@digitallcharity

digit<all>

Enriching your curriculum through the lens of climate action and sustainability

amazon future engineer

About Digit<all>

- A registered UK charity
- EU Code Week & Robotics Week hub for the UK
- Provides a focus on contextualised resources
- Experts in engaging girls in CS Education
- Utilises a group of volunteers and ambassadors from across the UK to support schools
- In the last two years the charity has reached over 1M young people
- The charity is driven by three amazing young women in technology
- Training and resources are developed by practicing teachers

Why climate as a context?

Pupils develop a responsibility and passion for the world around them

Helps pupils to understand scientific and computational issues in a cross-curriucular manner

Pupils become more resilient and understand the importance of mitigating the climate and environment crisis

Physical computing is a great way of helping pupils visualise and contextualise such issues with hands-on approaches

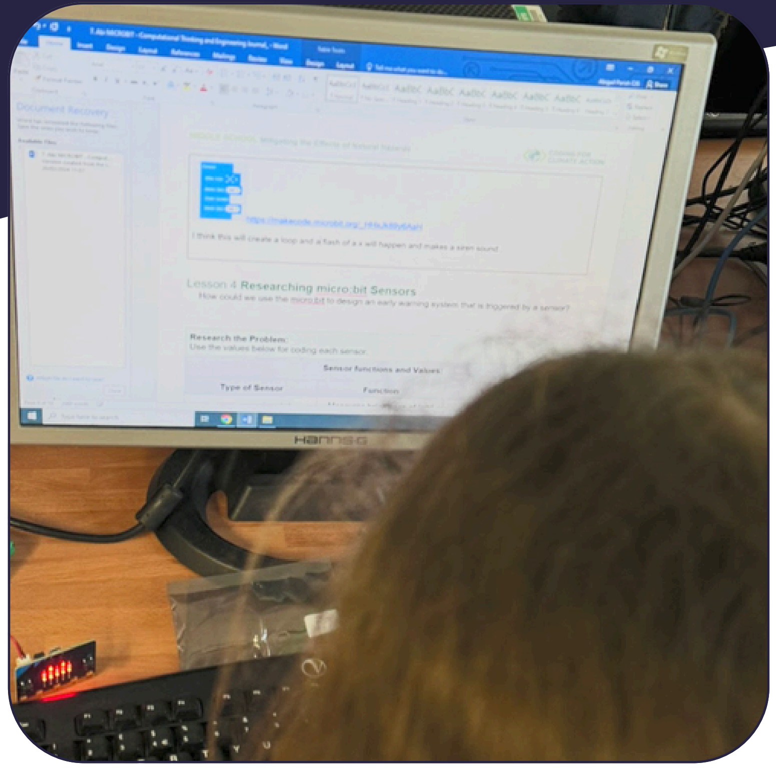
Contact with nature can improve the well being of pupils and develop their confidence, motivation and social skills (Children and Research Group)

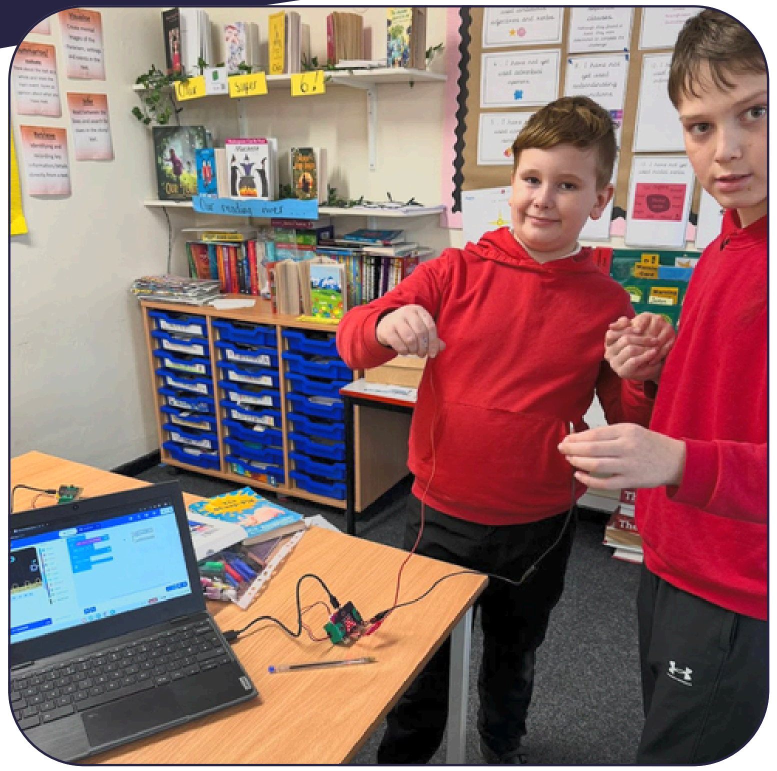
Introducing children to the diversity of nature can inspire them to pursue STEM subjects and careers

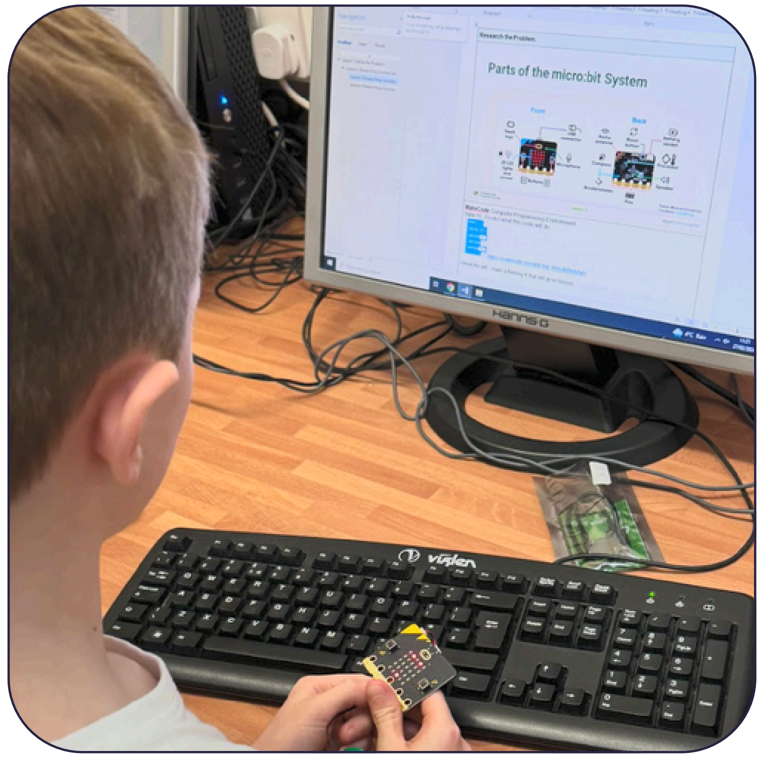
Coding for Climate Action

Overview

- A collaboation between Digit<all> and Amazon Future Engineer
- 6 lesson scheme of work for KS2 (and one for KS2)
- Mapped to the National & TeachComputing Curriculum
- Encourages students to code early warning systems with the micro:bit
- Helps pupils understand how natural hazards can be mititaged thorough the power of technology
- Helps pupils to work collaboatively and actively to tackle environmental challenges



