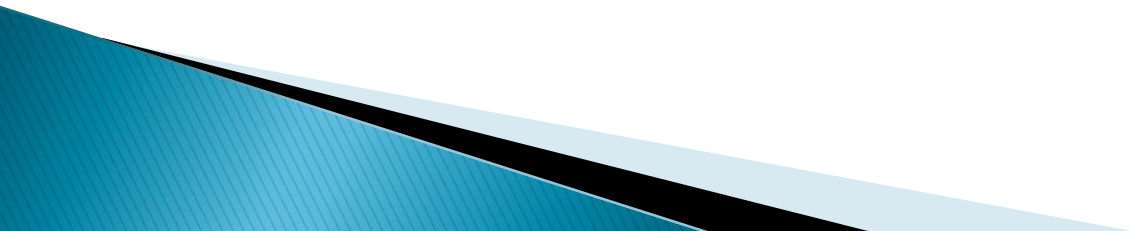




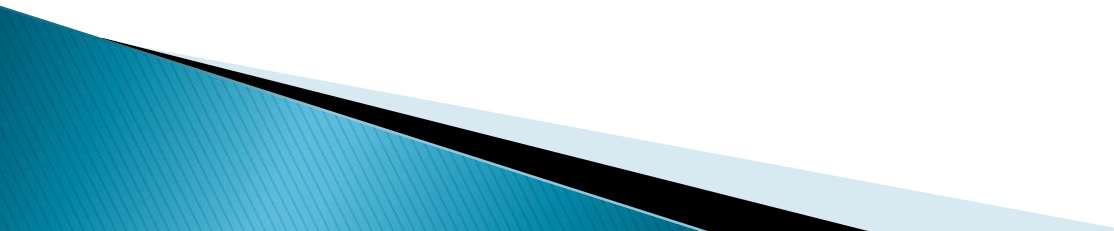
Coding With Robotics in the 21st Century

Steve Kellas
Ugborough Primary School, UK
ROBO21C: An ERASMUS+ Project

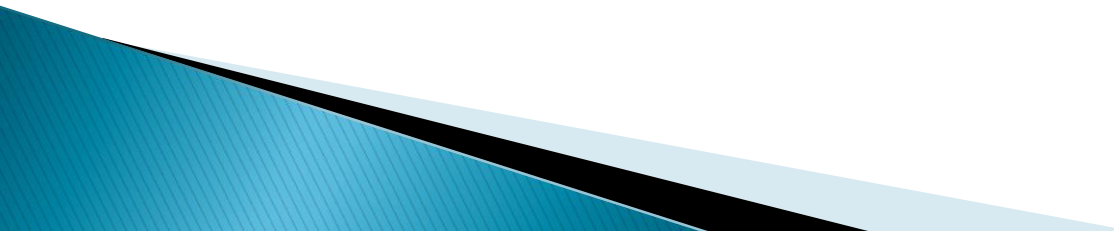
Coding is a 21st Century Skill



Talking Point 1:

- ▶ Q1. Have you ever created anything with code?
 - ▶ Q2. How are you coping with coding?
 - ▶ Q3. What has your most positive experience of teaching coding been so far?
- 

Coding Questions

- ▶ Q: Why coding?
 - ▶ Q: What is the language of coding?
 - ▶ Q: What are the key concepts to teach?
 - ▶ Q: What does progression in coding look like?
 - ▶ Q: What robots should we use and when should we use them?
 - ▶ Q: What are the ways we can teach children to code?
- 

Why coding?

- ▶ <https://youtu.be/nKlu9yen5nc>

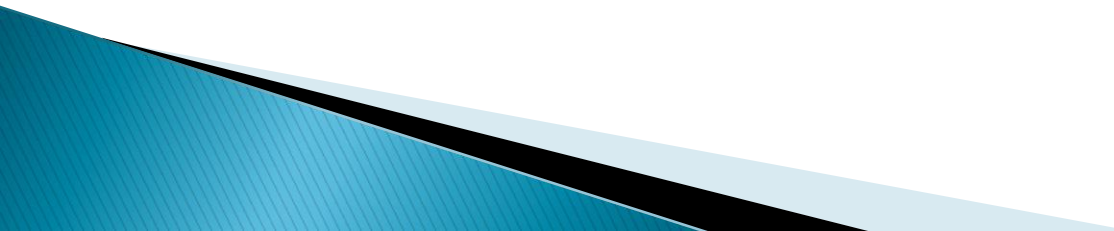
“Everybody in this country should learn how to program a computer... because it teaches you how to think.”

