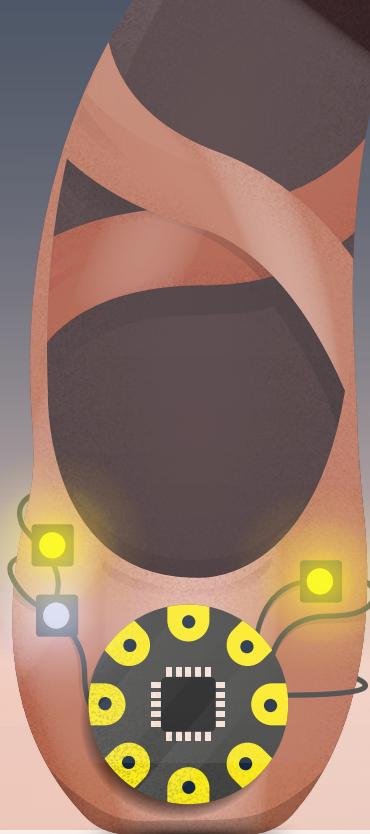


(Hello World)

Issue 9 | July 2019 | helloworld.cc

DATA DRIVEN DANCE

Coding meets ballet



THE MAKER MOVEMENT

Digital fabrication enters the art studio

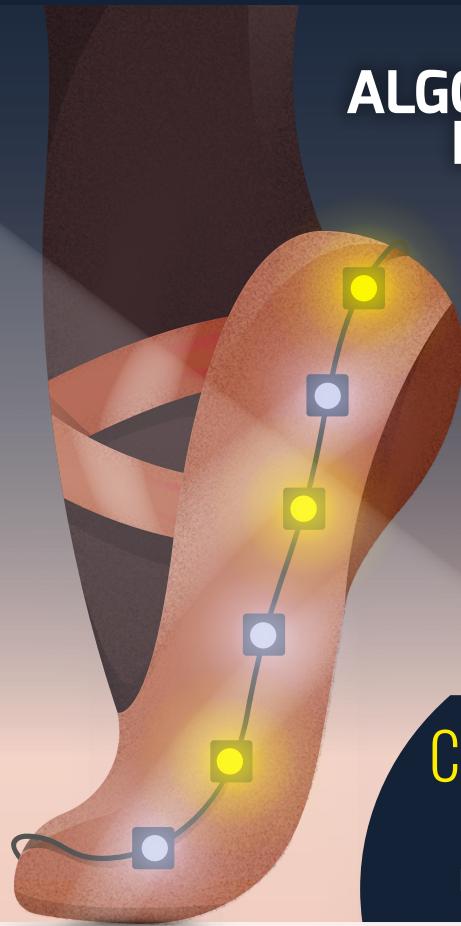
THE MAGAZINE FOR COMPUTING
& DIGITAL MAKING EDUCATORS

PRIMARY CODE CLUBS

Three educators share their benefits

ALGORITHMIC DRAWING

Activities for
all ages



CLASSROOM DISPLAYS

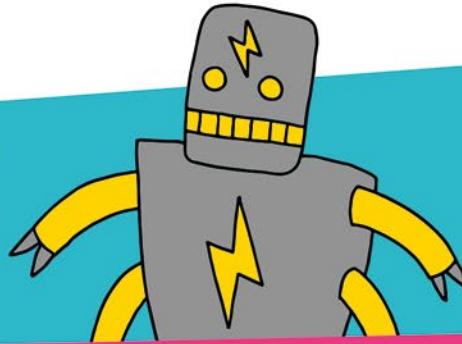
Inspiration and tips

COMPUTING & THE ARTS

Stories of inspirational projects and activities to engage your students

PLUS

SUMMER EVENTS FOR STUDENTS • DATABASES IN PYTHON • VIDEO GAMES IN CLASS • ROEHAMPTON REPORT
SCRATCH CONFERENCE EUROPE • THE LITTLE BOOK OF ALGORITHMS • CSFORALL • GENDER RESEARCH

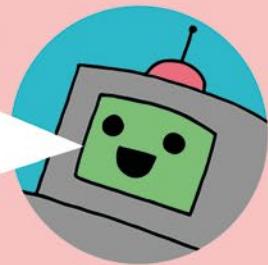


START A CODE CLUB IN YOUR SCHOOL!

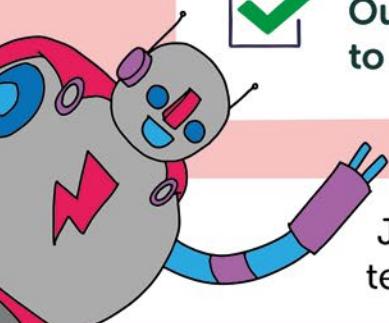
It's easy to get your school coding! Code Club supports educators to run coding clubs for 9- to 13-year-olds with free, step-by-step project guides for learning Scratch, Python, and HTML/CSS.

"I run a Code Club in our school for children in Year 5 and 6, which is always popular. Code Club activities encourage learning independence, and the children love to share the games they've coded with each other."

Jill, Teacher



- Code Club is free**
- Code Club offers engaging, hands-on activities that have been designed to spark curiosity and inspire creative thinking**
- Our free online training is perfect for teachers who are new to coding or are starting a Code Club for the first time**



Join the network of over 7000 clubs across the UK teaching more than 100,000 young people to code!

Get involved at codeclub.org





SUBSCRIBE
IN PRINT
FOR FREE
TURN TO PAGE
40

HELLO, WORLD!

Last month marked 60 years since C.P. Snow's influential Two Cultures lecture, in which he lamented the widening gap between the arts and sciences. And, in England at least, students are still sorted into disciplinary silos. A recent report from Cambridge Assessment found that in 2017, only 7% of students combined arts and science subjects in their A-level choices. What restrictions does this place on the next generation's views of the world and their ability to face societal challenges?

Computing is an inherently creative subject – every time a student writes a program or makes a device, they create something new. It offers a bridge between the technical and artistic worlds. I'm really excited that in this issue of *Hello World* we have been able to showcase some brilliant interdisciplinary projects, as well as ideas for creative

computing activities that you can try out with your students.

I'm also really happy to say that this issue of *Hello World* features more contributions from educators than ever before, as well as an unprecedented number of pages dedicated to lesson plans and activities. This reflects the feedback received in our 2018 reader survey and I'd love for these trends to continue. So, if you have an idea for a feature or lesson plan that we could include in the magazine, please do get in touch at

contact@helloworld.cc

or on Twitter

@HelloWorld_Edu.

We look forward to hearing from you!

Sian Williams Page

Editor



FEATURED THIS ISSUE



AISHA FLORES

Aisha is an Electronics Engineer, Developer, and the Founder of the Buttercup STEAM Camp in Atlanta, Georgia. Turn to page 30 for her discussion of how Buttercup STEAM aims to reduce the gender and racial gap in technology professions.



ERIN E. RILEY

Erin is Director of the Engineering and Design Lab at Greenwich Academy, a K-12 school for girls in Connecticut, and author of *The Art of Digital Fabrication*. She introduces projects that explore the intersection of coding, art, and digital fabrication on page 22.



PAUL CURZON

Paul is a Professor of Computer Science at Queen Mary University of London and co-founded both cs4fn and Teaching London Computing. Together with Jane Waite and Ged Gast, he explores the links between computer science and the arts on page 14.

EDITORIAL

Publishing Director

Russell Barnes
russell@raspberrypi.org

Editor

Sian Williams Page
sian@helloworld.cc

Contributing Editor

Miles Berry
miles@helloworld.cc

Sub Editors

Nicola King and Phil King

Subscriptions

Joshua Crossman

DESIGN

Critical Media

criticalmedia.co.uk

Head of Design

Lee Allen

Designers

Harriet Knight and Sam Ribbits

Photography:

Brian O'Halloran and Fiacre Muller / Raspberry Pi Foundation
Adobe Stock Photo

Cover illustration

Sam Alder

Graphics

Jack Willis

CONTRIBUTORS

Sam Aaron, Miles Berry, Giles Booth, George Boukeas, Mac Bowley, Tom Bowtell, Tom Bromwich, Ceredig Cattanach-Chell, Katherine Childs, Paul Curzon, Rebecca D'Cruz, Leigh Ann Delyser, Tom Evans, Aisha Flores, Lucia Flóriánová, Ged Gast, Lorna Gibson, Sway Grantham, Will Grey, Elliot Hall, Katie Henry, Deirdre Hodson, Panagiotis Kampylis, Peter Kemp, Fergus Kirkpatrick, Michael Költing, William Lau, Dylan Lewis, Dominic McKoy, Alan O'Dohnoe, Martin O'Hanlon, Eleanor Overland, John Parkin, Michelle Paull, Tamara Pearson, Simon Peyton Jones, Oliver Quinlan, Neil Rickus, Erin E. Riley, Laura Sach, Katie Sanders-Pope, Sue Sentance, Martin Sexton, Genevieve Smith-Nunes, John Stout, Nina Szymor, Scott Turner, Thomas Turner, Jane Waite, Matt Warne, Sian Wheatcroft, Gemma Wilson, Susan Wright

Supported by **ORACLE**

Hello World is a joint collaboration:



Raspberry Pi



This magazine is printed on paper sourced from sustainable forests and the printer operates an environmental management system which has been assessed as conforming to ISO 14001.

Hello World is published by Raspberry Pi (Trading) Ltd, Maurice Wilkes Building, St. John's Innovation Park, Cowley Road, Cambridge, CB4 0DS. The publisher, editor, and contributors accept no responsibility in respect of any omissions or errors relating to skills, products or services referred to in the magazine. Except where otherwise noted, content in this magazine is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0).



(HW) CONTENTS

THE MAKER MOVEMENT
Digital fabrication enters the art studio

22

COVER FEATURE

CREATIVE MICRO:BIT
How the device is breaking down barriers

28

A BOOK BROUGHT TO LIFE
An author shares her inspiring creation

26

ART IN MOTION
A physical computing project

34

NEWS, FEATURES, AND OPINION

6 NEWS

Funding for gender balance in computing, Tate Exchange, MIT Solve, and more

14 COMPUTING AND THE ARTS

How the two subjects are closer than they appear

18 BALLET MEETS CODE

Performances driven by data

20 DIGITAL GHOST HUNT

Immersive theatre for digital education

30 BUTTERCUP STEAM

Summer camps to reduce the gender and race gap in tech

32 ALGORITHMIC ART

Scratch activities based on the artist Sol LeWitt

36 SONIC PI

The inspiration behind the music creation tool

39 AN OVERHAUL IN WALES

A teacher from a Pioneer School shares his thoughts on the new curriculum

40 SUBSCRIBE IN PRINT TODAY!

42 INSIGHTS

Videos and self-explanation, and data literacy

45 ROEHAMPTON REPORT

The key findings with responses from educators and academics

52 PI WARS

A young team's teacher shares their experience



- 54 PRIMARY CODE CLUB**
How one teacher made it work
- 60 SPIRAL CURRICULUM**
Planning for progress
- 62 WRITING CODE**
Strategies to support students
- 64 DIGITAL EVALUATION**
The SELFIE tool from the European Commission

- 66 CSFORALL**
The systems supporting students
- 70 LITTLE BOOK OF ALGORITHMS**
A new free resource
- 87 RECURSIVE DOODLES**
Sketching and sorting algorithms
- 90 INSIDER'S GUIDE**
Inspiration for your classroom displays

CONVERSATION

- 94 YOUR QUESTIONS**
Answered by the Learning Team at the Raspberry Pi Foundation

- 95 LETTERS**
Feedback and thoughts from Hello World readers

REVIEWS

- 96 BOOKS ROUNDUP**
New book recommendations right this way

LEARNING

RESOURCES & LESSON PLANS

- 72 AN INTRODUCTION TO ALGORITHMS**
Painting at primary
- 74 THOMAS'S TANGLES**
A two-player game with a twist
- 76 BEETHOVEN MEETS SONIC PI**
A cross-curricular music lesson
- 78 FABRIC DESIGN**
Explore repeating patterns in Scratch
- 80 HACKING WEB CONTENT**
Introduce learners to HTML via fake news
- 82 PIXEL ART**
An unplugged activity
- 84 A FIRST DATABASE**
A scaffolded lesson in Python



The project hopes to build on the successes of informal learning contexts, such as Code Clubs, which often have a better gender balance than computing qualifications offered at school

A BOOST FOR GENDER PARITY

The Gender Balance in Computing Project has been awarded £2.4 million to investigate how to engage more girls in computing

Sue Sentance and Oliver Quinlan

It is well-known that computer science and other STEM subjects have a gender balance problem. Decades of research have been carried out in this area, looking at why women don't choose to specialise in computing-related subjects at school and university, or pursue related careers.

A variety of factors are likely to be responsible for girls choosing not to study computing, including not feeling they belong in the subject or its community, and not having any encouragement from parents or teachers.

In this context, we are delighted that the Department for Education in England has funded a new programme of research to investigate which interventions may be effective in school to both engage female students and to increase numbers selecting computer science at GCSE and A-level. Gender Balance in Computing (GBIC) is a collaboration between the consortium of the

Raspberry Pi Foundation, STEM Learning, BCS, The Chartered Institute for IT, and the Behavioural Insights Team, known for its work on 'nudges' to encourage certain behaviours. Operationally, the project will be led by the Raspberry Pi Foundation, with Apps for Good and Women in Science and Engineering (WISE) also working on the project. Trials will run from 2019–2022 in Key Stages 1–4, and over 15,000 students and 550 schools will be involved. It will be the largest national research effort to tackle this issue to date.

The interventions

There will be a number of interventions taking place over the next few years. It has been shown that interventions in this area should focus on a specific problem to be solved, and that is therefore the approach of the project. For example, girls outperform boys in computer science, but may be less likely to be encouraged

to continue with computing education or to take it in the first place. This lack of encouragement from parents, carers, and teachers may stem from negative stereotypes and false perceptions of career pathways. Therefore, one of the project's interventions will investigate whether a combination of encouraging influences, including teachers and parents, increases the number of girls choosing to study computer science.

Another intervention will focus on the fact that female students don't feel they belong in computing classes because they don't see other girls in those classes. Computing classes at GCSE and A-level, which historically have few or no girls taking the subject, may deter girls in lower years from selecting the subject. This can create a cycle of low participation because there is no 'critical mass' of girls participating and being the only one, or one of a few girls, studying computing may be off-putting.

- The project's interventions include trialling different pedagogical approaches, such as pair programming



GENDER RESEARCH IN STEM

The Gender Balance in Computing Project interventions have been developed on the basis of existing research on the factors affecting of girls' participation in other STEM subjects, as well as wider research on the influences on students' choices of subjects.

Key papers include:

- DeWitt, J., Archer, L., & Osborne, J. (2013). Nerdy, brainy and normal: Children's and parents' constructions of those who are highly engaged with science. *Research in Science Education*, 43(4), 1455-1476. (helloworld.cc/DeWitt2013)
- Good, C., Rattan, A., & Dweck, C. S. (2012). Why do women opt out? Sense of belonging and women's representation in mathematics. *Journal of personality and social psychology*, 102(4), 700. (helloworld.cc/Good2012)
- Hulleman, C. S., & Harackiewicz, J. M. (2009). Promoting interest and performance in high school science classes. *Science*, 326(5958), 1410-1412. (helloworld.cc/Hulleman2009)
- Goode, J., Estrella, R., Margolis, J. (2006). Lost in Translation: Gender and High School Computer Science. In Aspray, W. & Cohoon, J. M. (Eds.) *Women in IT: Reasons on the Reasons of Under-Representation*. Cambridge, MA: MIT Press (helloworld.cc/Goode2006)

THERE IS AN INCREASE IN GIRLS' MOTIVATION TO STUDY COMPUTING WHEN IT IS LINKED TO REAL-WORLD IMPACT

The results will hopefully tell whether girls are more likely to self-identify as a computing student after being exposed to peers or role models they can identify with.

There is also evidence that there is an increase in girls' motivation to study computing when it is linked to real-world impact. To address this, one intervention will see classes undertake contextualised projects where they create technology to solve real-world problems. Other initiatives include trialling more inclusive pedagogical approaches to teaching computing, including collaborative approaches such as pair programming, and the use of storytelling with Key Stage 1 children. One intervention involves extra-curricular contexts, where a better gender balance exists. It is possible that relating informal learning opportunities to computing as an academic subject or career choice could increase engagement with the subject.

The team bring a variety of skills and knowledge to this endeavour. The Raspberry Pi Foundation directly engages over 200,000 young people each week in networks of free computing clubs and competitions. The organisation has a track record of effectively engaging girls,

evidenced by the fact that 40% of the young people currently in Code Clubs are girls.

The Behavioural Insights Team has a proven track record of delivering high quality work in education, including delivering interventions with schools, colleges, and higher education institutions, and with young people – about their future careers and aspirations, the resources they draw upon to make these decisions, and the context in which decisions occur.

Apps for Good is an award-winning technology education charity with which young people learn how to create, build, market, and pitch innovative tech products to solve real-world problems they care about. Apps for Good has a strong track record of engaging girls in technology and helping to change their perceptions of computing: each year, at least 50% of the students taking an Apps for Good course are girls.

Women in Science and Engineering (WISE) runs large-scale programmes in primary schools, secondary schools, and colleges across the UK, with good evidence of impact on gender participation, in computing specifically and STEM more broadly.

The project is in its infancy and the interventions are 'pilots' because they

will be implemented with a sample of schools, with a focus on finding out which approaches have long-term impact. The trials will be run in a range of schools over a wide geographic area in England, and rolled out through the National Centre Computing Hubs.

If you think your school may be interested in participating in a trial, then complete an interest form at ncce.io/gbicregister. Updates of the project's progress, including reports describing the results of each intervention, will be shared at teachcomputing.org. The results of the project should enable schools to apply techniques in their own contexts to improve gender balance in computing. (HW)



AN INSPIRING SUMMER OF EVENTS

Isaac Computer Science is working with universities across England to host a series of events that encourage students to study Computer Science at university

Fergus Kirkpatrick

A new series of events designed to inspire students to study Computer Science degrees begins across England this summer. The Isaac Computer Science programme, which starts this July, includes a mixture of Discovery events, masterclasses, and revision sessions for students, as well as professional development days for teachers. The aim of the programme is to spark students' interest in the subject, stretch their attainment, and increase teachers' expertise at A-level. All of the events are free to attend, with the hope that as many teachers and students in England will take part in the programme as possible.

Events will be held at universities across the country, with the first dates announced at University of Bath, the University of



Cambridge, Newcastle University, Queen Mary University of London, and the University of Southampton. As part of the new National Centre for Computing Education, Isaac Computer Science is partnering with these institutions, and many more, to develop a full programme of events over the next three years.

Regional coverage and social mobility

A key aim of the Isaac Computer Science programme is to achieve a wide geographic reach for the events programme. The latest UCAS End of Cycle report 2018 illustrates the large regional disparities in students participating in higher education. According to UCAS, the proportion of university-age students participating in higher education from the most advantaged areas is 46.7%, compared to 19.7% in areas considered to be the most disadvantaged. Research from the BCS, the Chartered Institute of IT, has shown that careers in IT offer amazing potential for social mobility and the demand for computer science skills is

ISAAC COMPUTER SCIENCE EVENTS IN JULY 2019

1 JULY	Discovery Event, University of Cambridge
3 JULY	Teacher CPD, University of Bath
8 JULY	Teacher CPD, University of Southampton
8 JULY	Discovery Event, Queen Mary University of London
9 JULY	Teacher CPD, Newcastle University
11 JULY	Student masterclass, Newcastle University
16 JULY	Teacher CPD, Newcastle University

increasing in all parts of the country. This suggests that there is an opportunity to reach students across the country and encourage them to consider degrees that offer rewarding career opportunities, such as Computer Science.

Many of the career options open to students who choose to study Computer Science will be showcased at the Discovery events. Students will also hear what it is like to study Computer Science at university from current students and recent graduates. Some Isaac Computer Science events are planned with potential employers, including a Discovery event in Manchester in November hosted by PwC.

By partnering with universities with extensive regional networks and subject expertise, the Isaac Computer Science events programme aspires to achieve the necessary scale to have reach across the country, and build a network to support the mission of inspiring more students to study Computer Science at university.

Find out more

Registration is now open for the Isaac Computer Science events happening in July at helloworld.cc/isaac_events.

The Isaac Computer Science online learning platform is now available to teachers and students for use in the classroom and at home, at isaaccomputerscience.org. It is a free and accessible platform to support post-16 studies in Computer Science. (HW)



London Connected Learning Centre's annual Tate Exchange project

PRIMARY STUDENTS GET CREATIVE WITH TECH AT TATE MODERN

Hundreds of London primary students gathered at the annual Tate Exchange project

Michelle Pauli

It's a weekday in the spring term and, across level five of Tate Modern on London's South Bank, the hubbub of 120 primary schoolchildren can be heard. Some are making puppets for an animation, some are creating a Scratch-based soundscape, while others are working with an illustrator and app developer to fly a flock of birds across a screen. Some children have been engrossed in their tasks for hours, working with children from other schools they have only met that morning; some are wandering between activities, getting a flavour of each. All are seamlessly combining the digital and physical, the technical and the artistic – putting STEAM into action.

That was the scene at Tate last month when the London Connected Learning Centre (CLC), which supports schools in using digital technologies to enhance learning, held its annual Tate Exchange

project. It gives children the opportunity to interact in new ways with a huge variety of physical and digital objects in the context of the Tate space, interweaving the real and the virtual, connecting them to the artistic content of the gallery as they find their space in a community of artists. Tate Exchange is an ambitious 'open experiment' that allows organisations and members of the public to participate in Tate's creative process, using art as a way of addressing wider issues in the world. For three years now, London CLC has been using the opportunity to bring together schools, technology, and the arts in innovative ways.

This year the theme was 'movement' and, over two days, more than 200 children and teachers from local primary schools explored the relationship between natural and manufactured movement. The children programmed computers and

devices to create different types of physical movement, as well as using digital tools to create and capture animation. They were encouraged to investigate their own movement, as well as the power of physical and digital movement in relation to questions such as: what does it mean to give something movement? Can we use the movement of our bodies to produce an output? How can physical and digital/virtual movement intersect?

According to Sarah Horrocks, co-director of London CLC, "What is really exciting about this collaboration with Tate is how it encourages the children to think and make differently. It allows children and teachers alike to be more experimental, as having the freedom of Tate's rich galleries and artwork opens up a learning experience distinct from the normal school world, and provides plenty of space for trying things out and





■ Access to equipment and technology that aims to teach and inspire



■ Interactive and hands-on learning - technology in action!

"FOR MANY OF THE CHILDREN, IT WAS THE FIRST TIME THEY HAD EVER BEEN INSIDE AN ART GALLERY OR MUSEUM"

► developing ideas. It also helps to break down that oddly false dichotomy between the technical and the artistic."

The activities on offer ranged from using software to create sonic soundscapes triggered by body movement, to using Scratch and video motion sensors to draw patterns on screens with their movements. Children's illustrator Joe Lillington, who has created an app, Flocc, inspired by bird murmurations, encouraged the children to explore bird movements using programming in Scratch, sketching and making bird silhouettes that were then animated. Children also drew inspiration from three artworks in Tate's main gallery.

Creativity is at the heart of it all, regardless of the medium, says Horrocks. "What was perhaps unexpected was that the children tended not to make any distinction between digital and analogue activities. They were making music or they were making things happen on the computer, or they were making films, making images move – it was all, ultimately, about creation."

The setup allows the children to choose which activities they do and for how long,

an approach that London CLC draws on in all its cultural partnership projects. While that might seem to offer a dangerous freedom to flit, it also facilitates focus.

"There's a boy who was on Scratch all morning trying to make an animation, which is probably more time than we would ever give them in school to have a go at something. They have a long time to work at something and perfect it," explains Luke Parker, Assistant Head of Hitherfield Primary School.

A FUN WAY OF LEARNING

Or, as a child puts it, "It was fun because you got to create it by yourself and you didn't have anyone telling you how to do it."

Each workstation provides the children with an activity that they can work on and they learn through making and creating goals for themselves, scaffolded and supported by adults. The result is a phenomenal level of involvement and creativity. "What I notice as I walk around is that every child is 100% engaged in the activity they are doing. Everyone's really focused," says one of the class teachers. "With the activities, the children are using

so many different skills: there's a lot of problem-solving going on, children are learning new digital skills and working together collaboratively to do that. They are taking risks with some of the things that they do, a lot of it is new to them and they are just trying things out to see what happens – and that's the essence of creativity. It is an essential skill for life."

For London CLC and Tate Exchange, context is critical. For many of the children, it was the first time they had ever been inside an art gallery or museum, let alone one as physically imposing as Tate Modern. To be creating their own work in that

A SCRATCH PROJECT

One of the great things about Scratch 3 is how many of its exciting features have been bundled together in the handy block extension section. This can act as a series of signposts to guide you and your pupils to some interesting programming possibilities.



Dedicated workstations, where children can work on activities

space opened up a sense of ownership and possibility that simply viewing the art already in there could not do.

"One of the children said to me, 'I thought that we would just be walking round and looking at things, I didn't realise that we would be making our own art.' The combination of going and looking at what other people have created and then creating things yourself is very powerful," comments Horrocks.

This year London CLC also held its creative arts forum, a regular CPD session for arts educators with differing levels of digital experience, at the event, enabling them to observe the pupils, hear from visiting practitioners and creative industry experts, explore how digital technologies can support creative arts teaching and, after hours, get hands-on with the Tate Exchange activities the children were doing earlier in the day.

Of course, not every school is on Tate's doorstep. Yet Horrocks is keen to emphasise that, while schools in London have an exceptional level of access to rich cultural experiences, arts/cultural and digital collaborations can take place in any cultural organisation, however small or specialised.

"Even the smallest of towns has some kind of local museum or gallery, and it doesn't matter what size it is or what its focus is on, there will be an opportunity for arts/tech exploration," says Horrocks. "We also hold workshops for Reception and Year 1 children in the Garden Museum, where the pupils investigate the space using a variety of technologies, from magnifying glasses and binoculars, to digital microscopes and VR headsets, and then document their findings digitally. The key thing is to get in there and take up opportunities. For the children involved they can be life-changing."

AN INSPIRING EXPERIENCE

During the Tate Exchange project, the children made brilliant use of the video motion-sensing extension. They were invited to explore and adapt a script that made an invisible sprite which, on coming into contact with camera movement, could magically draw a pattern on the screen. You can try out the project by clicking the green flag, enabling your camera, and moving around to trigger their different designs to appear: helloworld.cc/tatescratch.

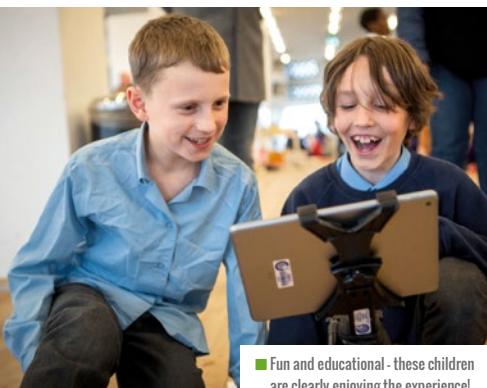
Fun and educational - these children are clearly enjoying the experience!

LIFELONG KINDERGARTEN

London CLC is a firm supporter of Scratch, created by Mitchel Resnick's MIT Media Lab research group. The free choice model for the Tate Exchange programme aligns with Resnick's key 'lifelong kindergarten' principles of projects, passion, peers, and play.