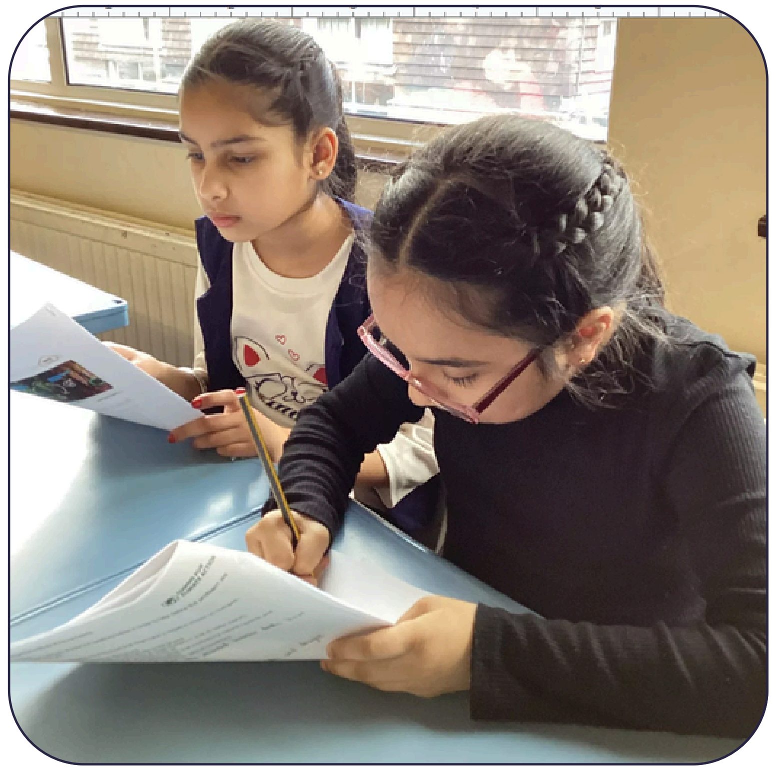


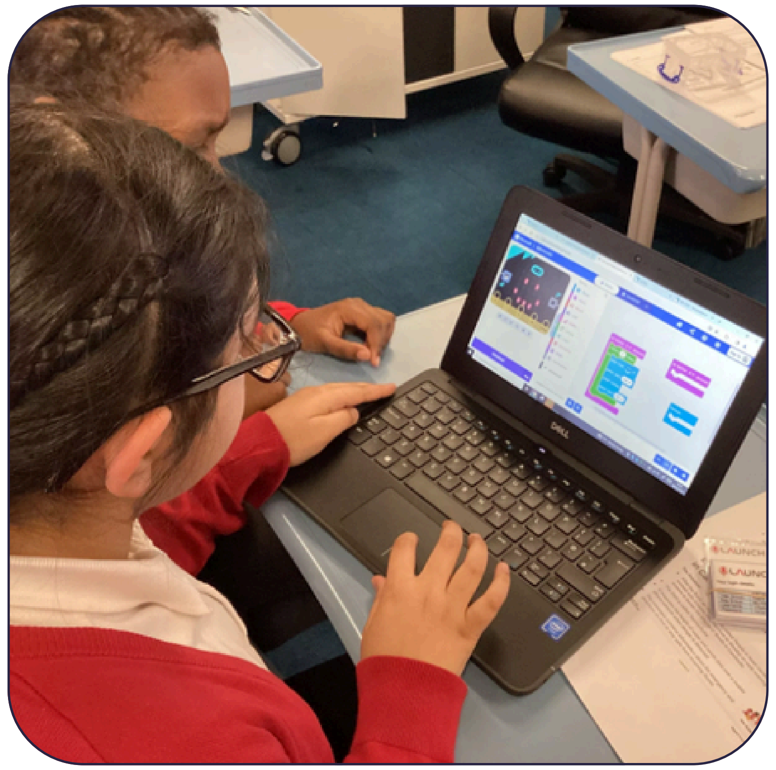


Coding for Climate Action

digit<all>

amazon future engineer





Coding for Climate Action

digit<all>

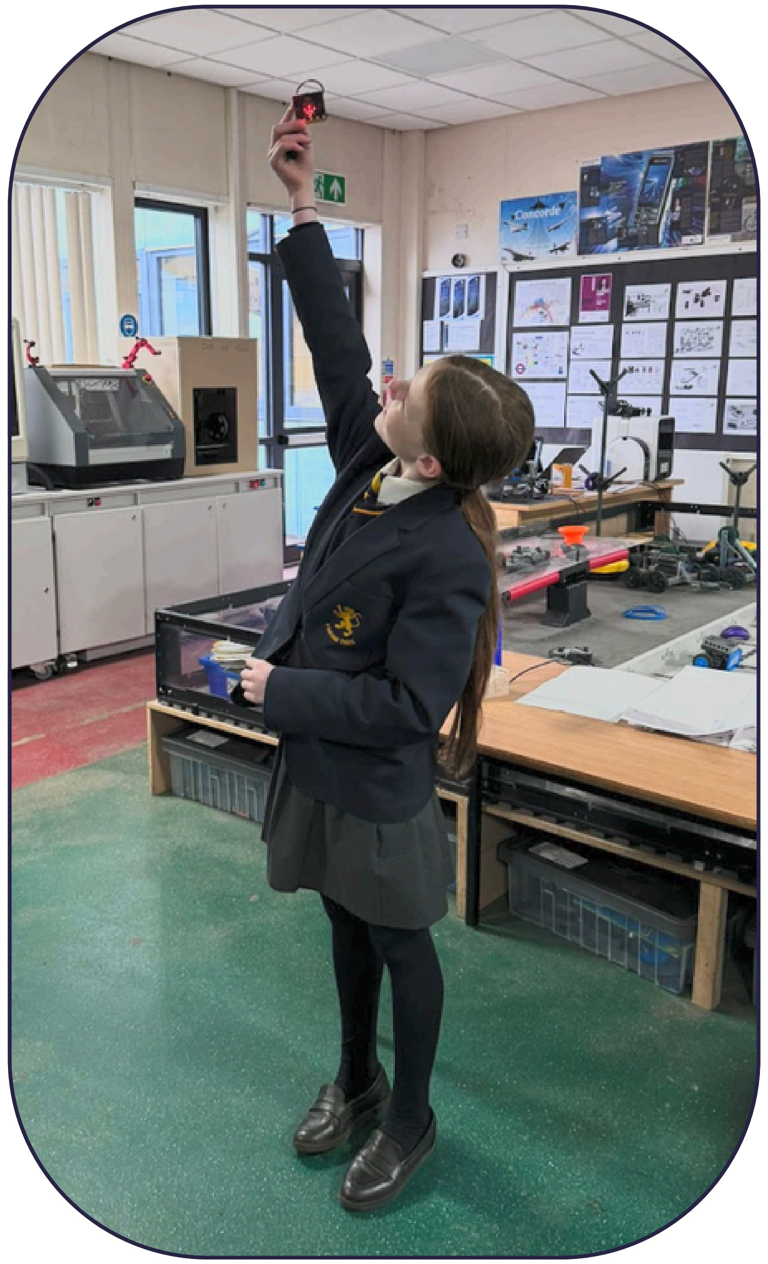
amazon future engineer

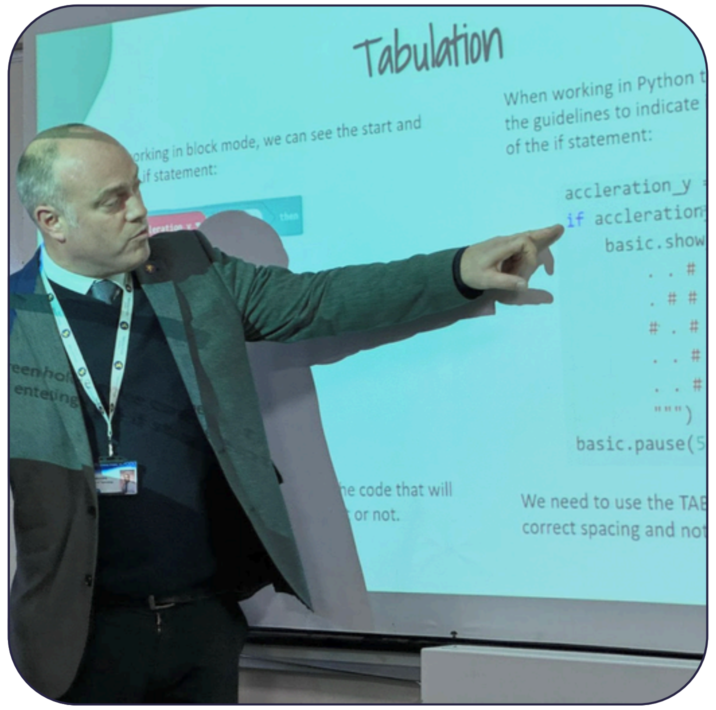
Overview

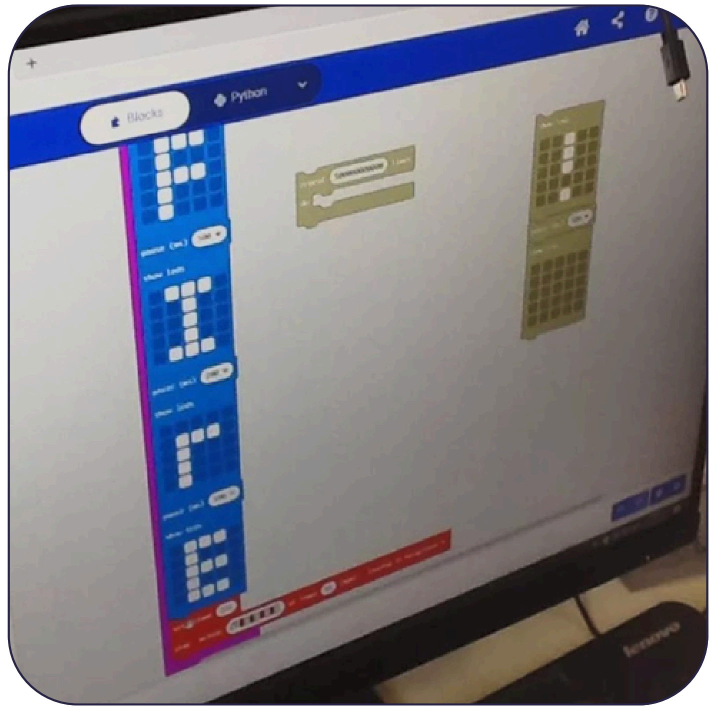
- Lesson 1: Defining the problem
- Lesson 2: Analysing existing solutions