- Steve Jobs

The National Curriculum

- ▶ The programmes of study require that by the end of Key Stage Two pupils can:
- ▶ 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

In other words...

- ▶ Design to a brief
 - ▶ Be a problem solver
 - ▶ Learn to work logically
-
- ▶ But why robotics?
- 

Lego WeDo and the curriculum

- ▶ **Science:** Trace the transmission of motion and transfer of energy through the machine. Identify the simple machines mechanisms at work in the models, including levers, gears, and pulleys. Become familiar with complex motion using a cam, worm gear, and crown gear. Understand that friction can affect the movement of the model. Understand and discuss criteria for a fair test. Consider the needs of living animals. Understand that animals use parts of their bodies as tools. Compare natural systems with mechanical systems.
- ▶ **Design & Technology:** Program and create a working model. Interpret 2D and 3D illustrations and models. Use software media to acquire information. Demonstrate the knowledge and operation of digital tools and technological systems.
- ▶ Build, program, and test the models. Modify a model's behavior by changing the mechanical system or by adding a sensor to provide feedback. Brainstorm to find creative alternative solutions. Learn to share ideas and work together.
- ▶ **Mathematics:** Measure time in seconds and tenths of a second. Estimate and measure the distance in centimetres or inches. Understand the concept of randomness. Compare the diameter and rotational speed. Understand and use numbers to represent the type of sounds played and the amount of time the motor turns on. Understand and use the distance between an object and to activate the motion sensor. Understand how the position of the model is measured by the tilt sensor. Understand and use numbers to measure and score qualitative characteristics.
- ▶ **English:** Communicate in spoken or written forms using appropriate vocabulary. Prepare and deliver a demonstration using a model. Use interview questions to find out information and write a story. Write a script with a dialogue. Write a logical sequence of events and create a story including main characters and dramatizing with visual and sound effects. Use technology to create and communicate ideas. Participate as knowledgeable, reflective members of the group and class.

Coding is all about language

- ▶ Logical
- ▶ Unambiguous instructions
- ▶ Use of conceptual language (logic) to create physical (concrete) outcomes

Terminology

- ▶ **Input:** **Input** can take a variety of forms, from commands you enter from the keyboard to data from another **computer** or device. A device that feeds data into a **computer**, such as a keyboard or mouse, is called an **input** device. (v.)The act of entering data into a **computer**. (Source: [What is Input? Webopedia Definition](#); *accessed 31 August 2016*)

Terminology

- ▶ **Output:** Anything that comes out of a computer. Output can be meaningful information or gibberish, and it can appear in a variety of forms -- as binary numbers, as characters, as pictures, and as printed pages. Output devices include display screens, loudspeakers, and printers. (v.) To give out. For example, display screens output images, printers output print, and loudspeakers output sounds.
- ▶ *Source: [What is Output? Webopedia Definition](#); accessed 31 August 2016)*

Terminology

- ▶ **Debug:** this means ‘troubleshooting’
 - In order to succeed in any computing, children need to be able to debug their programs and to adapt the examples in order to bring new ideas to life
 - This means they have to do the work – not you!
 - The goal is for them to develop the skill (habit) of debugging (i.e. problem solving) and not relying on others to solve it for them.

The language of coding

- ▶ Primary age range – we use ‘blocks’ of code
 - Blocks are pre-programmed sequences that perform a specific action
- ▶ This means they work on ‘everyday’ language
- ▶ How do I get the robot to move from here to there?
 - Start, Turn the wheels X times at speed of 0.5, then Stop

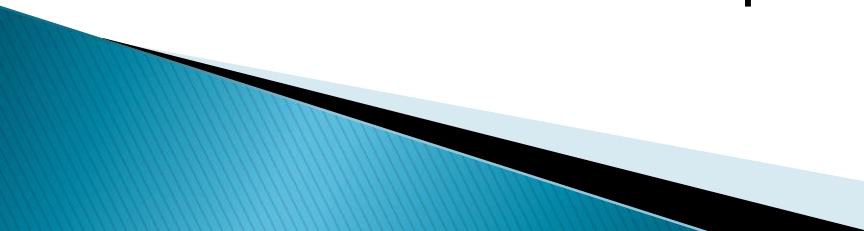
The language of coding (or logical reasoning)

- ▶ If / then / else
 - If X then do A, else do B
- ▶ And / or
 - Do A and B; Do A or B
- ▶ Repeat (loop) until / forever
 - Forever repeat this sequence (wait, check for movement, if movement, then close croc's mouth sequence, wait 10 seconds, then open croc's mouth sequence)
 - Repeat 10 times: When sensor touched, add 1 to score, when score =10 then end and display 'You Won!'.