Creativity is a key focus, in an exciting and stimulating environment



Project-wise, each workstation provided the children with an activity that they could work on, and they learned through scaffolded making and creating goals for themselves. Allowing children to self-select activities means that they can follow their passions and their interests. In terms of peers, the Tate Exchange experience sees children from a number of schools across London come together to experience the event. Throughout the day, they are encouraged to interact with their peers from other schools, sharing ideas and creating things together. Finally, the activities are designed to allow pupils to be playful and imaginative in their learning or, as Mitchel Resnick puts it, to be 'designing, creating, experimenting, and exploring'. (HW)



■ Purvi Shah pitches StoryWeaver during the Teachers & Educators session at Solve Challenge Finals in 2018.
Photo © Adam Schultz MIT Solve

INNOVATORS MEET AT MIT SOLVE

A global community gathers to tackle challenges in 21st century education

Nina Szymor

MIT Solve is an initiative of the Massachusetts Institute of Technology that identifies and supports the development of solutions to the world's most pressing problems. Each year, Solve sets a number of 'challenges' and invites entrepreneurs and organisations from around the world to propose innovative approaches to tackle them.

Last month, as part of the Flagship event held in Cambridge, Massachusetts, the Educators & Teachers challenge posed the question 'how can we provide accessible, personalised, and creative learning experiences for all?' Participants were asked to explore how technology can amplify the work of teachers, support new educators and learning facilitators, and help them bring better-quality education to their students.

Proposed solutions to this challenge included a multilingual communication platform to enable parent-teacher engagement (TalkingPoints), technology that reduces administrative tasks and gives

special education teachers the tools they need to focus on their students (Education Modified), a digital platform that provides free access to multilingual books for children (StoryWeaver), software that enables non-verbal, disabled children with learning

impairments to communicate (Livox), an AI-based platform that personalises learning to overcome the traditional one-size-fits-all model of education (CENTURY Tech), and many more. Read about the challenge and its solutions here: helloworld.cc/mitsolve. (HW)



■ Noëlla Coursaris Musunka, founder of Malaika, a nonprofit organisation that empowers Congolese girls and their communities through education and health programs, writes notes during the Teachers & Educators session at Solve Challenge Finals in 2018.
Photo Adam Schultz MIT Solve



Scratch Conference Europe is a great opportunity to increase your skills with the educational programming language

JOIN SCRATCH CONFERENCE EUROPE

Tickets are now available for the first event of its kind in the UK, which will take place in Cambridge on 23–25 August 2019

Tom Evans

Scratch Conference Europe is coming to the UK for the first time this August. The event provides a unique opportunity for teachers, education professionals, and volunteers to come together and discuss creative ways to use the Scratch language, start new collaborations, and take new knowledge back to their classrooms, coding clubs, and communities.

Over the course of three days, the Raspberry Pi Foundation will be hosting Scratch Conference Europe on the campus of Churchill College, Cambridge. Arriving on Friday 23 August, participants will have the opportunity to meet one another while trying out a series of Scratch project demonstrations by creators from both the UK and abroad. Saturday and Sunday will be packed with a huge variety of activities to help guests deepen their understanding of the Scratch language, see the latest innovations in the community, and connect with each other.

A unique conference for educators

Scratch Conference Europe will bring together educators and enthusiasts from all over the world to explore the very best ways of using the educational programming language – both inside and outside the classroom. The event aims to provide tremendous benefits to educators working in schools, as well as to volunteers running educational clubs. Guests will have the chance to interact directly with experts from the Scratch Foundation and discuss their experiences of using Scratch in educational settings.

The presentations will cover a wide variety of subjects and approaches to using Scratch. The packed schedule features talks, discussions, workshops, and poster sessions. Guests can look forward to discovering and discussing everything Scratch-related, from using Scratch alongside hardware, to approaches to introducing the language to new audiences, to innovative uses for it in the classroom, and much more besides. The

recently launched Scratch 3.0 will also be covered in detail.

Session leaders include members of the Scratch Foundation team themselves and Scratch experts from across the globe. The keynote speaker is Scratch founder Mitchel Resnick, Papert Professor of Learning Research at the MIT Media Lab.

A friendly community atmosphere

With most of the conference accommodation on site, guests will have plenty of opportunity to engage with peers, both during the days' scheduled activities and also in the mornings at breakfast in the dining hall and at evening meals. Entertainment will be provided on-site in the evenings, and the campus bar (known as The Butterly) will be open until late.

Visit helloworld.cc/scratchconference to find out more. And, to keep up to date with all the latest Scratch Conference Europe announcements, follow the Raspberry Pi Foundation on Twitter and Facebook. (HW)

ART, CREATIVITY, AND COMPUTER SCIENCE

Paul Curzon, Jane Waite, and Ged Gast argue that art and computing have a lot more in common than you might imagine

Both computing and art involve creativity, innovation, and imagination. And people who possess both artistic and computing skills can make wonderful things happen.

Art involves a creative process, and creating an emotionally or intellectually engaging work of art requires more than just skill with a chisel or paint brush and an understanding of the medium. It needs innovation, creativity, and imagination. To most people this may sound a million miles from the stereotype of the computer programmer, whose work is based on logical thinking. The links are deeper than you might imagine, though. Computer science involves great creativity, innovation, and imagination too.

As well as these high-level links in approaches, artists are increasingly using digital media and it opens new opportunities for interactive art for those with programming and electronics skills.



The original iPhone keyboard represented a breakthrough in its ability to have the software step in and help the user

The creative computer scientist

Programming is obviously creative in the loose sense that it involves constructing new things, but also in a more inventive sense. When programming, you are not creating things by rote, not following a fixed plan, but devising something that has never existed before. You may even completely change the way we do things if you creatively see opportunities.

Take the original iPhone keyboard. Prior to its launch, Apple had a problem. The virtual keyboard was unusable. The buttons were just too small. Lots of engineers worked on the problem, trying different keyboard layouts, but nothing worked. With the

launch looming, it looked dire. Without a usable keyboard the product would flop. Then someone thought about it differently. Rather than a different layout, they had the idea of writing a program that would predict which letter was most likely to come next, based on those gone before... and make the area of those keys larger. It worked and the iPhone went on to become one of the biggest ever successes. It took creativity, innovation, and imagination to come up with the product-making program, and skill with the medium to make it work. The best computer scientists

need creativity, innovation, and imagination – just as great artists do.

Art and the machine

Artists now have new media to play with. By using programs and electronics as media, they can create interactive art. For example, Soda Constructor was a program that just implemented a simple 2D line drawing program. However, it added in the laws of physics. Points were masses, lines were springs with rules based on

“ ARTISTS ARE INCREASINGLY USING DIGITAL MEDIA AND IT OPENS NEW OPPORTUNITIES

Newton's and Hooke's laws. Springs could also be turned into muscles that were given periodic energy boosts, making them stretch and contract. Users made a digital drawing and then switched on the laws of physics and their pictures would come alive. You could even make creatures, following all kinds of body shapes and means of locomotion, that moved around the screen. The result was an amazing program that won a BAFTA for interactive art.

Interactive programs create imagery on a computer screen, but digital art can also escape into the real world. Physical



■ The Photogrowth project simulates the behaviour of artificial ants while they travel on a canvas.
Images used with permission of CDV Lab. Source image: Just Be Yourself by Kirsten Sims

installations can be computer controlled, creating, for example, son et lumière (sound and light) shows where sculptures include light and sound effects controlled by sensors. Epic-scale examples include the work of artist Leo Villareal, turning the San Francisco Bay Bridge into a pulsating, ever-changing work of art by covering it in lights controlled by a computer. Di Mainstone had a different take on performance art as she turned bridges into art. She put digital sensors on bridge cables attached by lines to a performer's clothing. As the bridge vibrates to traffic and people, and the performer moves, the angle and length of the lines are measured and different sounds produced. Human and bridge become one augmented artistic instrument.

Another example of the use of electronics and digital creative practice in art is the wonderful Hive sculpture at Kew Gardens. It is a massive architectural-scale sculpture in the shape of a walk-in beehive. It is covered in lights that pulse, controlled by the activity of bees in hives elsewhere in the gardens.

This kind of art often uses very simple combinations of sensors, lights, and sounds with simple computer control. The artist is writing programs as an essential part of the creative process. As an artistic medium, it is now within the reach of school students, combining programming, electronics and art. Similar kinds of art can be made with simple sensor kits such as Arduino, Raspberry Pi, and so on. Of course, to make art requires a creative spark too. ▶



■ Artist Leo Villareal covered the San Francisco Bay Bridge with lights controlled by a computer



PAUL CURZON

Paul Curzon has been a member of CAS since its foundation. He is a Professor of Computer Science at Queen Mary University of London and co-founded both cs4fn and Teaching London Computing. He is author of the book *The Power of Computational Thinking*.



JANE WAITE

Jane Waite has worked both as a primary teacher and in industry as a developer. She worked on the Barefoot programme, was CAS London manager, and now undertakes research and coordinates teacher training and support programmes at Queen Mary University of London. She runs the QMUL CAS London Community of Practice.



GED GAST

Ged Gast is a visual arts specialist, education consultant, and Past President of the National Society for Education in Art and Design.



The Hive at Kew Gardens has a distinctive mesh frame constructed from 170,000 aluminium parts and 1000 LED lights

Algorithmic art

In interactive art, artists are just using computers and electronics as a medium, like a canvas, but in some works algorithms play a more fundamental role, generating the art themselves.

Arguably, artists have been using algorithms for a long time – following rules as part of the artistic process. Perhaps the most stunning examples of this are the geometric images of Islamic art – the word algorithm itself comes from the Persian scholar Al-Khwārizmī and his 9th century book on the algorithms behind our Hindu-Arabic numeral system. We are also all taught simple algorithms in art classes at school, dating back to the Renaissance, such as for getting perspective right. Maths, algorithms and art have been intertwined, at least informally, for centuries.

Taking this idea further leads to algorithmic or generative art. Now, rather than the work coming just from the creativity of an artist, the picture is generated by an algorithm that the artist creates. Different

pictures result because the algorithm has elements of randomness in it, or because the artist tweaks the starting parameters (the inputs to the algorithm). Or, the artist may devise variations on the algorithm, exploring the different kinds of result that emerge from different changes. So in algorithmic art, algorithms themselves become a medium for the artist to work with, just like organising colour and expressive marks on a surface, painting with oils, or constructing in clay. The artist is also providing judgement about which images to keep.

There are lots of places to get inspiration for the algorithms behind algorithmic art. One good source is in the algorithms of nature. A particularly creative example of this, from the University of Coimbra's CDV Lab, is a program called Photogrowth (helloworld.cc/photogrowth). It involves breeding a colony of virtual ants that deposit ink as they crawl over an initial image. The brighter colours of the starter picture act as food sources for the ants, and their trails lead other ants to those sources.

“

ALGORITHMS THEMSELVES BECOME A MEDIUM FOR THE ARTIST TO WORK WITH

If they don't find enough 'food' the ants die. If they do find enough 'food', however, they reproduce, leading to more ants to continue developing the picture. Here, the parameters you can set include things like the thickness of the trails the ants leave.

Many other kinds of algorithms can be used. Our article on algorithmic doodle art in this issue (pages 87-89) gives some simple and unplugged algorithmic art activities for the classroom, using algorithms that mirror the development processes of plants. In a sense, these approaches to computer art take abstract ideas for artistic algorithms. Students may find it intriguing to use this as a means to explore the way in which our minds generate doodles and links with how the rules of abstraction could, or might, function. Another way is to try to create a program that codifies the way human artists work... an algorithm for human art.

The art of AARON

AARON is a robotic painting device created by the artist Professor Harold Cohen. AARON's creations have appeared in art galleries worldwide, including London's Tate Modern gallery. Just as we learn the basics of art in school, AARON has been taught the rules of art, such as composition and perspective. It also knows about things in the world, such as how the parts of a person's body are connected. AARON has another kind of knowledge too: creative procedural knowledge. It knows how to follow the steps to create a painting. For example, it starts on the background of a new painting and works its way forward to the foreground. By following these in-built rules that codify artistic knowledge,

ART & COMPUTING RESOURCES

Teaching London Computing:
helloworld.cc/londonart

National Society for Education in
Art & Design:
helloworld.cc/nsead

Tech Pathways:
helloworld.cc/techpath

AARON creates novel pictures. Each picture AARON paints is different and AARON even chooses the subject to paint.

In creating AARON, Cohen made a great step forward in understanding creativity, and that is a mission researchers interested in understanding computational creativity continue to pursue. Ultimately, work like Cohen's is about writing programs that help us understand what it means to be human.

Computers and judgement

The next step for algorithmic art is to take the human out of the loop and let the computer itself judge which pictures are 'good'. This is a key part of creativity, and feeds back into a human's development as an artist. Great artists don't churn out lots of mediocre paintings, with just odd ones being chosen as brilliant. They get better, they develop a distinctive style of their own. This development as an artist goes hand in hand with that of their judgement and so their creativity. Computer scientists working in the area of computational creativity are exploring ways for algorithms to exercise this kind of judgement.

Ant art and the program Photogrowth explore one kind of algorithm to provide judgement: a kind of artificial intelligence algorithm called a genetic algorithm. This allows the program to take a further step so that the computer itself judges the images it produces and feeds that into a creative process. In this approach, the artist develops a 'fitness function' – an algorithm that gives images a score as to how well they conform to a particular aesthetic. The best images then 'breed': their parameter settings are used to create a new virtual ant colony, but with mutations – slight adjustments – to those winning settings. Over time, the ratings of the surviving paintings improve as only the best survive each round, just as animal species adapt to better survive in their environment.

There are many more links between computing, maths, creativity, and art. And we would argue that the world needs more computer scientists who are trained artists, and artists who are trained computer scientists. **(HW)**



Digital photography is now the only digital art offering at many secondary schools

COMPUTING AND ART IN THE CLASSROOM

Digital technology has been used in art and design classrooms since the mid 1980s, mainly as a tool for digital drawing, manipulating, and reproducing images or objects in different formats. The 1990s and early 2000s saw the greatest range of creative digital practice, with developments in digital video, animation, and even game design. Overlapping with this, the 1999 ICT curriculum included elements of art and design in developing ideas and making things happen, such as with digital photography, multimedia, and animation.

However, recent school budget cuts, an increased emphasis on the computer science elements of computing, and changes to secondary examinations have had a significant negative impact on digital art and design in class. Many secondary schools have phased out the specialist software shared by art and design departments, leaving mostly digital photography as the digital art offering. In computing departments, some new digital media courses are being introduced, such as Creative iMedia and BTEC Creative Digital Media Production.

Some craftspeople and designers use technology as their main creative tool: others use technology alongside traditional processes and techniques, such as the ceramicist Michael Eden. He uses CAD for rapid prototyping, and 3D printing in a plaster and gypsum material with a non-fired ceramic coating to create some of his digital forms. 3D design, multisensory work, and generative art have become part of the art and design landscape. Famous artists such as Bill Viola use film, electronic sound, and digital images; William Kentridge combines drawing and animation; Jenny Holzer creates immersive installations of words and pictures using programming. Creative industries such as web design, interface design, games development, advertising, marketing, animation, and film incorporate digital art and design.

Have classroom practices and examination requirements kept pace with the changes in the world of art and creative media? Some people have called for young learners to have earlier access to relevant information on digital media careers, and an increased exposure to the skills needed in the digital art and design profession and the creative industries. Furthermore, some suggest that increasing the computing content in art lessons could address the lack of boys opting to study art and design, and more art in computing lessons could contribute to improving the representation of girls in computer science.

Teachers can access support to increase digital art and design by joining the National Society for Education in Art & Design (NSEAD) at helloworld.cc/nsead. NSEAD has a new guidance document on digital art and design. Teachers can also make use of its digital audit tool to reflect on what actions and development are needed for their schools. When planning actions, resources can be found at Teaching London Computing (helloworld.cc/londonart), TechPathways (helloworld.cc/techpath), and in the other ideas presented in this *Hello World* issue on computing and the arts.

COMPUTING MEETS CLASSICAL BALLET

Genevieve Smith-Nunes wanted to bring art into the computer science classroom, so she created DataDrivenDance: a ballet that interrogates computer science theory, bioengineering, and digital ethics

DataDrivenDance is where the worlds of classical ballet and computing collide. It was born out of my childhood love for ballet, coupled with my passion for computing. Why ballet you might ask? Ballet has many similar characteristics of a programming language. The precision of classical ballet, with set structures, routines, rules, and terminology, is akin to a programming language. DataDrivenDance links the analogue nature of classical ballet with digital and virtual concept representation achieved through data visualisation, dance, and digital technologies. Traditionally, computer science concepts are represented

through diagrams, formulae, or equations; DataDrivenDance uses ballet, data visualisation, and electronics to represent those same concepts. For example, certain steps and motifs can only be performed with certain arm positions; similar to a function or method.

I started working on DataDrivenDance in 2011, when I was teaching ICT and acting as a member of the drafting group for the initial computing programme of study for England. I was motivated by a desire to build an interest in computing and reduce fear of the subject. I was also driven by my experiences of seeing girls having an interest and an ability in computing in Key

Stage 3 (ages 11–14), but choosing not to continue the subject at GCSE. I hoped that benefits would be felt by the dancers, the audience, the educators, and artists. The project won a Google RISE award for Computer Science Education in 2013, and has also since benefited from the supported of three Grants for the Arts Awards from Arts Council England.

I first realised that the idea of a ballet about computing was powerful back in 2013, after running DanceHack as part of the Brighton Digital Festival. The project brought together dancers and programmers. One participant said there was “great insight into the boundaries of digital design and innovation, particularly useful information on how to marry, and seamlessly integrate, graphic iconography remotely generated and human choreographic movement. This has enabled moving a current project on considerably.”

I perform many roles within the development of each ballet, working as a

DATADRIVENDANCE TECHNOLOGIES AND THEMES

[arra]stre, 2014: theory and concepts from the English Department of Education computing programme of study that used a Microsoft Kinect, data visualisation, and wearables.

[data]storm, 2015: Data transmission through weather data supplied by the Met Office. Additionally used projection mapping and live data streams.

[pain]byte, 2017: Chronic pain and biomedical engineering that used biometric sensors and a VR experience.

Singularity, 2020: A story of interstellar travel to highlight technical and ethical questions of communication and human body augmentation. (This will be performed live in Cambridge, UK - see datadrivendance.com for more information).

A key aim is for teachers and students to be able to replicate the technology. We use off-the-shelf products, developing tools to facilitate this. For example, we developed our Kinect server that captures data and exports it as a JSON file. We share code so others can develop their own projects: helloworld.cc/datastorm



Ballet shoes feature Arduino Gemma wearable microcontroller boards and a pressure-sensitive plastic film. Image: ReadySaltedCode CIC

producer through to a coder. I am not a lone wolf, working with amazing technical minds such as Alex Shaw, and the choreographer Camilla Lloyd. I have found that the role of the artist is similar to the code, in a constant cycle of creation, failure, and recreation. The iterative process allows emergent ideas, changes in choreography, all within a sphere of creativity. Failure is part of the process and nothing is perfect. Each ballet crosses many domains within the curriculum from computing, performing arts, science, design and technology, geography to RE.

All of our ballets are performed alongside a video in front of a live audience. This is the method we used to create each ballet, starting with the data first. Each ballet uses different technologies and themes.



The dancers represent debugging and algorithms Image: ReadySaltedCode CIC

role of data packets travelling through a network. Certain dancers only had access to slow bandwidths that trigger modified

choreography, depending on when they arrived at a certain node in the network. Choreography notation was written using node maps.

An interesting output was the use of node maps as a choreographic tool.

The theory for networks, virus simulations, and associated choreography came from social network theory and social information processing. This has ties with the English curriculum on data structures, and the implications of computer use and units in the OCR GCSE specification that require students to understand data structures, data types, data transmission, network structures, and protocols.

From the stage to the classroom

You could create your own ballet in a computing classroom, but room size generally doesn't allow for that. We think of the ballets as R&D platforms for computing in the classroom. We use Arduino and micro:bits in live performances, but the code has also been converted and simplified for a Crumble controller in a primary setting. Year 3 students have repurposed the 'wearables', creating disco lights using recycled bottles.

They have also added interaction using a touch or ultrasonic sensor, and the Crumble. Each ballet brings a new set of learning opportunities that are converted to classroom resources at helloworld.cc/ddd.

Bringing brainwaves to ballet

Our upcoming ballet, Singularity, will use brainwave data from an Emotiv headset to highlight the hidden nature of data. Singularity is about interstellar travel, augmented reality, communication, data, and morality, encouraging reflections on the ethical use of data and cognitive privacy. The streamed data is in the Open Sound Control format, as used in other applications such as Sonic Pi and Scratch. Last year, we gave a talk and workshop on Singularity at the Electromagnetic Field festival. The video (helloworld.cc/singularity) highlights some of our plans for the project. We will be sharing some of our datasets for educators and students to explore, as the cost of the headset is significant. (HW)



GENEVIEVE SMITH-NUNES

Genevieve (@pegleggen) is an artist, entrepreneur, and a lecturer at Roehampton University School of Education & Media, Culture and Language. She is the founder of ReadySaltedCode CIC.

11 DANCERS TOOK ON THE ROLE OF DATA PACKETS TRAVELLING THROUGH A NETWORK

Simulated networks

An example of theory performed through ballet and data visualisation is 'Networks' from the ballet [data]storm. The data visualisation represents a social network, with each node in the network representing a user who has the following characteristics:

- friendliness (how often they're likely to make friends with another user)
- chattiness (how often they're likely to send messages)
- category (the subject area they're most interested in)

The visualisations are coded in JavaScript, with random time intervals to generate the simulation based on the characteristics. All the rules stay the same throughout the simulation. At the same time, the dance (ballet) movements and wearables (LEDs) were choreographed/coded to accompany the data visualisation using network mapping techniques. Dancers took on the

SPIRITED ADVENTURES IN DIGITAL EDUCATION

Elliot Hall and **Tom Bowtell** have created The Digital Ghost Hunt: an immersive theatre and augmented reality experience that takes a narrative-driven approach to make digital education accessible

The Digital Ghost Hunt combines coding education, augmented reality, and live performance into an immersive storytelling experience. It begins when a normal school assembly is disrupted by the unscheduled arrival of Deputy Undersecretary Quill of the Ministry of Real Paranormal Hygiene, there to recruit students into the Department's Ghost Removal Section. She explains that the ministry needs students' help as children have the unique ability to see and interact with ghostly spirits.

Under the tutelage of Deputy Undersecretary Quill and Professor Bray (the Ministry's chief scientist), the young ghost-hunters learn how to program and use their own paranormal detectors. These allow students to discover ghostly traces, translate Morse code using flickering lights, and find messages left in ultraviolet ectoplasm. Meanwhile, the ghost communicates through a mixture of traditional theatrical effects and the poltergeist potential of smart home technology. Together, students uncover

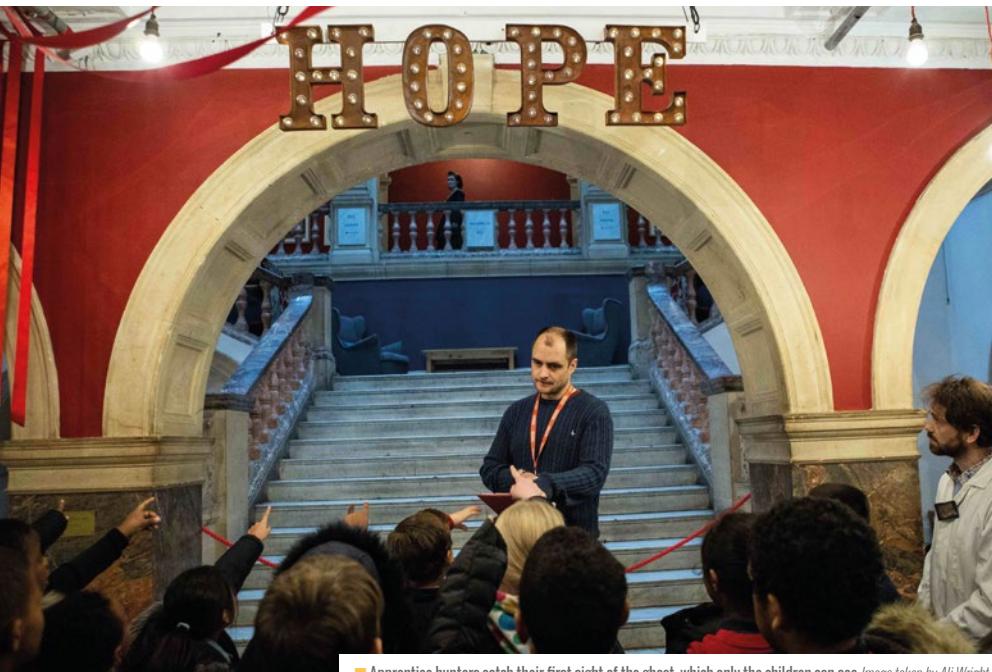
the ghost's identity, discover her reason for haunting the building, unmask a dastardly villain, find a stolen necklace, clear the ghost's name, right an old wrong, and finally set her free.

The project conducted two successful test performances at the Battersea Arts Centre in South London in November 2018, funded by a grant from AHRC's New Immersive Experiences Programme, led by Mary Krell of Sussex University. Its next outing will be at York Theatre Royal in August.

Adventures in Learning

The Digital Ghost Hunt arose out of a shared interest in putting experimentation and play at the centre of learning. We felt that the creative, tinkering spirit of earlier computing – learning how to program BASIC on an Atari 800XL to create a game, for example – was being supplanted by an didactic and prescriptive approach to digital learning. KIT Theatre's practice, creating classroom adventures which cast pupils as heroes in missions, is also driven by a less trammelled, experiment-led approach to learning.

We believe that the current computer science curriculum isn't engaging enough for students. We wanted to shift the context of how computer science is perceived, from something techy and boyish, back into a tool of the imagination that it should be. We did this by de-emphasising the technology itself and, instead, placed it in the larger context of a ghost story. The technology



Apprentice hunters catch their first sight of the ghost, which only the children can see *Image taken by Ali Wright*

becomes a tool to navigate the narrative world – a means to an end rather than an end in itself. This helps create a more welcoming space for students who are bored or intimidated by the computer lab; a space of performance, experiment, and play.

Ghosts and machines

The device we built for the students was the SEEK Ghost Detector, made from two microcomputers, the Raspberry Pi and a micro:bit, which Elliot stapled together. The micro:bit was the device's interface, which students programmed using the block-based language MakeCode. The Raspberry Pi handled the heavier technical requirements of the show, and communicated them to the micro:bit in a form students could use. The detector had no screen, only the micro:bit's LEDs. This meant that students' attention was focused on the environment and what the detector could tell them about it, rather than having their attention pulled to a screen to the exclusion of the 'real' world around them.

In addition to the detector, we used a Raspberry Pi to make ordinary smart

Adventures in Learning model, and its Theory of Change:

- **Disruption:** an unexpected event grabs attention, creating a new learning space
- **Mission:** a character directly asks pupils for their help in completing a mission
- **Achievement:** pupils receive training and are given agency to successfully complete the mission

The Ghost Hunt

During these lessons, Deputy Undersecretary Quill kept in touch with the students via email, and the chief scientist sent them instructional videos. Their work culminated in their first official assignment: a ghost haunting the Battersea Arts Centre – a 120-year-old former town hall. After arriving, students were split into four teams, working together. Two teams analysed evidence at headquarters, while the others went out into places in the building where we'd hidden ghostly traces that their detectors would discover. The students pooled their findings to learn the ghost's story, and then the teams swapped roles. The detectors were therefore



ELLIOTT HALL

Elliott Hall (@TheLastDantes) is Senior Research Software Engineer at King's Digital Lab.

He is also the author of three novels: *The First Stone*, *The Rapture*, and *The Children's Crusade*.



TOM BOWTELL

Tom Bowtell (@Tom_Bowtell) is the Director of creative education company KIT

Theatre. Tom's National Charity Award and BAFTA-winning work includes Wardrobes, which engaged hundreds of pupils in a multi-venue time-travelling adventure across Birmingham. More than 10,000 young people have participated in KIT's Adventures in Learning programme.