- Lesson 3: Building and EWS
- Lesson 4: Introducing sensors
- Lesson 5: Researching radio and pins
- Lesson 6: Earthwake early warning system

Complete with:

- Lesson slides with weaved PRIMM approaches
- Lesson plans
- NC and TCC mapping
- Activity journal for evidence capture
- Online recorded and live training resources and solutions



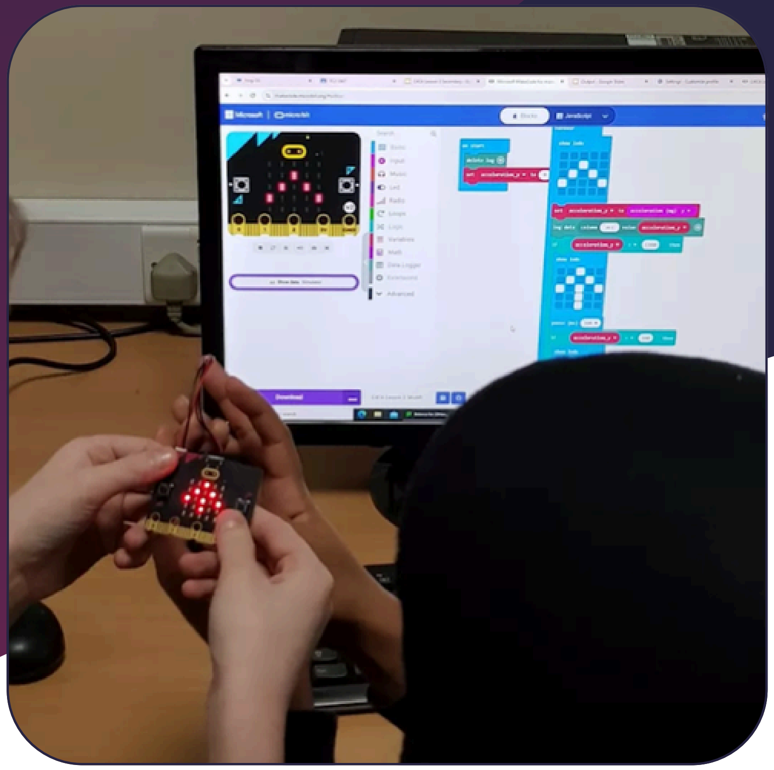
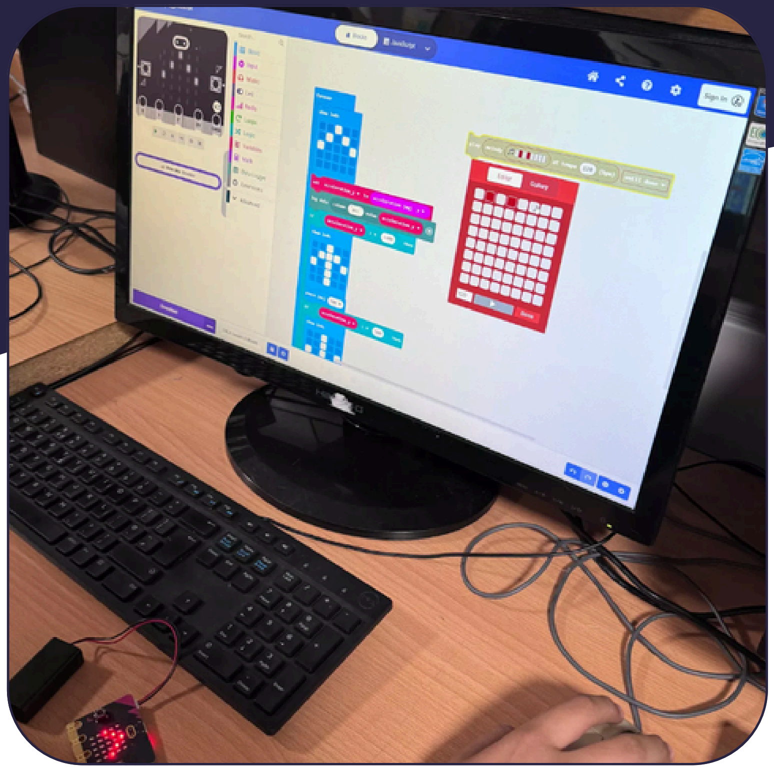


