Year Group: 5	Term: Spring 1 & 2	Topic: Computing Science

NC Links

- Design and write programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts.
- Use sequence, selections and repetition in programs; work with variables and various forms of input and output; generate appropriate inputs and predicted outputs to test programs.
- Use logical reasoning to explain how a simple algorithm works, detect and correct errors in algorithms and programs.

Other Curriculum Links

Topic Overview

Children will further understanding of computing science by using knowledge of algorithms and programming by using a range of activities and applications. Topic will begin by revisiting 'unplugged' activities using Barefoot Computing and/or CS Unplugged, to support children's understanding of computing science and the vocabulary behind it (use knowledge organisers to help explain concepts and vocabulary). Children will then progress onto the next stage of Code.org (Unit E). Class teachers can also use school ipads as an additional activity or exploration lesson.

Links to Rights Respecting

Article 17 - Every child has the right to reliable information from the media. This should be information that children can understand. Governments must help protect children from materials that could harm them.
Article 28 - Every child has the right to an education. Primary education must be free and different forms of secondary education must be available to every child. Discipline in schools must respect children's dignity and their rights.

Links to North East Ambition

Children will look at different careers within computing and how the subject is evolving every day. Teachers can reference jobs/companies that are recognised globally (apple, Microsoft etc.) or locally (Sunderland Nissan, Newcastle University, CAS). Jobs may include: Engineer, Game Designer, Cyber Crime Officer, Photographer, Video Animator, Office Worker etc.
GATSBY BENCHMARK 3
GATSBY BENCHMARK 4
GATSBY BENCHMARK 5

Possible Visits/Visitors

Trip using metro stations to create algorithms

Essential Subject Skills to be covered

- Set IF conditions for movements. Specify types of rotation giving the number of degrees.
- Change the position of objects between screen layers (send to back, bring to front).
- Upload sounds from a file and edit them. Add effects such as fade in and out and control their implementation.
- Combine the use of pens with movement to create interesting effects.
- Set events to control other events by 'broadcasting' information as a trigger.
- Use IF THEN ELSE conditions to control events or objects.
- Use a range of sensing tools (including proximity, user inputs, loudness and mouse position) to control events or actions.
- Use lists to create a set of variables.

Overall Learning Outcomes

By the end of this unit, children will learn about Ramp Up, Sprites, Digital Citzenship, Impacts of Computing, Nestled Loops and Functions

Learning Intentions (for use in self assessment at end of topic)

- To be able to set IF conditions for movements
- To be able to change the position of objects between screen layers
- To be able to upload sounds from a file and edit them.
- To add sound effects such as fade in and out and control their implementation.
- To be able to combine the use of pens with movement to create interesting effects.
- To set events to control other events by 'broadcasting' information as a trigger.
- To understand the difference between IF THEN ELSE conditions to control events or objects.
- To be able to use a range of sensing tools (including proximity, user inputs, loudness and mouse position) to control events or actions.
- To use lists to create a set of variables.

Year Group: 5	Term: Spring 1 & 2	Topic: Computing Science

Possible Activities

- **Barefoot Computing** – Code Cracking and Ranking Search Activity
- **CS Unplugged** – Sorting Networks p81-93 & Graph Colouring (Intractability) p132-140
- **Code.org** – (Unit E) Ramp up, Sprites, Digital Citizenship, Impacts of Computing, Nestled Loops, Functions
- ***'Underground Algorithms'*** http://www.animate2educate.co.uk/web/underground_algorithms_lanchester_ep/179234
  Pupils to work in pairs to devise a route from one underground station to another. Allow pupils time to independently create their algorithms before directing them to the precise nature of naming lines and colours, North/South/East/West, where to change etc. Create either the text (or text box) in the colour of the line but don't explain significance of this to pupils at the start of the lesson. Do they apply 'logical reasoning' and see link to the colour of the line that the station is on? Link to 'Tube Map' app. This can be firstly used to help find stations on the map and then can be used at the end of the lesson to find out if the most 'precise' route was created, or whether debugging is necessary to make the route more precise. Use large laminated London Underground map, along with counters to help with the physical element of creating the route algorithms.
- Continuation of activities from Year 4 using ***'Lightbot Jr'*** app.
- Review skills of ***'Scratch Junior'*** app from Year 3 and Year 4 and then introduce basic programming activities using desktop version of 'Scratch'. Lots of great activities and ideas on the Scratch Community website: https://scratch.mit.edu/

Suggested Strategies for Recording Learning

- Code.org
- Create algorithms for navigating around London Underground map
- Create own version of Underground Algorithms using local Metro service
- Complete own project using Scratch and Scratch Jr.
- Creating algorithms and debugging using ***'Lightbot Jr'*** app.

Year Group: 5	Term: Spring 1 & 2	Topic: Computing Science

Assessment

Ongoing assessment from guided activities, observations, discussions, questioning and work evidence. A suggested activity is:

- Code.org will track progress of children
- Create own algorithms using London Underground map.
- Code using Scratch and Scratch Jr.
- Use Lighthouse Jr. app to create algorithms and debug

	x.1	x.2	x.3
Motion	Beginning to understand the difference between and appropriately uses 'if', 'then' and 'else' statements.	Understands the difference between and appropriately uses 'if', 'then' and 'else' statements.	Understands *fully* the difference between and appropriately uses 'if', 'then' and 'else' statements.
Looks	Beginning to change the position of objects between layers.	Can regularly attempt to change the position of objects between layers with some accuracy.	Can regularly and independently change the position of objects between layers accurately.
Sound	Is beginning to upload and edit sounds from a file as well as adding fade in and out effects controlling their implementation.	Can upload and edit sounds from a file, as well as adding fade in and out effects controlling their implementation.	Can independently and accurately upload and edit sounds from a file, as well as adding fade in and out effects controlling their implementation.
Draw	Is beginning to develop combining the use of pens with movement to create interesting effects.	Can securely combine the use of pens with movement to create interesting effects.	Can independently and purposefully combine the use of pens with movement to create interesting effects.
Events	Is beginning to set events to control other events by 'broadcasting' information as a trigger.	Can set events to control other events by 'broadcasting' information as a trigger.	Can set a range of events to control other events by 'broadcasting' information as a trigger.

Control	Beginning to use IF THE ELSE conditions to control events or objects.	Regularly uses IF THE ELSE conditions to control events or objects.	Accurately and independently uses IF THE ELSE conditions to control events or objects.
Sensing	Beginning to use a range of sensing tools (including proximity, user inputs, loudness and mouse position) to control events or actions.	Use a range of sensing tools (including proximity, user inputs, loudness and mouse position) to control events or actions.	Can use a wide range of sensing tools (including proximity, user inputs, loudness and mouse position) to control events or actions.
Variables and lists	Is beginning to use lists to create a set of variables.	Can use lists to create a set of variables.	Can use lists to create a set of multiple variables.