STUDENTS TOOK PART IN A SERIES OF FOUR LESSONS AT SCHOOL, FOCUSED ON THINKING LIKE A PROGRAMMER

home technology into our poltergeist. It communicated with the students using effects such as smart bulbs that flashed in Morse code, which the students could then decode on their devices.

To program their detectors, students took part in a series of four lessons at school, focused on thinking like a programmer and the logic of computing. Two of the lessons featured significant time spent coding on the micro:bit. The first focused on reading code on paper, and students were asked to look out for any bugs. The second had students thinking about what the detector will do, and acting out the steps together, effectively 'performing' the algorithm. We based the process on KIT Theatre's

only one method of exploring the narrative world. But the fact that they'd learned some of the code gave students a confidence in using the detectors – a sense of ownership. During one performance, one of the students pointed to a detector and said: "I made that."

Future of the project

The project is now adapting the experience into a family show, in partnership with Pilot Theatre, premiering in York in summer 2019. We aim for it to become the core of an ecosystem of lessons, ideas, and activities – to engage audiences in the imaginative possibilities of digital technology.

You can find out more about the Digital Ghost Hunt at digitalghosthunt.com. (HW)

TIPS TO CREATE ADVENTUROUS LESSONS

- **Disrupt the rhythm of a normal lesson:** an email with a secret code arrives from a mysterious character, or an ordinary PowerPoint presentation is suddenly haunted or hacked.
- **Give pupils a mission.** Tell the story first, then add the syllabus later.
- **Get out of the classroom.** Change the context by taking the story out into the world
- **Collaborate with teachers to pool skills in technology with performance.**

THE MAKER MOVEMENT MOVES INTO THE SCHOOL ART STUDIO

Erin E. Riley explores the intersection of coding, art, and digital fabrication

People learn best through creating and sharing the objects they make. Every time someone makes art with any kind of material – clay, cardboard, or computer code – ideas are brought into the world. While ideas are being expressed, a parallel process is taking place, unlocking previous understandings and experiences and connecting them to new ones. This web of knowledge that develops through ‘doing’ is a cornerstone of progressive education. Education giants

like Seymour Papert, through the theory of constructionism, underscored the expressive power of computers as creative tools. Today we have an unprecedented opportunity to put these tools in the hands of the next generation.

So what does a contemporary art studio look like today? How do artists meld digital tools with more traditional ones? With the enthusiasm around the maker movement, and its growing influence on education, an exciting opportunity exists to forge new pathways for making art using digital technologies and fabrication.

Digital fabrication machines, such as laser cutters and 3D printers, were developed for engineering and commercial design, but also have direct applications in art and design in a school setting. Educators can now fold the knowledge of tools and materials for art, into the advancement of ideas in making. As the field of art and design continues to expand, it seems fitting that artists look to the design

processes used in engineering, architecture, and commercial design to inform their process. At the same time, elements of art and design bring relevance and beauty to school subjects.



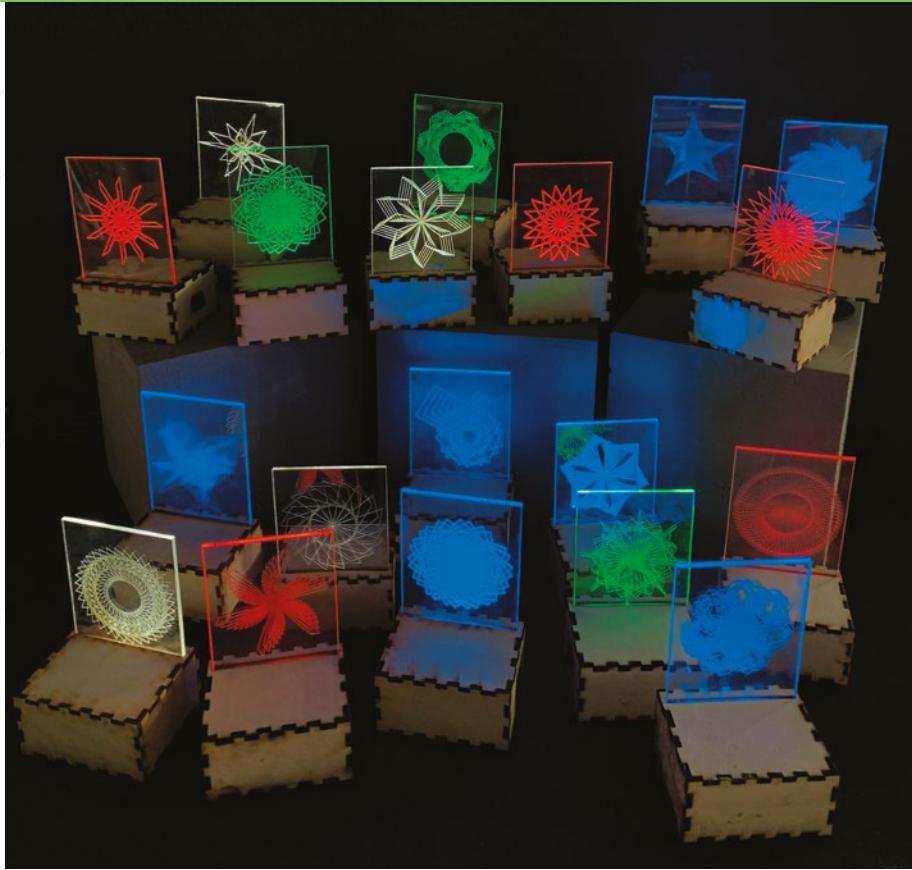
Turtle art on clay: drawings created with code and expressed as fabricated objects

ELEMENTS OF ART AND DESIGN BRING RELEVANCE AND BEAUTY TO SCHOOL SUBJECTS

Digital and physical materials combine

Studio practitioners needn’t abandon materiality when using digital technology. Fabrication produces objects from tangible materials and the art studio is a rich source of raw materials for projects. Digital design parallels physical design, and digital tools can mirror the push/pull responsiveness of physical materials. What we imagine and create digitally can be produced with machines that understand code, and the ingenuity of the artistic creation that is a product of that process ultimately depends upon the vision of the maker.

A makerspace is full of machines that can interpret drawing – the universal visual language – in familiar ways, but also can allow exploration of the very idea of what drawing means. Drawings can be created

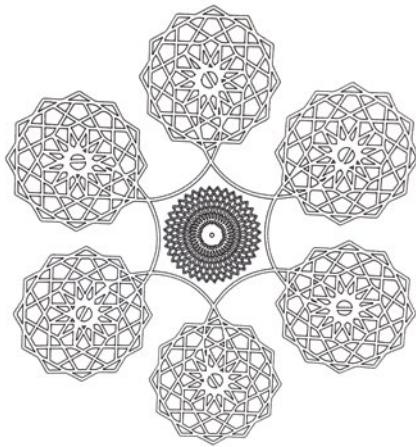


■ Edge-lit pieces of acrylic facilitate 'drawing with light'

with code and expressed as fabricated objects. We can substitute a pen for the cutting tool on a computer numerical control (CNC) machine or vinyl cutter, or extrude very thin prints on the 3D printer to make flat shapes. We can 'draw with light' by attaching a light to a programmed robot or laser-etch/score a design for an edge-lit piece of acrylic. Machines can be used to explore variance and imprecision in a process. A loose pen or varying the settings in a laser-cut etching or score line brings artistic line quality to a work. There are infinite combinations of materials, machines, and software awaiting the curious artist interested in machine drawing. And drawing is only one of the myriad artistic processes that can be explored and expanded using digital technology.

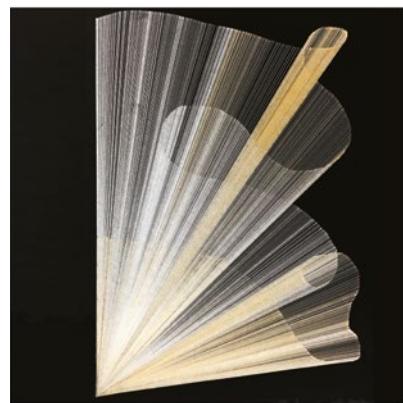
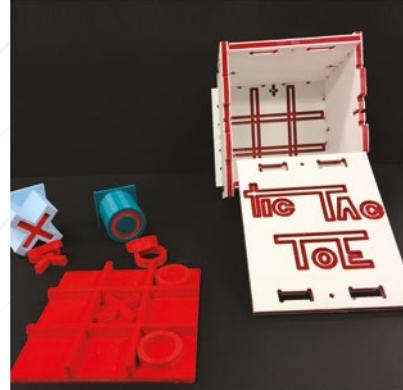
Design on purpose and by accident

Digital design parallels physical design. Pixels, or tiny elements of graphic information, are the elemental unit of a raster image, and have been a part of the art and design vocabulary since early computer graphics. When small,



■ Pens can be substituted for cutting tools

manipulating the pixels can be a painterly application of digital information. But as pixels increase in size, a gridded order emerges. Using the Pixel by Numbers Processing application created by Erik Nauman (helloworld.cc/pixels), students experience 'pixellation' as an artistic element while exploring digital tools. Students convert images into a 'colour by number' pixel map and the program generates a numbered grid with a colour



■ Merging physical art materials with digital design, fabrication, and code broadens the toolset for our artists, designers, and makers Credit: Erin Riley



Varying the settings in a laser-cut etching or score line brings an artistic line quality to a work

► key. The grid is in vector form and can be printed or exported for digital fabrication. Not only do students enjoy selecting images to convert to pixel art, they are excited by the process as their pixellated image emerges from the blank grid.

Finding beauty in mistakes or random actions, or 'mismethetics', is an accepted design approach in art. Like drips and drizzles



Erik Nauman's Pixel by Numbers tool allows students to convert images to pixel art



of action painting, data-bending a digital image by modifying image text, gives rise to an unexpected, glitchy result. Students

experiment with additions and deletions in the text file, and the raster image linked to the text file progressively degrades.

Artists interested in exploring the extent of variance of change in a design can use code. Some modern design applications have codable elements, allowing students to experiment with variables or random number generators in their designs. The unknown variable creates

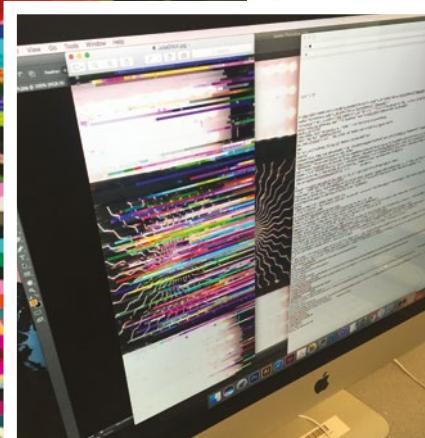
surprise combinations when the code is executed and the artist looks for the serendipitous result to capture and send to the fabrication machine.

The power of iterative design

The computer has the capacity to do things differently than the human hand. Sometimes what we want to achieve with our work requires algorithms. Similar to how a drafting machine can be a tool to aid precision, a computer program can do the same. We write code and the computer executes our intention. Designs for digital fabrication machines like 3D printers, laser cutters, vinyl cutters, and CNC machines are output as machine directions and generate



Students can experiment with 'glitches' in digital artworks just as they might explore the effects of drips and drizzles in action painting



toolpaths. These machines are programmed to exact locations and pair well with design ideas that are very precise.

For educators interested in STEAM, design and fabrication builds competencies across all areas. Maths, which is applied in art, is foundational to image and space building. Artists use maths to build compositions, patterns, or structurally solid 3D constructions. Students have the opportunity to test their designs in the physical world through fabrication. Mistakes in measuring, scale, and transformations show up in the model that is sent to be fabricated. Mistakes may result in failed prints, machines cutting in the wrong place, or machines not working at all. But as artists know, mistakes are not failures. Mistakes lead to new discoveries or are improved upon as modified designs reimagined and iterated upon.

We can equip our students with a new language for speaking about technologies

in relation to their work that is true to our fundamental methods of how we make and teach art. Digital materials and fabrication add to a vast menu of options for bringing impressive, and personally meaningful objects and ideas into the world. **(HW)**

Erin E. Riley's book, *The Art of Digital Fabrication*, is reviewed on page 96, and available to buy now at artofdigitalfabrication.com.



ERIN E. RILEY

Erin (@eeriley99) is Director of the Engineering and Design Lab at Greenwich Academy, a K-12 school for girls in Connecticut. Her work centres on STEAM - she collaborates with teachers in the development of hands-on making activities that intersect the arts, STEM subjects, and humanities.



ART AND CODE

Coding gives artists and makers a powerful tool for introducing pattern and complexity to designs. Digital fabrication machines found in makerspaces, like 3D printers, vinyl cutters, and laser cutters, can be used to draw, carve, and cut intricate designs that students make. Programming variability into code using parameters can amplify the iteration process with parametric design tools and open up possibilities with generative art.

BRING A BOOK TO LIFE

Sian Wheatcroft created an interactive story display to allow for a deeper exploration of her children's picture book

The task of promoting my first children's picture book, *This Bear, That Bear*, was a daunting one. At the time, I wasn't a teacher and the thought of standing in front of assembly halls and classrooms sounded terrifying. As well as reading the book to the children, I wanted to make my events interactive, using physical computing, and showing a creative side to coding and enabling a story to come to life in a different way than what they would typically see, i.e. animated retellings.

The plan

Coming from a tech-loving family, I naturally gravitated towards the Raspberry Pi, and found out about Bare Conductive and their PiCap. I first envisaged using their conductive paint on the canvas, enabling users to touch the paint to interact with the piece. It would be some sort of scene from the book, bringing some of the characters to life. I soon scrapped that idea, as I discovered that simply using copper tape on the back of the canvas was conductive enough, which also allowed me to add colour to the piece.

I enlisted the help of my two sons (two and five at the time) – they gladly supplied their voices to some of the bears and, my personal favourite on the canvas, the ghost. The final design features characters from the book – when children touch certain areas of the canvas, they hear the voices of the characters.

Getting the project up and running went pretty smoothly. I do regret making the piece so large, though, as it proved difficult to transport across the country, especially on the busy London Underground!

“THE PROJECT ADDED A WHOLE OTHER LAYER TO THE EVENTS I WAS TAKING PART IN”

Interactivity and props

The project added a whole other layer to the events I was taking part in. In schools, I would read the book and have props for the children to wear, allowing them to act out the book as I read aloud. The canvas then added further interaction, and it surprised me how excited the children were about it. They were also really curious and wanted to know how it worked. I enjoyed showing them the back of the canvas with all its





■ The back of the canvas shows the copper tape used to create a conductive surface

copper tape and crocodile clips. They were amazed by the fact it was all run on the Raspberry Pi – such a tiny computer!

Fast-forward a few years, and I now find myself in the classroom full-time as a newly qualified teacher. The canvas has recently moved out of the classroom cupboard into my newly developed makerspace, in the hope of a future project being born.

I teach in Year 3, so coding in Python or using the command line on Raspbian may be a little beyond my students. However, with a keen interest in project-based learning, I'm hoping to incorporate a host of cross-curricular activities with my students involving the canvas. I hope to instil a love for digital making in my students and, in turn, show senior leaders what can be done with such equipment and projects.

A literacy project

This work really lends itself to a literacy project that other educators could try. Perhaps you're reading a picture book or a more text-based piece: why not get the students to design the canvas using characters from the story? The project would also work equally well with

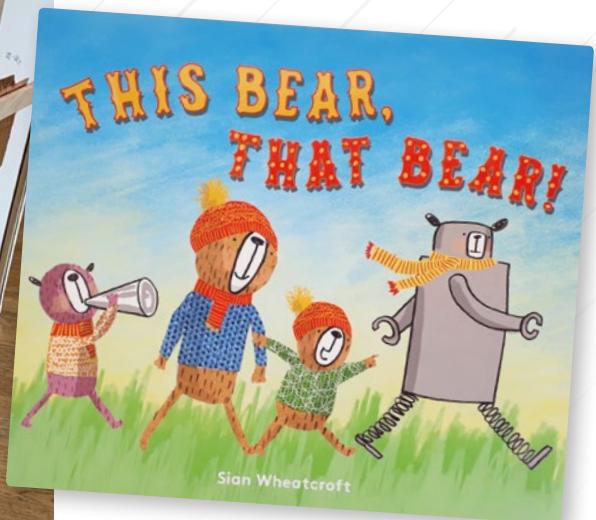
foundation subjects like history or science. Children could gather information onto the canvas, explaining how something works or how something happened. The age of the children would influence the level of involvement they had in the rest of the project's creation. The back end could be premade – older children could help with the copper tape and wiring, while younger children could stop at the design process.

Part of the project is getting the children to create sounds to go with their design, enabling deeper thinking of a story or topic. It's about a collaborative process with the teacher and students, followed by the sharing of their creation with the broader school community. [\(HW\)](#)



SIAN WHEATCROFT

Sian (@strawberrykitty) is a former children's illustrator, and now a newly trained primary school teacher. She is ICT lead in her school and runs a coding club and makerspace. She is a Raspberry Pi Certified Educator.



■ *This Bear, That Bear* - a children's picture book, written and illustrated by Sian

KIT LIST FOR AN INTERACTIVE CANVAS



- Raspberry Pi
- Crocodile clips
- Copper tape/conductive paint
- PiCap (A Pi HAT with conductive connectors)
- A portable speaker and an external battery

BREAKING DOWN BARRIERS WITH MICRO:BIT

Giles Booth shares the stories of artists and teachers who use the micro:bit to mix technology with traditional skills in ways that mutually benefit the worlds of art and computer science

Fine art and computer science may seem to be poles apart, but increasingly affordable and easy-to-use technology is helping blur the two worlds in ways that can inspire creativity in your classroom. The micro:bit is especially helpful here, as it can be coded in a familiar block environment, allowing pupils to get quick results using its built-in sensors and LED displays, yet it contains a microcontroller that can be used in more

GET CREATIVE WITH MICRO:BIT IN CLASS

- Get pupils to represent the world around them in 5x5 pixel art
- Create patterns and animate them
- Use the micro-bit's built-in accelerometer and light sensor to respond to the environment
- Attach tin foil pads with crocodile clips to objects and the micro-bit to create large touch-activated triggers appealing to young children
- Add sound by attaching headphones or amplified speakers and design sound patterns triggered by movement, touch, or light
- Use the radio function to link multiple micro-bits together in a networked display
- Get crafty and make your micro-bit art wearable, or just fluffier, using felt or wool
- Find more inspiration here: microbit.org/microbit-art

sophisticated ways with external sensors, displays, and even building networks of interactive art.

Yarnbombing with technology

Sam Topley is researcher and lecturer at De Montfort University, specialising in experimental music technology and community arts practice. Her 'Playground' project uses giant textile balls to make sound and music as they are played together. Each hand-crocheted ball contains a micro:bit; the balls connect wirelessly and transmit data to each other and Sam composed digital music which responds to movement.

Sam has used the balls to run workshops with learners aged between 2 and 25, some with sensory impairment, learning difficulties, communication difficulties, or complex medical conditions. The soft, tactile casings for the technology make playing with

them feel safe and calm and, when coding the music algorithms, she was careful to avoid any loud, sudden, or surprising sounds. Interactive technology like this, where there is no 'right' or 'wrong' way of using it, is perfect for this kind of environment.

As well as benefiting the learners who appreciate a sensory experience, Sam has found the project beneficial for developing her own art and further broaden its appeal and reach.

Finding algorithms in art

Digital artist Sean Clark, an international professor at Guangdong University of Technology, is passionate about the mutual benefits of art and technology and says programming should be taught by art teachers as much as maths teachers. Sean uses micro:bit in his workshops in China, alongside Scratch and simple materials like

cheap NeoPixel lights and laser-cut scrap cardboard. Students use Scratch to recreate patterns found in Cantonese ceramics. In following the ancient rules that govern which shapes can – and cannot – be combined, they learn about the history of art, but also the computational skills of decomposition, pattern recognition and algorithms in practical programming.



■ Sam Topley's crochet balls contain micro-bits that interact using radio and respond to movement, changing the music they play Credit: samantha-topley.co.uk

The micro:bit allows the creation of artefacts that reflect this, augmenting 3D structures and headdresses inspired by traditional art, with patterned-lighting responding to environmental triggers such as the sound of clapping. Sean says that the simplicity of micro:bit allows pupils to get quicker results than they could with some other microcontrollers, as you can create digital artefacts without adding any external hardware. Quick results are also helped by the micro:bit MakeCode editor being available in local languages so learners can be more independent.

MICRO:BIT ALLOWS PUPILS TO GET QUICKER RESULTS

Connected artworks

The open architecture of micro:bit MakeCode, allowing anyone to add their own blocks, is enabling another of Sean's projects, artThings. This is an infrastructure for 'Connected Digital Artworks' that can exchange information over the internet using the simple MQTT protocol.

Sean is currently working on a micro:bit MQTT gateway that will allow artworks to communicate with each other over the internet. You could create an installation in a school which would respond to pupils' movement and sound, and trigger light or sound in another installation in a school nearby – or thousands of miles away. Sean is developing artThings as an open standard through GitHub (helloworld.cc/artthings) and is looking for people to get involved.

Augmented fine art

Vahagn Matossian is a musician and product designer who is part of the Human Instruments project; they create technology such as the 'haptic baton' which allows visually impaired percussionists to feel the conductor's baton moves, and hence take part in an orchestral performance. Fellow Armenian artist Taline Temizian feels the



The mask uses a micro-bit to add movement sensors and lighting control. This can be used by a dancer to add extra expression to a performance Image courtesy of interactdigitalarts.uk/fe-artlab

digital realm is now almost inseparable from the analogue world, and she turned to Vahagn to help augment her work; Vahagn decided to use micro:bit for this because its simplicity enables speedy results.

In one work, drops of fragrance are blown by a fan and lit from above and below behind a white mesh that forms part of the canvas. Adafruit's Crickit and Kitronik's motor driver accessories were used with micro:bit to control the fans and lights. Another work used sensors connected to micro:bits to log the time people spent in front of the painting, and the distance, to modify the behaviour of lights on two NeoPixel rings.

They plan future shows where artworks communicate with each other using the micro:bit's simple radio function.

Getting arty in the classroom

Stu Lowe is educational technology and innovation specialist at Beacon Hill School in Hong Kong. His maker-centred lessons have children tinkering and building with a wide range of technology, from recycled cardboard and plastic junk to computers. The micro:bit is at the heart of many of creations; the simple block-coding interface married

with the ease of extending it into the physical world with sensors, sound, lights and motors enables children to go beyond just coding on screen, instead using technology to solve practical problems in the real world.

As well as engineering solutions, Stu's pupils have created art for art's sake: sound-activated animatronic heads that blend traditional papercraft with 3D printing and coding, and animated spirit animals using scrap cardboard, LEDs, and servo motors driven by micro:bit.

If you're inspired to make art in your classroom, the Micro:bit Educational Foundation would love to hear from you on Twitter @microbit_edu, or you can find us on Instagram, Facebook, and LinkedIn. (HW)



GILES BOOTH

Giles (@blogmywiki) is the Educational Content Manager for the Micro:bit Educational Foundation. He is a Raspberry Pi Certified Educator, and has taught Computer Science in both primary and secondary schools in London.

BUTTERCUP STEAM

Aisha Flores shares the how her steampunk coding camps fuse art and technology to engage girls

Buttercup STEAM Camps are unique week-long adventures for girls aged 8 to 14 to learn, build, and code in a fun and supportive environment. Students work hands-on creating steampunk-themed projects that incorporate elements of STEAM (science, technology, engineering, arts, and mathematics).

It is important to include the arts in STEM for several reasons, but I believe the most important reason is that art is what makes us well-rounded individuals. Regardless of the form of art we engage in – be it visual arts, performing, writing, music, or one of a many other forms of artistic expression – we can be creative in any part of our lives and it makes our world a better place.

I am an art lover, art collector, and arts advocate. I come from a family of artists and educators. It was an easy choice

to incorporate the arts into my love for technology. Art is a key part of our daily lives. Everything we use was once someone's conceptual art at some point. My husband and partner, Daniel Flores, is a professional artist and we founded Buttercup STEAM Camp, where we teach girls to build robots and code. We fuse technology and creative arts projects to spark their imaginations.

I recently read, "you cannot have smarts without the arts," and I believe this is true: creativity is what sparks innovation. We use creativity in STEAM to help children open their minds to the endless possibilities of utilising technology in new ways. How will they use a new coding skill? What else can they create after learning how to build a robot? Can they use it to help others, or create something that we haven't seen before?



■ Steampunk hat made for camp. Photo credit: Daniel Flores

STEAM for under-represented children

According to the latest STEM Index from the US News and World Report, incorporating the arts into STEM subjects is a proven method for onboarding and recruiting under-represented groups, particularly girls and children of colour, into STEM. The hands-on aspects of STEM have been a great way to engage students in what would normally be considered mundane concepts. STEM subjects have been typically the hardest to engage students in long-term. Allowing children to creatively approach these subjects, with the addition of hands-on projects, allows us to engage children in a way that is memorable and enduring. This gives us an opportunity to lead them to paths that were not previously explored.



■ Campers with self-made cosplay Laser Gauntlets. Photo credit: Daniel Flores



Working on Laser Gauntlets. Photo credit: Daniel Flores

" WE CAN BE CREATIVE IN ANY PART OF OUR LIVES AND IT MAKES OUR WORLD A BETTER PLACE

STEAM + cosplay

One of the ways we incorporate arts into STEM is through cosplay and steampunk. Steampunk is a subgenre of science fiction or science fantasy that incorporates technology and aesthetic designs, inspired by 19th century industrial steam-powered machinery. Taking inspiration from a retro-futuristic mash-up between a Victorian aesthetic and cyberpunk, steampunk encompasses a sense of longing not for the actual past, but for a futuristic past that never was. The DIY and repurposing aspects keep steampunk a flexible form of expression with a spirit of adventure, which we apply in our summer camp projects.

I have been a guest panellist at conferences such as Momocon and Dragoncon, where I have taught workshops on 'How to Light-Up Your Cosplay' with one of my good friends, Tanya Woods of ThrillBuilds. We teach people budget-friendly ways to add electronics and lights to costumes. Tanya visits Buttercup STEAM Camp to shows our girls how she uses her creativity for passion projects and also

how she uses technology in her costumes. Another cosplayer, Darin Hicks, has also visited our camp and discussed how he incorporated speakers and lights into his robotic costume inspired by the Overwatch character Lucio.

Why Buttercup STEAM?

Buttercup STEAM's mission is to reduce the gender and racial gap in technology professions by inspiring girls, and girls of colour, with an adventurous and memorable introduction to coding and robotics. Our goal is to help students see how they can create, make, and explore while solving problems with critical thinking and analytical skills. [\(HW\)](#)



AISHA L. FLORES

Aisha (@buttercupsteam) is an Electronics Engineer, Developer, and the founder of Buttercup STEAM Camp. She is a Raspberry Pi Certified Educator and is based in Atlanta, Georgia.

TEAPOT RACER



The Teapot Racer is a transformed RC (remote control) truck used in a Teapot race at Dragoncon 2018 in Atlanta, GA. Materials: NeoPixel LED ring, steampunk goggles, programmable microcontroller switch, power bank, ultrasonic humidifier (Baymax) for steam, RC truck powertrain, and gear embellishments. This is also a demo take-apart project on how to creatively repurpose electronics.

ART AND ALGORITHMS

Katharine Childs takes inspiration from the artist Sol LeWitt to create algorithmic art in primary computing lessons



At first sight, computing and art are an unlikely combination: computing seems precise, prepared and predictable; art seems creative, expressive, and ambiguous. Sol LeWitt (1928 – 2007) was an American artist who conceptualised that the instructions for creating art were as important as the finished piece. He gave his instructions and diagrams to other people so that they could follow them and create wall paintings.