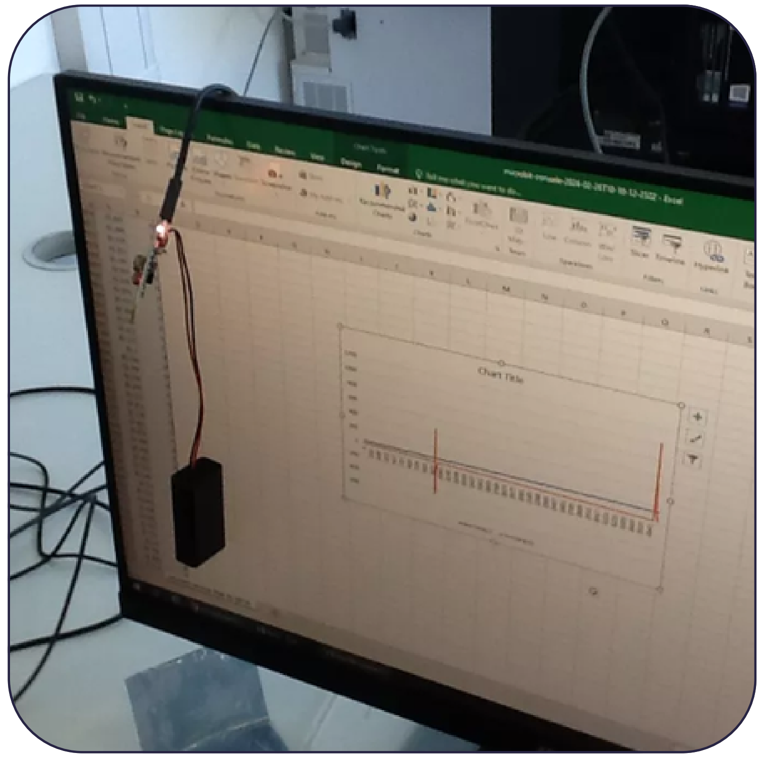
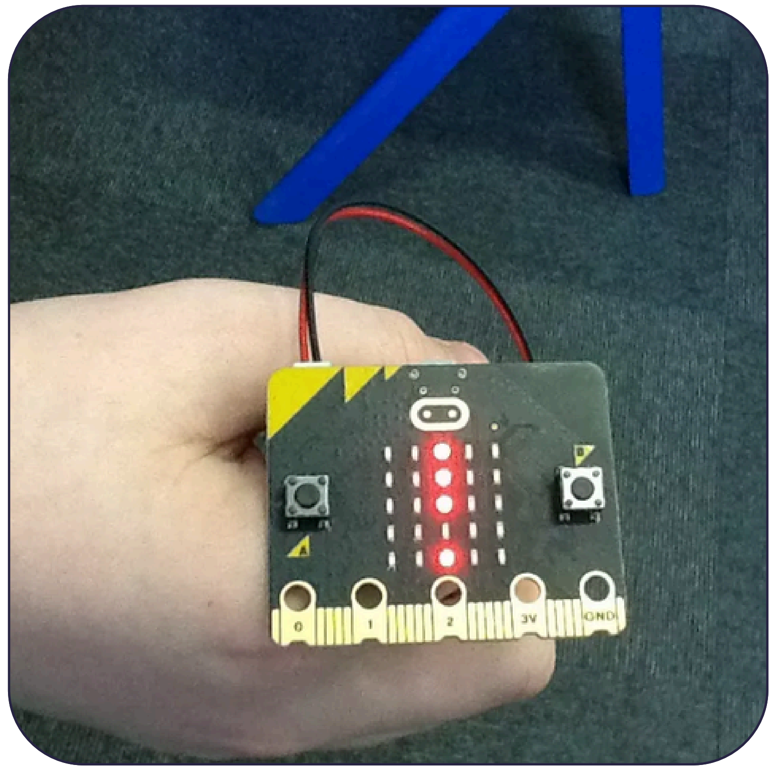


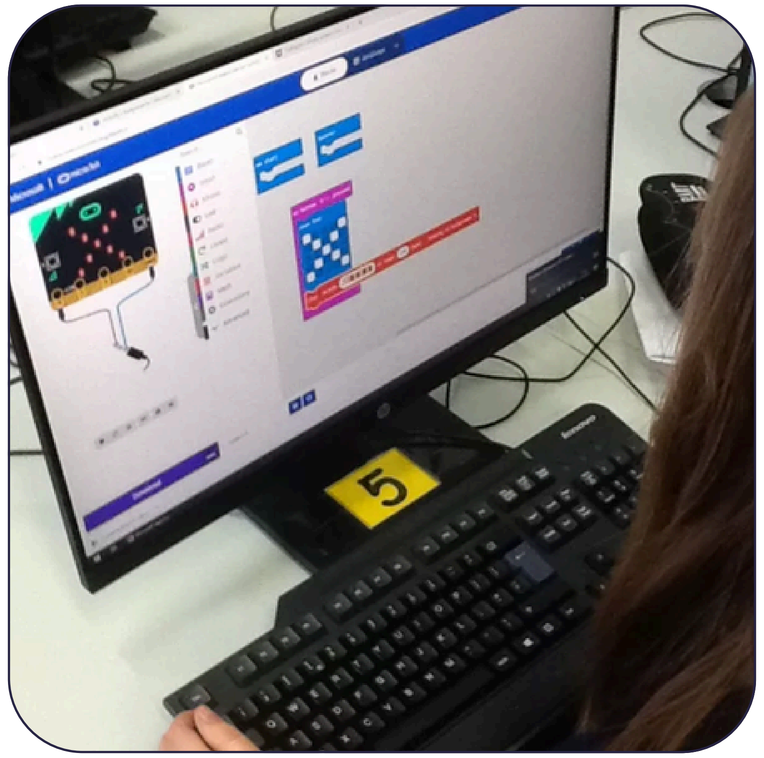
Coding for Climate Action

digit<all>

amazon future engineer




Beyond the curriculum

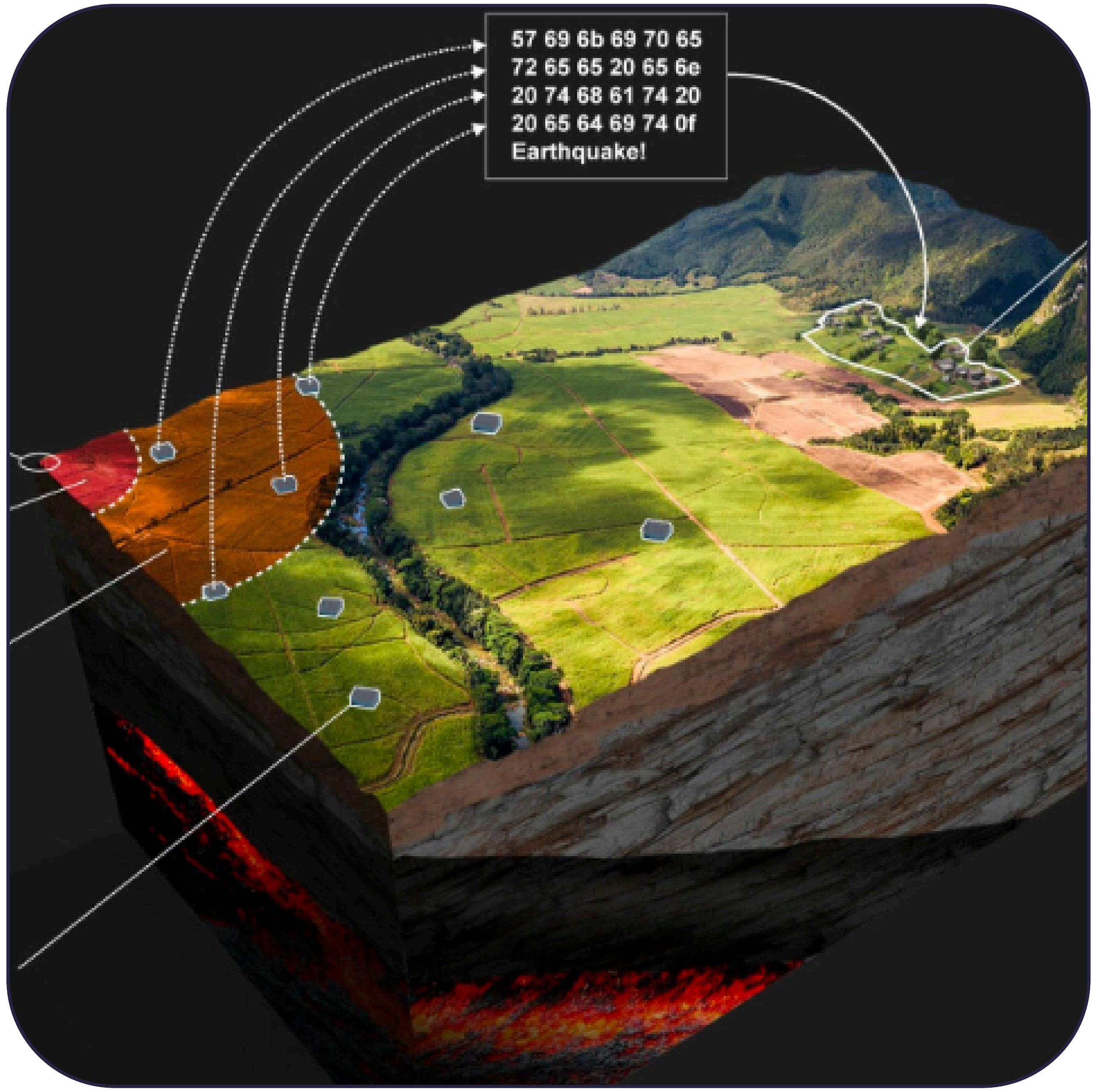
digit<all> amazon future engineer

Careers links

- Learning from careers and labour market information
- Addressing the needs of each student
- Linking curriculum learning to careers

