Year 5 – Programming A – Selection in physical computing

Unit introduction

In this unit, learners will use physical computing to explore the concept of selection in programming through the use of the Crumble programming environment. Learners will be introduced to a microcontroller (Crumble controller) and learn how to connect and program components (including output devices — LEDs and motors) through the application of their existing programming knowledge. Learners will be introduced to conditions as a means of controlling the flow of actions, and explore how these can be used in algorithms and programs through the use of an input device (push switch). Learners will make use of their knowledge of repetition and conditions when introduced to the concept of selection (through the 'if... then...' structure) and write algorithms and programs that utilise this concept. To conclude the unit, learners will design and make a working model of a fairground carousel that will incorporate their understanding of how the microcontroller and its components are connected, and how selection can be used to control the operation of the model. Throughout this unit, pupils will apply the stages of programming design.

There are two Year 5 programming units:

- Programming A – Selection in physical computing
- Programming B – Selection in quizzes

This is unit A, which should be delivered before unit B.

Lesson	Brief overview	Learning objectives and Success Criteria
1 Connecting Crumbles	In this lesson, learners will become familiar with the Crumble controller, some of its associated components, and the programming environment used to control it. They will explore how the items connect together to create a complete circuit, and how to construct programs that turn an LED on and off and set its colour. Learners will apply their understanding of repetition by identifying how their programs can be modified to make an LED flash continuously. https://teachcomputing.org/curriculum/key-stage-2/programming-a-selection-in-physical-computing/lesson-1-connecting-crumbles	To control a simple circuit connected to a computer • I can build a simple circuit to connect a microcontroller to a computer • I can program a microcontroller to light an LED • I can explain why I used an infinite loop
2 Combining output devices	In this lesson, learners will develop their knowledge of a Crumble controller further by connecting additional devices (another Sparkle and a motor) to the controller, and they will construct programs to control more than one of these. They will design sequences of actions for these output devices. They will then apply their understanding of repetition by using count-controlled loops when implementing their design as a program. https://teachcomputing.org/curriculum/key-stage-2/programming-a-selection-in-physical-computing/lesson-2-combining-output-devices	To write a program that includes count-controlled loops • I can connect more than one output device to a microcontroller • I can design sequences for given output devices • I can decide which output devices I control with a count-controlled loop
3 Controlling with conditions	In this lesson, learners will be introduced to conditions, and how they can be used in algorithms and programs to control their flow. They will identify conditions in statements, stating if they are true or false, and learn how they can be used to start and stop a set of actions. Learners will be introduced to a Crumble switch, and learn how it can provide the Crumble controller with an input that can be used as a condition. They will explore how to write programs that use an input as a condition, and use this knowledge to write a program that uses a condition to stop a repeating light pattern. https://teachcomputing.org/curriculum/key-stage-2/programming-a-selection-in-physical-computing/lesson-3-controlling-with-conditions	To explain that a loop can stop when a condition is met, eg number of times • I can explain that a condition is something that can be either true or false (eg whether a value is more than 10, or whether a button has been pressed) • I can experiment with a 'do until' loop • I can program a microcontroller to respond to an input
4 Starting with selection	In this lesson, learners will develop their understanding of how the flow of actions in algorithms and programs can be controlled by conditions. They will be introduced to selection, and learn to represent conditions and actions using the 'if… then…' structure. They will apply their understanding by using selection in an algorithm created to meet the requirements of a task. They will discover that infinite repetition is required when programming input devices to repeatedly check if a condition has been met. https://teachcomputing.org/curriculum/key-stage-2/programming-a-selection-in-physical-computing/lesson-4-starting-with-selection	To conclude that a loop can be used to repeatedly check whether a condition has been met • I can explain that a condition being met can start an action • I can identify a condition and an action in my project • I can use selection (an 'if… then…' statement) to direct the flow of a program
5 Drawing designs	In this lesson, learners will apply their understanding of microcontrollers, output devices, and selection when designing a project to meet the requirements of a given task. To ensure their understanding, they will identify how selection might be used in real-world situations, then they will consider how they can apply this knowledge when designing their project. They will produce detailed drawings to show how their model will be made and how they will connect the microcontroller to its components.	To design a physical project that includes selection • I can identify a condition to start an action (real world) • I can describe what my project will do (the task) • I can create a detailed drawing of my project

	https://teachcomputing.org/curriculum/key-stage-2/programming-a-selection-in-physical-computing/lesson-5-drawing-designs	
6 Writing and testing algorithms	In this final lesson of the unit, learners will build on the designs that they developed in Lesson 5 by creating an algorithm to meet the requirements of the given task. They will identify how they are going to use selection before writing their algorithm. They will then move into the code level to test their algorithm by implementing it as a program, running it, identifying any bugs, and returning to the algorithm to debug it where necessary. Finally, to conclude the unit, they will evaluate their algorithms and other areas of their designs. https://teachcomputing.org/curriculum/key-stage-2/programming-a-selection-in-physical-computing/lesson-6-writing-and-testing-algorithms	To create a controllable system that includes selection • I can write an algorithm to control lights and a motor • I can use selection to produce an intended outcome • I can test and debug my project

Subject knowledge

This unit focuses on physical computing that allows learners to control real-life events through the construction of programs. When learners undertake physical computing, they write programs that control real-world objects, like LEDs and motors, using a computer. The tangible effect of seeing the commands that they entered into a computer being carried out on a physical item, rather than on screen, can be highly motivational for learners. Physical computing also offers the opportunity to take a more project-based approach to learning, and allows learners to make choices about the purpose, design, and program of their product.

Throughout this unit, there are opportunities to demonstrate a concept within the Crumble programming software or play a video. Pedagogically, it is more beneficial to demonstrate the concepts to learners, as it allows for easier questioning and understanding. We recommend that you use the videos to see what to demonstrate, then show learners with a live demonstration, however, videos are provided on the slides if you wish to use them instead.

For this unit, you will need experience of constructing programs using the Crumble programming software (see the ‘Resources’ section at the end of this document). It uses the same drag-and-drop style as Scratch. You will need to write programs that turn LEDs (Sparkles) on and off, change LED colours, spin motors, use push switches as inputs, and combine a number of these peripherals. Additionally, you will be connecting the Crumble controller with battery packs, Sparkles, motors, and push switches. For further support on using Crumbles, see the Crumble ‘Getting Started’ guide at redfernelectronics.co.uk/crumble-getting-started.