I have developed primary computing activities based on a particular type of LeWitt's art. These comprise coloured stripes, and involve children writing their own algorithm to draw a stripy picture. An algorithm is a sequence of instructions or a set of rules to achieve something, in this case, some artwork. Once the algorithms have been written, children will work in pairs. One child can take the role of the artist who reads out their instructions and the other child will be the creator who carries out the directions exactly as they are instructed. It's worth noting that LeWitt made some of his instructions deliberately ambiguous, whereas in the classroom children will need to be precise with their



**KATHARINE
CHILDS**

Katharine (@IAmKatharineC) is a Programme Coordinator at the Raspberry Pi Foundation, and a former primary school teacher and network manager.

algorithms so that they can be interpreted by a computer, or a human artist working in a robot-like way.

Adapting across key stages

This idea works well as an unplugged activity by choosing art materials that can quickly create blocks of colour. For example, painting on large sheets of paper taped to the floor or using chalks on the playground tarmac. Just like LeWitt's art, the creative space needs to have some boundaries marked, so that children can use positional language to describe where to add the stripes. Horizontal or vertical stripes are the easiest to work with, and the activity can be made simpler by using only two colours, or made more complex by using stripes in a repeating pattern. Children will need to think carefully about when to put the pen (or paint or chalk) down onto the paper and when to pick it up again.

At Key Stage 2, the activity could also be used to write an algorithm to create art in Scratch, a free online programming environment (scratch.mit.edu). It is useful for children to do a small-scale, unplugged version of their design to help them plan their work and give them the structure of their algorithm. Although Scratch contains a paint editor for freehand drawing, the activity works best by programming a sprite to draw the stripes one by one on the stage. This will involve some careful calculations to determine the starting and finishing position of the sprite on each line, and pupils could use variables to store and update the values of these positions. As the sequence of instructions grows, children may notice that they repeatedly go to the starting position, put the pen down, draw the stripe, pick the pen up, then change the coordinates and the pen colour ready for the next line. Spotting this pattern helps to determine where to use a repeat loop in the algorithm.

Common misconceptions

When writing algorithms, pupils often underestimate how precise the instructions need to be so that a computer can understand them. This precision can be misunderstood in two ways:

- 1 Missing out steps. It's easy to make assumptions about what needs to happen to create a piece of art. In these activities, a common misconception is that children forget to explicitly say when to put the pen (or paintbrush or chalk) down on the paper and when to pick it up. Having someone else carry out their instructions is a good way of spotting these omissions, and can help children to debug their algorithm by finding and fixing errors in it.
- 2 Missing out detail. Children often make the steps in their algorithms too ambiguous. Adding in detail such as positional language can help to make algorithms more precise. This might include using vocabulary such as 'left, right, under, touching' at Key Stage 1, and Key Stage 2 pupils may be able to incorporate precise measurements. Watching someone else carry out their instructions is a useful way for children to think about how to improve their algorithm by adding precision.

Effective scaffolding

You can support pupils to write precise algorithms by scaffolding this activity with examples. A useful technique is to model how to decompose a task into small steps, including deciding what to do first. Pupils can write the first part of the algorithm, test it on a small-scale, check it for errors, amend what they have done so far, and then write some more. This 'run-test-fix-add' cycle is a good habit to develop and can be used as part of a toolbox of techniques to be displayed on tables or on the whiteboard.

There are many opportunities for 'gotcha' moments when writing an algorithm for someone else, so expect to hear comments such as, "I didn't mean that!". Modelling how to learn from failure at the start of the activity creates a classroom culture that mistakes aren't a problem in themselves, as long as you persevere and put them right. Classroom displays often include the final version of a piece of work, but displaying an algorithm including errors side-by-side with the debugged version sends a powerful message that it's OK to fail and learn.

Learning from failure is also important when pupils take the role of the creator and follow the algorithm. Some children may need support to understand that their role is to help the artist by working in a robot-like way, rather than adding in steps or detail.

At the end of this activity, the artwork and their algorithms make a great display. Match algorithm and art together, or jumble them and see if others can do the matching!

For further ideas for algorithmic art, turn to pages 74-75. And for more advanced concepts related to art, algorithms, and recursion, see pages 87-89. [\(HW\)](#)

FIND OUT MORE



More information about Sol LeWitt is at:
helloworld.cc/Tate_LeWitt

An unfinished basic version of a Scratch project referencing LeWitt's work:
helloworld.cc/LeWitt1

A completed Scratch project referencing LeWitt's work:
helloworld.cc/LeWitt2

ART IN MOTION

Tamara Pearson and **Katie Henry** integrated physical computing in art lessons to engage students who might not see themselves as computer scientists

In 2018, students from Sandtown Middle School in Atlanta, Georgia toured an American Civil Rights photography exhibit at the High Museum of Art, but this wasn't a typical museum gallery walk. The visit to the museum was the start of a semester-long project known as Art in Motion – a project-based curriculum designed to authentically integrate computer science into art classes. This program helps students who don't normally see themselves as part of the computer science pipeline to broaden their view of themselves as potential coders. The program begins with students selecting a meaningful work of art (online, in a museum, or anywhere) and creating a new work of art in response using a robotics kit and art supplies.



A student engaging with American Civil Rights photography during the Art in Motion project

Art in Motion is inspired by a project known as Moving Masterpieces developed by art teacher Jayne Sweet in Pittsburgh, Pennsylvania. Moving Masterpieces asks students to recreate a favourite work of art using robotics. Whether asking students to recreate an existing work of art or create something new altogether, here's a pathway you can follow to integrate robotics and computer science into your art classroom.

Check your resources

What robotics tools are already being used in your school? For Art in Motion, we used Hummingbird Robotics Kits, but micro:bits or similar would also work. The important thing here is to find out what hardware or software students are already using to create with robotics. If nothing is being used in your school, you could start with Scratch, a free, web-based application that you can use to integrate creativity and computer science into any classroom.

Does one year group seem to be using more physical computing tools than another? If this is your first time integrating computer science into your art classroom, you may want to start working with a class with the most computing experience first. However, for a project like Art in Motion, no prior experience is actually necessary.

Using tools already available in your school, and prior computing knowledge students bring to the project, will help to ensure success. But remember, this project is about helping students broaden their view

of themselves and computer science. If you continually rely on a small group of students who have attended an expensive summer camp or after-school robotics club to 'help you teach', you may be unintentionally putting up walls between yourself, the content, and the rest of the students.

Disrupting negative bias and stereotype with and for historically under-represented groups such as students of colour, girls, students with special needs, resource-poor and rural students, requires intentional effort on your part to be sure that these students see themselves represented in the process. Just because some of your students may have greater computing

TRUE CREATIVITY INVOLVES A SURPRISE ALONG THE WAY

knowledge than you doesn't mean you can throw good pedagogy out the window and let these students run the show with bad classroom practices. You don't want to say to your students, "Computer Science can be for anyone," then accidentally teach it's only for people who can afford summer camps or have care-givers who can provide transportation to after-school clubs.

Find out where the resources of hardware, software, and computing skill are in your school and use them to launch



■ One of the images students based their work on: Sanitation Workers Strike, Memphis, Tennessee, March 28, 1968 © Dr. Ernest C. Withers, Sr.

your project successfully – but remember, you know your students best, and are responsible for being sure they each have their voice heard in the process.

Get started

How will you invite students into the project? Students at Sandtown Middle School began with a trip to an art museum, but you can start anywhere. Try to see your local community through fresh eyes. What public art is within walking distance of your school, or even in your own school? For example, check your local park or library for murals or sculptures or connect with another part of your school's campus that displays art. The goal of Art in Motion is for students to create a work of art, using robotics, in response to something they experience.

How will you teach students to work in their new medium of robotics? A single mini-lesson on programming an LED to blink can go a long way for your learners. If you commit to ensuring that every student learns a few basic skills, they will likely teach themselves a lot more. Just like working with any other medium in your classroom, you'll need to allow time for students to practice with robotics. Set your students up for success by teaching a few basic skills, but let your students have

time to experiment with their new skills. Integrating LEDs and motors is a good start – to make projects more interactive, ask students to integrate sensors.

How will your students share their final projects? Think about where student projects originated, and try to reach that same audience with your students' new projects. Did your project begin in an art museum? Display student projects there. Did your project begin with a mural at a public library? Display student projects there. Students need an authentic audience with whom to



■ An 8th grade student creating original Art in Motion piece



TAMARA PEARSON

Tamara Pearson (@tampear) is Associate Director of School and Community Engagement at the Center for Education Integrating Science, Mathematics and Computing (CEISMO) at the Georgia Institute of Technology in Atlanta, Georgia.



KATIE HENRY

Katie Henry (@KatieHenryDays) is the Director of Learning for BirdBrain Technologies, creators of the Finch Robot and Hummingbird Robotics Kit, in Pittsburgh, Pennsylvania.

share their work. Invite your students into the planning process of sharing their work.

Be surprised

If this is your first time integrating the arts and computer science, you may not know what to expect. That's great! True creativity involves a surprise along the way; if you already know exactly what is going to happen, you might not be creating something new. Allow yourself to be surprised by giving up a little bit of control over the final product. (HW)

■ A student sketch detailing the design of their art project, including robotic components



CREATIVE CODING IN THE CLASSROOM WITH SONIC PI

Sam Aaron highlights the links between music and programming, and explains why he created Sonic Pi, a code-based music creation and performance tool

Code is one of the most creative media that humans have created. The initially obscure symbols of parenthesis and lambdas are not just deeply rooted in science and mathematics, they are the closest we have managed to get to casting the same kind of magical spells as Gandalf and Harry Potter. I believe that this provides a powerful means of engagement in our learning spaces. Through the magic of code, we are able to conjure up individually meaningful stories and learning experiences.

We are surrounded by magical experiences: from the sleight of hand of a stage magician making the ball disappear into thin air, to the wonder of seeing your favourite band perform on a big stage. It is these 'wow' moments that inspire us to pick up a magic book and learn the French drop, or to start jamming power chords on an old guitar. How might we create similar deep and lasting senses of wonder that will motivate people to practise and learn the fundamentals of programming?



SAM AARON

Sam (@samaaron) is the creator of Sonic Pi, an international public speaker, and live coding musician. Sam has performed by writing code live using Sonic Pi in nightclubs, music festivals, school assemblies, and the Royal Albert Hall.

Musical engines and notation

The histories of music and computers have been intricately woven together since the inception of computing machines, or 'engines' as Charles Babbage's powerful analytical engine was called. Back in 1842, the mathematician Ada Lovelace, who worked very closely with Babbage, saw the creative potential of these engines. Whilst these first engines had originally been designed to accurately solve hard maths problems, Ada dreamt about making music with them:

"...the engine might compose elaborate and scientific pieces of music of any degree of complexity or extent." Ada Lovelace, 1842.

Of course, today in 2019 much of our music, regardless of genre, has either been composed, produced, or mastered with a digital computer. Ada's dream came true.

It is possible to trace the history back even further. If you see coding as the art of writing sequences of special symbols that instruct a computer to do specific things, then musical composition is a very similar practice. In Western music, the symbols are black dots positioned on a stave of lines that tell the musician which notes to play and when. Intriguingly, if we trace the roots of Western music notation back to the Italian Benedictine monk Guido d'Arezzo, we find that the dots and lines system that modern orchestras use is just one of a number of notation systems he worked on. Some of the

others were much closer to what we might now see as code.

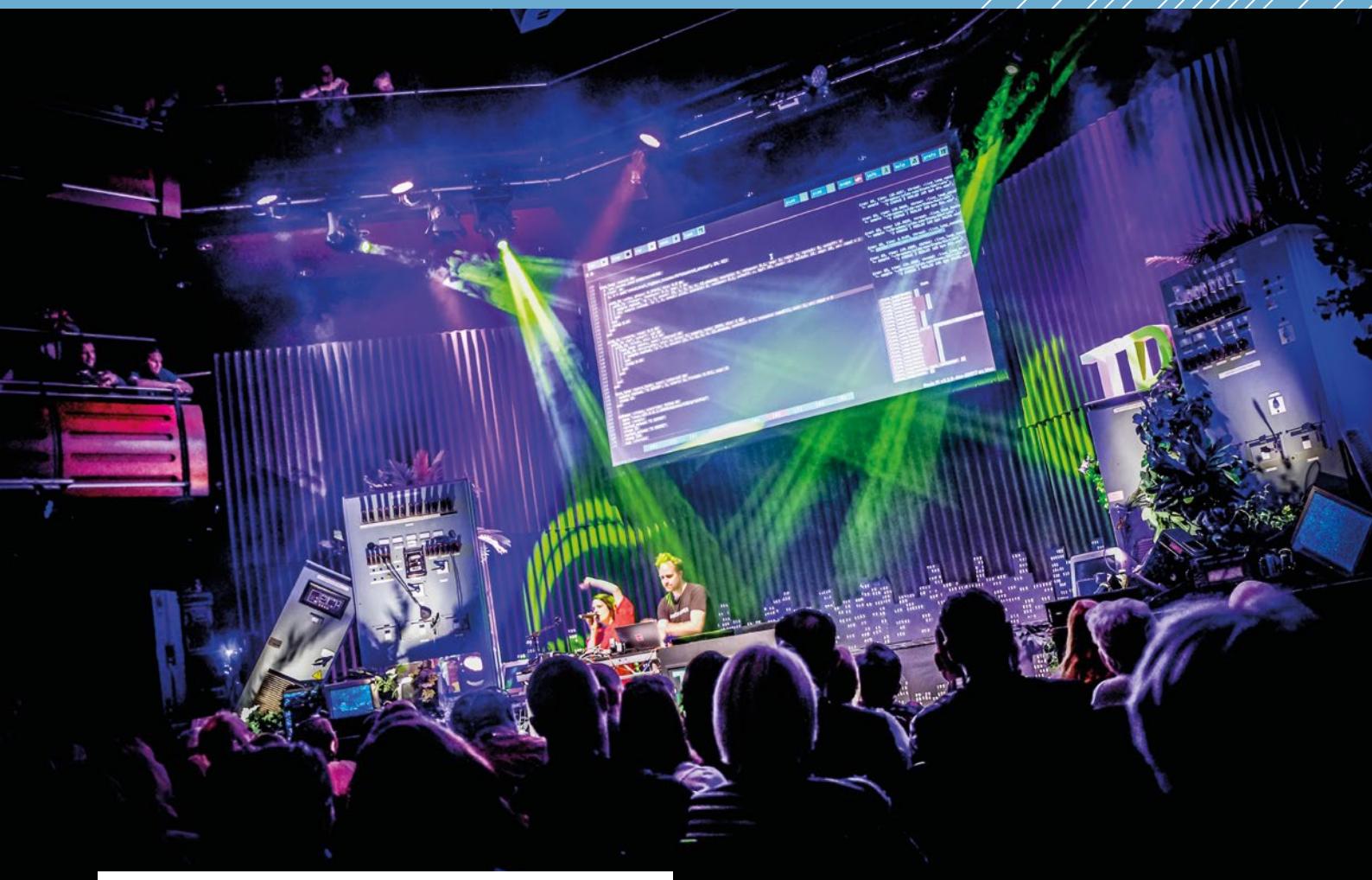
In education, magical, meaningful experiences with computers and programming languages have been explored since the late 1960s. Computer education pioneers Seymour Papert, Marvin Minsky, and Cynthia Solomon explored simple Lisp-based languages that moved pens over large pieces of paper. With just a few simple commands, it was possible to program the computer to draw any picture. They even experimented by extending their Logo language from drawing to music. Papert

**“SONIC PI HAS
UNDERGONE
ITERATIVE
IMPROVEMENTS”**

wrote about learning through experiencing the reconstruction of knowledge, rather than its transmission. Getting people to play with things directly was an important part of his group's work.

Live coding in the classroom

Sonic Pi is a code-based music creation and performance tool that builds on all of these ideas. Unlike the majority of computing education software, it is both simple enough to use for education and also powerful enough for professionals. It has been used



Jylда and Sam Aaron perform at the Thinking Digital Conference in the Sage Gateshead Photo: Tynesight Photos

to perform in international music festivals, used to compose in a range of styles from classical to EDM and heavy metal, and was even reviewed in the *Rolling Stone* magazine. It has a diverse community of over 1.5 million live coders, with a variety of backgrounds, all learning and sharing their ideas and thoughts through the medium of code. It is free to download for Mac, PC, and Raspberry Pi, and includes a friendly tutorial that assumes you know nothing about either code or music.

Sonic Pi was initially conceived as a response to the UK's newly released Computing curriculum in 2012. The goal was to find a motivating and fun way to teach the fundamentals of programming. It turns out that there is a lot in common and it's huge fun to explain sequencing as melody, iteration as rhythm, conditionals as musical variety. I developed the initial designs and first iterations of the platform with Carrie Anne Philbin, who brought a

teacher's perspective to the project. Since then, Sonic Pi has undergone iterative improvements, thanks to the feedback gained from observing learners and collaborating directly with educators in the classroom. A core design philosophy was to never add a feature that couldn't be easily taught to a ten-year-old child. This meant that most ideas had to be heavily refined and reworked until they were simple enough. Making things simple, whilst keeping them powerful, continues to be the hardest part of the project.

In order to provide the magical motivation, Sonic Pi's design was never limited to a pure focus on education. Ideally, there would be famous musicians and performers using Sonic Pi as a standard instrument alongside guitars, drums, vocals, synths, violins, etc. These performers would then act as motivational role models demonstrating the creative potential of code. For this to be possible, sufficient focus and effort

SONIC PI PERFORMANCES

Sonic Pi has been used to perform in a wide range of venues such as school halls, nightclubs, outdoor stages at musical festivals, college chapels, and prestigious music venues - for example, at the amazing Convo project, which brought 1000 children together in the Royal Albert Hall to perform an ambitious new composition by composer Charlotte Harding. The piece was written for traditional instruments, choirs, percussion, and Sonic Pi code. The pop-artist Jylда also performed with Sonic Pi in the Sage Gateshead for the Thinking Digital Conference, where she created a unique live-coded improvised remix of her song *Reeled*.

▶ therefore had to be placed on making it a powerful instrument whilst still keeping it simple enough for ten-year-olds to pick up. In addition to educators, I worked directly with a variety of different artists in classrooms, art galleries, studios, and venues in the early stages of Sonic Pi's development. This provided essential feedback, which enabled Sonic Pi to grow and ultimately flourish as a tool for creative expression.

There were a number of exciting and unexpected side effects of this dual focus on education and professional musicians. Many of the features are beneficial to both groups. For example, a lot of effort has been put into making error messages more friendly and useful (rather than being a huge complicated mess of jargon). This turns out

IT IS A REAL JOY TO SEE PEOPLE LEARN TO USE CODE TO EXPRESS THEMSELVES

to be very useful when you write a bug while performing in front of thousands of people. Additionally, functionality such as playing studio-quality audio samples, and providing access to live audio from the microphone, all turn out to make the learning experience more fun and meaningful.

The Sonic Pi community continues to grow and share compositions, lesson plans, and more. Much of this happens on our forum (in-thread.sonic-pi.net) which is home to a diverse group of people including educators, musicians, artists, and makers. It is a real joy to see people learn to use code to express themselves in new ways and for that in turn to inspire others to do the same.

Sonic Pi is available free to download on Mac, PC, or Raspberry Pi at sonic-pi.net. The website also features a tutorial. For ideas of how you could use Sonic Pi in the classroom, turn to pages 76–77. (HW)



■ Sonic Pi used as one of the instruments as part of Convo at the Royal Albert Hall Photo: Pete Jones

SOME FUN CAPABILITIES OF SONIC PI

From a computer science perspective, Sonic Pi provides the building blocks to teach you the basics as found in the UK's curriculum, such as sequencing, iteration, conditionals, functions, data structures, and algorithms. However, it also builds on a number of important and relevant concepts that have become adopted in mainstream industry, such as concurrency, events, pattern matching, distributed computing, and determinism - all while keeping things simple enough to explain to a child. Getting started is as simple as:

```
play 70
```

A melody can be constructed with one more command, `sleep`:

```
play 70
sleep 1
play 72
sleep 0.5
play 75
```

In this example, we play the note 70 (roughly the 70th note on a piano), wait for one second, play note 72, wait for half a second and then play note 75. What's interesting is that, with just two commands, we have access to virtually all of Western notation (which notes to play and when) and learners can code any melody they've ever heard. This leads to huge variety in expressive outcomes, whilst focusing on the same computing concept: sequencing in this case.

Taking ideas from the professional music world, we can also play back any recorded sound. Sonic Pi can play any audio file on your computer, but also has a number of sounds built-in to make things easy to get started:

```
sample :loop_amen
```

This code will play back the Amen drum break which was a pillar-stone of early hip-hop, drum and bass, and jungle genres. For example, a number of early hip-hop artists played this drum break back at half speed to give it a more laid-back feeling:

```
sample :loop_amen, rate: 0.5
```

In the 1990s, a number of music scenes burst out of new technology, which enabled artists to take drum breaks like this apart and reassemble in a different order. For example:

```
live_loop :jungle do
  sample :loop_amen, onset:pick
  sleep 0.125
end
```

In this example we introduce a basic loop called `:jungle` which picks a random drum hit from our audio sample, waits for an eighth of a second, and then picks another drum hit. This results in an endless stream of random drum beats to dance to whilst you experience what a loop is.



DYLAN LEWIS TEACHER

A CHANGE FOR WALES

Dylan Lewis shares his experience of teaching in a school that is leading the development of the new Welsh curriculum

The new curriculum in Wales, developed following a review by Professor Graham Donaldson and due to be introduced in 2022, is said to be one of the biggest shake-ups of education in the country in a generation. With a focus on skills embedded across the curriculum and the same level of importance placed on computer skills as literacy and numeracy, its announcement has had a mixed reception.

The current Welsh curriculum was developed and introduced in the late 1980s. This was a different era technologically, and a change was needed to enable students to thrive in a digital world. The digital competence framework introduces digital skills across the new curriculum, preparing students for the opportunities and risks that an online world presents. Like literacy and numeracy, digital skills will be a cross-curricular responsibility.

I am fortunate to work in a Pioneer School – a group of schools drawn from a mix of contexts across Wales, implementing and developing the new curriculum. One of the biggest transformations has been the move from single subjects to six new Areas of Learning and Experience (AoLE). Over the last year, I have been involved with developing the SciTECH AoLE – comprised of digital technology, design technology, and science skills. My work

has been focused specifically in digital technology. From the outset, my vision would be one which would focus on physical computing, to make the subject as engaging as possible by getting pupils interacting with hardware.

My quest started by purchasing a set of micro:bits. We wanted to ensure that each of the three subject areas within SciTECH had some input, to allow a seamless transition of skills throughout the subject areas. By working as a group, we have tried to ensure as many skills can be passed between departments to allow for purposeful learning. Working with the micro:bits has been incredibly engaging for our pupils, and has allowed for other spin-off projects, such as elastic band-powered cars that allow students to investigate speed, distance, and time, as well as developing their programming skills in Python.

Soon after the introduction of the new curriculum, I found that pupils were eager to enhance their learning in digital technology. My classroom is now packed during lunch hours, and students started by using their free time working on a project for Technocamps, a competition run by Welsh universities – which they went on to win.

The curriculum that allows our pupils to be ambitious learners who are unafraid of challenge. As with implementing any new curriculum, we have had to iron out a few issues along the way. All new initiatives come with problems but effective planning and rigorous evaluation has successfully allowed for progression.

The new curriculum starts in 2022, and I feel privileged to be part of a school that embraces innovative principles. This not only strengthens our pupils' learning experience, but could help revolutionise education as we know it. 

DYLAN LEWIS

Dylan is a teacher at Pontarddulais Comprehensive School in Swansea, South Wales.

SUBSCRIBE

SIGN UP TODAY FOR **FREE**



**SUBSCRIBE
TODAY**

- Get five term-time issues
- Have them delivered directly to your door
- Hello World is not available in stores!

(Hello World)



**FREE
IN PDF**
For anyone,
anywhere!

TO SUBSCRIBE VISIT:
helloworld.cc/subscribe

**Not a UK-based educator?
Buy in print for £6
Visit: store.rpiress.cc**

**SUBSCRIBE ON
APP STORES**

- Direct to your mobile
- For both Android & iPhone
- Back issues available



#INSIGHTS

LEARNING TO PROGRAM VIA VIDEOS AND SELF-EXPLANATION

STORY BY Lucia Flóriánová

Results from a small study of high school students in Denmark suggests that an approach that combines guided instructional videos with questions that prompt students to reflect on their learning can help them learn to program. The researchers have named the approach 'stepwise self-explanation'.

Novice learners face various difficulties while acquiring programming skills. In particular, they struggle to combine and use basic structures to build a program and apply acquired knowledge to new situations.

Making programming easier

Aureliano, Tedesco, and Caspersen (2016) explored how to make learning of programming easier for novices. They used an approach where students answered questions that prompted them to explain what they were learning as they watched instructional videos on how to build a program.

The video instructions were taken from 'The Joy of Code' series, which was developed to teach students to program in Java via the Greenfoot development tool.

The videos are structured according to the stepwise improvement framework, providing step-by-step guidance. After watching video instructions about a piece of code, students answered questions about its purpose and functionality. For example, they explained which part of code made an object move. The researchers compared this method with another approach, where students only self-studied the same videos but did not do any self-explanation. Although the research



Students watched videos from 'The Joy of Code' series

STEPWISE IMPROVEMENT

Stepwise improvement is a framework that advises to learn programming by developing small pieces of code in a systematic and incremental way. It encompasses three types of activities that programmers use to build a program: extension, refinement, and restructuring.

Stepwise self-explanation is an approach to learning programming that combines the stepwise improvement framework with self-explanations. In this study, this was done through instructional videos and a list of questions about them.

“ THE STEPWISE IMPROVEMENT FRAMEWORK IS LIKE TAKING STUDENTS FOR A GUIDED TOUR

cannot confirm how significant the difference between the two groups is, there was a positive correlation between students' self-explanations and their final results. Students also left positive feedback.

What makes the approach work?

Caspersen and Kölking (2009) explain that the stepwise improvement framework is like taking students for a guided tour instead of leaving them to walk around randomly – it draws attention to the most important aspects of programming education. It breaks down the content and gives structure to instruction materials that guide learners through programming processes. However, the structure of the instruction materials itself is not enough to make novice programmers learn, as they do not yet know how to study for the needs of a programming course. This can be mitigated through the use of

self-explanation, which guides students further through the instruction materials. Previous studies have found that students who generated explanations of instructions or programs learnt more and scored better in problem-solving.

Students enjoyed this method of learning. They appreciated that the videos presented the content step by step, and thus found them easy to follow. They also liked that they could rewind or re-watch videos when needed. Groups using self-explanation commented that the question prompts helped them to remember things and summarise what they'd learned.

Stepwise self-explanation in the classroom

Efficient teaching of programming takes students through the right level of detail when they are ready for it, but also prompts

them to think about what the code is doing as they learn. Stepwise self-explanation is one way to provide a balance between teachers' guidance and students' cognitive engagement. Providing a small number of questions for students to answer can direct their attention to the most important learning objectives and computing concepts. Using instructional videos that are structured around small pieces of code guide is not only engaging, but also guides students through a new topic giving them the opportunity to come back to the parts they missed or about which they are unsure. (HW)

FURTHER READING

- ✓ Aureliano, Tedesco, and Caspersen (2016): helloworld.cc/aureliano2016
- ✓ Caspersen and Kölking (2009): helloworld.cc/caspersen2009
- ✓ The Joy of Code videos: helloworld.cc/joyofcode

Well-placed questions can direct students' attention



DEFINING DATA LITERACY

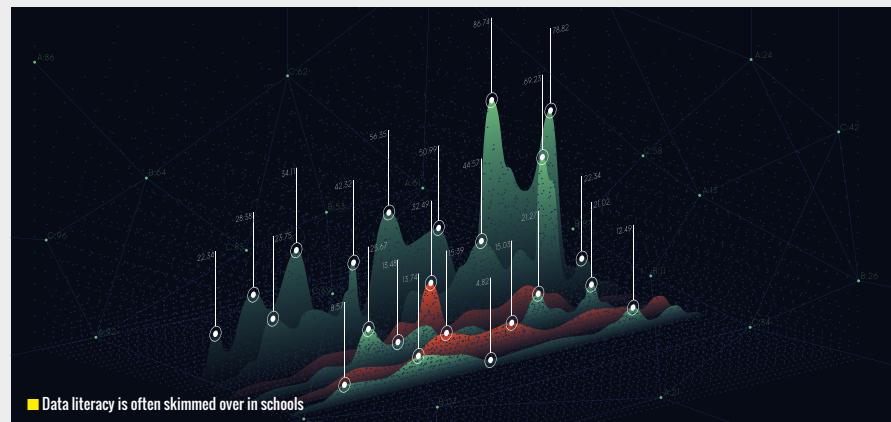
STORY BY Oliver Quinlan

The use of data is a powerful and ever increasing force in the modern world. Vast datasets drive many of the applications of computing that affect our day-to-day lives, and it is increasingly accepted that modern scientific research is a data-driven field.