Skills ready

- Problem solving
- Creativity
- Staying positive
- Aiming high
- Leadership
- Teamwork



Programming competencies developed - primary

Programming skills

- Explain what an infinite loop does
- Design sequences that use count-controlled loops
- Design sequences that use conditional loops
- Explain that a condition is either true or false
- Use selection and direct the flow of a program
- Modify a condition of a program
- Use selection in an infinite loop
- Show that a condition can direct program flow

Control structures

- Create a simple circuit and connect to a computer
- Program a microcontroller to make an LED switch on
- Connet more than one output component to a microcontroller
- Program a microcontroller to respond to input
- Transfer program to a controllable device
- Experiment with different physical inputs

Algorithms

- Identify conditions in a program
- Identify the condition and outcomes of an if..else statement
- Explain that program flow can branch according to a condition

Programming competencies developed - secondary

Programming skills

- Use an IDE to write and execute a Python program
- Locate and connect common syntax errors
- Walkthrough a sequence and sketch the state and output
- Use variables and counters
- Use Boolean values as flags
- Arrange program statements in a sequence
- Use selection to control programe flow
- Combine iteration and selection
- Use functions and parameters

Control structures

- Arrange program statements in a sequence
- Use selection to control the flow of program execution
- Combine iteration and selection

Data types and structures

- Use input and output
- User operators and data types
- Use if else and relational operators
- Use if elif statements
- Use while loops
- Use the logical operators AND and NOT

Linking phenomenon and problem

Lesson standards alignment

Unit problem: How can we design Early Warning Systems to monitor more than one exception to improve its reliability?

Phenomenon: Natural hazards are increasing due to climate change but Early Warning Systems are helping protect people and saving lives.

Science and engineering practices	Disciplinary core ideas (mapped to PoS)	Crosscutting concepts
Asking questions and defining problems Students will reflect on the reliability and effectiveness of their prototypes. They will extend the functionality to measure acceleration change in two directions.	**3-3** Use two or more programming languages, at least one of which is textual, to solve a variety of computational problems; make appropriate use of data structures [for example, lists, tables or arrays]; design and develop modular programs that use procedures or functions **3-4** Understand simple Boolean logic [for example, AND, OR and NOT] and some of its uses in circuits and programming; understand how numbers can be represented in binary, and be able to carry out simple operations on binary numbers [for example, binary addition, and conversion between binary and decimal] **3-6** Understand how instructions are stored and executed within a computer system; understand how data of various types (including text, sounds and pictures) can be represented and manipulated digitally, in the form of binary digits **3-7** 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 **3-8** Create, re-use, revise and re-purpose digital artefacts for a given audience, with attention to trustworthiness, design and usability	**Cause and effect** Cause and effect relationships are routinely identified, tested, and used to explain change.

Each lesson plan provides a link to both the discipline of Computing and Science education

Lesson Number	Lesson Name	Lesson Objectives
1	Define the problem	• I can identify similarities and difference between natural hazards • I can identify the relationship between heat and evaporation • I can describe how Early Warning Systems can be used to mitigate the effect of natural hazards
2	Analysing Existing Solutions	• I can describe the differences between the parts of a system and the functions of those parts. • I can research the part and functions of three different existing technologies • I can explain the part and functions of three existing technologies
3	Build an Early Warning System	• I can identify the differences between hardware and software. • I can identify input, output and repetition in code. • I can modify and complete code that utilises input, output and repetition and variables
4	Introducing Sensors	• I can identify the sensors on a Micro:bit. • I can design and modify programs using sensors. • I can modify and complete code that utilises input, output, count controlled loops and and variables
5	Researching Micro:bit Pins	• I can describe how a simple electrical circuit works. • I can describe how a circuit can be controlled by a physical device. • I can use selection in a program to produce and intended outcome
6	Researching Micro:bit Radios	• I can describe that Micro:bits can send data to one another using radio signals. • I can read the code that is used to send data between Micro:bits. • I can use if/else statements in my code to produce one of two intended outcomes.
7	Earthquake EWS	• I can identify core programming constructs input, output, variables, repetition and selection. • I can write code to achieve a specific outcome. • I can read and predict the outcomes of code

Label	Year 5 Teach Computing Programming Units	Covered in Code for Climate Action
	Programming A – Selection in physical computing	
CS	Create a simple circuit and connect to a computer	Lesson 5
CS	Program a microcontroller to make an LED switch on	Lesson 3, Lesson 4, Lesson 5, Lesson 6, Lesson 7
CS	Connect more than one output component to a microcontroller	Lesson 5, Lesson 7
CS	Program a microcontroller to respond to an input	Lesson 3, Lesson 4, Lesson 5, Lesson 6, Lesson 7
PG	Explain what an infinite loop does.	Lesson 3, Lesson 5, Lesson 6, Lesson 7
PG	Design sequences that use count-controlled loops	Lesson 4
PG	Design a conditional loop	Not Covered
PG	Explain that a condition is either true or false	Lesson 5, Lesson 6, Lesson 7
PG	Use selection (an ' if...then..' statement) to direct the flow of a program	Lesson 5, Lesson 6, Lesson 7
PG	Use selection to produce an intended outcome	Lesson 5, Lesson 6, Lesson 7
DD	Describe what my project will do	Lesson 7
DD	Test and debug my project	Lesson 4, Lesson 5, Lesson 6, Lesson 7
	Programming B – Selection in quizzes	
AL	Identify conditions in a program	Lesson 4, Lesson 5, Lesson 6, Lesson 7
AL	Identify the condition and outcomes in an 'if... then... else...' statement	Lesson 5, Lesson 6, Lesson 7
AL	Explain that program flow can branch according to a condition	Lesson 5, Lesson 6, Lesson 7
PG	Modify a condition in a program	Lesson 4, Lesson 5, Lesson 6, Lesson 7
PG	Use selection in an infinite loop to check a condition	Lesson 5, Lesson 6, Lesson 7
PG	Show that a condition can direct program flow in one of two ways	Lesson 5, Lesson 6, Lesson 7
PG	Identify the outcome of user input in an algorithm	Lesson 3, Lesson 4, Lesson 5, Lesson 6
DD	Outline a given task	Lesson 7
DD	Test my program	Lesson 4, Lesson 5, Lesson 6, Lesson 7
DD	Identify ways the program could be improved	Lesson 3, Lesson 4

Mapping to computinational thinking

Computational thinking

- Abstraction: Students will use abstraction to make choices. They will decide on an additional tolerance level to inform output, by viewing data graphs.
- Decomposition: Students will break down challenges into constituent parts and use code samples to construct an improved solution.
- Algorithmic thinking: Students will explore how to extend a simple selection statement to include additional pathways through the use of if...elseif statements.
- Generalisation: Students will refer to their previous solution to identify refinements to improve the reliability of the solution using variables.
- Evaluation: Students will evaluate a solution upon testing to decide how they can make the solution more reusable.

