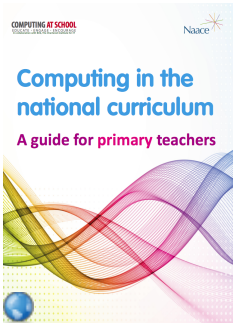
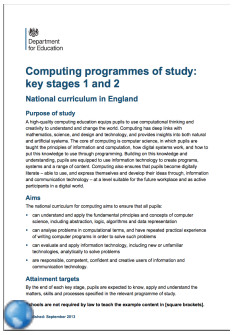
The Computer Study Aspect of the Computing Curriculum

A basic overview with suggested applications





Louise Wade

Computer Studies Aspect

Key stage 1

Pupils should be taught to:

understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions

create and debug simple programs

use logical reasoning to predict the behaviour of simple programs.

Key stage 2

Pupils should be taught to:

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

EYFS

- to be inquisitive
- to ask questions
- to explore for themselves
- to test things out
- to look for solutions

Learning to Follow Instructions

Action Games

Learning to Follow Directions

- Forward
- Backwards
- Stop
- Go
- Turn around

Program a Simply Floor Robot Bee-Bot

Move forwards and backwards towards a specific location e.g. On a number line, alphabet square, ect.

Making Choices on a Computer

LGfL Resources



Unplugged

Outside & Inside Play Areas

Remote Control Cars & Toys

iPad Apps

Duck Duck Moose

More Trucks

Block a Doodle

Walkie- Talkies & Phones

Busy Bundle1

Scratch Junior

Toys for the Home Corner

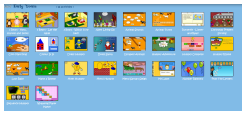
Digital Recording Equipment to Capture and Review Learning

LONDON GRID FOR LEARNING

EYFS

Turtle

Animate



26 activities

Follow the cloud
Music mixer
Monster scale
Monster grid
Colour choir
Colliding cogs
Crazy cogs
Falling wall
Puzzling paths
Talent show
Block-a-doodle-doo
Path peril
Helicopter rescue
Hen hunt
Road blocks
Tunnel trouble
Maze
Little Red Ridding Hood
The Gingerbread Man
Goldilocks and the Three Bears

KS1

Jun 21-18:43

Year One

Understand what algorithms are.

Record the instruction for making a cup of tea

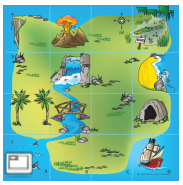
Algorithm

An **algorithm** is a precisely defined procedure – a **sequence** of instructions, or a set of rules, for performing a specific task (e.g. instructions for changing a wheel or making a cup of tea)

Bitesize

Sequencing an activity

Create simple programs.



Plan and record the sequence of instructions needed to follow a route.

Cross curricular resources

Use 2Go to Program an on Screen Turtle

Unplugged

CAS Barefoot supported by BT

Tinkering

Memory aids

Using the Apps to plan and program a sprite / character to complete a task.

Busy Bundle1

Block a Doodle

Kodable

Path Puzzler

Scratch Junior

Bee-Bot

LONDON GRID FOR LEARNING

Year One

Turtle

Animate

JIT

Visual

Look at lesson plans

Cheese Sniffer
Compass Points
Controlling Round a Route
Drawing with a Control Toy
Fly Catcher
Lily Hop
Drawing

Follow the cloud
Music mixer
Monster scale
Monster grid
Colour choir
Colliding cogs
Crazy cogs
Falling wall
Puzzling paths
Talent show
Block-a-doodle-doo
Path peril
Helicopter rescue
Hen hunt
Road blocks
Tunnel trouble
Maze
Little Red Ridding Hood
The Gingerbread Man
Goldilocks and the Three Bears

Year Two

Understand that algorithms are implemented as programs on digital devices.

Record 'How to...' instruction for digital devices.

Understand that programs execute by following precise and unambiguous instructions

Using values to program a Pro-Bot to move around a route.