While many computing educators and researchers recognise the importance of data literacy, it is an area that is often not covered in great detail in computing curricula in schools. Researchers Andreas Grillenberger and Ralf Romeike have set out to develop a competency model for data literacy to help educators to address this important area.

Content and process

The model has been created based on a review of a wide range of research literature into data management, data science, data ethics, and the practicalities of handling and processing data. It splits data literacy into two categories: content and process. However, these are interleaved to emphasise the strong links between them. The intention is that students should be supported to develop strong skills in working with data, and that these skills should be underpinned by a good



understanding of the nature of the content area for the data they are working with.

The links between adjacent content and process areas are perhaps most obvious, for example the content area 'data and information (C1)' and the process area 'gathering, modelling, and cleansing (P1)'. All content and process areas can be usefully linked, though, so understanding 'data ethics and protection (C4)' is also important when 'gathering, modelling and cleansing (P1)'.

The framework in action

The authors used their model to design a series of lessons that were aimed at

developing students' understanding of both the content areas and the process areas. One lesson that clearly illustrated this approach included an activity where students analysed anonymised data from Portuguese students. The data contained information such as the students' habits and family situations, alongside their exam grades.

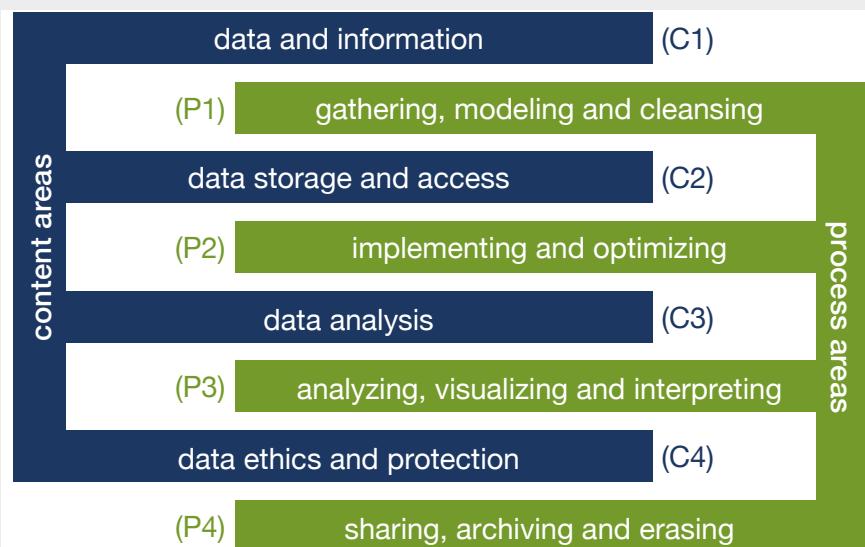
The lesson encouraged students to explore the data using the tool Orange, which allows analysis using a graphical interface. Students were able to explore aspects of the data that interested them, and develop their 'analysing, visualising, and interpreting' processes. What they found also prompted discussions around the limits of data and the ethics of using factors such as family situation to predict student achievement. The lesson developed practical skills, but also engaged students in exploration of the nature of data as content by using this interleaved model.

Developing data literacy

This data literacy competency model provides a framework for educators to plan to develop data literacy that is based in research. The model supports educators to consider how they develop data literacy in ways that allow students to develop practical skills and a deep understanding of the content involved in data literacy.

You can read the full paper on this work at helloworld.cc/Grillenberger2018. (HW)

■ The data literacy competence framework by Grillenberger and Romeike





PETER KEMP LECTURER

THE ROEHAMPTON REPORT

A pre-release snapshot of the Roehampton Annual Computing Education Report hit headlines in the UK last month, with the BBC reporting, “Computing in schools in ‘steep decline’”. Here, the lead author of the report, **Peter Kemp**, outlines its findings

The 2014 change in curriculum from ICT to computing was welcomed by people working in schools, who felt it an educational right of all students to understand how computers work and how to program them. This change was accompanied by the introduction of a GCSE in Computer Science. Our latest report (helloworld.cc/Roehampton2019) shows that, in 2018, there were over 70,000 students sitting GCSE Computer Science, and 79% of students were in a school where the GCSE was on offer, both up slightly on 2017. We are right to celebrate this, but at the same time we have to recognise worrying digital divides that correspond with these changes.

In 2018, 81% of schools serving the richest communities in England had students entering the GCSE CS exam, compared to only 69% of those schools

PETER KEMP

Peter (@peterejkemp) is a Senior Lecturer in Computing Education at the University of Roehampton, where he runs the PGCE. His research includes work into digital equity and digital creativity. The Roehampton Report has been running since 2017.

PERSPECTIVE: CEREDIG CATTANACH-CHELL

SUBJECT ADVISOR IN COMPUTER SCIENCE, IT AND CREATIVE IMEDIA, OCR (OXFORD CAMBRIDGE AND RSA EXAMINATIONS)



Despite the challenges for schools starting to teach computing after years of teaching ICT, I believe the picture is more positive than the report suggests.

OCR reintroduced the GCSE in Computer Science a few years ago and the numbers of students taking our GCSE continues to grow. And there are more vocational options available for schools too, such as our Cambridge National in Information Technologies and in Creative iMedia. The number of students taking these, as well as the Computer Science A-level, is growing rapidly.

With the emerging teacher training and support available through the NCCE, I feel that the determination of teachers to upskill and deliver the new computing curriculum to students can be recognised and applauded. The future is positive!

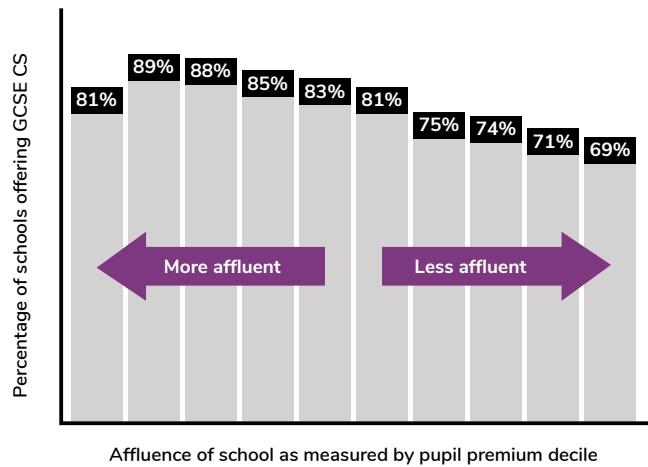
PERSPECTIVE: ELEANOR OVERLAND

SENIOR LECTURER IN COMPUTING EDUCATION, MANCHESTER METROPOLITAN UNIVERSITY



As part of my research, I try to look at the stories behind the data. For example, in one of the schools I have been carrying out research, they did have a difficult time. They could not recruit a computing teacher but had capacity within their expressive arts department. As a logistical necessity, the Year 7 computing lessons were replaced with 'digital art' - art teachers using computing classrooms. This could be interpreted as a disaster for computing education in the school, or an ingenious way to develop a more creative approach. The data will reflect it as a reduction in computing curriculum time, but individual stories might make us think about it in a different way.

“
IT WAS NEVER THE INTENTION THAT COMPUTER SCIENCE WOULD REPLACE ICT



■ Schools in areas of England with the highest proportions of children living in low income households were least likely to offer GCSE Computer Science in 2018

➤ serving the poorest communities. These worrying statistics support our previous research showing that poorer students were far less likely to take CS than their richer peers. The overall number of schools offering GCSE CS was up slightly from 59% in 2017 to 61% in 2018. However, there are concerning changes among girls-only comprehensive schools, with 19% of those offering the exam in 2017 dropping it in 2018.

Computing incorporates computer science, information technology, and digital literacy: the old elements of ICT remain present and important. It was never the intention that computer science would replace ICT, and our research looks beyond just GCSE Computer Science to all computing education. Most shocking, maybe, is the reduction in the number of hours of computing,

PERSPECTIVE: SUE SENTANCE

CHIEF LEARNING OFFICER AT THE RASPBERRY PI FOUNDATION



The Roehampton Report shows that GCSE Computer Science is gaining traction in school, with an increase in students sitting their exams in the subject between 2017 and 2018. This is to be welcomed, especially given that it is such a young subject and we have so much still to learn about what it should include, and how it should be taught and assessed. Once the impacts of the National Centre for Computing Education are felt, I believe we will see an even greater level of engagement with the subject at GCSE, thanks to this massive investment in the subject.

However, we also need to look at digital education in England more broadly than uptake at GCSE. In England, we are fortunate to have a space for computing in the curriculum that many other countries don't have. From the age of five, children have access to a wide range of skills and knowledge that will help them to become digitally literate.

The report also highlights issues around gender balance in computing. The UK government recently announced investment to trial initiatives that aim to increase the uptake of female students at GCSE and A-level Computer Science, further details of which can be found on pages 6-7.



■ While the report shows an apparent decline in the number of hours spent on computing in the classroom, some argue the data should be treated with caution

measured as a total of ICT, computer science, and other digital subjects, taught in schools across the country. Our research shows a 26% decrease in the hours at Key Stage 3 (ages 11-14) and a massive 47% decrease at Key Stage 4 (ages 14-16) over the five years from 2012 to 2017. For those students not taking a qualification in computing, it looks likely they will be getting no computing education past the age of 13 or 14: for some, that start studying for the GCSE a year early, this will mean no computing education from as young as 12. This decline began before the introduction of the new curriculum, but has continued since then. Further reductions appear highly likely going forward, with the steep drop in qualifications in 2018, and the removal of the ICT GCSE in 2019.

In 2018, 43,000 students were in a school where no computing qualification was offered, almost doubling from 24,500 in 2017. Accompanying this is the steep decline in the number of qualifications taken by

PERSPECTIVE: WILLIAM LAU

LEADER OF KEY STAGE 4 COMPUTING AT CENTRAL FOUNDATION BOYS' SCHOOL AND AUTHOR OF 'TEACHING COMPUTING IN SECONDARY SCHOOLS'



The Roehampton Report brings to light a few trends. The data provides support to views mentioned anecdotally on social media by computing teachers. Firstly, it is great to hear about the increased uptake in computer science at Key Stage 4. However, this has been at the cost of fewer students being offered a digital education at Key Stages 4 and 3. This can be explained by the removal of GCSE qualifications in IT/Digital Literacy. The resulting decline in hours and number of students securing a broad digital education is somewhat concerning. I think we can all be advocates of computer science, whilst also raising awareness for the need of a balanced Key Stage 4 computing offering which includes elements of IT and Digital Literacy. The trickle-down to Key Stage 3 would ensure more students leave school with some form of digital education.

PERSPECTIVE: MICHAEL KÖLLING

*PROFESSOR OF COMPUTER SCIENCE,
KING'S COLLEGE LONDON*



I would caution that the data in the Roehampton Report must be interpreted with care and awareness of how it might be misread. The introduction of the report, which also serves as a summary, makes some general statements pointing out some

steep 'decreases' in various measures and uses terms such as 'decline', which are likely to be picked up and presented as the overall conclusion. We have seen this leading to headlines such as the BBC's, "Computing in schools in 'steep decline'" and similar summaries elsewhere. This is highly misleading.

The introduction of Computer Science in UK schools is not failing, as the secondary reporting seems to suggest. The misleading picture has two sources: First, the umbrella-reporting of Computer Science, IT, ICT, and digital literacy under the single term 'Computing' is not helpful. These are very different subjects. The 'decline' is in the ICT area, which was expected, and not in Computer Science, as it is being interpreted by the BBC and elsewhere. Reporting these different disciplines under one headline is like bundling physics and woodwork under a single name, and making statements about them as if they were the same.

The second problem is timing: many of the efforts to support computer science, including training and funding, have not yet taken effect. Observing effects in teaching interventions in multi-year qualifications takes time. It is not surprising that effects are not yet visible - reporting this in language that makes it sound like failure is unhelpful.

The report is in danger of doing us a disservice: to the casual reader it seems to create the headline, 'The introduction of computing in UK schools has failed', while in reality we are going through a difficult transition phase that we knew was coming.

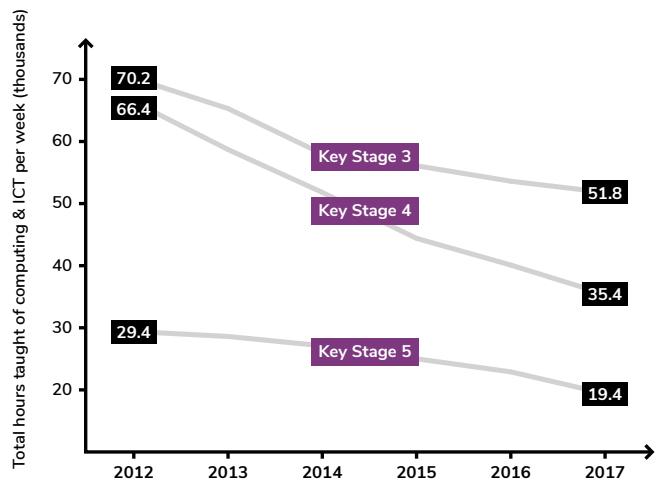
In all, the call to action in the report, warning that we must take care not to leave pupils behind, is important, should be applauded, and the developments closely watched. The summaries, however, paint a picture that may be unnecessarily bleak.



ECDL SIGNIFIED AT LEAST SOME FORM OF DIGITAL EDUCATION WHERE THERE WILL NOW, FOR MANY, BE NONE

students, dropping by 144,000 or 45% in a year. This can be largely explained by schools no longer offering the European Computer Driving Licence (ECDL), a qualification that few would argue should ever have featured in league tables. However, we cannot just dismiss this finding: for all its many faults, ECDL signified at least some form of digital education where there will now, for many, be none.

Is it too early to talk about the impact of curriculum change? I don't think so. While the reasons are complex, it looks likely that hundreds of thousands of young people, particularly girls and poorer students, are less likely to have a digital education over the next few years. **(HW)**



■ The most recent figures show a decline in the hours dedicated to teaching computing (measured as a total of ICT, computer science, and other digital subjects) at Key Stages 3-5



MILES BERRY LECTURER

WE NEED A GCSE IN COMPUTING

Let's replace Computer Science with a qualification every student can enjoy

En gland's education inspectorate, Ofsted, are turning their attention to the curriculum, recognising, at last, the importance of its breadth and its intent. In their new school inspection handbook, they state that a curriculum should be rooted in a consensus about the skills and knowledge that pupils need and that it should remain as broad as possible for as long as possible. I think they're right.

How should school leaders decide what should be taught? Curriculum design should be about preparation for the responsibilities and opportunities of life, further study and employment, and about passing on the best that has been thought and said to the next generation.

These priorities underpinned the design of the English national curriculum for computing. This includes much that pupils will find useful, but also things which might perhaps spark joy. Like the Royal Society report that preceded it, the curriculum identifies three elements of computing: computer science (including programming), information technology, and digital literacy. It's helpful to think of these as the foundations, applications, and implications of computing. What we have here is a statutory entitlement to a rounded education in computing knowledge and skills for children and young people from 5 to 16.

However, curriculum standards are only a starting point. Teaching matters. So does assessment. At times, it can seem that assessment is all that matters. For computing, it's increasingly clear that this is a problem – the removal of GCSE ICT and a cluttered, overly theoretical GCSE in Computer Science seems to have resulted in an

unbalanced experience of computing at Key Stage 4, fewer students leaving school with any computing qualification, and very unwelcome gender, socioeconomic, and academic attainment biases amongst those that do. This is despite the national curriculum's requirement that at Key Stage 4, "All pupils must have the opportunity to study aspects of information technology and computer science at sufficient depth to allow them to progress to higher levels of study or to a professional career."

What then is to be done? Improving the adoption, teaching and uptake of GCSE Computer Science amongst schools, teachers and students will help. However, we now ought to rethink the GCSE, creating a qualification which reflects the whole breadth of computing, which recognises the creative and human elements of the subject, alongside the scientific ones.

I have in mind a qualification that could sit at the core of the Key Stage 4 curriculum, covering things 'essential for learning and life', framing this as an up-to-date, broad specification covering principles and practice for programming, data, media, the web, security and ethics, with an intake that is far more inclusive and representative than what we see at the moment. [\(HW\)](#)

MILES BERRY

Miles (@mberry) is Principal Lecturer in Computing Education at the University of Roehampton. He's a member of the Raspberry Pi Foundation, and serves on the boards of CAS and CSTA.



SIMON PEYTON JONES PRINCIPAL RESEARCHER

A RESPONSE TO ROEHAMPTON

The Roehampton Report gives us some valuable insights into the state of computing education in England. We should use these insights to inform a thoughtful debate

The recent Roehampton Report makes for sobering reading: it looks as if everything is going in the wrong direction. There are real grounds for concern, but it's too early to panic.

The context

The National Curriculum Programmes of Study for computing sets the aim that all pupils:

- can understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms, and data representation
- can analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems
- can evaluate and apply information technology, including new or unfamiliar technologies, analytically to solve problems
- are responsible, competent, confident, and creative users of information and communication technology

SIMON PEYTON JONES

Simon Peyton Jones is a Principal Researcher at Microsoft Research and an Honorary Professor of the Computing Science Department at Glasgow University. He chairs Computing at School and the National Centre for Computing Education.

Notice 'all pupils'!

Ours may be the only country with the aspirational goal that all children encounter computing both as users and as a subject discipline in its own right. In particular, it establishes computer science as a foundational subject discipline that, like natural science, every child should learn at an elementary level, starting at primary school. Why? Because everyone – from plumbers to brain surgeons – needs an elementary understanding of the digital world that surrounds them, so they can bend it to their will not the other way around. It's a big vision.

A system in transition

Establishing an entirely new school subject, from primary onwards, is a pretty challenging undertaking for our schools, and especially for our teachers: it asks them to step up to a subject in which many of them have not been trained.

Simultaneously, the landscape of Key Stage 4 qualifications has undergone a substantial overhaul, across all subjects but particularly in computing: the grading system for all GCSEs has changed; the content of the GCSE in Computer Science has been revised, the GCSE in ICT has been withdrawn outright, the set of non-GCSE qualifications in computing – so-called Technical Awards – has been radically culled; and practical work no longer contributes directly to exam marks.

And that is not all: four years after the introduction of the new curriculum in 2014, the government launched the

National Centre for Computing Education in late 2018, specifically to support and equip schools to give their pupils an inspirational, world-class education in computing. This is very good news, but it's too early to expect to see impact.

Given all this turmoil, we should be cautious about what conclusions we can draw from the figures in the Roehampton Report. For example, the examination data it contains reflect choices made by schools and pupils three years ago, in spring 2016, before the NCCE was even thought of; and much of the decrease reflects the withdrawal of ECDL (the European Computer Driving Licence) from the school performance tables, where it should never have been in the first place.

The drop in curriculum time devoted to computing is more concerning, and I welcome having quantitative data on this question.

MOST OF THESE QUALIFICATIONS ARE NEW, AS IS THE CURRICULUM ITSELF

Cohort groups

Looking specifically at Key Stage 4 (age 14–16), which is the focus of the examination data in the Roehampton Report, it may help to consider at the needs of different groups:

- **Computer scientists.** Some young people will elect to take a GCSE in Computer Science. I am thrilled that we have such a qualification at age 16 at all (it did not exist until 2010). One might worry about some of the details (Is the content right? Is it too hard compared with other GCSEs?) but it is a challenging, high-status qualification that leads into A-level. Less happily, after several years of rapid growth, the proportion of students taking GCSE CS has flattened off at 12%. I would like to see that increase very substantially (double, or more) and it's an area of specific focus for the NCCE.
- **Applied computing.** What about students who are interested in computing, but more in its applications than in computer science per se? There is no longer a GCSE in this territory, which I regret, but there are the Technical Awards. These too have been reformed, and they are not

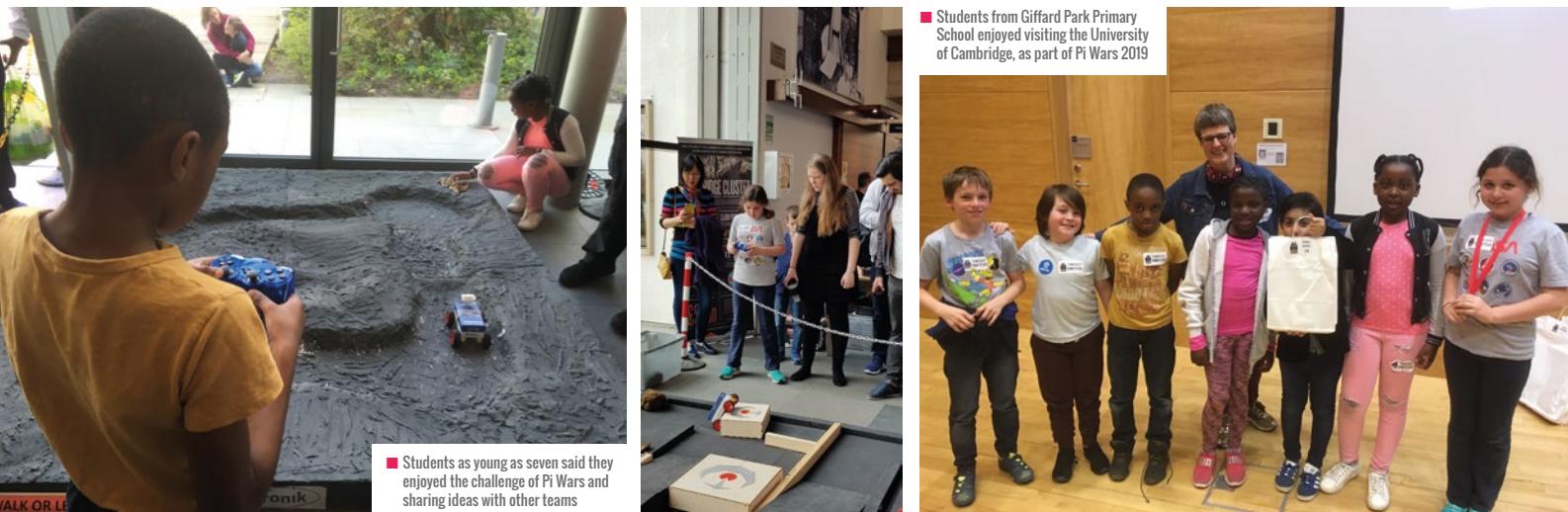
easy. But the pathway exists, the qualifications count for school league table points, and uptake is growing rapidly. Will all schools offer this route? Does it meet the needs of those applied-computing students?

- **Non-specialists.** Finally, we have the young people who, given a limited number options at KS4, choose art or history rather than computer science. Fair enough! But computing is one of a handful of 'foundation subjects' that every student is supposed to study right up to age 16. In practice, though, this requirement is simply ignored: at KS4, schools offer no education other than study leading to a KS4 qualification that counts for league-table points. For them, no pathway exists in computing.

There are big, structural issues here, and one should beware of knee-jerk reactions. Education is complicated. But neither should we sit on our hands.

- Many of these qualifications are new, as is the curriculum itself. Most significantly, the NCCE itself is brand new, and I think it will have a huge impact over time: as our teachers become more skilled and confident, students will find computing more rewarding and attractive. So we should monitor the situation closely, but move cautiously: there is much goodness in the pipeline, and repeated changes of direction are both demoralising for schools and confusing for students.
- We should debate what qualifications we actually want at KS4. Miles Berry's article in this issue is a contribution to that debate. I don't think we have a consensus in our community yet, and without that consensus it is hard to persuade government to change anything.
- We should make the case for a meaningful, substantive computing education at KS4 for students who (quite legitimately) do not want to take a high-stakes KS4 qualification; and we should rejig the incentives that schools face so that they can offer such a pathway without disadvantaging the student or the school. One way to do so might be for Ofsted to explicitly check that good and outstanding schools are doing this, something that both the BCS and Royal Society proposed in their responses to Ofsted's recent consultation.

The Roehampton Report gives us truly valuable insight. Rather than beat our breasts (though there is cause to do some of that), let's use that insight to inspire debate and thoughtful action. 



ROBOTS, COMPETITIONS, AND EXTRA-CURRICULAR OPPORTUNITIES

Sway Grantham shares the joys of running a coding club for primary students and her experiences of taking the youngest team to Pi Wars 2019

Teaching is a hard job with lots of demands on your time. Extra-curricular activities can seem like another burden in a busy week. However, they are also an opportunity to work with your most inspired, engaged, and interested learners. I recently took seven primary pupils to Pi Wars, an annual robotics competition, as the culmination of an after-school

club which had been running for three months. The competition involved children designing, building, and programming a robot to compete in seven space-themed challenges, with additional points awarded for blogging, artistic merit, and technical understanding. What they took away from this, however, was far more than ten hours' of 'extra computing lessons'.

Age is just a number

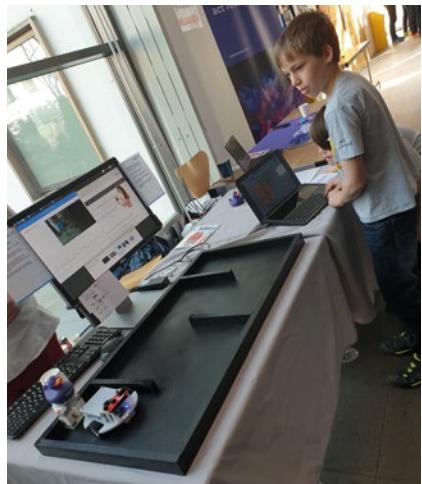
The British education system, for logistical reasons, puts a lot of emphasis on age. When you're a year older, you learn the next year's content and each child in the year will be the same age. This often gives children two misconceptions:

1. Older children/adults will always know more than them on a topic;
2. They're 'too young' for some knowledge.

The children in my club are aged seven to eleven years old, all from different classes. Some had attended Pi Wars the previous year, others had not, yet they all supported each other throughout the club and the

competition, and were the youngest team to enter. Yet, rather than being intimidated and put off, they loved having the opportunity to work with other people, talk to them about their projects, and learn together, even making some suggestions to older teams. Some of the children reported that their favourite part of the experience was how helpful the other teams were, and how happy they were to share ideas.

Entering Pi Wars gave the children an opportunity to see that children of all ages could bring their own unique skills to the table. They were aware that you need to learn some things before others – we didn't do sensors this year as we were focusing on a working robot – yet they acknowledged they could go on to learn this next. They also frequently suggested that 'anyone could do it', because there's so much helpful content online. This is a very different attitude to that which you experience in school, where the teacher provides the knowledge and they define your next steps.



