Subject: Computing	Year: 3 – Summer 2 – **Programming B – Events and Actions**
National Curriculum objectives • Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts; • Use sequence, selection, and repetition in programs; work with variables and various forms of input and output; • Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs; • 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.	
To begin this unit, the children should have already learnt: Year 1 – Moving a Robot (Spring 1) Floor robots have buttons which help us to direct them. We can use algorithms (a set of guidelines to perform a task) to program floor robots along routes. Year 1 – Introduction to Animation (Summer 2) Programming is when we make a set of instructions for computers to follow. *ScratchJr* is a program that we can use in order to code our own stories and animations. Year 2 We can create simple quizzes in *ScratchJr* where the user can select an answer by clicking on a sprite. An outcome occurs when the sprite is clicked.	**The learning in this unit will prepare the children to learn these things in the future:** Year 4 Count-controlled and infinite loops can be used to create different examples of repetition in games: using repeat and loop operator blocks in *ScratchJr* can make our programs more logical and efficient by running code continuously or for a set number of times. Year 5 '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. Year 6 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.
Key Enquiry Question How do 'events' affect sprites' move or act in *ScratchJr*? What directions could your sprite be programmed to move? Why have your sprite ____ size? What blocks can you use to set up your program every time? How could adding _____ block change the outcome? What can you do to check a program works? Who might benefit from having a pen trail following the sprite in the main? Does your project work successfully and could it be improved?	**The Big Idea:** 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.

To achieve ARE, pupils will need to be secure in the following knowledge:	
By the end of this unit, children will know: • Programs start because of an input; • What a sequence is; • A program contains a series of commands; • The sequence of a program is a process; • The order of commands can affect a program’s output; • Different sequences can achieve the same output; • Different sequences can achieve different outputs.	**Vocabulary:** Programming; *Scratch Jr*.; command; algorithm; sprite; home; block; stage; background; app (introduced in Y1). Sequence; quiz; debugging (introduced in Y2). **Code; events; motion; trialling.**
By the end of this unit, children will be able to do: • Build a sequence of commands; • Combine commands in a program; • Order commands in a program; • Create a sequence of commands to produce a given outcome.	**Useful Resources:** **Online training courses** **Raspberry Pi online training courses** *ScratchJr* for iPads and/or computers.

Y3

COMPUTING: PROGRAMMING KNOWLEDGE ORGANISER

Overview

Events and Actions in Scratch

- Programming is when we make a set of instructions for computers to follow.

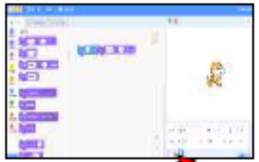
-Scratch is a program that we can use in order to code our own stories and animations. We can use event and action command blocks in order to make sprites carry out acts when certain prompts take place.

-We use algorithms (a set of instructions to p erform a task) to sequence movements, actions and sounds in order to program effective animations.

The Basics of Scratch

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

-Scratch helps us to learn how to use programming language, whilst also being creative and using problem-solving skills.



There are three main areas in Scratch:

-The Blocks Palette (on the left) contain all of the different blocks: puzzle piece commands which control the animation.

-Code Area (in the middle) is where the blocks are placed to create a program.

-Stage with Sprite (right) is where the output of the program is presented. The sprite is the character.

Adding/Removing Sprites: This can be done here, at the bottom of the stage. There are many sprites to choose from.

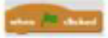
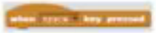
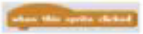
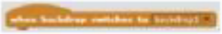
Attributes: There are three attributes of the sprite which we can change to make our animation: Code, Costumes, Sounds.

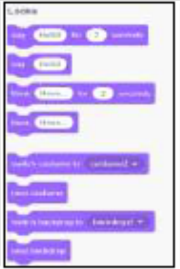
-**Backdrops**: Backdrops can be added by clicking on this icon (bottom right of the screen, below the stage).

Event and Action Blocks

-**Event Blocks:** Event blocks are coloured yellow and are used to sense different events that happen, e.g. the green flag being clicked, when a key is pressed, or when a sprite is pressed. They are needed for every project.

-**Action Blocks:** Action blocks include 'Motion' blocks (coloured blue), 'Sound' blocks (pink) and 'Looks' blocks (purple). They make the sprite move, make sounds and change appearance when the event is triggered.

 when clicked	Putting this at the beginning of your sprites programming will mean the program will only start when the green flag is clicked.
 when space key pressed	This will begin a sprites actions when the selected key from the drop down bar is pressed. The drop down bar allows you to select any key on the computer keyboard. Meaning you can control your sprites movements by setting up movements in connection with different keys.
 when this sprite clicked	A sprites actions will start when you click on it. For example: If that picture was the correct answer to a question you could get it to move.
 when backdrop switches to backdrop1	This will start a sprites action when the back drop changes. Using the drop down bar you can select any backdrop you have in the selected backdrop library for your program. For more info on backdrops see page on creating and selecting backdrops.



Sequencing and Algorithms

-A **sequence** is a pattern or process in which one thing follows another. In Scratch, blocks can stack vertically on top of one another to create sequences.

-Designing an **algorithm** (set of instructions for performing a task) will help you to program the sequence that you require.

-**Programming** is when we move the blocks into the position (based on our algorithm design). Programming uses a code that the computer can understand.

Trialling and Debugging

-Programmers do not put their computer programs straight to work. They **trial** them first to find any errors:

-Sequence errors: An instruction in the sequence is wrong or in the wrong place.

-Keying errors: Typing in the wrong code.

-Logical errors: Mistakes in plan/thinking.

-If your algorithm does not work correctly the first time, remember to **debug** it.

Important Vocabulary

Programming Scratch Blocks Commands Code Events Motion Sequence Trialling Debugging