Program using a complexsequence of instructions.

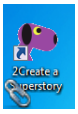
Record instructions.

Explore alternative programming sequences.

Write a simple program to fly a rocket sprite to the moon and back.

Debug simple programs

Debug a sequence of instructions.



Create and debug a simple animation.

Use logical reasoning to predict the behaviour of simple programs

Add another sprite by duplicating it.

Edit the script to control itusing different keyboardinputs.

Game Design

iPad Apps

Path Puzzler

Kodable

Scratch Junior

Bee-Bot

Daisy the Dinosaur

Tynker

LONDON GRID FOR LEARNING

Year Two

Turtle

Animate

JIT

Visual Logo

Look at lesson plans

Spider Web
Mole Maze
Chameleon
Cinderella story boarder
Anansi story boarder
Minibeast sorter
Minibeast sorter tree

Follow the cloud
Music mixer
Monster scale
Monster grid
Colour choir
Colliding cogs
Crazy cogs
Falling wall
Puzzling paths
Talent show
Block-a-doodle-doo
Path peril
Helicopter rescue
Hen hunt
Road blocks
Tunnel trouble
Maze
Little Red Ridding Hood
The Gingerbread Man
Goldilocks and the Three Bears

KS2

Jun 21-18:43

Year Three

Write programs that accomplish specific goals.
Use sequence in programs.
Work with various forms of output.
Work with various forms of input

Scratch

- Edit an existing file and explore how to program a sprite to draw a series of shapes.
- Use the repeat control to make the program more efficient.
- Explore repeating a repeated sequence of instructions.


Square


Square Repeat


Pattern

Lego Wedo

Program models to perform specific tasks in a sequence.
Edit the programs to change or extend the sequence.
Edit the programs to change or extend the output.
Edit the programs to change the input.
Edit the programs to change the in put.
use of 'Repeat' command.

2Create a Superstory

Create an Animation Sequence

Pro-Bot

- Include degrees of turn in program.
- Break complex sequences down into a manageable set of instructions.
- Write down instructions before programming Pro-Bot. Test and edit.
- Find the most efficient way to complete a route.
- Use a pen to draw output.
- Program Pro-bot to draw simple regular shapes and annotate them.
- Use the repeat function to program the Pro-Bot to draw a square.

iPad Apps


Path Puzzler

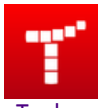

Bee-Bot Pyramid


ALEX


Move the Turtle


Cato's Hike


Hopscotch


Tynker


My Robot Friend


iLogo

LONDON GRID FOR LEARNING

Year Three

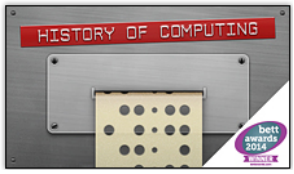
JIT

Visual

Logo

Look at lesson plans

Have to activate MyDrive



Year Four

Design programs that accomplish specific goals

Design and create programs

Debug programs that accomplish specific goals

Use repetition in programs

Use logical reasoning to detect and correct errors in programs
Control or simulate physical systems

Pro-Bot

- Program Pro- Bot to follow a procedure.
- Program Pro- Bot's lights to turn on when the light levels falls.
- Program Pro Bot to to reverse into a parking space.
- Program Pro-Bot to maneuver around a route

Scratch

Make a simple maze game for Year 2 children.

Create the game in stages

Create graphics

Create sequence of events.

Use 'If ' and 'Forever' programming blocks.

Use the sensing command.

Test and Debug

Simple Car Game

Simple car
game 2 levels

Lego Wedo

Build a model with sensors.

Write a sequence of instructions to control a model based on conditional commands ('If the motion sensor detects movement the motor will turn anti- clockwise.').

Use repeat commands and add loops to your sequence.

Using what you know of the program adapt and edit a sequence to create a morecomplex program.

