DIGITAL SCHOOLHOUSE together with NINTENDO SWITCH

+44 (0) 203 534 0580
info@digitalschoolhouse.org.uk

Teaching Guide.

Funky Maths: Creating a calculator

Introduction

This workshop addresses the concepts of simple algebraic equations and variables by teaching pupils to create their own calculator using the scratch programming environment.

Pupils are introduced to the day through the use of a CS4FN activity “The Australian Magician’s Dream”. This introduces them to the idea of instructions and algorithms. After learning the trick for themselves, the pupils apply decomposition skills to identify how the various aspects of a calculator work. Using what they have learnt they attempt to write their own algorithms for the different buttons on a calculator interface.

After having prepared their algorithms, pupils spend time developing their skills in scratch and begin by making a simple addition calculator with sprites of their own design. Pupils discuss interface design for the calculator and how they can use the Scratch commands add further interactivity. Further modifications and functionality are added to the calculator as the pupils become more confident.

+44 (0) 207 534 0586
contact@digitalschoolhouse.org.uk

Computing Programmes of Study Links

2.1 design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts

2.2 use sequence, selection, and repetition in programs; work with variables and various forms of input and output

2.3 use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs

3.1 design, use and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems

Progression Pathway bands covered

ALG = Algorithms: Pink, Yellow, Orange, Blue

Reference	
PA1	Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically.
PA2	Understands that computers need precise instructions.
PA3	Demonstrates care and precision to avoid errors
YA1	Understands that algorithms are implemented on digital devices as programs
YA2	Designs simple algorithms using loops, and selection i.e. if statements.
YA3	Uses logical reasoning to predict outcomes.
YA4	Detects and corrects errors i.e. debugging, in algorithms.
OA1	Designs solutions (algorithms) that use repetition and two-way selection i.e. if, then and else.
OA2	Uses diagrams to express solutions.
OA3	Uses logical reasoning to predict outputs, showing an awareness of inputs.
BA2	Designs solutions by decomposing a problem and creates a sub-solution for each of these parts.

+44 (0) 203 534 0580
info@digitalschoolhouse.org.uk

P&D = Programming & Development: Pink, Yellow, Orange, Blue

Reference

PP1	Knows that users can develop their own programs and can demonstrate this by creating a simple program in an environment that does not rely on text
PP2	Executes, checks and changes programs
PP3	Understands that programs execute by following precise instructions
YP1	Uses arithmetic operators, if statements, and loops, within programs.
YP2	Uses logical reasoning to predict the behaviour of programs
YP3	Detects and corrects simple semantic errors i.e. debugging, in programs.
OP1	Creates programs that implement algorithms to achieve given goals.
OP2	Declares and assigns variables.
OP3	Uses post-tested loop e.g. 'until', and a sequence of selection statements in programs, including an if, then and else statement.
BP2	Uses a variable and relational operators within a loop to govern termination.
BP3	Designs, writes and debugs modular programs using procedures.

+44 (0) 207 534 0586
contact@digitalschoolhouse.org.uk

Computational Thinking Strands

AL – Algorithmic Thinking

Ref.	Activity
A1	Writing instructions that if followed in a given order (sequences) achieve a desired effect
A2	Writing instructions that use arithmetic and logical operations to achieve a desired effect
A3	Writing instructions that store, move and manipulate data to achieve a desired effect; (variables and assignment)
A4	Writing instructions that choose between different constituent instructions (selection) to achieve a desired effect;
A5	Writing instructions that repeat groups of constituent instructions (loops/iteration) to achieve a desired effect;
A6	Grouping and naming a collection of instructions that do a well-defined task to make a new instruction (subroutines, procedures, functions, methods);

AB – Abstraction

Ref.	Activity
Ab1	Reducing complexity by removing unnecessary detail;
Ab2	Choosing a way to represent artefacts (whether objects, problems, processes or systems) to allow it to be manipulated in useful ways;
Ab3	Hiding the full complexity of an artefact, whether objects, problems, processes, solutions, systems (hiding functional complexity);