(example from session 3 of Coding for Climate Action (Secondary)

Recognising championing schools

- Certificates for lead schools
- Badges for delivery partners, volunteers and teachers


digit<all>

Share your achievement on social media and tag us @digitallcharity

Coding For Climate Action

Lead School

PROUDLY PRESENTED TO

#1#

COMPLETED:

30/01/2024

Matt Hewlett
Community and Education Lead

Derek Peaple
Trustee, Chair

amazon future engineer

CODING FOR CLIMATE ACTION

digit<all> amazon future engineer

Key Climate for Coding Action features

Real world contexts

- Develops self-efficacy in girls
- Provide a 'hook' to engage in the activity
- 'Empower' students to collaborate
- Enable the cross-curricular link between Computing and the Science curriculum
- Provides context to prototyping and testing

Radio transmission

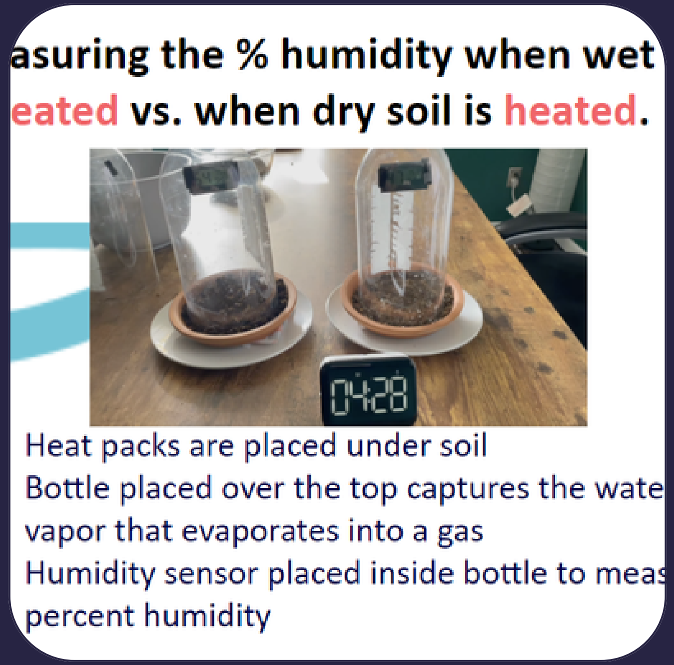
- Enables pupils to get active
- Promotes teamwork and collaboration
- Provides opportunities for pair programming approaches
- Develops computational thinking, especially decomposition and evaluation

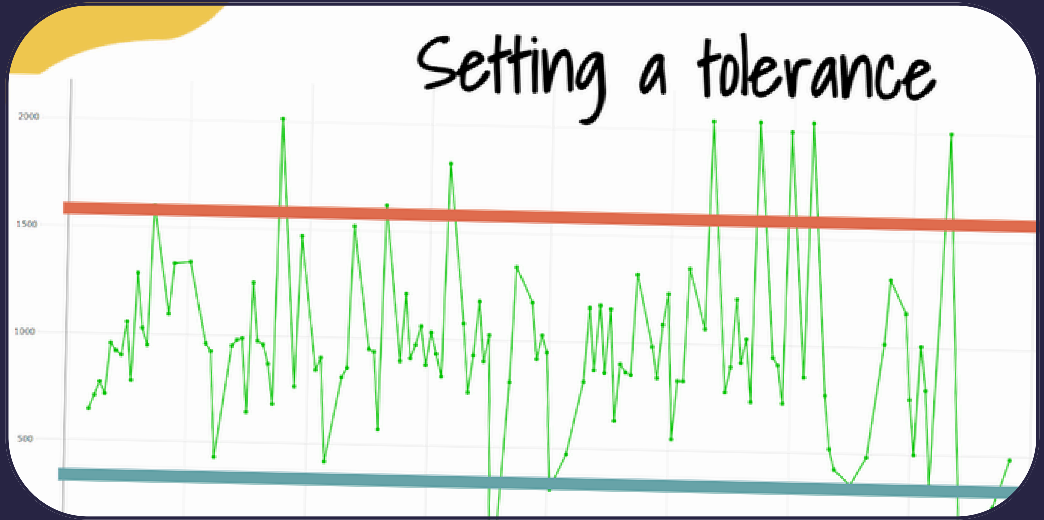
Data logging

- Provides a link from physical capture through to data analysis
- Data logged can be used to inform setting on tolerances
- Enables data comparisons with other pupils and groups

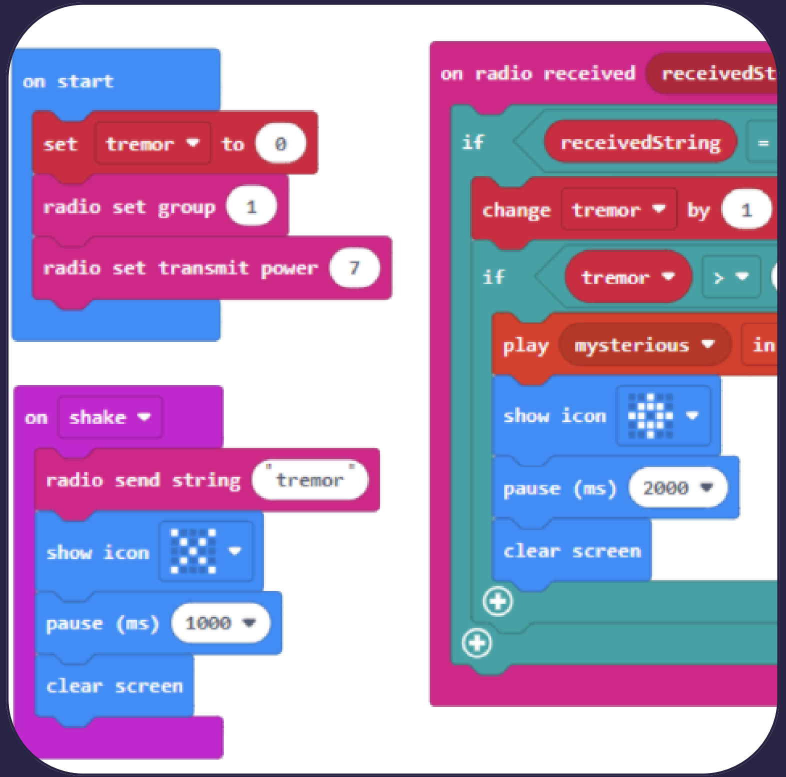
Resources

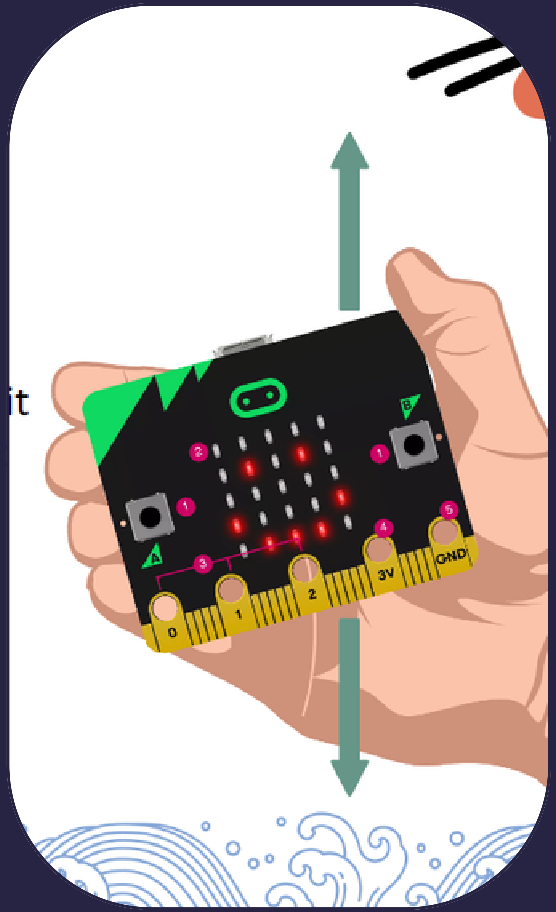
- Bridging block to text programming
- Consideration of accessibility
- Active use of the micro:bit
- Cross-curricular links with Science and Computing
- Supported by live and recorded training
- Hour of Code version






```
accleration_y = 0
if accleration_y >= 225
    basic.show_leds("""
        . . # . .
        . # # # .
        # . # . #
        . . # . .
        . . # . .
        """)
basic.pause(50)
```





Hour of Code

We've packaged the unit of work as two individual Hour of Code activities

This allows community leads and teachers to create a simple experience or introduction to the Coding for Climate Action units

Activities include the use of PRIMM and Fuller techniques as well as Parsons problems.