Debug program.

iPad Apps

Tynker

Move the Turtle

Cato's Hike

Hopscotch

iLogo

My Robot Friend

Minecraft
PE

Nov 24-3:56 PM

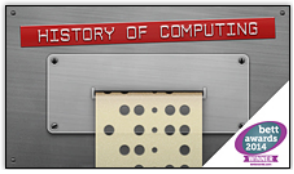
LONDON GRID FOR LEARNING

Year Four

[j2code] [code] repeat bett AWARDS 2015 FINALIST

JIT
Visual
Logo
Look at lesson plans

NEW AppMaker
Have to activate MyDrive



NEW WebTech tutor

Year Five

Solve problems by decomposing them into smaller parts
Use selection in programs
Work with variables

Bitesize Bitesize

Lego Mindstorms

Programme the floor robot to;

- Move around a square
- Revise park into a specific location
- Complete an obstacle course
- Plan, record, edit and amend programs.

Scratch or Kodu

- Create a scoring game in stages.
- Create variables.
- Include selection ('if' statements).
- Include iteration
- Test, edit and amend.

EV3

SCRATCH

Chomp game

Kodu

http://www.kodugamelab.com/hour-of-code/

CAS Barefoot supported by BT

The BLACK Chamber

Learn the basics of HTML and other coding lan

codecademy playto

Code Monster

iPad Apps

Hopscotch My Robot Friend iLogo Minecraft PE Hakitzu Cargo Bot Khan Academy

LONDON GRID FOR LEARNING

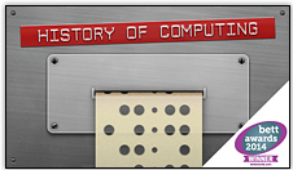
Year Five

[j2code] [code] repeat bett AWARDS 2015 FINALIST

JIT
Visual
Logo
Look at lesson plans

NEW AppMaker

Have to activate MyDrive



NEW WebTech tutor

Year Six

Robotics

Lego Mindstorm

- Using sensors, create and program a moon buggy that is able to sense obstacles and take preventative action.
- Use repeat and if statements to do this.
- Programme the robot to use stop and wait commands.

EV3

Developing projects using a Rasberry Pi

Raspberry Pi

Scratch or Kodu

- Plan and create a multi-level game.
- Include a time and score variable.
- Peer test your game
- Publish game for evaluation.

Kodu

Designing and creating an app

MIT App Inventor

appshed creative apps generation

Python

- Become familiar the Python interface and some simple statements.
- Practice how to write simple statements.
- Create a simple program.
- Play games in Python.
- Edit a simple existing file in Python.

python

Code Monster

iPad Apps

Hakitzu

Cargo Bot

Khan Academy

AppCraft

meComputer

Simduino

LONDON GRID FOR LEARNING

Year Six

[j2code] [code] repeat bett AWARDS 2015 FINALIST

JIT

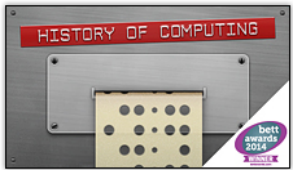
Visual

Logo

Look at lesson plans

NEW AppMaker

Have to activate MyDrive



NEW WebTech tutor

Appendix

Computational Thinking Connections

The Benefits of Computational Thinking

Five Steps of Programming

Programming Vocabulary

Programming Resources

Learning Programming

Research

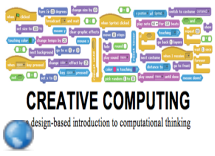
Sep 4-10:15 am

Computational Thinking Connections

The following tables summarise the computational thinking framework and define its constituent components.
From ' Creative Computing' (Scratch Ed Team, 2011)

Computational Concepts

Concept	Description
sequence	identifying a series of steps for a task
loops	running the same sequence multiple times
parallelism	making things happen at the same time
events	one thing causing another thing to happen
conditionals	making decisions based on conditions
operators	support for mathematical and logical expressions
data	storing, retrieving, and updating values



Computational Practices

Practice	Description
being iterative and incremental	developing a little bit, then trying it out, then developing some more
testing and debugging	making sure that things work – and finding and fixing mistakes
reusing and remixing	making something by building on what others – or you – have done
abstracting and modularising	building something large by putting together collections of smaller parts

Computational Perspectives

Perspective	Description
expressing	realising that computation is a medium of creation -"I can create."
connecting	recognising the power of creating with and for others -"I can do different things when I have access to others."
questioning	feeling empowered to ask questions about the world -"I can (use computation to) ask questions to make sense of (computational things in) the world."