TOP TIPS: CLUBS WITHOUT WORKLOAD

1. Don't over-plan - clubs are not lessons and therefore they do not need to be structured like them. Get the children to suggest what they'd like to do 'next week' and all you need is an idea of topics/activities, no minute by minute plan.
2. Resourcing - don't spend hours trying to get everything you need ready before your club after a full day of teaching. Pupils can come with you to get resources you need, or if your school allows, they can be sent in small groups to get bits themselves!
3. Tone - making the 'feel' of a club different from a lesson reduces the expectations of pupils and teachers to maintain the rigid structure of lessons. Do you need to stand at the front or can you all gather around a table? Do the children need to listen to you or can they discuss it together? Do you need to 'lead' the session or have they got ideas? Can they find answers online or in books? You don't have to be the most knowledgeable person in the room!

PI WARS GAVE AN OPPORTUNITY TO SEE THAT CHILDREN OF ALL AGES COULD BRING THEIR OWN UNIQUE SKILLS TO THE TABLE

There's more to life than sport

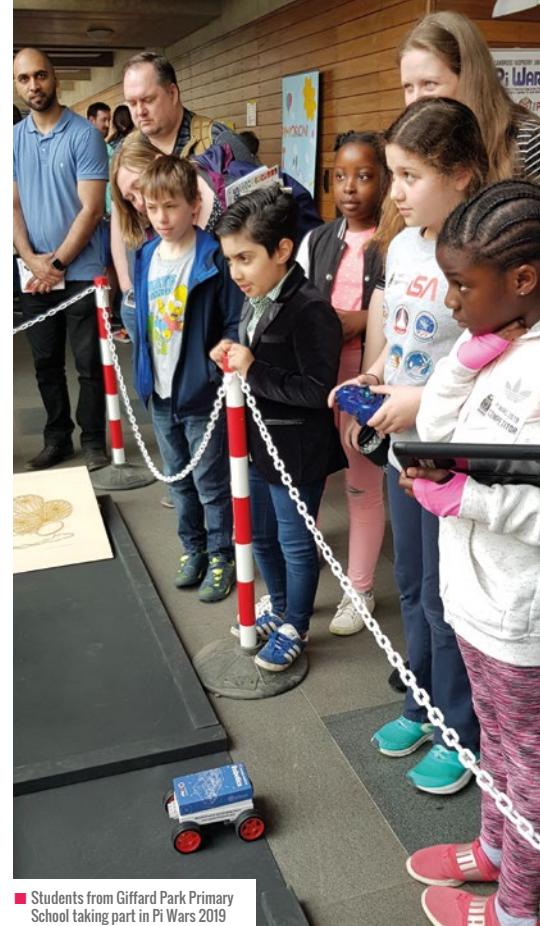
Sport is a really important part of growing up. It offers children a space to develop their teamwork and communication skills, builds resilience, all whilst keeping them fit and healthy. Yet, nearly all schools have a huge imbalance in the extra-curricular and competitive opportunities offered in sport versus other areas. The children who attended Pi Wars spoke positively about getting an opportunity to compete and develop their skills in computing as being a more 'unique' opportunity, but one where they were also able to develop academically. Once pupil stated, "We usually do a physical challenge, but this time we are doing a mental or robotic challenge which gives us tons more experience."

As a teacher, this opportunity for challenge is one aspect I love about running clubs. There are certain constraints on the type of projects you can do in a classroom due to curriculum coverage and managing 30 different children at a time, yet a club allows you this freedom. From attending one club, the children felt they'd had a 'creative experience' improving their art and design, design technology, and computing skills whilst having a chance to do an activity 'with clear purpose' working towards the competition. It was so clear that these

children valued the variety a computing club offered and the experience of competing – and if it helps them develop their skills at the same time, that's a bonus! Equally, every child spoke of the 'challenge' of attending – not in a negative, or frustrated way, but as an opportunity to take part in something where they did not know if they would succeed. How often do we allow the children to fail, while reassuring them that it's the experience they learn from, not the success?

School and beyond

The children I work with are very young, and have so much education ahead of them that their careers aren't really a consideration at this point, yet they saw older children and adult volunteers who were really invested in taking these skills towards future career paths. The children enjoyed talking to the adults, who had their own technology, about how they got started and how they would explain how things worked to them. They also saw adults who were passionate about learning even though they'd left school, something which some of them struggled to understand – why would you choose to learn after school stopped it being compulsory? The experience gave them another perspective on the purpose of learning. Finally, Pi Wars is held in the William Gates



Students from Giffard Park Primary School taking part in Pi Wars 2019

Building, home of Computer Science at the University of Cambridge, which is such a special opportunity for the children. Before the event, we talked about universities and what it was like to go to one and how it was similar and different to the school they know. It made them feel like they were more grown-up, just by being allowed into an 'adult' environment. This, being combined with such a supportive learning environment, has led to the children asking what they have to do to be allowed to go to universities like the one we visited. (HW)



SWAY GRANTHAM

Sway (@SwayGrantham) is a Computing Teacher at Giffard Park Primary School in Milton Keynes and Primary Curriculum Manager for the National Centre for Computing Education. She is a Raspberry Pi Certified Educator and a Specialist Leader in Education.



REALISING THE POTENTIAL FOR PROGRAMMING IN PRIMARY SCHOOL

John Parkin reflects on his experience of setting up an after-school Code Club for ten- and eleven-year-old children

As a newly appointed Computing Leader in a primary school in Cambridgeshire, I wanted to set up an after-school Code Club. My aims for the club were to help raise the profile of computing in the school, foster enthusiasm for the subject, and to develop my own understanding of the software available for computing.

Code Club is an after-school programme for children to learn about programming in an informal environment. Code Club provides a range of different projects ranging from Scratch to micro:bits. The sessions are run by teachers, or by volunteers from beyond the school, for children aged 9 to 13. Best of all, the resources are free to use.

I was eager for my Code Club to enthuse pupils about computing, but when I was planning the sessions I was unsure of how many Year 5 and 6 children would be interested in joining. To my amazement, the club was oversubscribed. However, it was manageable to get everybody into the club. As I had not taught the children before, I did not know their previous computing experience, so I gave them a simple activity from the Code Club projects. The children flew through and were soon elaborating and extending the activity. This first session told me that the children were more proficient at using Scratch than I expected, so I needed to select some more challenging projects!

Code Club organised a Global Coding Competition, and I thought this would be a

challenge that would motivate the children. We were learning about Sonic Pi, a music synthesizer which uses text-based coding. The children were excited to be entering the competition, and created some interesting music about their local community. The pupils had to vote for their favourite piece of music created, as we were only allowed one entry as a club. And a few weeks later, my pupils were delighted to hear that they had won a prize for their composition.

The Code Club activities have motivated children to do computing outside of school. Some have downloaded the free software used at school, and others have received a Raspberry Pi as a birthday present. This is fantastic to hear, as children must be enjoying what they are doing in school.

Code Club provides clear projects for children to work through

JOHN'S TIPS FOR PRIMARY CODE CLUBS

- Don't plan too far ahead for sessions - this will allow you to adapt to the interests and skills of the students.
- Try activities you wouldn't normally use in class.
- Give children the choice of working with friends or by themselves.
- Speak to children in an assembly, ideally with club members, about why they should sign up.

An opportunity for professional development

Most of my teaching experience has been in the Early Years Foundation Stage and Key Stage 1 (children aged four to seven), so a big part of my wish to set up a Code Club was to develop my own knowledge of computing in Key Stage 2 (seven- to eleven-year-olds). I was confident at using Bee-Bots, apps such as Kodable and Daisy the Dinosaur, along with Scratch at a fairly basic level. Through Code Club, I wanted to learn alongside the club members how to use Scratch at a level beyond that of a seven-year-old and have a chance to explore new software, such as Sonic Pi.

HOW TO START A CODE CLUB AT YOUR SCHOOL

Code Club supports educators to run coding clubs for 9- to 13-year-olds with free, step-by-step project guides for learning Scratch, Python, and HTML/CSS. The free online training is perfect for teachers who are new to coding or are starting a Code Club for the first time.

Find out more at codeclub.org and follow Code Club on Facebook at: facebook.com/codeclubuk, and Twitter: @CodeClub

“ BY SPENDING TIME IN THE CODE CLUB, I HAVE REALISED THE POTENTIAL FOR PROGRAMMING IN A PRIMARY SCHOOL

Code Club provides clear projects for children to work through for different software and hardware, such as Scratch, Sonic Pi, and Raspberry Pi computers. I could spend a short period going through the instructions and develop my understanding of the resource. I found that in the informal setting of a Code Club, I felt confident to try different ideas and software where I could learn with the pupils in the club, as opposed to a lesson during the school day where, with a classroom of 30 children, there is more pressure to make sure everything goes according to plan. Code Club gave me the time and space to have fun and take risks with computing, which has had an impact on my classroom teaching.

The projects created by Code Club have introduced me to new software, such as Blender, which is a program which allows primary-aged children to make 3D models and explore digital animation. I am planning to use this with the group this term. This is something I never imagined I would be doing in a school!

As the Computing Leader in the school, Code Club has been beneficial for understanding what computing looks like

in my context. By teaching children from different classes, I have learnt about their abilities and interests.

In addition to being a primary teacher, I am a lecturer in primary education. As I have found Code Club projects so useful for my own professional development, I have used them with undergraduates to teach them about Python.

Overall, I have found setting up a Code Club an immensely positive experience. Not only has it helped enthuse children about computing and raised the profile of the subject, but also it has been a useful professional development opportunity for me personally. By spending time in the Code Club, I have realised the potential for programming in a primary school. **(HW)**



JOHN PARKIN

John is a Senior Lecturer Practitioner at Anglia Ruskin University in Cambridge, where he teaches on the BA course in Primary Education Studies. He also teaches at Bourn Church of England Primary Academy near Cambridge.

The all-girls' Code Club at RGS The Grange is led by students



GIRLS CAN CODE

Matt Warne shares how a group of girls at his school have encouraged their peers take up coding in their spare time

Last summer, a group of pupils from my school approached me about a simple idea that they felt would be really popular with the school community. They wanted to run and oversee their own Code Club at lunchtimes, but there was one catch... it would be for girls only!

Pupil leadership is certainly not a new concept at RGS The Grange, as the school actively promotes an array of opportunities, ranging from French ambassadors to Digital Leaders. The thought process behind this is to give pupils a chance to take on responsibility within school and develop their personal skills within a specific role.

Milly, Daya, Lily, Sophia, Megan, and Rosie were regular attendees at the Thursday evening Code Club and, despite being outnumbered by boys, always

worked hard to develop their confidence and skills with Scratch projects. The girls were extremely keen to attract more of their female peers into the club, and wanted to encourage more girls to get involved in learning how to code. The Girls Can Code leaders also wanted to give something back to the school community from which they benefited so greatly.

"The Thursday Club was packed full of boys and we thought we needed more girls to learn how to code" – Rosie, Code Club Ambassador.

The girls came to me with this idea of launching their very own Code Club and without hesitation, I gave them a lunchtime slot which I could help facilitate. We then planned the logistics behind launching a club. The girls marketed the club through

a talk in assembly, posters, and even video trailers to entice people to sign up and get involved.

In September 2018, the girls launched their new Code Club. The club was aimed at young girls from Years 3 and 4, led by the female Code Club ambassadors. Giving up their lunchtime every Friday to run the club, the girls have been supporting up to 40 young and passionate seven- and eight-year-olds, who have been inspired by these young leaders. Despite there only being 21 computers in the IT suite, the girls promote a collaborative environment, and pair-up younger members of the group, so they can learn from each other.

Often the young leaders will start sessions with computing trivia quizzes, through apps such as Quizlet or Kahoot! The session



MATT WARNE

Matt (@MattWarne) is the Head of Computing and IT at RGS The Grange, an independent preparatory school for boys and girls aged 2-11 in Worcester. He is a CAS Master Teacher, and champions the Worcester Primary Community.

then progresses onto looking at the next Code Club project, where the leaders will offer tips and advice, before allowing the young attendees to immerse themselves in the joy of programming. The Code Club ambassadors always finish with a 'show-and-tell' session where they talk about things that have gone well (or not!), and then award the prestigious 'Girls Code The World' sticker to the chosen pupil of the session.



■ Young leaders at RGS The Grange spend their lunchtimes supporting up to 40 younger students in their Code Club

certificates had only been attainable by Year 5 and 6 pupils, yet these inspired coders are grasping a thorough understanding of key concepts and applying them into their weekly lessons.

Inspirational leaders

What the club leaders have achieved is inspirational, and is a testament to their passion for coding and making programming

the girls have used as a process to refine their skills. Whereas in the early days of the club they would turn up and try something new in a lesson, the girls now pre-test their ideas before putting them into practice, and they are confident they will add something of benefit to the sessions.

"I know more about coding now, and I enjoy the session and supporting people with Scratch and Hour of Code" – Milly, Code Club Young Ambassador.

"It inspired me to help other people to code. People are looking up to me, I like helping people, and that's a good feeling" – Lily, Code Club Young Ambassador.

What's next?

We wanted to know if the young ambassadors had thought about how they could use coding in their future career.

Rosie wants to be a computer scientist and knows that this club will really help her achieve this – "Code is part of my life. I like to code, I do it as a hobby and it's fun!"

Millie said that coding is fun and that she is going to carry on learning so she can continue to improve her skills, and Daya shared that she knows that her coding experience is something that she can take with her into other industries.

Having girls like the team at RGS The Grange leading the way for other young female coders, the future is in safe hands. We can't wait to hear what they plan to work on next – whatever it is, we know they have the determination and drive to succeed! **(HW)**

IT INSPIRED ME TO HELP OTHER PEOPLE TO CODE. PEOPLE ARE LOOKING UP TO ME, I LIKE HELPING PEOPLE... THAT'S A GOOD FEELING

The club's impact

With support and guidance, the girls work through Code Club projects. They are currently looking at Chatbot, an AI programmable robot that uses variables to store answers, written by the user, and respond with them in conversation – the fourth instalment of the Code Club Scratch Module 1 certificate.

The impact has been remarkable: girls who attend the club are making exceptional progress in the classroom – potentially thanks to a boost in their self-confidence in the subject. The girls demonstrate an increased self-belief to tinker when programming, with a reduced fear over making mistakes. Prior to this club, Module 1

available to the whole school community. The girls have had a hugely positive impact and have really developed their own interpersonal and leadership skills through delivering the project. We have noticed that the young leaders are starting to demonstrate initiative and improved organisational skills – this change is partly due to the fact that the room full of young coders are simply reliant on them. The other factor might be that, although I helped to set up the club, I endeavour not take over or undermine the girls when they are presenting, demonstrating, or organising the group. If, as educators, we intervene at every possible correctable moment, pupils are losing the ability and right to make mistakes – these are the real moments of learning that



THE POWER OF VIDEO GAMES

Tom Bromwich developed games-based lessons to develop the skills of his Year 3 students

Over the past few years, my school has been carrying out a project in Year 3 focused on investigating the positive effect video games can have in the classroom. As a gamer, I have always thought that games were a resource woefully undervalued by the majority of teachers, and that when used effectively can be far more immersive than video clips and pictures that we use every day.

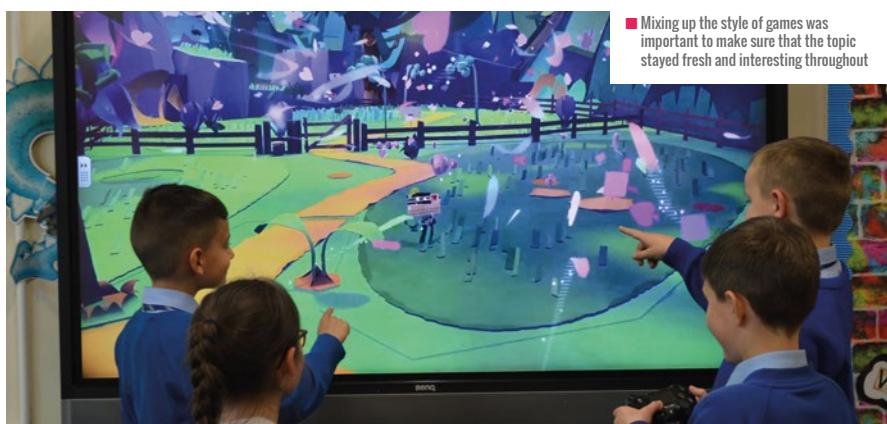
When planning the topic, it was always incredibly important to make sure that the games chosen would actually help the children with their learning, rather than being there simply as a novelty. This meant there was a bit of experimenting with a range of games in the planning stage, which I am not going to pretend was anything less than a huge amount of fun.

It was less about choosing well-known games that the children may have heard of,

but choosing games that would not only help develop their written work, but also their imagination, curiosity, and creativity. This meant picking games such as Tearaway, Journey, and Overcooked, as well as slightly better known titles such as The Legend of Zelda.

A source of inspiration

Incorporating video games into the classroom has led to some of the most imaginative, engrossing, and interesting pieces of writing that I have ever read from a Year 3 class. The fact that the children knew they were going to be showcasing their





TOM BROMWICH

Tom is a Year 3 teacher at Cooper and Jordan School in Aldridge. He is the ICT co-ordinator and has been teaching for nine years.

work through blogs, and also sharing it with the gaming community, has led to a huge improvement in their writing.

Carrying out this topic has also led to the children having some opportunities I don't think would have been possible without the focus on gaming. They have spoken to a huge range of people from the gaming community, all of whom have been amazingly helpful. From curators at the V&A Museum, to games developers from around the world, everyone that they contacted was fantastic, offering brilliant answers and advice to their questions. Emailing experts in a field, I think, is something that we try to do in every topic because there are so many brilliant people out there.

At the end of the topic, the children created persuasive letters to show others how useful the games have been. They have sent these letters to Blue Peter (and received badges in return!), our local MP, as well as others from the gaming community. The children have loved receiving responses.

Perhaps the best thing that has come from the topic, though, is the recent work the children have carried out with the charity, Special Effect. After speaking to their CEO via Skype, the children then had the chance to speak to people through the use of an AV1 robot. They also decided to try to raise money by creating an e-book containing all the work they carried out, which they are currently selling to parents to raise money for the charity. Everybody in the year group was amazed when we were the top-selling children's book on iTunes, even beating Harry Potter! (HW)

VIDEO GAMES IN THE CLASSROOM: THE STUDENTS' VIEWS

"Playing Overcooked was a good experience because we learnt more about teamwork, decision making, and friendship. It was really important that we communicated with each other and stopped each other getting too stressed. We loved playing the game, even though we don't want to become chefs when we are older!" - Nieve and Kaira

"In The Legend of Zelda, we felt amazed because it was like we were actually there in the game! We both felt like we were stood in a dark, dingy cavern, only to walk through and find columns of lava shooting up to the roof. It was terrifying, but really helped us with our story-telling!" - Alfie and Harry

"Tearway is a game made up of beautiful colours, set in a world of imagination where everything is made out of paper. The main character is a letter which needs to be delivered, and it really helped us develop our vocabulary as we described the vibrant setting!" - Lola and Lily

"I am really grateful that, through the gaming topic, I got to meet the students from Rocklands Special School. My attitude really changed after going to their school: I realised how lucky I am and not to take things for granted. Also, it was amazing to experience their facilities and to meet them. Rocklands is a fantastic school, full of compassionate, caring staff and their pupils are the nicest people I have ever met." - Kyle

"We made a book for charity and I really enjoyed doing it. Not only because I got to do lots of art, which I love, but also because we were raising money for Special Effect, who do amazing things! It was awesome seeing my own work get published." - Isla-Mae

"Journey is a very calm game, full of mystery. We really loved watching the character glide slowly across the colossal sand dunes. After the using the game, we packed our writing with questions and new vocabulary to add mystery. It was a superb game to use in class!" - Lottie and Daisy

"When we were asked to use the AV1 robot, I was a bit anxious, but when it arrived that turned to excitement. As soon as the eyes flickered on, we immediately got to know each other. The experience was incredible and if you get the chance to use one you should: it was an amazing way to make new friends." - Ylena

"Using video games in class has been awesome. It is the best topic that I have ever done in school because, when I first started playing, I realised there was so much more to them and it made lessons better. I think all schools should use games because they improved my work a lot!" - Louis



■ Games like Overcooked helped the children develop their teamwork, communication, and decision-making skills



REBECCA D'CRUZ TEACHER

A SPIRAL CURRICULUM

Teaching programming can seem like an uphill struggle. **Rebecca D'Cruz** advocates a spiral approach to engage pupils and build confidence

I was pretty lazy at school. I knew I could get by without too much effort, so I did. My reports were a litany of 'could do better'. That was until I just happened upon a friend with a Sinclair ZX Spectrum; my world changed!

I was brought up in Kenya and was a teenager in the 1980s. I had never even heard of personal computers, let alone come across one. I visited a friend who had just come back from England, spent the afternoon tinkering under his guidance, and was absolutely hooked. That was the beginning of my love of programming: hours were spent writing animations and games; an O-level followed and then a career.

BIBLIOGRAPHY

Bruner, J. S., 1960. *The Process of Education*. 1 ed. Cambridge, Massachusetts: Harvard University Press.
helloworld.cc/Bruner1960

Sentance, S. & Csizmadia, A., 2015. *Teachers' perspectives on successful strategies for teaching Computing in school*. IFIP TCS.
helloworld.cc/SentanceS2015

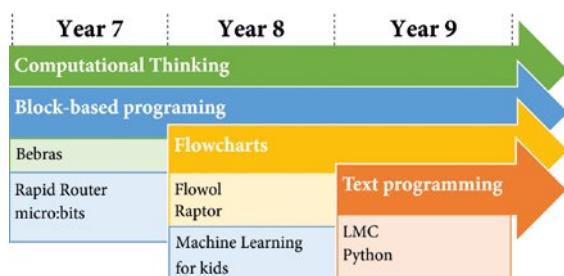
Shaffer, D., Doube, W., Tuovinen, J., 2003. *Applying cognitive load theory to computer science education*. International Journal of Computer and Information Engineering, p333-346.
helloworld.cc/Shaffer2003

Why is programming hard?

After starting teaching following a first career in software development, it took me a while to understand why pupils can find programming so hard. I have been teaching age groups from nine to adult for over ten years now and I think I finally have it: programming is about learning a language, yes, but it is much more about a way of thinking and complex concepts can't be taught in one hit. You must use a spiral approach.

This is hardly a new idea. Bruner said in 1960 that pupils need to learn fundamental principles, extend these through enquiry, make their own connections, and that the curriculum should revisit these ideas over time, adding complexity. Much research since has supported these findings, including recent work by Sue Sentance and Andrew Csizmadia.

In early programming tasks, pupils are being asked pretty simple questions: count to ten, work out the ticket price, etc. and, given a piece of paper, will happily give you the answers. Ask them to explain the method though, much less write that in an unfamiliar formal language, and the cracks will begin to appear. We need to be careful not to introduce complex algorithms or text-based programming too early. Shaffer points out that if we ask pupils to think computationally and express this using a formal language all at once, we will run into issues with cognitive load and many will decide it's not for them and give up. Pupils have often come to rely on learned responses, can be anxious about taking risks and, frankly, a bit reluctant to think for themselves. However, there is nothing much



■ A plan for progression in programming

more satisfying than a “Yes! I’ve got it” from a previously unenthusiastic or struggling pupil. So how do we help pupils in this process?

I am in my third year as Head of Computing, and have developed our scheme of work from scratch – no pun intended. In teaching programming using Python with my Year 9s last term – the first cohort I have taught right through from Year 7 – pupils were saying, “This is just like when we used flowcharts, Miss, isn’t it?”, “Isn’t this question just like this other one?”, all of which is music to my ears!

Same ideas, different contexts

I have threaded programming throughout my Computing curriculum and my approach is very much about questioning, encouraging, and extending. My hope is that by presenting computational thinking, programming, and debugging in lots of different contexts, pupils will see the patterns and find text-based programming much less intimidating. Here is a summary of the programming elements of my curriculum:

We take part in the Bebras Computational Thinking challenge from Year 7 upwards. These logical thinking puzzles are engaging, stimulating, and quite challenging – my thanks go to Chris Roffey for his phenomenal work on developing them! We then spend the spring term on our first programming unit. We cover some theory, looking at decomposition, algorithms outside the world of computers, and then use Code for Life’s Rapid Router for a first look at sequence, selection, and iteration.

In Year 8, we use Flowol as part of a Control Systems unit. Flowol is great for an introduction to flowcharting because, although the mimics might look a little dated, they are like physical computing without the batteries! It is much easier to persuade pupils to test their algorithms when they are turning on lights, controlling trains, etc. In the spring term, we use Dale Lane’s (IBM) excellent resources for machine learning. Naturally the pupils’ programs are simple, but the fact that they are interacting with a supercomputer is phenomenal!

Computer Science is optional in Year 9 and we have a large new intake, so we need to start with some programming. We use Rapid Router again, backtracking

RESOURCES

- Bebras Computational Thinking Challenge (Free) - helloworld.cc/bebras
- Rapid Router (Free, includes teaching resources) - helloworld.cc/rapidrouter
- micro-bit development environment (Free, micro-bits start at around £13 each) - helloworld.cc/microbit
- Flowol (£300 for a secondary site licence) - helloworld.cc/flowol
- Machine Learning for Kids (Free) - helloworld.cc/mlforkids
- Little Man Computer (RISC version, Free) - helloworld.cc/littleman
- Raptor (Free) - helloworld.cc/raptor
- Python 3 (Free) - helloworld.cc/python
- Codio (Prices vary) - helloworld.cc/codio
- HackerRank (Free in education) - helloworld.cc/hackerrank

a little for new pupils and allowing the old hands to pick up where they left off. Rapid Router is an excellent resource for teaching programming with debugging hints, a clear development pathway and sneaky solutions for teachers. I really want pupils to understand that Computer Science is a serious, academic subject, so part of our Computer Fundamentals unit (hardware, logic gates, binary, etc.) involves some assembly programming using Peter Higginson’s fabulous Little Man Computer. This allows pupils to start to get a grasp of the inner workings of the CPU and compels them to think hard to write even simple programs. Raptor is a tool for creating and running flowcharts that are much closer to the more formal flowcharts you see at GCSE-level with variables, functions, etc. All of this leads to a whole term spent on Python programming. I have written a course in Codio which reiterates and expands on the taught lesson and use HackerRank for topic exercises and tests, largely because it forces pupils to thoroughly test their code – something that can otherwise be difficult to achieve.

So, has it been successful? I think so. I have seen positive progression amongst pupils and my Year 9s have developed a better grasp of programming than previous years. The logical thinking and resilience needed to be a successful programmer have far-reaching benefits, and I hope this approach helps to make this an achievable goal for all. (HW)

REBECCA D’CRUZ

Rebecca (@RebeccaDCz) is Head of Computing at St Albans School and an Educational Consultant at Ocado Technologies. She is also an NCCE Facilitator.



■ Learning to write code can be a challenge for students, but there are evidence-based approaches to support them

HOW TO SUPPORT YOUR STUDENTS TO WRITE CODE

Will Grey shares some evidence-based approaches for teaching students to write code

For many children, writing code can be a daunting prospect. To help our students learn to write code, we can use a range of scaffolded pedagogies. Initially, these approaches take ownership of the code away from the students, giving them confidence to explore and experiment. Gradually, as the students gain confidence, we can reduce the amount of support until they are able to write their own programs independently.

In the previous issue of *Hello World* (issue 8, pages 58–59), I shared approaches to support pupils learning to read code. These included activities such as explaining, predicting, and tracing code, as well as live demonstrations with worked examples. Now, I'll share some evidence-based approaches to support pupils who are learning to write their own code.

Fixing broken code

Children can find and fix common syntax, runtime, and logical errors in a piece of code. These errors might include missing

brackets, missing speech marks, spelling mistakes, or missing variables declarations. The pupils can use the programming language's translator to help find the errors by making the fixes and then running the code to test that it works.

Pupils will need to be made aware of how the errors present themselves – this is unique to each translator and language and is not always obvious. For instance, in the default IDLE Python interpreter, it can be unclear where each error is located. Once pupils are familiar with a variety of common error messages, their causes, and how to fix them, they will be able to debug their own code with greater confidence and resilience. For extra challenge, debugging exercises can be carried out on paper, where there is no feedback from the translator.