Primary:

- Focus on micro:bit sensors
- Build an EWS to detect earthquakes using the accelerometer
- Choose a challenge to create an EWS for landslides, forest fires or droughts

Secondary:

- Focus on micro:bit data logging
- Build an EWS to detect flooding through high waves
- Set tolerances and use radio to transmit messages between two micro:bits

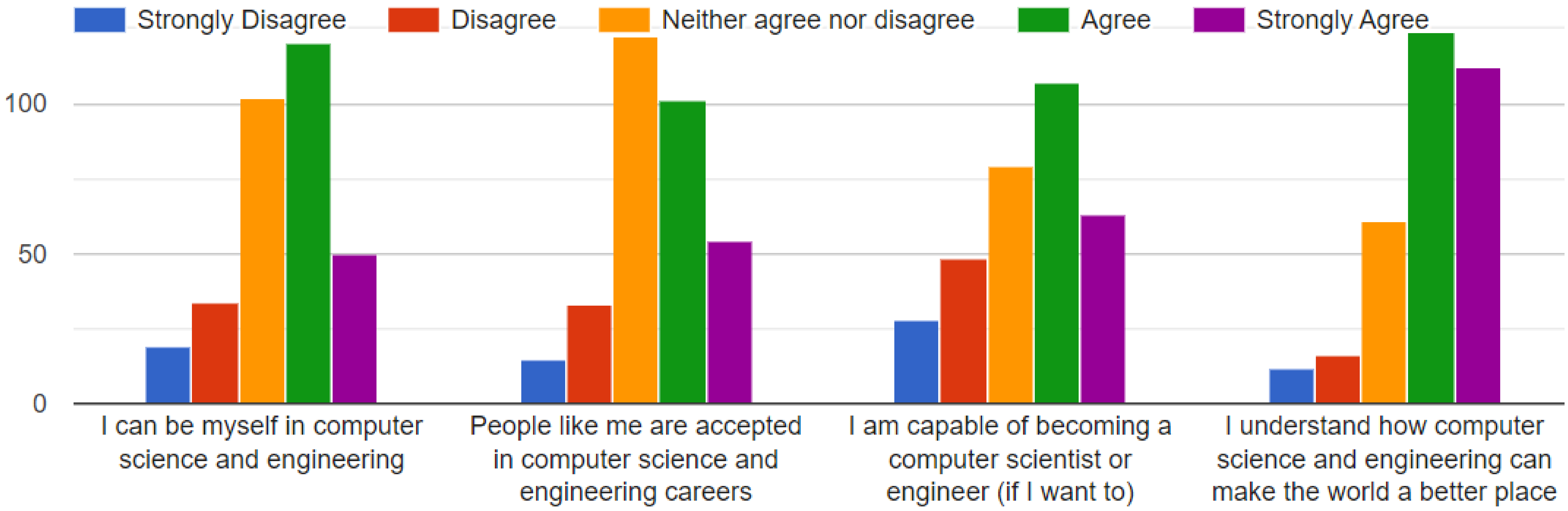
Coding for Climate Action

Recipe for success?

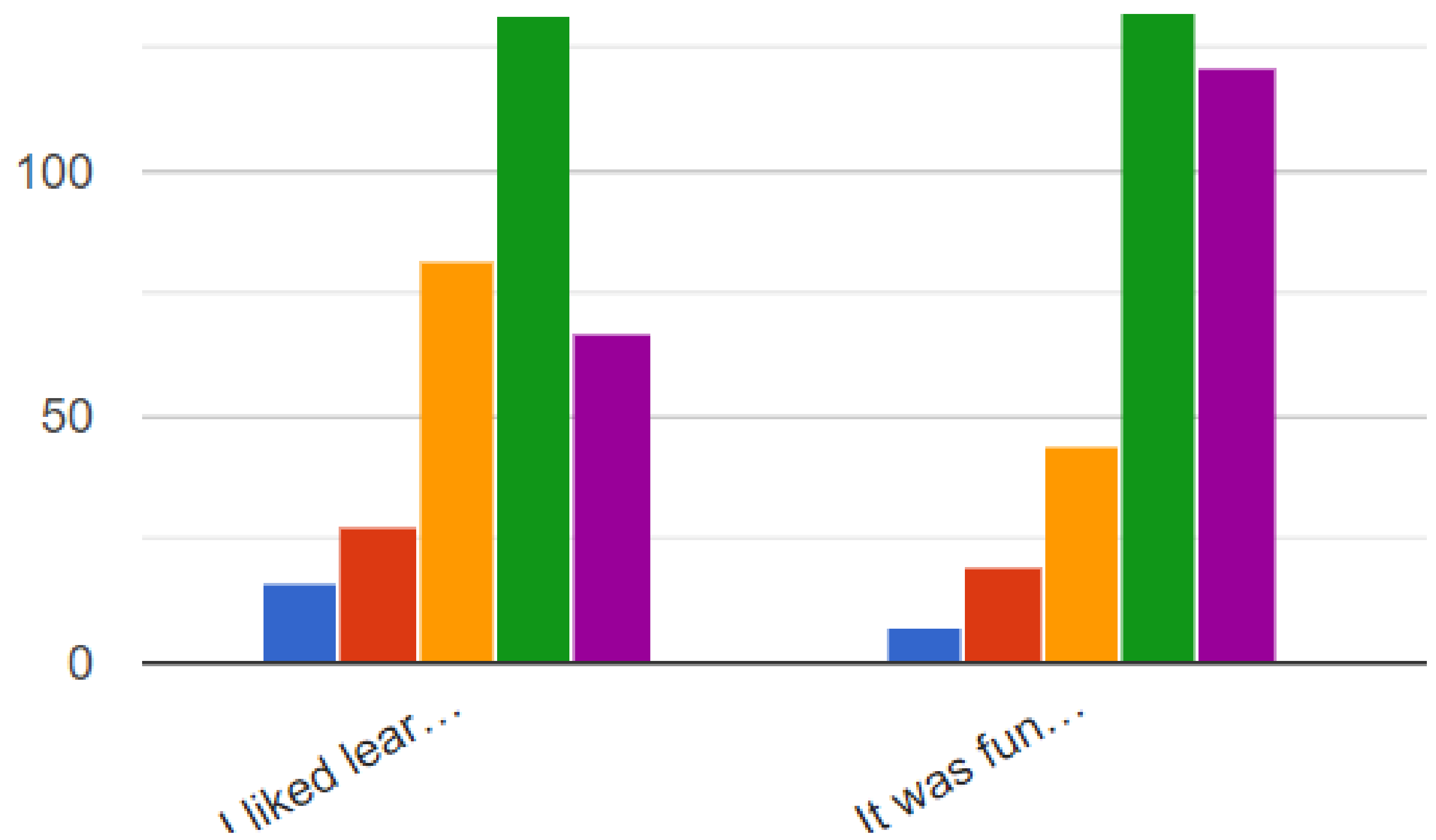
- Dynamic and adaptive resources
- Ensure resources developed with girls in mind
- Encourage activity and contextualisation
- Weave pedagogy approaches such as Fuller and PRIMM
- Computational thinking by stealth
- Pair programming and collaboration opportunities
- Project-based learning, programming, data, engineering and presentation
- Multiple models - units of work, community packages and Hour of Code
- Clear curriculum mapping
- Various training approaches