The Benefits of Computational Thinking

1. Facilitates new ways of seeing existing problems.
2. Emphasises creating knowledge rather than using information [Phillips 2008]
3. Presents possibilities for creatively solving problems [Phillips 2008]
4. Facilitates innovation

Taken from Computational thinking across the curriculum: A conceptual framework 2009 DePaul University

Nov 21-4:02 PM

Five Steps of Programming

1. Clarify Programming Needs
2. Design the Program
3. Code the Program
4. Test the Program
5. Document and Maintain

In order to design a program for a computer, you must determine three basic elements:

The instructions that must be performed.

The order in which those instructions are to be performed.

The fixed data required to perform the instructions.

Mar 16-10:50 AM

Programming Vocabulary

Digital
The input, stored program and output are all encoded as numbers, making these devices 'digital'.

Operators
Support for mathematical and logical expressions

Decomposing
Aspect of computational thinking, in which large problems are broken down into small tasks.

Debug
To detect and correct the errors in a computer program.

Program
a stored set of instructions encoded in a language understood by the computer that does some form of computation, processing input and/ or stored data to generate output.

Loops
A loop is a sequence of instructions that is continually repeated until a certain condition is reached. Each pass through the loop is called an iteration.

Algorithm
An **algorithm** is a precisely defined procedure – a **sequence** of instructions, or a set of rules, for performing a specific task (e.g. instructions for changing a wheel or making a cup of tea)

Making a cup of tea

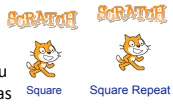
Thinking Algorithmically
When you write a program you need to have a clear idea of what it will do and how it should do it. This is where algorithms come in, and is an integral part of the craft of programming.

Abstraction
In computer science, **abstraction** is the process by which data and programs are defined with a representation similar in form to its meaning (semantics), while hiding away the implementation details. Abstraction tries to reduce and factor out details so that the programmer can focus on a few concepts at a time.

Iteration
in computing is the repetition of a block of statements within a computer program

Data
Stores, retrieves, and updates values.

Efficient Solutions
Finding ways that guarantee you get a task done in as few steps as possible.



Recursion
in computer science is a method where the solution to a problem depends on solutions to smaller instances of the same problem (as opposed to iteration).It calls upon its self.

Conditionals
Making decisions based on conditions

Events
One thing causing another thing to happen.

Selection
Refers to instructions such as *if ... then ... otherwise* decisions in which the operation (what the program does) depends on whether or not certain conditions are met.

Sequence
to place programming instructions in order, with each executed one after the other.

Repetition
A programming construct in which one or more instructions are repeated, perhaps a certain number of times, until a condition is satisfied or until the program is stopped.

Logic Reasoning
A systematic approach to solving problems or deducing information using a set of universally applicable and totally reliable rules.

Variables
a way in which computer programs can store, retrieve or change simple data, such as a score, the time left, or the user's name.

Parallelism
Making things happen at the same time.

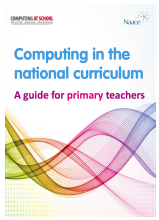
Computation
use of computing technology in information processing. - Wikipedia

Computational Thinking
It is a collection of diverse skills to do with problem solving that result from studying the nature of computation.