Completion problems and pseudocode

We can give children partial solutions to complete. This has scope for a variety of levels of differentiation: as students progress, we can gradually leave out more

code. Initially, students might be given most of the code but with parts of some statements missing. Then, we might remove statements completely. Next, we could remove several lines or complete blocks of code. Finally, we could remove all code and only provide comments to the pupils. If these comments are detailed enough, then this is like converting from pseudocode into working code.

At some point, children will need to plan their own code with pseudocode. It is difficult and does require the application of high-level computational thinking, where pupils perform abstraction and code design. A good approach is to get pupils to write the comments for each line of their program before they start coding.

Parson's puzzles

Parson's puzzles are problems in which the code is presented to children all jumbled up. The aim of the activity is to put the instructions into the correct order so that the whole code performs a specified task.

FURTHER READING

In this issue:

Learn to program with *The Little Book of Algorithms* by William Lau, pages 70-71

Previously in Hello World:

Read before you write by Will Grey, issue 8, pages 58-59

Cognitive load theory in the computing classroom by Phil Bagge, issue 8, pages 34-35

Parson's in practice by James Robinson, issue 8, page 87

The trouble with pseudocode by John Woollard, issue 7, pages 40-41

Programming pedagogies: PRIMM by Oliver Quinlan, issue 5, page 25

Python Cheat Sheet

Printing strings	Output	Comments
print ("Hello World")	Hello World	#This is a comment.
print ("Hello", "World")	Hello World	# It helps you or others to
print ("Hello" + "World")	HelloWorld	# understand the code better.
		# It does not affect how the program works.

Variables

apples=10 # This is an integer
name="Frodo Baggins" # This is a string

Input variables

Name = input ("what is your name?")
Age=int(input("How old are you?"))

Print variables

apples=10
oranges=3
print (apples+oranges)
output: 13

If ... else ...

```
if (score < 12):  
    print "Grade U"  
elif (score >= 12 and score < 16):  
    print "Grade Pass"  
else:  
    print "Grade Distinction"
```

Operators

<	Less than
>	Greater than
>=	Greater than or equal to
<=	Less than or equal to
!=	Does not equal
==	Equals

For loops

```
for i in range(4):  
    print (i)
```

Output

0
1
2
3

While loops

```
x=0  
while x<4:  
    print (x)  
    x=x+1
```

A Python crib sheet that students can refer to when writing code

code and gain increasing ownership of the code as they progress.

If we consistently apply scaffolded approaches for reading and writing code in our lessons, we can make the experience of learning to code for children a lot less intimidating, and a good deal more enjoyable. [\(HW\)](#)

WE CANNOT EXPECT CHILDREN TO BE ABLE TO WRITE CODE BEFORE THEY HAVE LEARNED HOW TO READ CODE

syntax of constructs with some examples is a helpful aid. Pupils should be encouraged to organise any code that they write so that they can reuse the code in the future. The ability to recognise code that can be adapted to a new context is an important skill.

Pair programming and collaboration

Social constructivist approaches that use collaboration and pair programming, where two students work together on a single computer, can support students who are learning to write code. Within each pair, one student does the coding and the other observes, reviews and gives feedback. Students in each pair should regularly switch roles.

Planning lessons

We cannot expect children to be able to write code before they have learned how to read code. Through careful planning of lessons and sequences of lessons, we can slowly remove the scaffolding as pupils gain greater independence as they progress from reading and using code, through modifying code to ultimately writing their own code to solve problems.

The great news is that the PRIMM (predict, run, investigate, modify, and make) model offers a framework around which we can plan our lessons. For each lesson episode within this structure, we can pick and mix from our assortment of techniques where pupils read code before they write



WILL GREY

Will is Head of Computing at Comberton Village College, and coordinator of the CAS South Cambridgeshire Community of Practice. He has developed a comprehensive set of resources for A-level and GCSE computer science at helloworld.cc/grey.

SELFIE: UNLOCKING DIGITAL POTENTIAL IN SCHOOLS

How well is your school using digital technologies in teaching and learning?

Deirdre Hodson and **Panagiotis Kampylis** introduce SELFIE – a free online tool from the European Commission that can provide a rigorous answer

How are we using technology in teaching, learning, and assessment? The SELFIE tool, launched in October 2018, can be used free of charge by any primary, secondary, or further education school in any country to address this overarching question as to how they are using digital technologies for teaching and learning.

SELFIE, which is developed by the European Commission with a team of experts from across Europe, involves the whole school community – teachers, school leaders, and students – in reflecting on a series of questions and statements on technology use in six areas:

- Teaching and learning: pedagogical uses of technology



Each school receives a tailor-made, interactive report which provides both in-depth data and quick insights into strengths and weaknesses

- Assessment: using digital technology for providing feedback
- Student digital skills: including digital creativity, safe and responsible online behaviour
- Teacher professional development: opportunities for peer-to-peer learning, online training, and face-to-face courses
- Infrastructure: availability of devices, internet access, and technical support
- Leadership: whether the school has a digital strategy and/or involves teachers and students in discussions on technology use

The questions and statements are different for each user group and take around 15 minutes for students to complete, and 20–25 minutes for teachers and leaders. A simpler version of the questions for younger students, between nine and twelve years old, is also included. The anonymity of all responses is respected throughout the process.

Schools can customise SELFIE, adding from a list of optional questions and also creating questions, depending on school size, needs, and context. SELFIE is available in over 30 languages and students and teachers can select the language in which they would like to answer the questions.

A tailored report for every school

Once the questions have been answered by school leaders, teachers, and students, the school receives an automatically generated report with detailed insights on technology use for teaching and learning. The tailor-made report is interactive and data can be

displayed in different ways, for example by area (teaching and learning, infrastructure etc.) or by group of respondents.

These findings can help a school see what's working well and less well, and identify areas where action is needed. Are teachers happy with the training they are receiving? What do students think of the infrastructure? Does the school have a vision for how it wants to use technology and, if so, has this been communicated to staff and students?

The report can be used to start a discussion in the school, develop an improvement plan, and monitor progress over time. SELFIE can be taken on a regular basis and the results can be compared to the previous reports. The report is for the school only and can only be shared if the school chooses to do so itself. On completion of SELFIE, the school can receive a digital badge and participants can receive personalised certificates.

How to use SELFIE

Schools wishing to use the tool should nominate a teacher or small team of teachers to coordinate the exercise at the school. They then register the school on the SELFIE platform helloworld.cc/Selfie – this takes just a few minutes. The school coordinators decide on who will answer the questions with guidance from the tool on how many students and teachers should take part, and how to select them and when and for how long the exercise will run in the school. The response period should run for minimum one week and maximum three. The school coordinators then set up

#SELFIE_EU



■ SELFIE evaluates schools' use of technology in relation to six areas

and customise the SELFIE questions for the school. There are a set of core questions common to every school. After this, schools can add from a list of optional statements and questions or add their own. Setting up SELFIE is straightforward and a step-by-step guide is provided for the school coordinators. Once the questions are set up, links are generated which can be sent to school leaders, teachers, and students, who then reply to the questions anonymously.

A cross-European project

SELFIE (which stands for Self-reflection on Effective Learning by Fostering Innovation through Educational technology) is a European project, developed by the European Commission's digital education team in cooperation with students, teachers, school leaders, policy makers, and researchers from across Europe, and provided to schools anywhere in the world, free of charge. The tool was developed to help schools across Europe who face the challenge of how to embed technologies for teaching and learning in a purposeful way.

SELFIE helps schools take the focus away from infrastructure and devices to considerations of teaching and learning, assessment, teacher professional development, and leadership. It allows voices to be heard – including students'.

SELFIE is not a survey/benchmarking tool. It doesn't rank schools, but allows them to see where they are in terms of using digital technologies for teaching and learning.

FEEDBACK ON SELFIE

"We have created a vision for digital learning in our school by talking to each other. SELFIE made consensus easier because it prompted us to have detailed discussions about digital learning. Our approach to digital learning is that it becomes almost an invisible aspect of school, that technology is there when the children need it the most and when it is going to improve the learning outcomes of each individual child. An added bonus was that it empowered us to discuss and plan how computational thinking and coding can be developed on a continuum throughout the school."

- **Seán Gallagher, School Principal, Ireland**

"We are using digital technologies for many years, but we have not had a proper evaluation of how we are doing, our weaknesses, and what the students think about technology use at our school. Through SELFIE, we know that we can improve things like data protection, safer use of the internet, and social networks. We also need to boost the confidence of teachers with more training and support in using technology."

- **Carmen Lazaro, School Principal, Spain**

"I think SELFIE is a very important tool as it invites the school communities to start a discussion on how they can improve the use of technologies. In this process it is very important to hear also the viewpoints of students. Because someone can think that they use technologies effectively in a school, but maybe students have a totally different opinion. The fact that SELFIE invites schools to start a dialogue with students is fantastic."

- **Maeve Galvin, Student, 16 years old, Ireland**

It is a great opportunity for us, and it is very important that our viewpoints are valued. We are the key people for building, step-by-step, a better use of technologies for learning.

- **Student, Spain**

Technology is present everywhere in our lives so, for us, it is natural to have technology in our classroom.

- **Student, Spain**

Next steps for SELFIE

A first conference on the tool took place in Madrid in April 2019. It brought together teachers, school leaders, students, policy makers, and researchers from 35 countries. The aim was to look at how the tool could be developed over coming years, and what support is needed for schools using SELFIE.

Future plans include offering online training to schools, and guidance on how to create a digital action plan, while the SELFIE team continue to make improvements to the tool.

Since launch, 330,000 students, teachers, and school leaders across the globe have already used the tool. Feedback has been excellent, with nine out of ten staff saying they would recommend it to a colleague.

Find out more

Details of how to use the SELFIE tool can be found at helloworld.cc/Selfie, and you can contact the SELFIE team at JRC-EAC-SELFIE-TOOL@ec.europa.eu (HW)



DEIRDRE HODSON

Deirdre Hodson (@deirdrehodson) works on the innovation in education team of the European Commission. She is also a student on the MSc programme in digital education at the University of Edinburgh.

PANAGIOTIS KAMPYLIS

Panagiotis Kampylis (@pankampylis) is a researcher and project manager at the Joint Research Centre, the European Commission's science and knowledge service, working on projects on learning and skills in the digital age.

A MOVEMENT TO BRING COMPUTER SCIENCE TO ALL STUDENTS

Leigh Ann DeLyser outlines the systems CSforALL is putting in place to ensure computer science is an integral part of the educational experience of all students in the US, from kindergarten through to high school graduation

Computer science education has long been a teacher-led movement. Since the 1980s, creative and enthusiastic teachers have seen computers and computer science as a way to teach problem-solving, excite students, and empower kids to be creators.

As computer science transforms from a speciality skill to a critical literacy in this digital world, it seems everyone else has caught on to what those early teachers knew: this is something kids can do. In today's world, computers are no longer something on the side. They do not exist only in isolated rooms or specialised areas – but instead pervade the way we work, live, communicate, and play. In the US, as in many countries globally, our education systems are embarking on a mission to make them a critical part of how we learn.

In this next phase, we will need those early teacher leaders, but cannot rely on them alone. We need our systems to help build pipelines for students and teachers to

develop not just introductory knowledge, but rather mastery in computer science.

Local educational systems

Great curricula do not appear in a vacuum. Educators, both in and out of school, rely on local systems to provide time, space, and students for their courses. At CSforALL, we believe that critical systems change is needed to allow teachers to focus on instruction.

I was a high school mathematics and computer science teacher for ten years before shifting to teaching at the college level. During that time, I spent eight years at one school, building a CS programme, a 40-member robotics team, and an AP (Advanced Placement) Computer Science class that had significant enrolment every year. When I was offered a position at a university, I worked with the district to find my replacement, and even had overlap time (unheard of in education). Sadly, just a year after I left, former students reached out indicating that the school was closing down the CS classes. I never realised how much my own personal efforts were contributing to CS education in that school.

Now, at CSforALL, it is about more than just supporting new teachers – it is about creating the infrastructure to truly serve ALL students equitably. If we believe CS education is a kindergarten through high school subject, than administration needs to be engaged in the support for and critical

planning of elementary and secondary pathways. Individual teacher leaders cannot do this alone.

This is why we work with local school districts through our SCRIPT programme (Strategic CSforALL Resource & Implementation Planning Tool – csforall.org/script) – a framework to guide teams of district administrators, school leaders, and educators through a series of collaborative visioning, self-assessment, and goal-setting exercises to create or expand upon a computer science education implementation plan for their students. And we engage with state leaders through the Expanding Computing Education Pathways project (ecepalliance.org) to think systematically about how to support teachers and students, and to improve and broaden participation in computing education.

Community systems: creating learning spaces

CSforALL believes that the ALL doesn't just represent all students, but their educators – both in the classroom and outside of school – and communities too. Education is inherently a local endeavour, and we strive to reach every zip code in America, and every region on the planet.

Our communities are the vital ingredients to move beyond isolated experiences for students and teachers, to sustained learning trajectories for both. In NYC, one

LEIGH ANN DELYSER

Leigh Ann (@lsudol) has spent her career building the K-12 computer science field. As a managing partner of CSforALL (csforall.org), she oversees programmes and strategic planning and supervises research to build support for high-quality CS education at all levels.



School districts joined CSforALL in January 2019 to create plans for CS education

of the very first things we did was start a meetup group for computer science education. That meetup group now has over 2000 members, and is now the NYC chapter of the Computer Science Teachers Association. The group counts not only classroom teachers as members, it is also a collection of informal educators, museum staff, and even professionals who want to support kids learning of CS. Together, the diverse group explores topics and builds knowledge and capacity in the community for high-quality computer science education.

We believe so strongly in the local model that we are launching a new programme, EcoSystemsforCS (csforall.org/ecosystems), meant to help catalyse local communities, and bring all the stakeholders to the table to support computer science education for all students and educators.

Teacher training and development

A third key component of CSforALL is teacher training and professional development. How we prepare teachers for the classrooms of today needs to include computer science as a key fundamental skill. Teacher training programmes in colleges and universities are a key component of

the systems that will ensure high-quality computer science education for all.

In 2018, in collaboration with other computer science education leaders, I co-authored a report entitled, 'Priming the Computer Science Teacher Pump' (helloworld.cc/cspump). This made specific

system where the best pedagogy does not just sit in academic papers, but makes its way into classrooms. The RPPforCS (csforall.org/rppforcs) project is hoping to do that – building a community of projects studying computer science education, funded by the National Science Foundation.

CSFORALL IS WORKING ON THE SYSTEMS THAT NEED TO CHANGE TO SUPPORT YOUR LEADERSHIP

recommendations for engaging teacher preparation programmes in computing education, with a range of focus areas varying from equipping all teachers with the necessary knowledge through to defining opportunities for interdisciplinary coursework for interested computer science education PhD candidates. The recommendations made in the report are meant to begin the systems change progress in higher education. It is crucial to support not just the orientation of teachers to CS education, but the development of master teachers.

Additionally, CSforALL wants to connect educators with researchers and create a

How to get involved

Systems change is the keystone of changing our education system so it can bring rigorous, equitable, and sustainable computer science education to all. And you can get involved! Teachers or supporters in the US can join the Computer Science Teachers Association (csteachers.org). Your district or out-of-school-time programme can join CSforALL (csforall.org), and you can work with your partners to make a commitment for the CSforALL Summit this October. CSforALL is working on the systems that need to change to support your leadership and we have resources to support that work. (HW)

MAKE AN APP IN A WEEK

Finding work experience in tech can be tricky, so **Martin Sexton** and **Katie Sanders-Pope** decided to set a group of their students a challenge

Meaningful work experience for students interested in programming can be incredibly difficult to source. So, last year, our school invited a small group of Computer Science and Maths students to join our Director of Sixth Form for a briefing on the first morning of their work experience week. The students were posed a challenge: to create an app through which the academy can share information directly to students' phones.

This project would help students with an interest in working in the technology industry to see how developing a piece of software for an end user works in reality. The students would be able to carry out the full software development life cycle and face real issues when trying to meet the requirements of the client.

A cross-platform framework

One of the biggest issues with developing apps is that iOS and Android use different languages. The students researched this problem and found a few options.

The first of these was Microsoft's Xamarin. This is a cross-platform way to create apps, but it had a few issues that made it unsuitable. For example, the students would have to be coding in either C# or Ruby, which none of them had used. This would have meant most of their time would be taken up learning the language and their deadline would not be met. There is also limited support for features in the free version.

The students therefore decided to use PhoneGap. PhoneGap is a free and

A STUDENT'S VIEW BY FINLAY BOYLE

FINLAY BOYLE IS A SIXTH-FORM STUDENT AT MILDENHALL COLLEGE ACADEMY AND WAS THE LEAD DEVELOPER OF HIS SCHOOL'S INFORMATION-SHARING APP

Creating an app was not something that any of us had done before, let alone in a week. We wanted the app to be something that would appeal to students, was aesthetically pleasing, and simple to use. As we did not have much time to attempt to build a prototype, we had to prioritise and plan effectively.

We spent approximately two hours on the first day planning and researching what we were going to do. We created a mind-map to visualise the different components of the app, as we realised that we would need to create two separate applications: one for mobile devices, and a web application for managing the content.

Once we were satisfied that we had a basic understanding of our objectives for the app, we began to implement it. Since we had done minimal planning, we decided to continually review our work as we created it to ensure that we would be able to integrate each section together. As a team, we decided to use GitHub to manage our code so that we could each produce different bits of the project at the same time.

In the team, Callum and Dan focused on researching and setting up PhoneGap, which we would use for the main app for students and staff. Freddie and Jake began to think about designs for the app and the admin web app. Whilst they did this, I began preparing the back-end. Since the main app and admin web app



would need to simultaneously send and receive data, we decided that a secure API would be the best method for the main app to fetch data from the server, whilst the admin app would upload content via PHP directly on the web server. To plan this, I created a rough overview of each of the endpoints we would need and designed the login flow so that we could secure it.

Due to the time constraints, we decided to repurpose some of the code from my A-level coursework to reduce the time spent on layout and coding of the admin pages. This allowed us to move on to creating new features for the app instead of spending time rewriting the base for our code.

We Have a Course for You



- Top 10% nationally for progress
- Level 2 & 3 courses available
- 38 different courses to choose from
- Sixth Form centre

■ Students from Mildenhall College Academy with the code behind their app, which is now used by the school to share information with students



open-source cross-platform framework for building apps using web technologies: HTML, CSS, and JavaScript. PhoneGap allows the source code for the app you are building to be hosted on your computer and then run via the PhoneGap app, allowing for quick testing. This was essential for the prototyping and allowed the team to swiftly alter the design and understand how the app would look on a phone or tablet. Using HTML and CSS made designing straightforward: it was a simple process for the students to turn their ideas into a functional user interface.

The app in action

One essential component the students had to learn about in order to use it on both the web app and main app was AJAX. This was important because the students did not want the app to freeze every time it was waiting on the server – students wouldn't use the app if they had to wait upwards of ten seconds to use it! AJAX allowed the team to overcome this by using asynchronous calls, which then used callback functions to execute once the requests had been completed. This allowed data to be dynamically loaded whilst the user was browsing the app, and then automatically update the display with the new data without disrupting the user experience. This was also used on the

PHONEGAP

PhoneGap is a platform which enables developers to use existing web development skills to create apps for iOS, Windows, and Android. It uses a combination of HTML, CSS, and JavaScript and distributes to each of the three main vendor app stores.

Full details of how to use PhoneGap can be found at helloworld.cc/phonegap.

admin site to allow them to search, update, and delete user data.

The academy was thrilled with the app, which incorporated the functionality requested. Now, staff are able to share daily student announcements specific to student groups, and maintain a web links page. The targeted student announcements allowed teachers to send daily study tips to Year 13 during the Easter break, encouraging students to use their time wisely!

The students are keen to take on the app development and increase its functionality. One idea is to integrate individual student timetables into the app linked to the student signed in. Staff are extremely proud of the determination shown by the students in creating the app, and are delighted to share their achievements! (HW)



MARTIN SEXTON

Martin Sexton (@MCACompSci) is the Lead for Computer Science and STEM Coordinator at Mildenhall College Academy in Suffolk, England. He is also the CAS Community leader for Mildenhall and a Barefoot volunteer.



KATIE SANDERS-POPE

Katie Sanders-Pope is the Director of Sixth Form at Mildenhall College Academy.

LEARN TO PROGRAM WITH THE LITTLE BOOK OF ALGORITHMS

William Lau introduces a free resource to help your students develop fluency in their programming

HOW TO USE THE BOOK

In lessons, I have found it useful to go through one or two algorithms at the front of the book with my students, talking through the programs line by line. I will then ask students to go to the associated challenge at the back of the book. Once they have drafted the program by hand, they can then check that the program runs using an IDE. Alternatively, students may choose to work through the book independently in class or at home. Like all good exercise books, the answers are provided in the back.

Many teachers and students of computer science will have asked themselves the question, 'Why is learning programming so difficult?'

I still find aspects of programming challenging and I am continually learning. After teaching programming for the past seven years, I noticed that only a minority of my students felt confident enough to program independently after two years of instruction. Upon realising this, I knew I had to change my pedagogy.

I had read Scott Portnoff's article calling to reframe the introductory computer programming course as a language course, and I witnessed a pattern where my bilingual students generally do better at programming than students who only speak

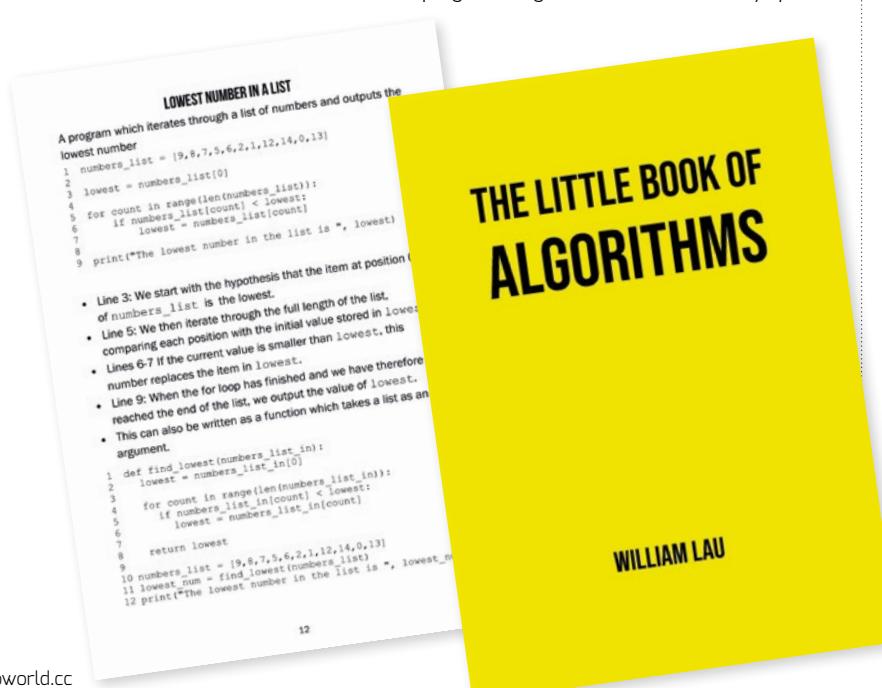
one language at home. Portnoff's article in issue 7 of *Hello World* reminded me of his work, and convinced me that students do need to memorise the syntax of some key programming constructs, e.g. if statements, while loops, and for loops.

Learning some basic structures by heart will decrease cognitive load and enable students to practise programming more fluently. Two further questions then came to mind, 'which programs do students need to learn by rote?' and 'which programs do all students need exposure to?' Portnoff's work was my starting point for developing *The Little Book of Algorithms*.

A canon for programmers

As a student of hip-hop culture, I came across Joseph Schloss's book *Foundation*, where he writes about a musical canon that exists for dancers. To add to this theory, Jonathan Sacks argues that a literary canon is essential to a culture. In linking these three ideas together, I thought about creating a canon for programmers. Perhaps there is a set of programs which represent algorithms that every computer science student should familiarise themselves with?

I started to compile a list of programs, based on my experience as a teacher and examiner. Some of these would be short five-line programs which would be worth repeating until they are committed to memory. However, learning some of the longer programs by heart would be both challenging and futile. Programmers in the real world do not remember entire



programs, and they rely on previous solutions created by themselves and others. The reality is that programmers also rely on websites such as Stack Overflow and GitHub. Therefore, I do not expect my own students to remember long programs and, to help them develop fluency with tackling longer algorithms, I added some challenges based on this canon of programs.

The challenges should help learners understand the programs by applying them. In the first version of the book, I have provided all the programs at the front of

frustrate you at times; learning programming is similar to learning a musical instrument. Both involve practice and making lots of mistakes. Both also require perseverance to develop fluency. Keep going!

Download *The Little Book of Algorithms* for free

I have followed the publishing philosophy of Al Sweigart: "I write books to teach beginners to code. I put them online for free, because programming is too valuable and needs to be accessible to all...". Therefore, free PDF

FURTHER READING

- Portnoff, Scott R. 'The introductory computer programming course is first and foremost a language course.' *Inroads* 9 (2018): p34-52 (helloworld.cc/portnoff2018).
- Sacks, J. *The Home We Build Together, Recreating Society*. Bloomsbury Academic (2009) (helloworld.cc/sacks2009).
- Schloss, Joseph G. *Foundation, B-boys, B-girls, and Hip-hop Culture in New York*. Oxford University Press, USA (2009) (helloworld.cc/schloss2009).

" MY PLAN IS FOR FUTURE EDITIONS OF THE BOOK TO BE FORMATTED IN A MANNER SIMILAR TO A MATHS TEXTBOOK

the book and all the challenges at the back. However, my plan is for future editions of the book to be formatted in a manner similar to a maths textbook, where students are given examples and then given a series of exercises to try immediately afterwards.

The Little Book of Algorithms will hopefully help students and teachers to practise and develop fluency in their programming. The book is not a substitute for other teaching resources, and I still think a teacher is a student's most valuable resource. The book should be another resource that helps students on this challenging journey towards becoming a competent programmer. My advice to learners is that programming will

copies of the book can be downloaded at helloworld.cc/lboa. Or, if you wish to buy a printed copy, these are available for £5 on Amazon (helloworld.cc/buy_lboa).

The work is released under a Creative Commons licence, meaning educators are free to use and adapt as they like. Alan Phoenix Holland, Head of Computer Science and iMedia at Sawston Village College, is rewriting the book with programs presented in C#. I would love to hear from further potential collaborators, or anybody with comments – you can get in touch with me at William.lau@computingatschool.org.uk. And please let *Hello World* know what you think too @HelloWorld_Edu! (HW)



WILLIAM LAU

William (@MrLauLearning) leads Key Stage 4 Computing at Central Foundation Boys' School. He is a CAS Master Teacher and the author of *Teaching Computing in Secondary Schools* and *The Little Book of Algorithms*.

AGE RANGE

4-8

LESSON TYPE

Unplugged

REQUIREMENTS

- Large piece of paper/card
- Paints of different colours
- Paintbrushes
- Stopwatches

HEALTH & SAFETY

- Check for allergies
- Ensure students are free to move, no trip hazards
- Painting overalls

OBJECTIVES

✓ What do art and algorithms have in common?

✓ How fast is a computer?

✓ What are humans good at?

PRIMARY ART AND ALGORITHMS

Gemma Wilson shares a creative lesson that introduces the concept of algorithms

The aim of my first year teaching in a primary setting has been to develop a bespoke computing curriculum for our students, focusing on a balance of unplugged and plugged learning to develop excellent computational thinking skills. As part of this, students in Years 1 and 2 (ages 5–7) have been learning that algorithms are a set of specific steps to solve a problem. We have discussed how search engines like Google use search and sort algorithms to make it easier for us to find information.