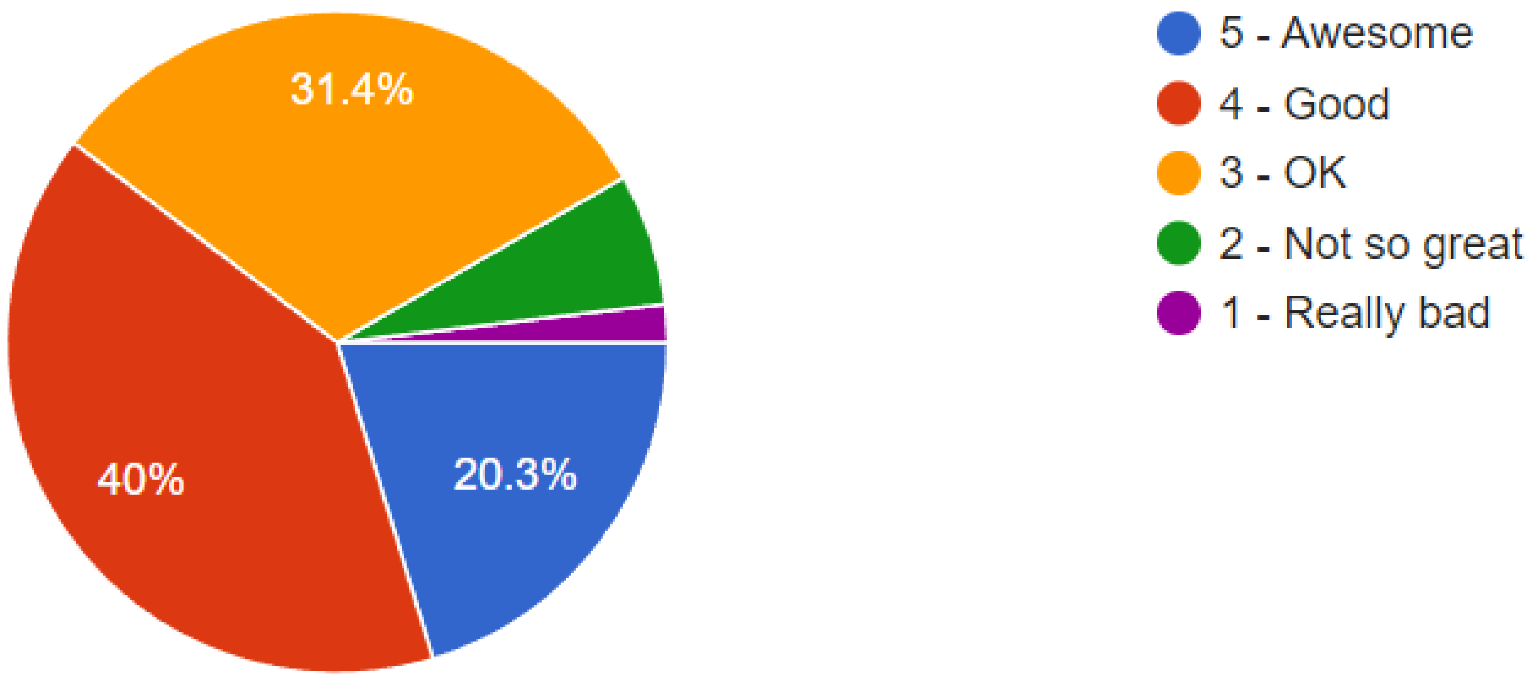
Prototyping and testing through colllaboration with the micro:bit

What the children say



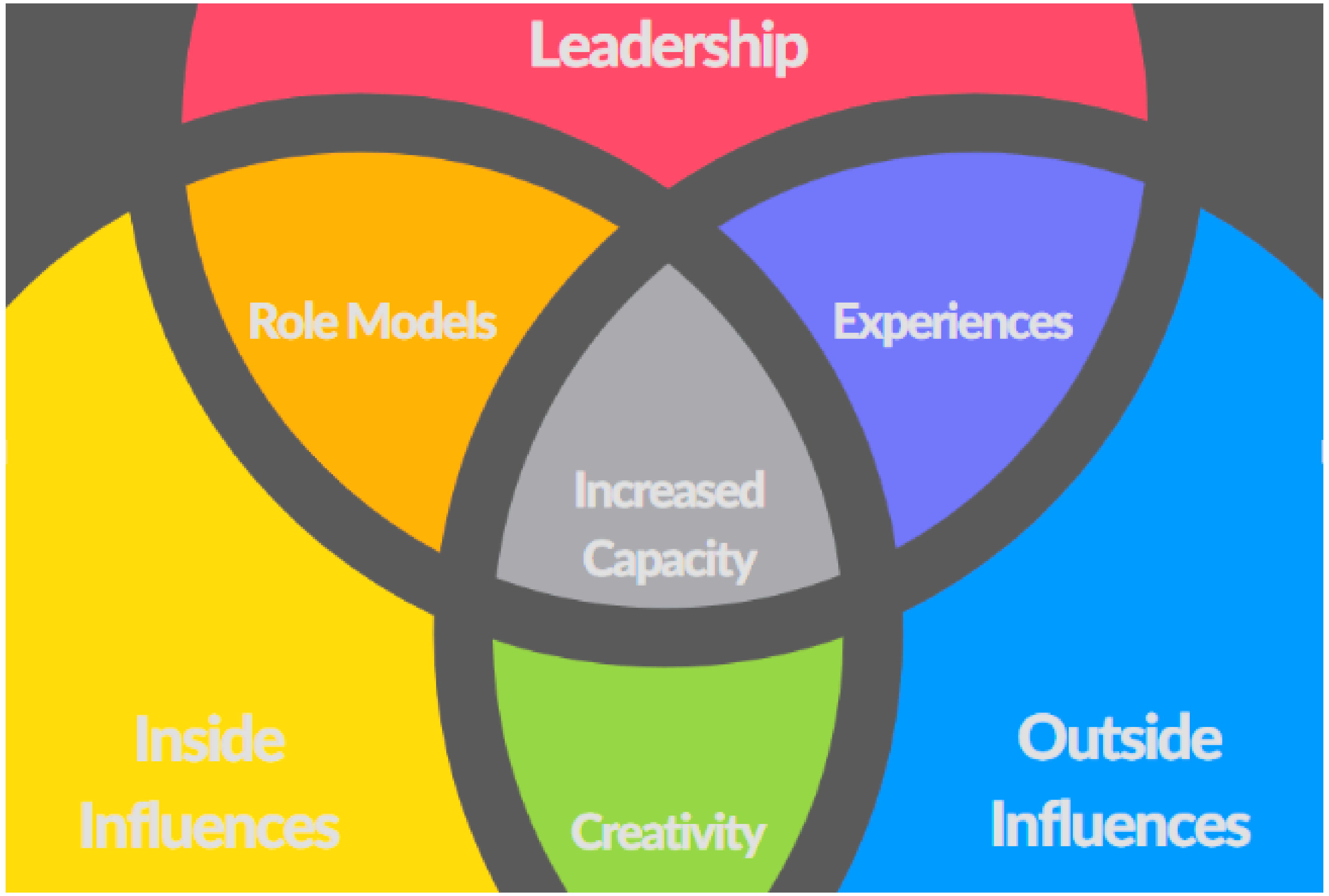
Rating the unit of work





"I liked learning computer science" and "It was fun to code with the micro:bit"

digit<all>



Resources developed to increase engagement of girls

- The resources develop self-efficacy of girls through real-wold contexts
- Girls have opportunity to collaborate, building EWS in pairs or in groups using radio functionality
- Strong focus on communication, prototyping and testing
- The ‘hook’ is provided through real world contexts including saving lives
- Opportunities for creative thinking and ‘open’ challenges, project-based learning
- Focus on choice and ownership