Nov 22-1:27 PM

Curriculum Resources

DfE Computing programmes of study KS 1 & 2
CAS Computing in the National Curriculum
Computing at School (CAS)
Computer Science for Fun
Exploring Computational Thinking and Problem Solving
Computer Science Unplugged
Naace



Learning Programming

Kodu Game Lab
Kodu Teachers Tutorial
Python
Python School
Small Basic
Hackety Hack
Try Ruby
GameMaker
Stagecast
Crystal Rainforest

Programming Resources

Scratch Ed
An Introduction to Computing Science, Starting from Scratch - The Royal Society of Edinburgh
The Khan Academy
Answers for Young People -Berners-Lee, T
Teaching Kids Programming
CodeAcademy
Code-It
The Raspberry Pi Manuel - CAS
Cracking the Code - BBC

Research

Evaluation of Computer Games Developed by Primary School Children to Gauge Understanding of Programming Concepts -GaLa
Computer Science: A curriculum for Schools - Computing at School Working Group
New frameworks for studying and assessing the development of computational thinking - Brennan & Renick
Shut down or restart? The way forward for computing in UK schools- The Royal Society

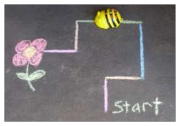
2.

Back

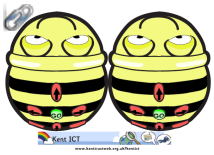
Bee-Bot and Trails

Small Alphabet Cards

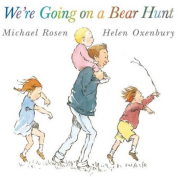
Large Alphabet Cards

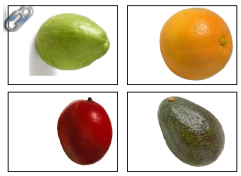


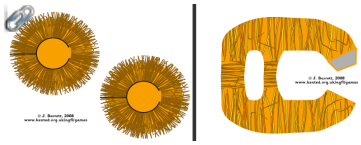
Bee-Bot and Counting



Bee-Bot and Journeys







Back

Challenge

What can you make your Bee-Bot do?

How man

Can you make your Bee-Bot move forwards?

1 2 3 4 5 6 7 8 9 10

How would you write these instructions down?

Mar 3-16:42

Back

2.

1.

2.

3.

4.

5.

I filled the kettle and turned it on.

When the kettle boiled I poured the boiled water into the mug.

I added milk.

I put a tea bag in a mug.

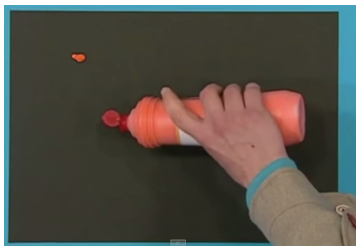
I drank the tea.

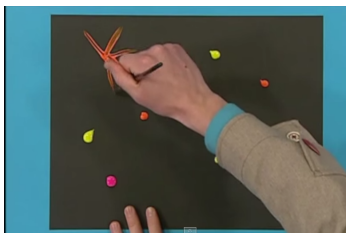
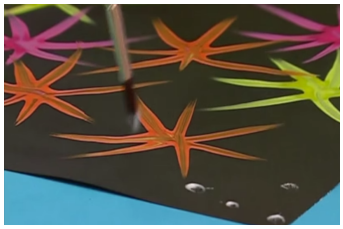
Printable activity for children

Apr 21 - 21:42

Back

Sequencing an activity



Talk through the sequence of how to make a fire work picture with your partner.

Could the presenter created this picture if we changed the sequence?

Ask the pupils to look at this sequence of images.

Briefly talk through the images.

Move some of the images around and ask if the firework picture could have been created if the sequence was changed. Change the sequence and ask would happen.

Nov 15-10:00

Back

Sequencing the pictures in the correct order

Can you sequence these pictures in the correct order?

1.

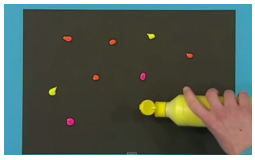
2.

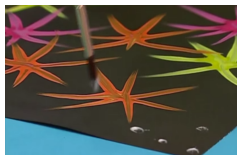
3.