Have you ever created something with code?

- ▶ Let's do it now!
- ▶ <https://studio.code.org/hoc/1>
- ▶ How many puzzles can you complete?
- ▶ What skills do these puzzles introduce?

What skills have you used in coding?

- ▶ Following instructions
 - ▶ Solving simple step problems (e.g. get from A to B)
 - ▶ Precise, unambiguous instructions
 - ▶ Create and debug simple programs
 - ▶ Predict behaviour using logical reasoning (e.g. if I follow these instructions where will I get to?)
 - ▶ Use sequence selection and repetition
 - ▶ Solve problems by decomposing into simple parts
 - ▶ Following and generating algorithms
 - ▶ Debugging
 - ▶ Programming, monitoring and controlling products
 - ▶ Use variables of input and output
- 

Let's try again!

- ▶ design, write and debug programs that accomplish specific goals,
 - including controlling or simulating physical systems; solve problems by decomposing them into smaller parts
- ▶ Create a pen and paper program that will simulate opening a set of automatic sliding doors (like those you find on a lift).

Key skills

▶ Sequence

- An algorithm – a series of steps that accomplish a task

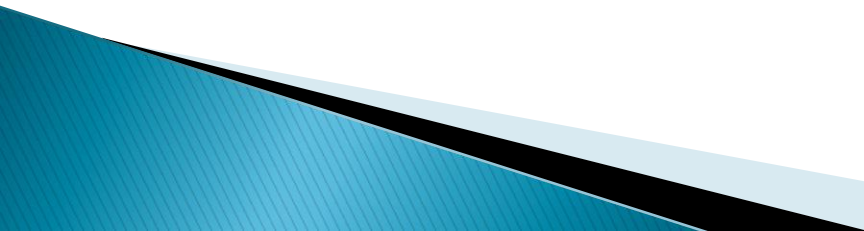
▶ Selection

- A question or choice where, depending on the answer, the robot (computer) takes a particular course of action (If sees 'red' then turn left or if sees 'blue' then turn right, else go straight.

▶ Repetition (loop)

- Repeats a step a set number of times (or duration), may be a 'wait' for some input or part of a selection or sequence (go forward 5 times, then stop and wait)

Key skill – decomposition

- ▶ Computers require clear, unambiguous commands in order to operate correctly
 - ▶ We must teach children the skill of breaking down a problem (Go from A to B) into smaller, more meaningful steps.
 - How does the robot move?
 - How many times must the wheel turn in order to move that far?
 - When should it stop?
 - When should it start?
 - ▶ This is logical thinking at its core
- 

Progression of skills

Beebot and Bluebot (R and KS1)

- ▶ Physical manipulation (assemble Lego and pressing buttons)
- ▶ Following instructions
- ▶ Solving simple step problems (e.g. get Bot from A to B)

Easily allows progression to:

- ▶ Precise, unambiguous instructions (get Little Jimmy's robot from A to B manoeuvring around obstacles)
- ▶ Create and debug simple programs (it didn't work Miss!)
- ▶ Predict behaviour using logical reasoning (e.g. if I follow these instructions where will Bluebot get to?)

Bluebot iPad App

- ▶ Key skills: Logical reasoning, sequence, repetition, create and debug

close

Explore Mode



Step by Step



Basic Programm...



Repeats



45 Degree Turns



A vertical toolbar containing 13 icons. From top to bottom: a green shield with a white up arrow, a green shield with a white up arrow, a green shield with a white up arrow, a green shield with a white right arrow, a green shield with a white up arrow, a green shield with a white up arrow, a green shield with a white up arrow, a green shield with a white right arrow, a green shield with a white up arrow, a green shield with a white up arrow, a green shield with a white up arrow, a green shield with a white up arrow, and a solid green square at the bottom.



A cluster of control buttons arranged in a grid. The top row has an orange up arrow. The middle row has an orange left arrow, a green circle with the text "GO", and an orange right arrow. The bottom row has a blue circle with a white "X", an orange down arrow, and a blue circle with a white pause symbol.

