

Computing Knowledge Organiser

Year 1 & 2

Computer Science



Key Knowledge

Computer science will introduce children of all ages to understanding how computers and networks work. It is about using computational thinking to solve problems and make things for a purpose.

It will also give all children the opportunity to learn basic computer programming, from simple floor robots in Years 1 and 2, right up to creating on-screen computer games and programmes by Year 6

Key Learning

- Physically follow & give each other give each other forward, backward & turn (right-angle) instructions to move around
- Begin to identify an algorithm to achieve a specific purpose
- Begin to predict what will happen for a short sequence of instructions in a program
- Execute a program on a floor robot to achieve an algorithm
- Begin to predict what will happen for a short sequence of instructions in a program
- Create simple algorithms using Scratch to make a sprite move

Key Vocabulary

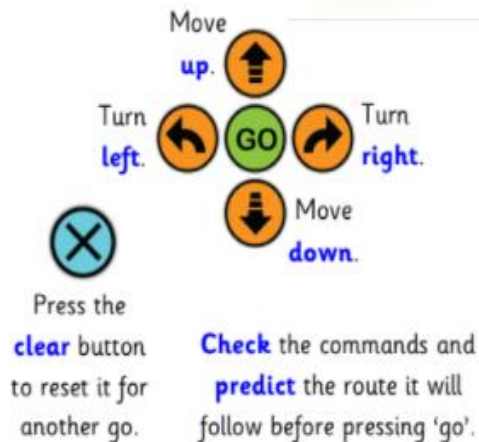
algorithm	sprite
debugging	floor robot
event	testing
execute	object

Key Skills

- I can use and follow instructions
- I can control a floor robot
- I can design a simple program
- I can write some code
- I can work out why it may not work

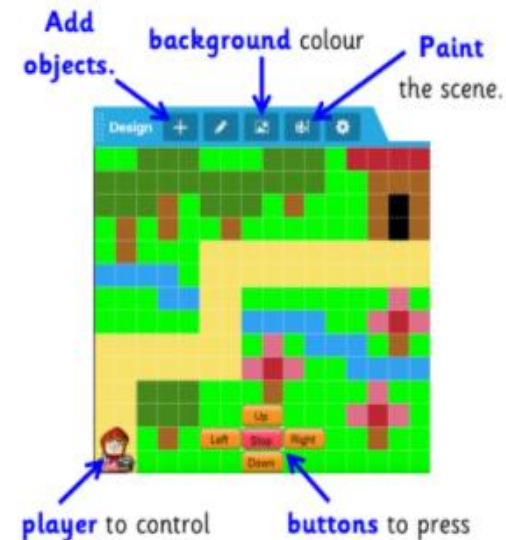
Floor Robots

You can **control** how a floor robot moves by giving it a **sequence of commands** to follow.



Designing a Program

The **user interface** is the graphics the user sees and interacts with.



Coding a Program



Computing Knowledge Organiser

Year 3 & 4

Computer science



Key Knowledge

Computer science will introduce children of all ages to understanding how computers and networks work. It is about using computational thinking to solve problems and make things for a purpose.

It will also give all children the opportunity to learn basic computer programming, from simple floor robots in Years 1 and 2, right up to creating on-screen computer games and programmes by Year 6

Key Learning

- Plan & enter a sequence of instructions on a robot specifying distance & turn to achieve specific outcomes, debug the sequence where necessary
- Create simple algorithms in Scratch to make sprites interact
- Create algorithms in Scratch to tell a story
- Use Repeat and If to determine how an algorithm will function
- Begin to correct errors (debug) as they program devices & actions on screen, & identify bugs in algorithms written by others

Key Vocabulary

algorithm	block	language
command	control	collaboration
debug	decomposition	encrypted
execute	input	output loops
manipulate	organise	program selection
repetition	scripted	sprite
sequence	simulation	packets of data
software	store	system
program	physical	
repetition	retrieve	
reverse engineer		

Key Skills

- I can write simple algorithms
- I can program an on screen turtle
- I know that one action can affect others
- I can debug a program when it does not work correctly

Writing Algorithms

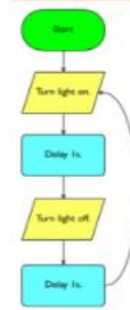
- An **algorithm** is a set of instructions to do a task, written in everyday language, in order.
- A **flowchart** shows how these steps are linked together in a sequence.
- A computer program precisely follows (**executes**) the steps of an algorithm.



Algorithm

- Switch the crossing light on.
- Wait 1 second.
- Switch the crossing light off.
- Wait 1 second.
- Go back to the first step.

