DIGITAL SCHOOLHOUSE

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

Enrichment Day Teaching Guide

Generating Art: Creating a Shape Calculator in Scratch

Introduction

Helping students understand the importance and relevance of calculating the internal angles of shapes is something that can become a challenge to deliver. This project aims to address that whilst teaching computational thinking concepts i.e. problem solving. The Digital Schoolhouse has worked with the Langley Grammar School Maths and IT/Computing departments to ensure that the project meets the curriculum needs of KS2 teachers but also gives pupils the opportunity to experience teaching techniques used at KS3.

The Digital Schoolhouse teacher begins the lesson by drawing out a range of shapes on individual whiteboards. The purpose of doing this is so that the class can analyse the structure of the shapes and consider how they knew to draw the shape and if it was correct i.e. they know that shapes have a particular number of sides. The Digital Schoolhouse teacher then helps the children to problem solve the internal angle of the shape by dividing 360 degrees by the number of sides.

The pupils then experience drawing shapes and images by following a series of instructions, as well as providing their own instructions. These unplugged activities introduce pupils to the concept of algorithms and programming and helps them understand the precise nature of programming.

The Digital Schoolhouse teacher then introduces algebra by using the computing concepts of 'constants' and 'variables' with inputs and the pen up/pen down tool to draw the shape in scratch. Pupils are then encouraged to make their shape calculator more user friendly by adding animation to the output of the answer so that a character walks across the screen and say the answer in a speech bubble. Extension activities at the end of the day allow students to adapt and extend their programme for other purposes, a range of activities are suggested thereby allowing pupils to choose their own modifications to their programme.

DIGITAL SCHOOLHOUSE

+44 (0) 203 534 0580
info@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

3.3. use two or more programming languages, at least one of which is textual, to solve a variety of computational problems; make appropriate use of data structures [for example, lists, tables or arrays]; design and develop modular programs that use procedures or functions

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.

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

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.
BA1	Shows an awareness of tasks best completed by humans or computers.
BA2	Designs solutions by decomposing a problem and creates a sub-solution for each of these parts.
BA3	Recognises that different solutions exist for the same problem.

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.
BP1	Understands the difference between, and appropriately uses if and if, then and else statements.
BP2	Uses a variable and relational operators within a loop to govern termination.
BP3	Designs, writes and debugs modular programs using procedures.
BP4	Knows that a procedure can be used to hide the detail with sub-solution.

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);

GE – Generalisation

Ref.	Activity
G1	Identifying patterns and commonalities in problems, processes, solutions, or data.
G2	Adapting solutions or parts of solutions so they apply to a whole class of similar problems;
G3	Transferring ideas and solutions from one problem area to another

DE – Decomposition

Ref.	Activity
D1	Breaking down artefacts (whether objects, problems, processes, solutions, systems or abstractions) into constituent parts to make them easier to work with

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

Learning Outcomes

1. Be able to follow an algorithm to create an image, and then use the results to judge the effectiveness of those instructions
2. Be able to generate an algorithm to produce a graphical image
3. Be able to test the and evaluate the effectiveness of an algorithm written by themselves or their peers
4. To understand that shapes are constructed of a number of sides, and internal angles
5. To understand and be able to implement the simple formula (Number_of_Sides/360 = Internal_Angle) in order to create a shape
6. To understand that the formula used to calculate an internal angle can be adapted to create a number of varying shapes and graphical images
7. To be able to write an algorithm which implements the formula to calculate an internal angle
8. To be able to write an algorithm that draws 2D shapes, with some pupils building 3D shapes
9. To be able to implement an algorithm that draws 2D shapes, with some pupils building 3D shapes
10. To be able to independently familiarise themselves with the Scratch programming environment
11. To be able to develop a programme which draws 2D (and 3D) shapes
12. To be able to develop a programme which uses variables to create 2D (and 3D) shapes
13. To be able to use loops, variables and calculations within their programme to implement a solution that they've designed
14. To be able to use pre-defined functions/procedures to improve the effectiveness and design of their programme
15. To be able to adapt their algorithms and programming to modify their solution to meet a new set of requirements