4.

5.

6.





Ask pupils to sequences these images on the IWB.

Explain that if you follow the sequence of instructions in the correct order you will always produced a firework picture.

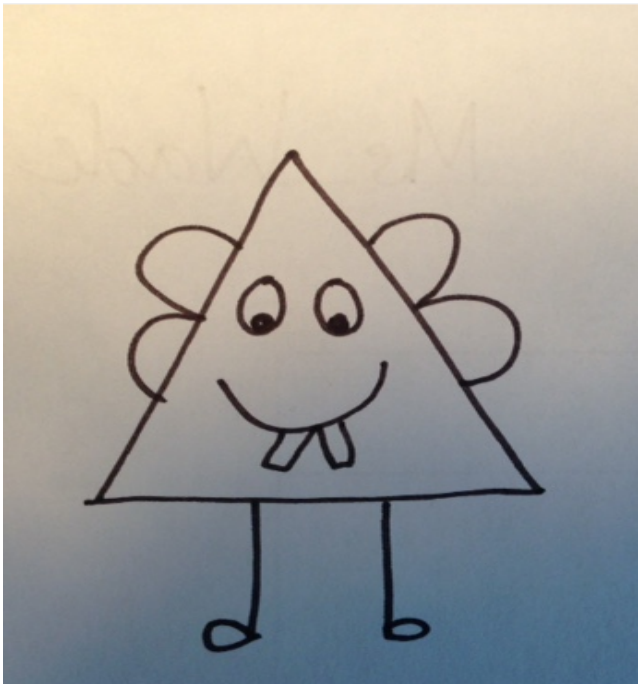
Explain this is an algorithm.

You may wish the pupils to carryout this activities individually. Click here a printable copy of the paper version of the activity

