDGE ORGANISER

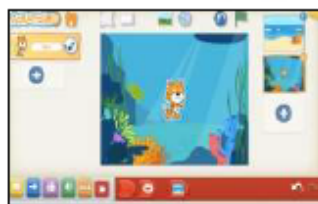


### Overview



#### Animations in Scratch Jr.

- **Programming** is when we make a set of instructions for computers to follow.
- Scratch Jr. is a program that we can use in order to code our own stories and animations. It involves sprites (characters on the screen).
- We use **algorithms** (a set of instructions to perform a task) to program the sprite to do different things.

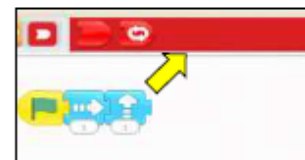


### Sequencing

- **Sequences:** -A sequence is a pattern or process in which one thing follows another. In Scratch Jr. we can stack blocks together side by side in order to create programs made up of sequences.



- **Deleting Blocks:** Blocks can be removed from programs by dragging them from the programming area back into the blocks palette.



- **Repeating Blocks:** For something to happen more than once, we can change the number underneath the block.



- **Running the Code:** Run your animation by tapping the full screen icon, and then the green flag.



### The Basics of Scratch Jr.

- **What is Scratch Jr?** Scratch is a website/ app that lets us code our own stories, games and animations.

- **Sprites:** Scratch Jr. uses characters called sprites. The main sprite is a cat called Scratch.

- **Home:** Clicking on the house takes you 'home' to your project screen.



#### Getting Started

- The **+** (right) starts a new project.



- These (right) are the **programming blocks**. We drag them into the **programming area** (right). Clicking the block in the area makes the sprite perform on the **stage**.



- **Background:** Backgrounds are added by clicking this icon (right).



- **Start Blocks:** Start blocks are yellow. These are used to start/run programs.

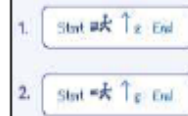


- **End Blocks:** End blocks are red. These show what happens at the end of your program.



### Algorithms and Programming

- An **algorithm** is a set of instructions for performing a task. Designing an algorithm can help us to make the sprite do the things that we want it to do.



- **Programming** is when we move the blocks into the position (based on our algorithm design). Our programming codes the sprite to perform the actions.



### Debugging

- Sometimes, things don't work exactly how we want them to the first time. This may be a problem with our algorithm, or we could have made a mistake in our programming.



- If the animation does not work correctly the first time, remember to **debug** it. This means finding and fixing the problems.



### Important Vocabulary

Programming

Scratch Jr.

Sprite

Home

Command

Block

Stage

Background

Algorithm

App