

Subject: Computing	Year: 6 – Spring 1 – <b>Programming A – Variables in Games</b>
<p><b>National Curriculum objectives</b></p> <ul style="list-style-type: none"> <li>• Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts;</li> <li>• Use sequence, selection, and repetition in programs; work with variables and various forms of input and output;</li> <li>• Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs;</li> <li>• 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.</li> </ul>	
<p><b>To begin this unit, the children should have already learnt:</b></p> <p><u>Year 1 &amp; 2</u> Programming is when we make a set of instructions for computers to follow. Robots, such as floor robots like Bee-bots, are one type of machine that can follow programs. We can use algorithms (a set of guidelines to perform a task) to program floor robots along routes and correct ‘debug’ mistakes in algorithms.</p> <p><i>ScratchJr</i> is a programming application.</p> <p><u>Year 3</u> <i>ScratchJr</i> is a programming environment with three main areas: The Blocks Palette; Code Area; and the Stage with Sprite. <i>ScratchJr</i> can be used to create sequences using sounds.</p> <p><u>Year 4</u> Instead of typing in the code to create each individual step, we can save time by repeating a sequence of instructions. We use the ‘repeat’ function and create ‘infinite’ or ‘count-controlled’ loops.</p> <p><u>Year 5</u> Microcontrollers control real-life objects (like LEDs and motors ) through the construction of programs. Conditions are a means of controlling the flow of actions in a program.</p>	<p><b>The learning in this unit will prepare the children to learn these things in the future:</b></p> <p>National Curriculum Objectives at KS3:</p> <ul style="list-style-type: none"> <li>• Design, use and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems;</li> <li>• Understand several key algorithms that reflect computational thinking [for example, ones for sorting and searching]; use logical reasoning to compare the utility of alternative algorithms for the same problem;</li> <li>• Understand the hardware and software components that make up computer systems, and how they communicate with one another and with other systems;</li> <li>• Undertake creative projects that involve selecting, using, and combining multiple applications, preferably across a range of devices, to achieve challenging goals, including collecting and analysing data and meeting the needs of known users;</li> <li>• Understand a range of ways to use technology safely, respectfully, responsibly and securely, including protecting their online identity and privacy; recognise inappropriate content, contact and conduct and know how to report concerns.</li> </ul>
<p><b><u>Key Enquiry Question</u></b></p> <p>What examples of variables can you identify? What do you predict will happen when you change this (variable)? Why does a variable need a name and value? Can you explain the role of the variables in your project? What features of your partner’s project is particularly successful and what could be improved?</p>	<p><b><u>The Big Idea:</u></b></p> <p>A variable is something that is changeable. A variable can be set and changed throughout the running of a program. Programmers will apply the Use-Modify-Create model: learners will experiment with variables in an existing project, then modify them, before they create their own project.</p>

**To achieve ARE, pupils will need to be secure in the following knowledge:**

**By the end of this unit, children will know:**

- A 'variable' is something that is changeable;
- examples of information that is variable, for example, a football score during a match;
- A variable can be used in a program, e.g. 'score';
- A program variable as a placeholder in memory for a single value;
- A variable has a name and a value;
- The value of a variable can be used by a program;
- The value of a variable can be updated;
- Variables can hold numbers (integers) or letters (strings);
- That a variable can be set as a constant (fixed value);
- The importance of setting up a variable at the start of a program (initialisation);
- Why, if you change the value of a variable, you cannot access the previous value (cannot undo);
- The name of a variable needs to be unique.

**Vocabulary:**

Programmed; algorithm; button; direction; forward; backward; robot; left; right; route; design; chunking; error; debugging (introduced in KS1).

Scratch; blocks; commands; code; sprite; stage; costume; backdrop; debugging (introduced in Y3).

Logo; codes; infinite loop; count-controlled loop (introduced in Y4).

Programming; circuit; condition-controlled loop; LED (introduced in Y5).

**Motor; modify.**

**By the end of this unit, children will be able to do:**

- Identify a variable in an existing program;
- Experiment with the value of an existing variable;
- Choose a name that identifies the role of a variable to make it easier for humans to understand it;
- Decide where in a program to place a variable;
- Update a variable with a user input (and use an event in a program to update a variable);
- Use a variable in a conditional statement to control the flow of a program;
- Use the same variable in more than one location in a program.

**Useful Resources:**

*Scratch* software.

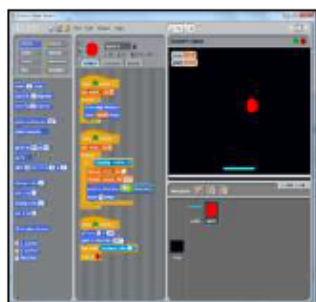
**Online training courses**

- [Get Started Teaching Computing in Primary Schools: Preparing to Teach 5- to 11-Year-Olds](#)
- [Programming Pedagogy in Primary Schools: Developing Computing Teaching](#)
- [Teaching Programming to 5- to 11-Year-Olds](#)



### Overview

#### Variables in Games



- Programming is when we make and input a set of instructions for computers to follow.
- Variables are changeable elements of a program. Scratch is one app in which we can explore variables.
- We use algorithms which we can plan, model, trial and debug, in order to create accurate command sequences, that enable variables to be enacted in games.

### Basic Variables

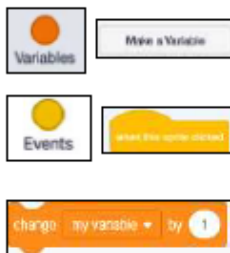
- **Variables:** A variable is something that is changeable. A variable can be set and changed throughout the running of a program.

In computer programming we use variables to store information that might change and can be used later in our program. E.g. in a game a variable could be the current score of the player; we would add 1 to the variable whenever the player gained a point.






































#### Making Variables in Scratch – The Basics


- Select 'Variables' (dark orange circle) from the menu on the left. Either choose from the available variables or 'Make A Variable.'
- Select 'Events' (light orange circle) from the menu on the left. Choose what needs to happen for the variable to change. E.g. 'When this sprite clicked' or 'when space key pressed.'
- Select 'Variables' again from the menu on the left. Choose what will happen when the event happens, e.g. 'change score by 1' (to add a point) or 'change score by -1' to remove a point.




### More Complex Variables

- Variables should always have a value and an appropriate name.
- **Adding Callouts:** Select 'Looks' from the menu on the left. Add it to the variable program. Edit the text to change the callout.              
- **Adding Motion:** Many games require sprites to change position. This is achieved using the 'Motion' commands. Select 'Motion' from the menu on the left. Choose from the available motion commands.           
- **Adding Sound:** Many games require sprites to change position. This is achieved using the 'Sound' commands. Select 'Sound' from the menu on the left. Choose from the available motion commands.         
- **Adding Comments:** Comments are a good way of showing that you understand what your code is doing. Right click on the block that you want to comment on, and add in your comment. 

### Sequencing and Algorithms

- A **sequence** is a pattern or process in which one thing follows another. 
- We design **algorithms** (sets of instructions for performing a task) to help us program sequences involving multiple output devices (e.g. LEDs and motors).
- **Programming** is the process of keying in the code recognized by the computer into the software (using your algorithm).

### 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

Variable

Scratch

Events

Code

LED

Algorithm

Motor

Modify

Debugging