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

Session Overview

SESSION 1

Session Content / Activity	Resources Used	Prog. Pathway	Comp. Thinking	Computing POS Link
Welcome and introduction	Welcome.pptx			
Use mini whiteboards to ask students to draw a range of shapes, such as triangles, squares, hexagons, octagons etc. For each, draw students into conversation discussing the structure of the shape and how they knew they drew it correctly. Tease out discussion and knowledge of internal angles and the mathematical formula to calculate it correctly.	MiniWhiteboards.pdf Scratch Calculator.pptx	ALG PA1, YA3, OA3	A1, G1, D1	
Using the mini whiteboards, show slide 2 – ask students to draw the shape as each instruction comes up on the board Students hold up their shapes for all to see – discuss why their drawings are so different. Show slide 3 – going through the instructions Discuss the reasons behind this using slide 4. Discuss the importance of precision and the correct sequence of instructions.	MiniWhiteboards.pdf Scratch Calculator.pptx	ALG PA1 – PA3, YA3 YA4, OA2, OA3, BA1, BA2	A1, G1, D1	2.1, 2.3
Move onto the “Marching Orders” activity from Slide 5 – 12. This activity is taken from CS Unplugged and a PDF file describing the delivery of the activity is included in this pack. It is recommended that you ‘freeze’ the display on the IWB so that the class cannot see the diagram that is being described to them. Reveal the image only after they have completed the instructions. Asking the class to hold up their image (use the mini whiteboards) allows them to compare their drawing with their peers. You may wish to have a short discussion of how the instructions could have been improved to produce better results. The images have repeating shapes – students may pick up on this naturally and use that within their instructions; for those that	MiniWhiteboards.pdf Scratch Calculator.pptx Programming_languages.pdf	ALG PA1 – PA3, YA3 YA4, OA2, OA3, BA1, BA2	A1, A4, A5, G1, G2, G3, D1	2.1, 2.3

do not, you may wish to introduce/highlight this to them and 'suggest' ways in which they could reduce the length of their algorithm.				
Ask pupils to work through the 'your turn' activity on slide 13 & 14. They can get as adventurous as they like with their drawing and if they wish they can use multiple colours. Work with them to enable them to consider their instructions carefully. Perhaps establish key words that the pupils could use to write out their instructions. Leaving time for the testing and debugging and evaluation of their algorithms is important here. It is important for students to enter a discussion into this area even if they do not have time to re-write an improved version	MiniWhiteboards.pdf Scratch Calculator.pptx	ALG PA1 – PA3, YA2, YA3 YA4, OA1, OA2, OA3, BA1, BA2, BA3	A1, A2, A3, A4, A5, Ab1, Ab2, Ab3, G1, D1	2.1, 2.2, 2.3
As pupils may already have experienced some Scratch programming, it is a good idea to end this session by allowing them to independently explore Scratch. Whether they have or haven't used it before they can identify key features they haven't used before. Session 2 can begin by pupils demonstrating what they have discovered in Scratch and this will serve as a useful introduction to the skills development section as well as enabling you to identify their skills and prior knowledge in this area.	Scratch Calculator.pptx	P&D PP3	G1, D1	2.1, 2.3

SESSION 2

Session Content / Activity	Resources Used	Prog. Pathway	Comp. Thinking	Computing POS Link
Pupil demonstration of key features/skills that they discovered in Scratch during their independent exploration time. Use this to introduce the Scratch environment and put into context why they are moving onto the computer.	Scratch Calculator.pptx			2.1, 2.3
Use slides 16 – 21 to introduce students to key skills in Scratch which will enable them to draw shapes. Slide 21 asks students to investigate how to draw a circle. This is a good way to introduce loops and repetition. Can they apply the same principle to improve the instructions for a square and a triangle?	Scratch Calculator.pptx	ALG PA1 – PA3, YA1-YA4, OA1-OA3, BA1-BA3 P&D	A5, G1, G2, G3	2.1, 2.3