Nov 15-10:35

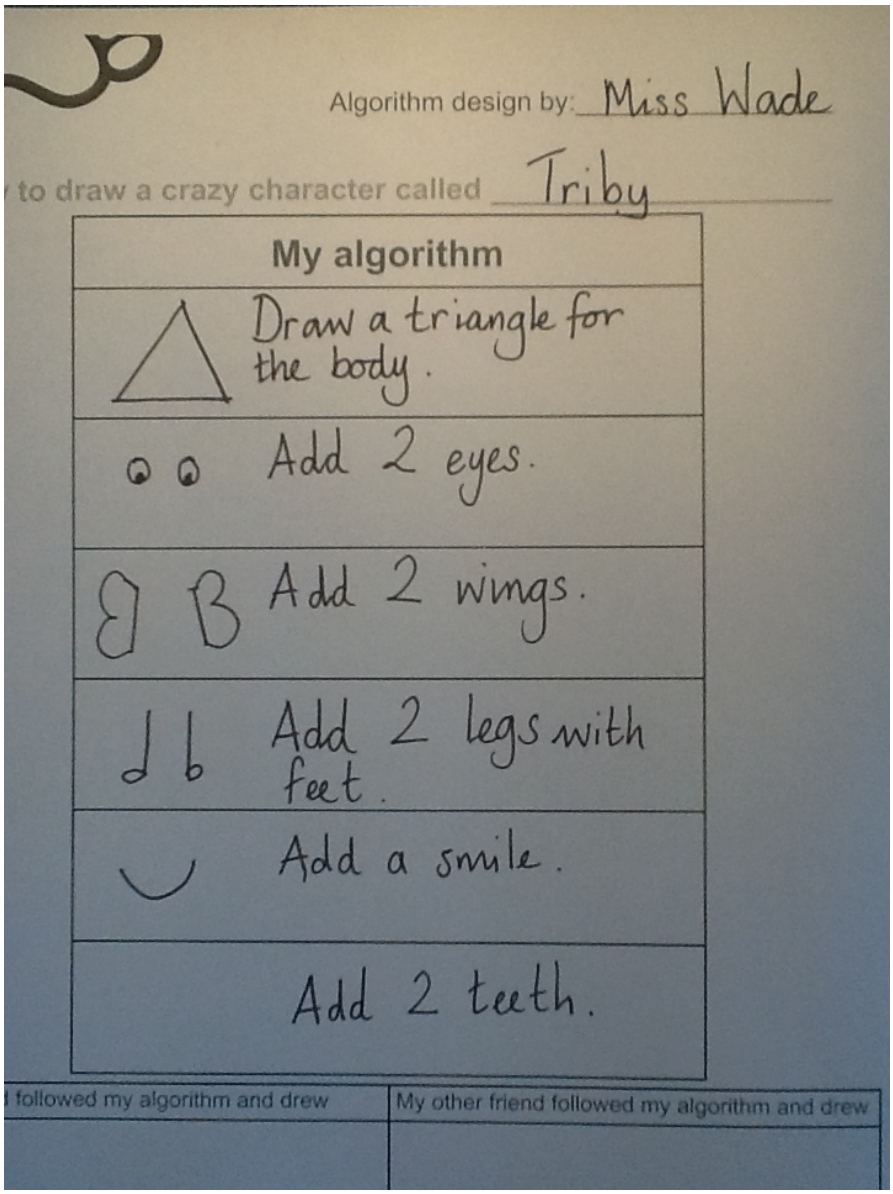
Back

My crazy shape character Triby



Nov 15-12:21

Back



Dual page display.

Ask the pupils to help you debug your algorithm.

Ask are there any instructions we could add to make it better.

Model an example.

Ask the pupils to discuss this with a partner how to improve my algorithm.

Use the pupil's suggestions to edit the algorithm.

Back

Crazy shape character algorithm.

Explain to the pupils that they are going to create a crazy shape algorithm which other pupils are going to follow.

We are going to be doing this because that is what we are going to be presenting but first we have to make sure we can write clear instructions to follow.

Explain that they are first going to follow your crazy shape character algorithm.

Explain they are going to follow the algorithm and draw my crazy character on their wipe boards.

Read out the algorithm line by line.

Ask the pupils to show you their character once they have completed their drawing.

Reveal your character.

Ask the pupils in pairs to compare and discuss the similarities and differences between the two drawings

Select a few drawing to compare with your drawings.

Back

What is an algorithm?

A sequence of instructions that is carried out in a particular order and always gives us the same outcome each time is called an

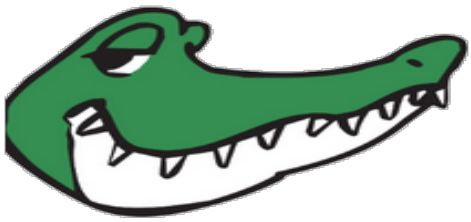
algorithm.

Algorithms are very important to people and computers.

To help us remember what an algorithm we can say the algorithm chant.

