together with

+44 (0) 207 534 0580
contact@digitalschoolhouse.org.uk

Teaching Guide.

Get with the Algo-rhythm

Introduction

The ‘Computing through Dance’ project was developed by the Digital Schoolhouse and Langley Grammar School’s ICT Department to appeal to girls and incorporate computing in an innovative way into the curriculum. The project starts by creating flow charts of instructions to perform dance moves of a well know dance like; the Hokey Cokey, the rugby team (New Zealand And Tonga) Hucker, Michael Jackson Moon Walk or a Tudor dance which many children study in Key Stage 2. The initial objective is to develop the understanding of a sequence and appreciate the importance of accurate instructions. Loops are then introduced for repeated instructions within the dance. A hilarious video clip of an animated character doing the Hokey Cokey is used but pupils should be encouraged to volunteer.

Next, the concept of selection is introduced with the introduction of a question; if the whistle blows, then freeze in a pose, else perform the dance. Once the concepts have been understood, students then have to create a dance with four dance moves; the dance must include at least one repeat and a pose for when the whistle blows. The dance sequence is written in a flow chart.

As space can be an issue in any classroom, students are given rules for the dance moves: they have to remain on their chairs and the dance moves are only with the upper body. The class can vote on a music style from some given samples. The students can record their dances using video cameras. Peer feedback is next; the videos are watched by the class while the students present their flow charts. A score is given on the clarity of instruction, accuracy of sequence, use of repeats, use of a question and overall quality of dance moves.

The ‘Computing through Dance’ project then evolves into using Scratch. Pupils choose a dance character or import images of themselves to perform a sequence of dance moves and by building on the students’ previous understanding in the kinaesthetic activity they are able to include repeats and a selection question. This project can be extended to add a variable to determine the number of times the dance sequences is repeated and later to introduce the concept of procedures for the more complex dance moves.

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

Progression Pathway bands covered

Write band abbreviation, full name followed by coloured levels/paths i.e.

ALG = Algorithms: Pink, Yellow, Orange, Blue

DDR = Data & Data Representation: Pink, Yellow, Orange

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.

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.
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

Computational Thinking Strands

AL – Algorithmic Thinking

Ref.	Activity
A1	Writing instructions that if followed in a given order (sequences) achieve a desired effect
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);
A13	Creating algorithmic descriptions of real world processes so as to better understand them (computational modelling);

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);

EV – Evaluation

Ref.	Activity
E1	Assessing that an algorithm is fit for purpose;
E2	Assessing whether an algorithm does the right thing (functional correctness);
E3	Designing and running test plans and interpreting the results (testing);
E4	Assessment whether the performance of an algorithm is good enough;
E5	Comparing the performance of algorithms that do the same thing;
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);

Learning Outcomes

1. To understand what an algorithm is
2. Be able to follow an existing algorithm in order to test the outcomes
3. Be able to write an algorithm to achieve a desired result
4. Be able to make simple suggestions to help improve the effectiveness of an algorithm
5. To be able to represent an algorithm as a flowchart
6. To understand what is meant by a 'procedure'
7. To be able to include the use of procedures when writing algorithms
8. To understand the concept of repetition and be able to recognise opportunities for repetition within algorithms
9. To understand the concept of selection and be able to implement selection statements within an algorithm
10. To be able to evaluate the effectiveness of an algorithm
11. To be able to implement a pre-written algorithm within a programming language such as Scratch
12. To be able to able to create a dance sequence in Scratch using sequence, selection and repetition.
13. To be able to edit and import costumes into Scratch
14. To be able to improve the effectiveness of a program by implementing procedures.

Session Overview

SESSION 1

Session Content / Activity	Resources Used	Prog. Pathway	Comp. Thinking	Computing POS Link
Welcome, Introductions General information about the day, including any Health and Safety information. Begin with some ice breaker activities	DSH_WelcomeIntroduction.ppt			
Show "The Haka – New Zealand Vs Tonga" video Discuss: why is everyone able to follow the dance so easily? Can you do the Haka? What are the steps involved?	The Haka.mp4	ALG PA1	A1	2.1
Introduce topic for the day. Do "A Thunk" activity	TeachingDance.pptx			
Do the 21 Card Trick – as described in the CS4FN magic book from page 6 onwards. Turn it into an elaborate show. Ask the students if they can work out how the trick works	Magic Book.pdf	ALG PA1,	Ab3	2.1
Display slide 5 – tell pupils to work in pairs to repeat the trick following the instructions. Did it work? Was there enough detail? Were the instructions specific enough? Move onto slide 6 – it might be worth giving pupils a few minutes to study the algorithm and consider their answers in pairs/small groups	TeachingDance.pptx	ALG PA1 – PA3, YA3, YA4, OA3	A1, A13, Ab1, Ab2, E1, E2, E3, E4, E10	2.1, 2.3
Class feedback will hopefully lead to some pupils suggesting that instructions 3 -5 are repeated 3 times. Use this to introduce the concept of repetition. Show slide 6 and then 7 to show how using repetition in our programming we can shorten our algorithm while making it easier to understand	TeachingDance.pptx	ALG PA1 – PA3, YA1- YA4, OA1 – OA4	A1, A5, A13, Ab3, E1	2.1, 2.2, 2.3