		PP1 – PP3, YP1 – YP3, OP1 – OP3		
Slide 22 – practice loops	Scratch Calculator.pptx		A5	2.1, 2.3
"What's Wrong with this?" – slide 23 and 24. Distribute the worksheet "whats wrong with it". The image is taken from the CS Unplugged activity in the morning. Ask students to see if they can identify why the code doesn't produce the image. Encourage them to implement and test the code for themselves.	Scratch Calculator.pptx WhatsWrongWithIt.docx WhatsWrongWithIt.pdf	ALG PA1 – PA3, YA1-YA4, OA1-OA3, BA1-BA3 P&D PP1 – PP3, YP1 – YP3, OP1 – OP3	A1, A2, A3, A4, A5, Ab2, D1	2.1, 2.3
Slide 25 – discuss the answer with the pupils. Use this opportunity to discuss the function/procedure blocks. What are they doing and how do they help? Pupils create their own version of the solution – can they modify it? Extension – write the code to produce a drawing of another CS Unplugged diagram.	WhatsWrongWithIt.docx Scratch Calculator.pptx	ALG PA1 – PA3, YA1-YA4, OA1-OA3, BA1-BA3 P&D PP1 – PP3, YP1 – YP3, OP1 – OP3, BP1 – BP4	A1, A2, A3, A4, A5, A6	2.1, 2.3, 3.3
Introduce the shape calculator activity on slide 26 and 27. Use slides 28 - 34 to discuss calculations and variables. Encourage pupils to first write out the algorithm for their program before implementing it independently.	Scratch Calculator.pptx	ALG PA1 – PA3, YA1-YA4, OA1-OA3, BA1-BA3 P&D PP1 – PP3, YP1 – YP3, OP1 – OP3	A1, A2, A3, A4, A5, A6, Ab2, G1, G2, G3, D1	2.1, 2.2, 2.3, 3.3

SESSION 3

Session Content / Activity	Resources Used	Prog. Pathway	Comp. Thinking	Computing POS Link
Display a possible solution on slide 35 for the shape calculator. Why does this solution work? What does it use? How does it compare to your own attempts?	Scratch Calculator.pptx		D1	2.1, 2.2, 2.3
Allow pupils time to complete their own solution, and possibly refine their attempts based on the solution seen on screen.	Scratch Calculator.pptx	ALG PA1 – PA3, YA1-YA4, OA1-OA3, BA1-BA3 P&D PP1 – PP3, YP1 – YP3, OP1 – OP3, BP1 – BP4	A1, A2, A3, A4, A5, A6, Ab2, G1, G2, G3, D1	2.1, 2.2, 2.3, 3.3
Slides 36 & 37 – how can we adapt our program further? Discuss with pupils how one algorithm may solve any number of problems. Once again in groups, ask the pupils to discuss and consider how their existing programme could be modified to solve a new problem.	Scratch Calculator.pptx	ALG PA1 – PA3, YA1-YA4, OA1-OA3, BA1-BA3 P&D PP1 – PP3, YP1 – YP3, OP1 – OP3, BP1 – BP4	A1, A2, A3, A4, A5, A6, Ab1, Ab2, Ab3, G1, G2, G3, D1	2.1, 2.2, 2.3, 3.3
Allow pupils time to extend and modify their programme.	Scratch Calculator.pptx			2.1, 2.2, 2.3, 3.3

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

Files/Resources

Filename	Resource Type	Purpose/Description
2 dimensional	Sub-Folder containing images and Scratch files	A collection of images and files showing how to create an array of 2D images with the source code
3 dimensional	Sub-Folder containing images and Scratch files	A collection of images and files showing how to create an array of 3D images with the source code
Instruction Logo to Scratch	Information sheet	Available as a pdf and a .doc version. A useful helpsheet connecting the syntax across both languages
Mini Whiteboard	Photocopiable Template	Available as a pdf and a .doc version. Print and laminate to allow students to use as their personal whiteboard
Programming_languages	Activity Pack	CS Unplugged document outlining their "Marching Orders" programming activity
Shape Calculator	PowerPoint	Main teaching PowerPoint presentation show
Shape Calculator	Video	A video tutorial
Shape Calculator	.sb – scratch source file	Original source file for the solution
WhatsWrongWithIt	Worksheet	Available as a pdf and a .doc version. A worksheet which allows students to debug existing code.
WhatsWrongWithIt	.sb – scratch source file	Programmed solution with the incorrect code
WhatsWrongWithIt – answer	.sb – scratch source file	Programmed solution with the correct code

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.