+44 (0) 207 534 0586
contact@digitalschoolhouse.org.uk

EV – Evaluation

Ref.	Activity
E1	Assessing that an algorithm is fit for purpose
E2	Assessing whether an algorithm does the right thing (functional correctness);
E4	Assessment whether the performance of an algorithm is good enough;
E5	Comparing the performance of algorithms that do the same thing;
E6	Making trade-offs between conflicting demands;
E7	Assessment of whether a system is easy for people to use (usability);
E8	Assessment of whether a system gives an appropriately positive experience when used (user experience);
E9	Assessment of any of the above against set criteria;
E10	Stepping through algorithms/code step by step to work out what they do (dry run / tracing);
E15	Assessing whether a solution meets the specification (criteria);
E16	Assessing whether a product meets general performance criteria (heuristics)

+44 (0) 207 534 0586
contact@digitalschoolhouse.org.uk

Learning Outcomes

1. Be able to solve simple mathematical problems mentally to be able to test the accuracy and functionality of their calculator
2. Be able to use simple formulae
3. To explain and understand the concept of variables
4. To be able to use variables in simple formulae
5. To be able to create a calculator in Scratch
6. To be able to design and edit their own sprites in Scratch
7. To be able to program the sprites within scratch using simple constructs such as sequence, selection and repetition.
8. To be able to use an existing calculator and identify the algorithms in use
9. To be able to design a calculator
10. To be able to implement given algorithms to create a programmable calculator

+44 (0) 207 534 0586
contact@digitalschoolhouse.org.uk

Session Overview

SESSION 1

Session Content / Activity	Resources Used	Prog. Pathway	Comp. Thinking	Computing POS Link
Welcome and introductions to the day.	DSH_WelcomeIntroduction.pptx			
Introduce the aims for the day and deliver the magic trick. Set the tone and context for the trick. Encourage students to work out how the trick operates. Then use the CS4FN materials to explain the self-working algorithm. Explain the mathematical concept behind the trick	Slides-australianmagician.pdf Activity-magic-australianmagician.pdf	ALG PA1, PA2, PA3, YA1, YA2, YA3, YA4, OA2, OA3		2.3
Draw the links between the magic trick and algorithms. Explain that mathematical formulae are also just algorithms. Perhaps ask them to write out the steps for simple mathematical formulae and calculations.	Funky Maths Calculator.ppt	ALG PA2, YA1,		2.3, 3.1
Ask students how they think calculators work. Explain that each of the buttons are programmed with their own set of instructions. So, what do each of the buttons do? What would the instructions be? Initiate this as a class discussion first, before distributing the worksheet. Alternatively, pupils could work in groups and brainstorm their algorithms using the mini whiteboards. Once set on task, encourage pupils to write down the steps specifically. They might begin using numbers to write out an actual calculation, however, move onto a discussion of variables. What could they use to represent different the different numbers instead?	MiniWhiteboards.pdf How do calculators work.pdf How do calculators work.docx Funky Maths Calculator.ppt MiniWhiteboard.pdf	ALG PA1, PA2, PA3, YA1, YA2, YA3, YA4, OA1, OA2, OA3, BA2		2.1, 2.3, 3.1
Initiate a discussion around the pupil's algorithms. Ask groups to share their algorithms with each other. One group can test	Funky Maths Calculator.ppt How do calculators work.pdf	ALG		2.1, 2.3, 3.1

+44 (0) 207 534 0586
contact@digitalschoolhouse.org.uk

another's algorithms by dry running it. Allow them time to modify their algorithms based on feedback.		PA1, PA2, PA3, YA1, YA2, YA3, YA4, OA1, OA2, OA3, BA2	
Begin the skills development required to build the calculator. Introduce the interface and key features.	Funky Maths Calculator.ppt	P&D PP1, PP3, YP1, OP1, OP2, BP2	2.2, 2.3
Pupils create/edit key sprites required to create the calculator			2.2, 2.3

SESSION 2

Session Content / Activity	Resources Used	Prog. Pathway	Comp. Thinking	Computing POS Link
Remind pupils about the concept of variables and demonstrate how they are used within scratch.	Funky Maths Calculator.ppt	P&D OP2, BP2		2.2
Use slides 16 – 19 to help students create a simple program that adds two numbers together	Funky Maths Calculator.ppt	P&D PP1, PP2, PP3, YP1, YP2, YP3, OP1, OP2, OP3		2.1, 2.2, 2.3, 3.1
Use slides 19 – 27 to modify the script and create a CE button. Discuss the 'broadcast' feature of scratch and explain what it does.	Funky Maths Calculator.ppt Help Videos	P&D PP1, PP2, PP3, YP1, YP2, YP3, OP1, OP2, OP3, BP2, BP3		2.1, 2.2, 2.3, 3.1
Help the pupils modify their script to integrate the broadcast option and generate a better algorithm.	Help Videos	P&D PP1, PP2, PP3, YP1, YP2, YP3, OP1, OP2, OP3, BP2, BP3		2.1, 2.2, 2.3, 3.1