Slide 7 can lead to greater discussion of the flowchart – abstraction has been used to hide some of the specific details; but what are these details? You may want to ask students to consider these and attempt to write them out. This will hopefully lead to them developing the idea of procedures; you can guide them in this direction if you wish or work with whatever they suggest.	TeachingDance.pptx MiniWhiteboard.pdf	ALG PA1 – PA3, YA1- YA4, OA1 – OA4	A1, A6, A13, Ab1, E1, E2	2.1, 2.3
Work through slides 10 – 18. Slide 12 shows a short (14 sec) video clip of the Hokey Cokey dance. These slides will reinforce the idea of sequence in algorithms, representing them as diagrams and procedures. Judge your level of discussion based on students abilities. For some students they may find these activities easy and will simply allow them to consolidate their learning from the 21 card trick activities. However, for other students this will reinforce their learning and understanding of the key concepts. Vary the time spent on this according to the needs of your students. It is important that they understand the key ideas being introduced in this session.	TeachingDance.pptx	ALG PA1 – PA3, YA1- YA4, OA1 – OA4	A1, A4, A5, A6, A13, Ab1, Ab2, Ab3	2.1, 2.2, 2.3
Work through slides 19 – 22. This part of the activity begins with a video clip of a dance sequence. You can use one of the videos included in this pack or pick one of your own. Selecting a music video that is currently popular amongst the pupils can be a good way to raise motivation and the 'fun factor'. However, be careful when choosing your video to ensure the sequences aren't too complicated. The students will need some space to carry out these activities. You can move them to an appropriate venue, or restrict their movements.	TeachingDance.pptx Video selection	ALG PA1 – PA3, YA1- YA4, OA1 – OA4	A1, A4, A5, A6, A13, Ab1, Ab2, Ab3	2.1, 2.2, 2.3

SESSION 2

Session Content / Activity	Resources Used	Prog. Pathway	Comp. Thinking	Computing POS Link
In this session you will be working through slides 23 – 31. This section of slides introduces two important concepts: repetition and selection. Repetition and the implementation of loops will already have been introduced to students through the 21 Card Trick, but it will be reinforced here. Judge the time spent by the needs of the students. It is worth spending time allowing them to write and refine their algorithms. Use a simple video recording device (i.e. Flip Camera) to record the dance performance of each group. At the end of the session it is important that students present their algorithm and their dance to the rest of the class. Use the feedback sheet to enable the class members to record their feedback. There is room for a 6th criterion – use suggestions from the class to identify a good 6th criterion for everyone to use (or perhaps allow each pupil to add their own – it will create useful discussion at the end about the value of different feedback criteria to judge an algorithm).	TeachingDance.pptx MiniWhiteboard.pdf Dance Feedback Worksheet (doc & pdf)	ALG PA1 – PA3, YA1 – YA4, OA1 – OA3	A1, A4, A5, A6, A13, Ab1, Ab2, Ab3, E1 – E5, E10, E15	2.1, 2.2, 2.3
End the session by discussing how pupils will now be moving onto Scratch to create a program of their dance routines. Gather information about pupils prior knowledge of Scratch. Introduce them to the environment and basic skills. If there is enough time, you can ask students to 'explore' Scratch; ask them to 'discover' or 'find out how to…' use various features within Scratch. The pupils can then demonstrate what they have discovered to the class using the IWB.		P&D PP1, PP3, YP2, YP3, OP1,	A1, A4, A5, A6	2.1, 2.2

SESSION 3

Session Content / Activity	Resources Used	Prog. Pathway	Comp. Thinking	Computing POS Link
The main focus of this session is creating the Scratch programme to implement the dance routine. It will involve working through Slides 32 – 48 in the slideshow, at a pace that suits the class. Using the order suggested, use a pedagogy that suits the needs of the class to work through the refinement of their program. It is a good idea for pupils to try and implement the algorithm they wrote in Session 2. You may wish to give them time to improve their algorithm based on the class feedback given. The help videos demonstrate different elements of Scratch and these should be made available to pupils for them to view at their own convenience. You may wish to allow for time at the end of the session to allow pupils to view the completed outcomes. Suggestion If time and facilities allow, the following variation is a good one to implement. Take photos of students as they perform their dance algorithm. If they are implementing their dance algorithm from session 2, then these could be done during their performance. Or alternatively, take photos of students as they pose. These would then be imported into Scratch and used as costumes for their program. Add a variable which allows the user to select how many times the routine is performed.	TeachingDance.pptx Help Videos: • Video 1 D...mp4 • Video 2 D...mp4 • Video 3 D...mp4 • Video 4 D...mp4 • Video 5 D...mp4 • Video 6 D...mp4 • Video 7 D...mp4	P&D PP1 – PP3, YP1 – YP3, OP1 – OP3	A1, A4, A5, A6, A13, Ab1, Ab2, Ab3, E1 – E5, E10, E15	2.1, 2.2, 2.3

Files/Resources

Filename	Resource Type	Purpose/Description
WelcomeIntroduction.ppt	PowerPoint	PowerPoint to be used at the start of the day
TeachingDance.pptx	PowerPoint	Main teaching PowerPoint
MiniWhiteboard.pdf (.docx)	Photocopiable	To be printed and laminated for use as a mini whiteboard
Dance Feedback Worksheet.docx (.pdf)	Worksheet	A worksheet to be used by the pupils
Video 1 Dance – Import Costumes.mp4	Video Tutorial	Video tutorial providing support on Scratch programming
Video 2 Dance – Rename.mp4	Video Tutorial	Video tutorial providing support on Scratch programming
Video 3 Dance – Sequence.mp4	Video Tutorial	Video tutorial providing support on Scratch programming
Video 4 Dance – Procedure.mp4	Video Tutorial	Video tutorial providing support on Scratch programming
Video 5 Dance – Repeating Procedure.mp4	Video Tutorial	Video tutorial providing support on Scratch programming
Video 6 Dance – Selection.mp4	Video Tutorial	Video tutorial providing support on Scratch programming
Video 7 Dance – Duplication & Backgrounds.mp4	Video Tutorial	Video tutorial providing support on Scratch programming

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.