Alligator Algorithm Chant

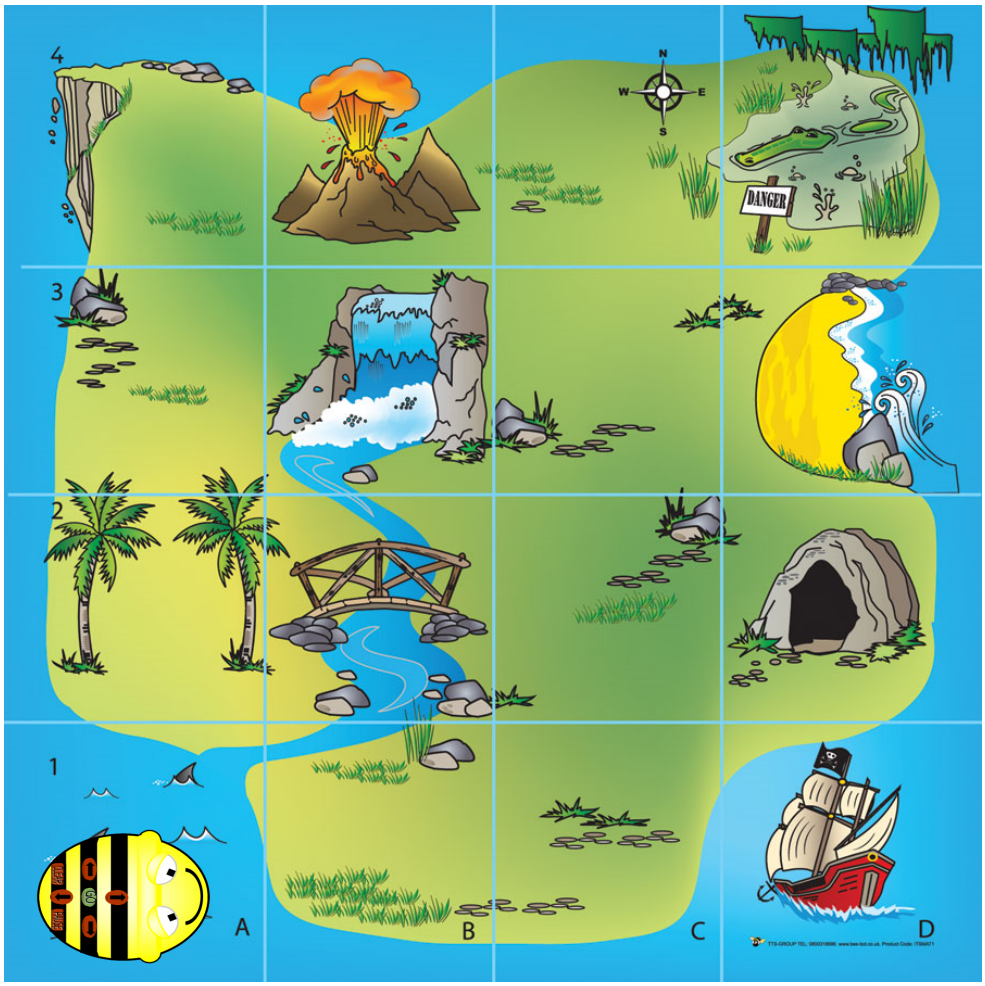
Alligator Algorithm, step, step, step, she's very, very bossy but she gets things done, how to draw a character, how to open the door, Alligator algorithm think of some more?



May 7-13:27

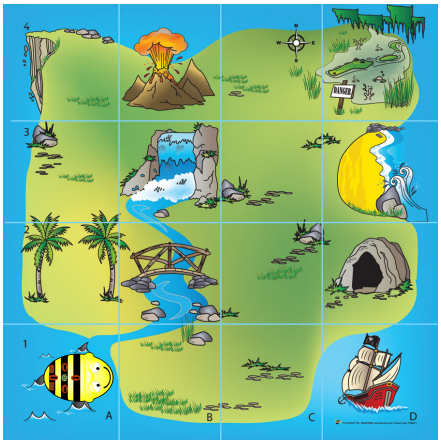
Back

Programming the pirate robot



N.B Use dual page display

Back



From	To	Code

The Task

Introduce pupils to the map and explain what everything is.

Explain that the Bee-bot is a pirate robot and just like our other pirate it is on an island looking for treasure.

Explain that i n groups we are going to plan, program and write down the algorithm in code.

Model how this is done;

- One pupil will plan the route to a specific location using the laminated bee-bot.
- One will write the algorithmic code on a wipe board.
- One will use this to program the bee-bot.

The pupils will then swap roles.

Attachments

KS1-Crazy-Character-Algorithms-worksheet-Barefoot-Computing2.pdf

Bee bot Treasure map programming book.docx

Bee Bot Class Set.docx

A Day Made of Glass... Made possible by Corning. (2011) - YouTube.url

Jack and monsters.2iy

Great Fire of London.2c2

Fire Engine.2c2

Square.LGO

Storyboard.docx