Device Bluetooth is switched off

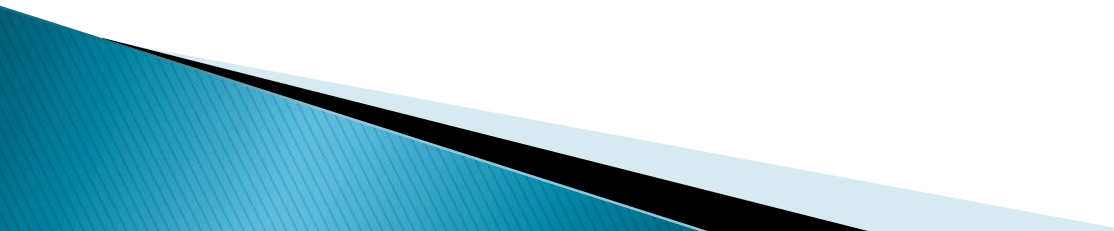


A control panel featuring a large orange button with a white upward-pointing arrow. To its right is a red volume control slider with a white plus sign at the top, a white minus sign at the bottom, and the number '3' in the center. A small red 'X' icon is positioned above the slider.

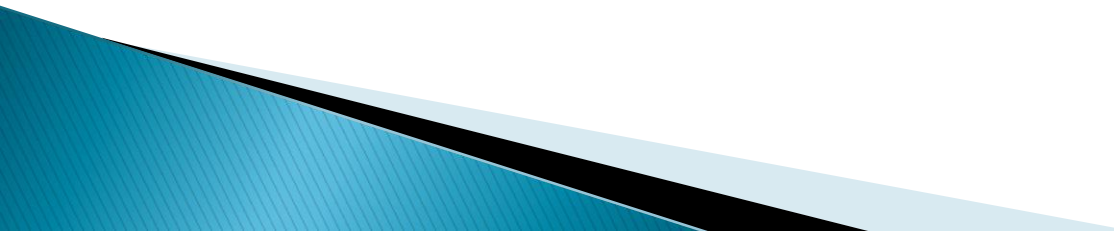


A set of seven circular control buttons arranged in a grid. The top row contains an orange button with a white upward arrow, a green button with the text 'GO', and an orange button with a white rightward arrow. The bottom row contains a blue button with a white 'X', an orange button with a white downward arrow, and a blue button with a white pause symbol.

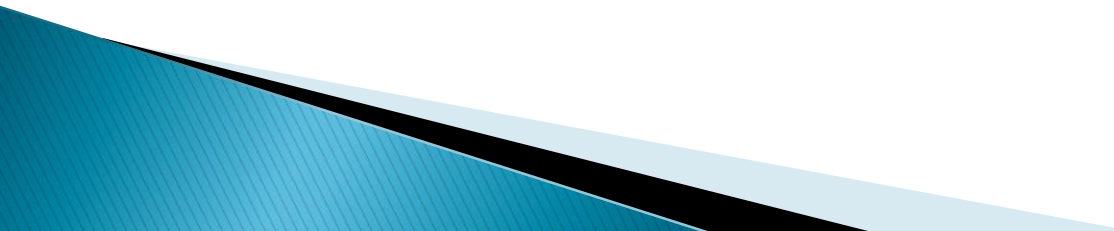
Lego WeDo

- ▶ Drag and drop sequencing
 - ▶ Repeats and Selection (loops and wait commands)
 - ▶ Input / Output: tilt sensor, motion sensor; output = behaviour, but we can change this
 - ▶ Teacher's guide
- 

Lego MindStorm

- ▶ Drag and drop blocks
 - ▶ Input sequences via laptop software or on the brick itself
 - ▶ Follow step by step, then experiment!
 - ▶ Debugging will follow experiments, but this is good coding practise.
- 

What is the best way to teach coding?

- ▶ Follow examples and adapt
 - ▶ Imagine, Plan, Design, Test, Iterate
 - ▶ Ad Hoc (try a small thing, test it, add to it)
 - ▶ All 3?
 - ▶ When would these be appropriate?
 - ▶ Language of coding should be embedded from an early age
 - ▶ Debugging must be encouraged from R upwards
- 

Next Steps

- ▶ Complete your Hour of Code
 - ▶ Read Teacher's Guide to WeDo and MindStorm
 - Many great ideas, lesson plans and activities already to go
 - ▶ Play with coding robots too! It's fun.
 - ▶ Be creative with how you use robots across the curriculum (they're not just for 'computing lessons')
- 