+44 (0) 207 534 0586
contact@digitalschoolhouse.org.uk

Pupils test their own solutions to ensure they work. Encourage them to think about how they could extend their solutions and independently 'experiment' with possible modifications.	Funky Maths Calculator.ppt	P&D PP2, PP3, YP2, YP3, BP3	2.1, 2.3, 3.1

SESSION 3

Session Content / Activity	Resources Used	Prog. Pathway	Comp. Thinking	Computing POS Link
Initiate a class discussion to recap session 2. Encourage pupils to share their ideas for expanding their solution and their independent investigations. Building in a system to enable them to demonstrate their solution to the class is a good way to encourage them to explain their work and programming. It will also help generate ideas for the rest of the group.	Funky Maths Calculator.ppt	P&D PP1, PP2, PP3, YP1, YP2, YP3, OP1, OP2, OP3, BP2, BP3	A1, E7…	2.1, 2.3, 3.1
Work through slides 28 – 33 to guide pupils with ideas and possibilities for modifying their solution. Enable them to add animation and additional features to their solution.	Funky Maths Calculator.ppt	P&D PP1, PP2, PP3, YP1, YP2, YP3, OP1, OP2, OP3, BP2, BP3 ALG BA2		2.1, 2.2, 2.3, 3.1
Once complete, ask pupils to test their own solutions and complete the self-assessment worksheet. This is a good opportunity to ask pupils to compare their final solutions with the algorithms they wrote at the start of the day. Initiate a class discussion around the key things that pupils discovered in their comparisons. Ask individual pupils to look at: - Similarities between their algorithm and solution	Funky Maths Calculator.ppt Calculator_SelfAssessment.doc How do calculators work.doc	P&D PP2, PP3, YP2, YP3, BP3 ALG PA1, YA1, YA3, YA4, OA1, OA3, OA2, BA2		2.1, 2.3, 3.1

+44 (0) 207 534 0586
contact@digitalschoolhouse.org.uk

- Differences between their algorithm and solution - If there is a difference between their programming and their initial algorithm, then which one is better and why? Structure of the code			
Allow pupils to share and demonstrate their solutions before completing the evaluation worksheet.	Calculator_Evaluation.doc		2.1, 3.1
Recap the concepts covered during the day and point out key programming concepts covered. Can pupils recall and recognise what they are? Things to discuss: - Variables - Selection statements - Sequence statements - Iteration - Algorithm - Procedures/functions		<u>P&D</u> PP1, PP2, PP3, YP1, YP2, YP3, OP1, OP2, OP3, BP2, BP3 ALG PA1, PA2, PA3, YA1, YA2, YA3, YA4, OA1, OA2, OA3, BA2	2.2, 2.3

+44 (0) 207 534 0586
contact@digitalschoolhouse.org.uk

Files/Resources

Filename	Resource Type	Purpose/Description
Activity-magic-australianmagician	Teaching Guide	CS4FN resource guide on how to deliver the Australian Magician activity
Calculator_Evaluation.doc	Worksheet	Evaluation document for the day
Calculator_SelfAssessment.doc	Worksheet	Self-Assessment worksheet
Funky Maths Calculator.ppt	Presentation	Teaching presentation resource
How do Calculators work.doc	Worksheet	Algorithm design worksheet
How do Calculators work.pdf	Worksheet	Printable format for the algorithm design worksheet
Scratch teaching maths Algebra.mp4	Help Video	Video tutorial
SJ Calculator Extra.sb	Scratch source file	For use in demonstrations
SJ Calculator final.sb	Scratch source file	For use in demonstrations
SJ Calculator.sb	Scratch source file	For use in demonstrations
Slides-australianmagic.pdf	PDF Presentation	CS4FN resource – printable format for presentation
Background images for scratch	Sub-folder	Contains a range of images for use in the project
Videos	Sub-folder	Contains a range of support videos for use as tutorials

PLEASE NOTE: The activities outlined in this workshop pack are a suggested outline of how the workshop can be delivered. It is envisaged that teachers will adapt the resources and the organisation of them according to the needs of their class.

Digital Schoolhouse, 18a Blackbull Yard, 24-28 Hatton Wall, London EC1N 8JH.