Linked learning theme: Year 5 – Spring 1

Computing – Programming A – Selection in physical computing

Prior Learning

- Learners will use their prior knowledge of block coding to support their learning and development of skills during this unit of learning. In particular, children will reflect on their learning from the Year 4 programming A and B units which focus on repetition, counted- and infinite loops. They will be encouraged to use what they learned in Years 1 and 2 also, revisiting the Moving a robot and the Robot algorithms units.

Core knowledge

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 it to control components (including output devices — LEDs and motors). Learners will be introduced to conditions as a means of controlling the flow of actions in a program. 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 demonstrate 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, learners will apply the stages of programming design.

Key skills

- Program a Crumble controller.
- Use infinite loops to achieve a purpose.
- Use sequencing for a purpose.
- Use conditions to structure a program.
- Design and plan for the effects of a unique project.
- Write and debug an algorithm for a functioning project.

Vocabulary

Crumble Controller
Program
Infinite loop
Circuit
LED
Motor
Count-controlled loops
Conditions
Algorithm
Debug

Learning Outcomes

- To control a simple circuit connected to a computer.
- To write a program that includes count-controlled loops.
- To explain that a loop can stop when a condition is met.
- To explain that a loop can be used to repeatedly check whether a condition has been met.
- To design a physical project that includes selection.
- To create a program that controls a physical computing project.