Flowchart



Conditional Events (Selection)

Selection is a way of making a program automatically choose to run some code **when a specific condition is met**.

If _____ happens, then run this code: _____

If the rabbit touches the brown fence, then stop it.

If 30 seconds have passed, then show a baddy.

On-Screen Turtle Programming

A floor robot can be **simulated** on a computer using a screen turtle, programmed using similar commands.

Variables

Programs store **data that can change** (or vary) in a variable, like a score counter or a timer.

If the pirate gets the coin, then move the coin to a new place and add 10 points to the score.

If the pirate touches the snake, then reset the score to zero and stop the game.

Computing Knowledge Organiser

Year 5 & 6

Computer Science



Key Knowledge

Computer science will introduce children of all ages to understanding how computers and networks work. It is about using computational thinking to solve problems and make things for a purpose.

It will also give all children the opportunity to learn basic computer programming, from simple floor robots in Years 1 and 2, right up to creating on-screen computer games and programmes by Year 6.

Key Learning

- Record in some detail the steps (the algorithm) that are required to achieve an outcome & refer to this when programming
- Increase confidence in the process to plan, program, test & review a program
- Use variables and subroutines to create transferable code

Key Vocabulary

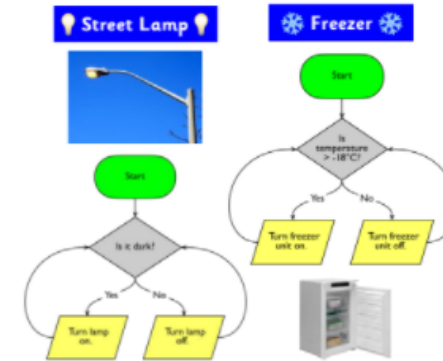
algorithm	block	language
command	control	collaboration
debug	decomposition	encrypted
execute	hardware	HTTP
input	IP address	output
loops	manipulate	organise
program	repetition	scripted
selection	sequence	simulation
sprite	software	store
packets of data	program	physical system
repetition	retrieve	

Key Skills

- I can use algorithms to create a simple program for a game
- I can debug a program to work out what went wrong
- I can use and evaluate programs I have made

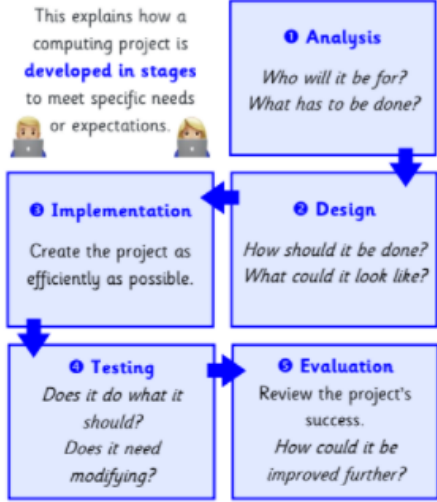
Flowcharts of Control Systems

Control systems constantly monitor **sensors** (input devices detecting changes in: light, temperature, movement etc.) and, using **conditional events**, decide when to respond by triggering **actuators** (output devices like: bulbs, motors, buzzers etc.).

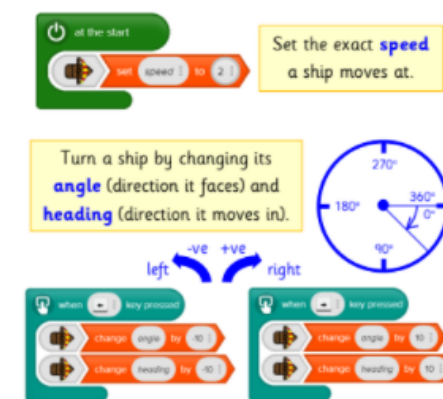


The Systems Lifecycle

This explains how a computing project is **developed in stages** to meet specific needs or expectations.

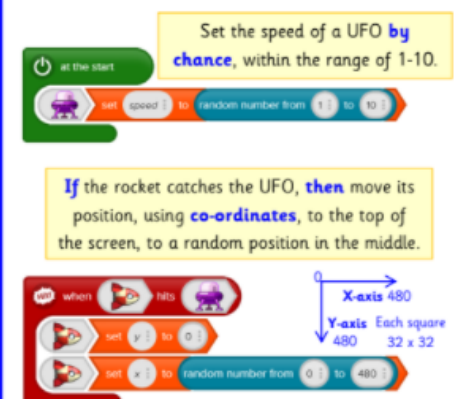


Programming with Numbers



Logical Reasoning Solving a problem carefully.
Tinkering Changing things to see what happens.

Random Numbers



Decomposition Breaking down into parts.
Abstraction Removing unnecessary detail.