The students were then instructed to 'be the computer' and that each could choose their own motif, such as a dot, a circle around a dot, the first letter of their name, or drawing squares in two different colours, and triangles inside each square.

The children were to start drawing when I said 'RUN' to signify the computer turning on. When I said 'END' the pupils stopped drawing and moved on one step to their right. The computer also ran fast and slow so the children could observe how the art changed.

The pupils got stuck in to the task, filling up large pieces of cardboard with their colourful algorithmic artwork.



■ The students each have their own motif

ASSESSMENT

A class discussion or written responses:

- ✓ Review art produced
- ✓ Discuss the patterns created
- ✓ How did it feel to be part of the process?
- ✓ What do humans bring to art that is different to computers?



GEMMA WILSON

Gemma is in her first year teaching primary-level computer science at The Mead School in Tunbridge Wells. She has over ten years' experience in secondary school science and leadership roles.

things that humans bring to this task – such as creativity, emotion, and interpretation.

They also discussed the fact that a computer can never experience real feelings – it can be programmed to respond to feelings or analyse artwork, but will never experience art the same way as humans do. **(HW)**

ACTIVITY 1: WHAT IS AN ALGORITHM?

5 minutes

Ask students to describe the term algorithm using everyday examples, if you have introduced the term before.

Or

- Start by talking with students about what algorithms are. Explain to them they are now the computer. Each one will get their own set of instructions to follow.
- Examples of instructions: Draw a dot. Draw a circle around a dot. Draw a line connecting all red objects. Draw the first letter of your name. Draw squares in two different colours. Draw triangles inside each square.



FURTHER READING

✓ This lesson plan was based on an activity from Hello Ruby: helloworld.cc/helloruby

ACTIVITY 2: CREATING ALGORITHMIC ART 20 minutes

- Ask students to choose a paintbrush and colour, and spread them evenly around the paper/card.
- When you say RUN, the computer turns on and each child starts drawing. When you say END, each child stops and moves one stop to their right. I gave the students a specific time limit, e.g. 1 minute.
- Run the 'computer' fast and slow. How does the art change? Having music play in the background might also help to focus the children on the activity – using different tempos to support the speed of the 'computer'.
- Ideally, you may let the artwork dry for a day and make the second layer. However,

we continued to work on the piece after a discussion about new algorithms the computer could use. Discuss what makes an algorithm easy to follow.

- Allow students to run the computer one at a time. Did the artwork look different to how they expected? Why?

ACTIVITY 3: A DISCUSSION OF ART MADE BY ALGORITHMS 10 minutes

Discuss the following points in groups or as a class.

- How long would it take for a computer to generate digitally similar piece on an art work given the exact same instructions? Not very long – it could probably do it in nanoseconds. A computer is very good at following instructions, and very fast. It would always win against humans.
- What do humans bring? Creativity. Emotion. Interpretation.
- Ask the students how the artwork makes them feel? Acknowledge the feelings. Explain that a computer can never experience real feelings – it can be programmed to respond to feelings or to analyse artwork from a database, but it will never experience art the same way humans.

DIFFERENTIATION

Support: To support students with special needs, you could differentiate by shape allocated to each student, or use photocopies of pictures to stick down.

Stretch and challenge: Students could use complex shapes, letter formations, and add variation. Allow students to design their own algorithms. Allow students to run the 'computer' independently.

AGE RANGE

7-10

LESSON TYPE

Unplugged

REQUIREMENTS

Per team (two to three students)

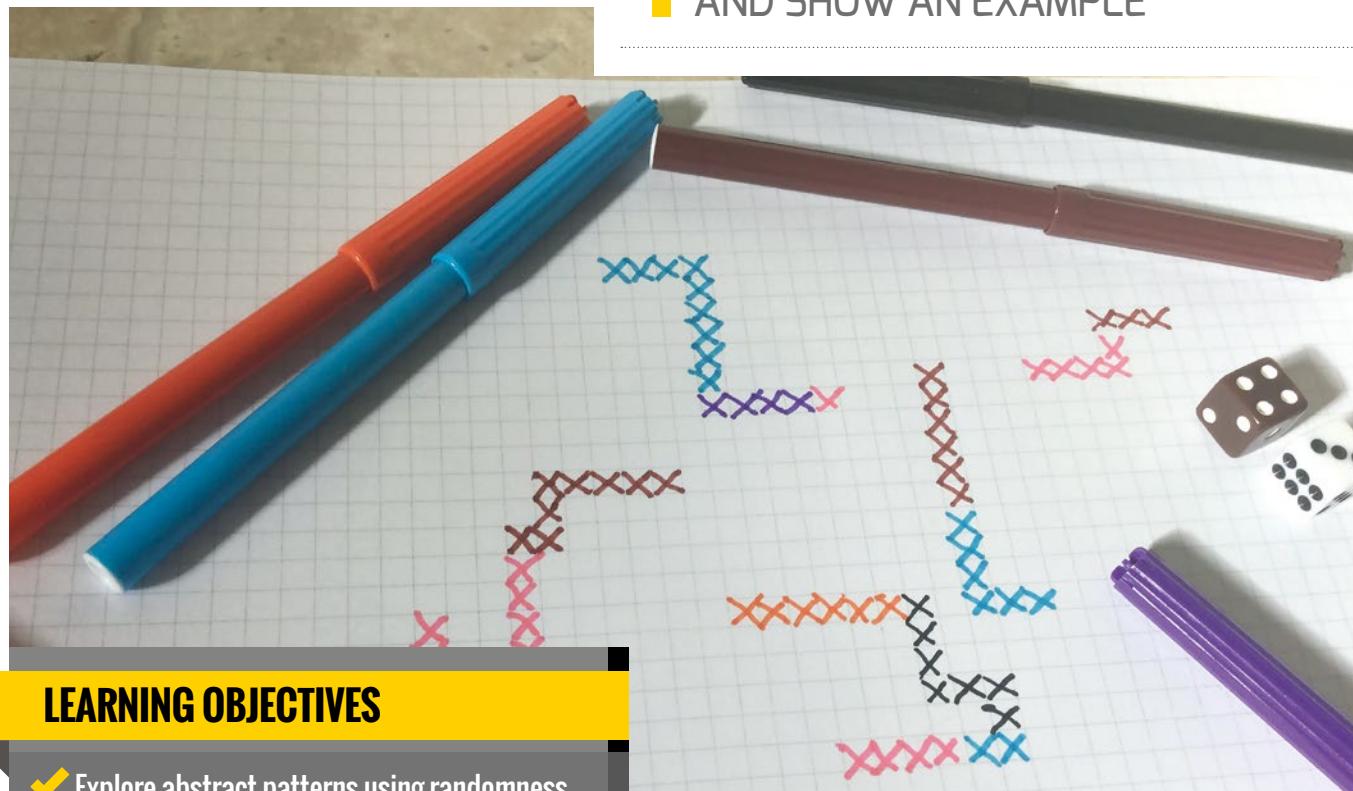
- Squared paper
- Six coloured pens or pencils
- One or two dice

THOMAS'S TANGLES

Algorithmic thinking doesn't just happen when coding. **Scott Turner** and his son **Thomas** have developed a game that uses an algorithm to produce drawings

The aim of this game is to use an algorithm to generate an image and see the main algorithmic constructs in action.

A FEW MINUTES MAY BE NEEDED TO GO THROUGH THE ALGORITHM AND SHOW AN EXAMPLE



LEARNING OBJECTIVES

- ✓ Explore abstract patterns using randomness within an algorithm.

After this activity, students should be able to:

- ✓ Use an algorithm that uses randomness
- ✓ Produce a picture by using an algorithm

Students may need some guidance on interpreting the 'check for blocks' part of the algorithm/subroutine: essentially it is just either if the move is off the page or the blocks have already been filled in, then find a new starting point

ACTIVITY 1: PRODUCE YOUR TANGLES 30 minutes

Get your students into pairs, each with a copy of the algorithm, a sheet of squared paper, pens, and dice. A few minutes may be needed to go through the algorithm and show an example.

Working in pairs, ask the children to take turns with the two roles.

- Person A: Rolls the dice and reads out the instructions – using the algorithm.
- Person B: Is the ‘robot’ carrying out the instructions.
- When the starting or central square is blocked and a new central square is needed, the roles of A and B swap (so A is the ‘robot’, and B rolls the dice and reads out the instruction).
- The roles keep swapping.
- Pedagogically, the approach is inspired by pair programming and this activity has elements of the Run and Investigation parts of PRIMM.

Algorithm

Start from a random square – call it the centre square

Repeat until end of game

If die roll = 1

Roll die for number of moves
Check for blocks
If not blocked then
move die roll number of steps up
the page

If die roll = 2

Roll die for number of moves
Check for blocks
If not blocked then
move die roll number of steps down
the page

If die roll = 3

Roll die for number of moves
Check for blocks
If not blocked then
move die roll number of steps to
the left

If die roll = 4

Roll die for number of moves
Check for blocks
If not blocked then
move die roll number of steps to
the right

If die roll = 5

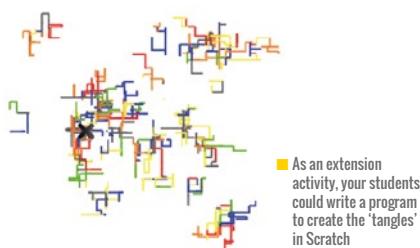
Roll die
If die = 1 change colour to Red
If die = 2 change colour to Blue
If die = 3 change colour to Black
If die = 4 change colour to Red
If die = 5 change colour to Orange
If die = 6 change colour to Yellow

If die roll = 6

Return to current centre square

Check for blocks:

If number of free blocks in the direction < number of moves, choose a new centre square.



As an extension activity, your students could write a program to create the ‘tangles’ in Scratch



THOMAS TURNER AND SCOTT TURNER

Thomas Turner has an interest in both playing and creating games. He is 14 years old.

Scott Turner (@scottturneruon) is Principal Lecturer in Computing at the University of Northampton. He is also a Code Club leader and volunteer.

ASSESSMENT

Some suggestions for questions:

- ✓ What part of this algorithm uses selection?
- ✓ How do we know what will be repeated?
- ✓ How would you modify to improve the algorithm?
- ✓ How could we alter the algorithm to use 12 colours instead of 6?

DIFFERENTIATION

For a follow-on activity, can you build part of this in Scratch? As a suggestion, build it so that only a single ‘spiral’ of one colour is used.

FURTHER READING

- ✓ A simple Scratch version of the game: helloworld.cc/tangles
- ✓ Scott Turner and Katharine Childs have written a chapter on artists in the book *Teaching Computing Unplugged in Primary Schools*: helloworld.cc/primaryup

ACTIVITY 2: REVIEW, MODIFY AND MAKE 30 minutes

- Share examples of the children’s work around the class.
- What changes would you make to the algorithm?
- Make the changes to the algorithm (and, if there is time, repeat on a new piece of paper using the modified algorithm).

AGE RANGE

10-11

RESOURCE TYPE

Music & technology lesson

REQUIREMENTS

- Sonic Pi - either on Raspberry Pis, or free to use on Mac or PC from sonic-pi.net

BEETHOVEN MEETS SONIC PI

Dominick McKoy collaborated with a music teacher to bring live coding to a lesson on the composer

Musical instruments are one of the oldest technologies in history. But, why isn't technology and music explored more as a collaborative experience in our schools, and how would a lesson plan even work? I have had successes with integrating technology into the music classroom by using Sonic Pi.

Step 1: Assuring an engaging experience without piling on work
New technology can be overwhelming, and part of showcasing all of its benefits is demystifying its complexity. I first spoke to Emily Kneur, our music teacher at the time, about how we could schedule a demo lesson together using the Raspberry Pi to help students experience what it

might have been like to compose music as Ludwig Beethoven did – without hearing it! She was nevertheless intrigued, especially when I offered to construct and facilitate the lesson.

Step 2: Getting all the materials (the resourceful way)

Funding was an issue, so I did what most teachers have to do: scrounge, pillage, salvage, borrow, and buy. For this demo I chose two Raspberry Pi 3s from MCM Electronics, which were delivered within a week. Then came the scrounging:

- microSD card: Luckily, I had old Android phones that had 8GB microSD cards in them, SCORE! But, if you don't have spare SD cards, here are a few things you can do: you can ask around, check in with your technology department, you can purchase an 8GB microSD card from Amazon for \$10, or you can purchase cards that come with Raspbian pre-installed. The Raspbian image is available on the Raspberry Pi Foundation website, as well as detailed instructions on how prepare the microSD card.

- Keyboard and mice: I borrowed a couple of keyboards and mice from our desktop computers in our main office. The monitors were tricky since they required VGA cables and the Raspberry Pi required

an HDMI cable. Fortunately, with some Google trickery and another eight bucks on Amazon, I was able to make it work. I used my Kindle charger to power the board.

Step 3: Make a memorable experience

Emily and I discussed the key points of the lesson – learning about the famous composer Beethoven's history, exhibiting some of his music, and exposing students to the experience of composing music as Beethoven. I set up two Raspberry Pi stations on opposite sides of the room, each with chairs and a stack of note cards. I hadn't had the pleasure of teaching this group of students yet, but they were excited all the same about the lesson since they were told it would involve some form of technology.

As the students entered the room, they were welcomed by the sound of Beethoven's Moonlight Sonata and me, their guest teacher, in a very odd-looking disguise including a wig! I let them all come into the room, but I did not let them sit down – instead I barked what seemed like unreasonable orders, bumped rudely into students, greeted them in what seemed like a sour tone, and as they would respond, I would walk away abruptly as if nothing was said. The teacher was already prepared and started prompting the students to try to explain my arbitrarily rude behaviour, while I began to be more

IT WAS TIME TO COMPOSE MUSIC LIKE BEETHOVEN DID IN HIS LATER LIFE

THE CHALLENGE

- ✓ Describe the story of Ludwig van Beethoven
- ✓ Use Sonic Pi to compose music



DOMINICK MCKOY

Dominick is a writer, software engineer at AndPlus, and Raspberry Pi Certified Educator who specialises in cross-curricular teaching and learning using technology. He has previously taught ecology and robotics to elementary school students, mathematics, and science courses at Rutgers University and worked as an Educational Technology Specialist.

deliberate about trying to read lips and telling them to speak slowly and clearly (and of course trying my hardest not to sway to the music).

It was an amazing 'a-ha' moment for the students who finally realised that I was depicting the behaviour of a particular deaf man, with a reputation as a difficult character, as they also realised that the song they were listening to the whole time was of course composed by the man I was trying to portray. We took a moment to laugh at my ridiculous disguise, then discussed a brief history of Ludwig van Beethoven.

We spent the first 15 minutes exploring the history of Beethoven and his music, entertaining questions, and imprinting on the students the magnitude of Beethoven's accomplishments. Then, I told them it was time to compose music like Beethoven did in his later life: without the ability to hear it. The Raspberry Pi was the perfect choice since it has the Sonic Pi program on its stock Raspbian image, and allowed the students to write code that programs the music. I split them into groups and told them we would



have a competition to see whose composition would make the top ten list on today's radio. I showed them the basics of how to code their notes and then played a coded version of the Super Mario Bros theme song, to further their engagement and anticipation.

Then they were given about 15 minutes to compose their masterpiece, each member of the group responsible for at least five separate notes. I was intrigued by the type of questions the students had immediately – asking about chords, crescendos, rests, loops, and samples – and how intricate each of their five lines of code was becoming. They were really getting into it, with no clue of what their composition sounded like. When the time was up, each judge (which were me, the student, and their teacher) was given a pair of headphones to listen to each song. The students got a kick out of seeing our expressions as we grimaced through the incoherent symphonies of our fifth grade music class. We spent a few minutes discussing what it was like to compose music without hearing it (and of course,

how it feels to still not be able to hear it). Then I played Beethoven's 5th Symphony, emphasising that at the time of composing the masterpiece, Beethoven was struggling with the loss of his hearing. The room fell silent in reverie as the song played through to the sound of the bell.

Step 4: Teach to fish and follow up

The day after the lesson, I sent all the students both compositions in an MP3 format and showed them where they could explore more. Emily enjoyed it so much, we did it again for her next fifth grade class, who received it with a similarly intrigued response. I even had parents approach me about how they could purchase a Raspberry Pi for their children.

All in all, this case study was immensely successful. The Raspberry Pi provided an excellent way to enhance a traditional classroom, using technology as a tool for exploration and engagement. It was exciting to create an amazing cross-curricular learning experience that was not only memorable for our students, but at an affordable cost. (HW)

Beethoven's *Sonata No.17*. The lesson makes use of Sonic Pi, a free code-based music creation and performance tool

AGE RANGE

11-13

RESOURCE TYPE

Visual / block-based coding

REQUIREMENTS

- Scratch 3
- Code templates: helloworld.cc/fabricpatterns
- Lots of creativity!

CODE YOUR OWN FABRIC DESIGN

Susan Wright and **Lorna Gibson** share a lesson that combines fashion and coding to make a digital sketchbook of beautiful fabric designs

A fashion and textiles topic is not an obvious place for a coding activity, but that is exactly what we did and the impact on the young learners was brilliant. Fashion provides a great opportunity to think about maths in the real world. Young learners are often surprised to learn that fashion designers use

mathematics heavily: from measuring models, scaling designs, through to calculating the cost of fabric. Fashion is also driven by fabric design which is based on pattern and repetition – perfect elements of a coding activity.

So, we devised this activity in Scratch 3.0 that sees students code geometric shapes in loops to

build amazing fabric designs. The lesson provided a fun and open-ended activity, allowing the learners to be as creative as they wanted. At the same time, they were exploring shape, angles, loops, and colour in a context that demonstrates the real-world applications of their learning. **(HW)**

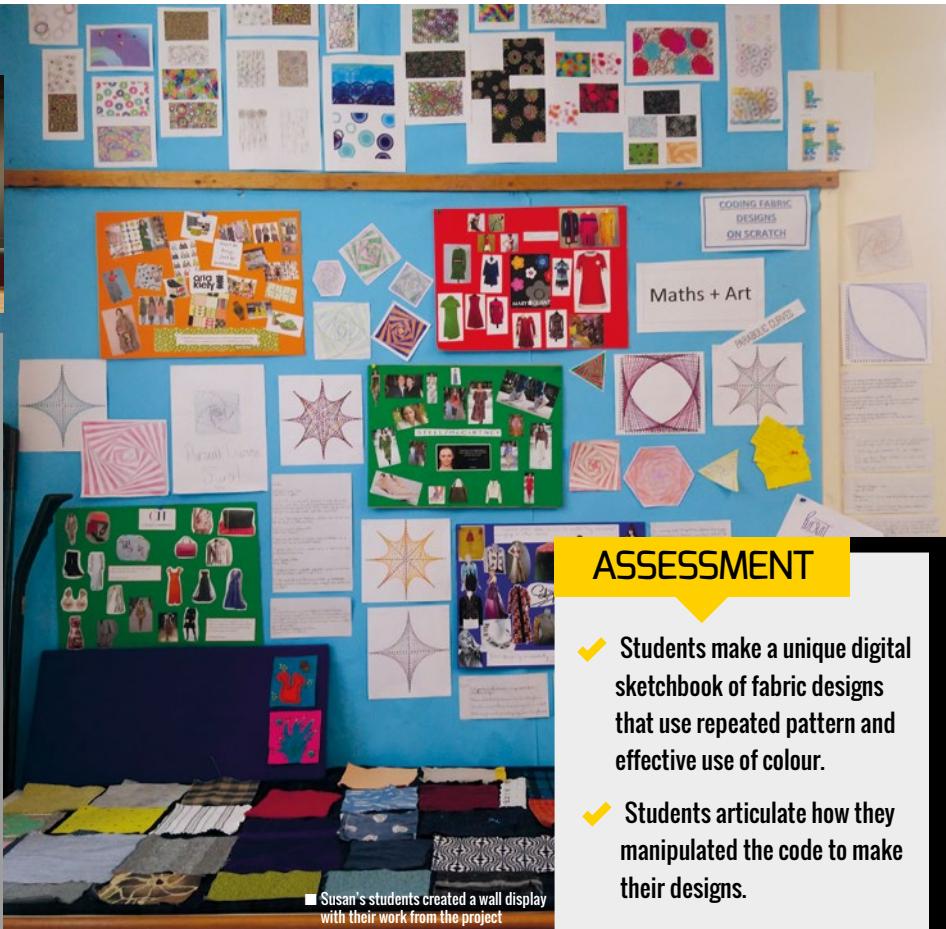
Students visited V&A Dundee for inspiration
© Hufton+Crow



SUSAN WRIGHT AND LORNA GIBSON

Susan Wright is a primary school teacher in North East Fife with a passion for using STEM activities to develop expressive arts in the classroom.

Lorna Gibson (@CodeClubLorna) is a Programme Manager for Code Club, a STEM Ambassador and a Raspberry Pi Certified Educator.



ASSESSMENT

- ✓ Students make a unique digital sketchbook of fabric designs that use repeated pattern and effective use of colour.
- ✓ Students articulate how they manipulated the code to make their designs.

PRE-CODING ACTIVITY: INSPIRATION FIRST

The first task is to gather some design inspiration. For us, this involved a field trip to the Scottish Design Gallery at the V&A Dundee. If that is too far for you, don't worry: there may well be an appropriate, accessible museum near you that features examples of fabric and fashion. And if that is not possible, there are a number of superb online resources, such as this one by National Museums Scotland (helloworld.cc/nms) that you can use.

Learners were asked to look carefully at various items, examining the shape and style of the garment, as well as looking specifically at the design of the fabric. They were encouraged to look at pattern (is it the same design or group of designs that appear across the fabric?), placement (are they regularly/uniformly distributed or is it irregular/random?), size (are all elements of the design the same size or different sizes?), and colour (are all the elements different or are they the same in different colours?).

Learners made notes in a design sketchbook which they used throughout the field trip and across the topic. In groups, they shared their findings and what they liked the best.



A Fair Isle jumper on display at V&A Dundee - students studied patterns in the garments

CODING FABRIC DESIGNS

Back in school, they were ready to take some of the inspiration from their field trip and think about becoming their own fabric designers. Their brief was to code a pattern-based fabric, but it could take on any of the features they had seen during their inspiration session, such as random or uniform placement of motifs, size, and colour.

The students were provided with three template patterns of code (available at helloworld.cc/fabricpatterns):

- Template 1 provided code that drew small, multicoloured arcs. Learners were encouraged to make changes to the repeat, move steps and the degrees blocks to support understanding of their effect on the screen.
- Templates 2 and 3 were very similar, to encourage them to explore how small changes can make a big difference.
- Unlike template 1, where the pattern is drawn randomly, templates 2 and 3 use the 'when space key pressed' block to demonstrate user-controlled

placement. Both also introduce looping and repeating geometrical shapes to make more intricate designs.

- Template 2 again uses the colour change in the loop to make it multicoloured, with template 3 setting the colour to a particular hue.
- Learners were encouraged to manipulate the numbers to determine cause and effect and save the changes they like the most.

Once the learners had been given ample opportunity to experiment with the starter code, they started to design their own fabrics. They were encouraged to save their designs into a document to make a digital sketchbook.

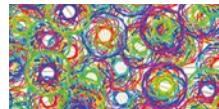
This was an open-ended activity that allowed them to be very creative, all the while being free to manipulate elements in the code to explore and understand the cause and effect on shape. The activity was engaging and enjoyable to the young learners. It also nicely led to a wider exploration of maths and art within the classroom, such as parabolic curves which in turn led to a beautiful wall display.

LEARNING OBJECTIVES

- ✓ Develop an awareness of the link between maths and art by creating unique fabric patterns
- ✓ Explore problem-solving strategies to meet textile challenges with block-based coding
- ✓ Create, develop, and evaluate solutions in response to a design challenge

DIFFERENTIATION

You could use coding buddies where appropriate. The open-ended nature of the activity meant that learners who needed more of a challenge could simply explore more challenging designs.



Examples of students' patterns created in Scratch

AGE RANGE

10-13

YEAR GROUP

UK Year 6-8,
US Grade 7-9

LESSON TYPE

Text-based
web design

REQUIREMENTS

- Desktop / laptop computer
- Modern, up-to-date web browser (Chrome or Firefox preferred)
- Access to goggles. mozilla.org

HACKING CONTENT ON THE WEB

Mozilla's X-Ray Goggles allows students to manipulate web pages. **Neil Rickus** highlights how this can provide a hook and an introduction to HTML

As children access more content online and the tools available to spread 'fake news' become more sophisticated, the need for young people to critically evaluate the information presented to them has never been greater. Within school, there are limited, age-appropriate examples to share with children, particularly with young pupils, which can make the subject challenging to teach. Related to this, many curricula introduce children to HTML (Hypertext Markup Language – a text-based language, consisting of keywords and tags, which tells a web browser how to display a page) in order to produce their own websites, which can be daunting to newcomers and difficult to debug. Mozilla's X-Ray Goggles (goggles.mozilla.org) is a web-based tool that helps us address these issues through enabling learners to modify existing webpages and share them online.

This lesson was effective in removing some of the barriers students face when using HTML for the first time, such as the need to use a wide range of tags to produce even a basic page. It also gives pupils a choice in the output they produce.

Taught successfully with pupils aged 10-13, the lesson could be used with older learners too. Younger pupils may be able to access the lesson, although they will need significant experience navigating web-based content and editing text. **(HW)**

FIRST STEPS IN REMIXING – 20 MINS

In order to use the X-Ray Goggles, pupils should drag a button from the from the X-Ray Goggles website on to the browser's bookmarks bar, which makes the tool available when visiting other sites.

⊕ Drag and drop me in your bookmark bar

When clicked, the button allows pupils to view the HTML used to display a page section, such as some text or an image, rather than a whole page. For example, hovering the mouse over a popular news site's headline would show the use of the `<h1>`, or 'heading 1' tag.

`<h1>Climate change: UK 'can cut emissions to nearly zero' by 2050 </h1>`

Clicking on the text takes this one step further, by displaying the tags used and the associated text. This can then be edited and made to appear as if part of the 'real' webpage. Using the headline above, you can demonstrate to pupils how we can make the climate change targets in the headline more challenging, along with altering other parts of the page.

Climate change: UK 'can cut emissions to nearly zero' by 2045

By Neil Rickus
Doesn't work for the BBC

By altering page sections in this manner, pupils can quickly produce engaging output and become more aware of a webpage's structure – without the need to start from scratch.

Pupils can alter text on a webpage of their choice. They could also change the tags used, or even add extra tags to the page content, and see how this affects the way the page is presented. Mozilla has a useful 'HTML Cheatsheet' (helloworld.cc/html_cheats), which can be used as a reference guide.

HEALTH AND SAFETY REQUIREMENTS

Teachers should ensure suitable measures are in place to minimise the risk of pupils accessing unsuitable content online and they are aware of the processes to follow should pupils access inappropriate material. Pupils should be reminded of your school's e-safety policy (or similar) and what to do if they feel unsafe when using a digital device.

MOVING ON TO IMAGES – 25 MINS

After text within a page has been altered, pupils can go on to changing a page's images. Once the X-Ray Goggle button is clicked and an image selected, pupils are able to edit the `` tag's attributes, including the image's URL, or web address, within the 'src' attribute.

Climate change: UK 'can cut emissions to nearly zero' by 2045

By Neil Rickus
Doesn't work for the BBC

2 May 2019 | Share



Climate change: UK 'can cut emissions to nearly zero' by 2045

By Neil Rickus
Doesn't work for the BBC

2 May 2019 | Share



With Mozilla X-Ray Goggles, it's possible to replace any image on a website - as seen here: replacing the relevant photograph of Natural History Museum with that of a cat.

In theory, any image from the web can be used, although the use of the tool provides an opportunity to discuss copyright and image with pupils. For example, Pixabay (pixabay.com) and Wikimedia Commons (helloworld.cc/wikimedia) have reusable and royalty-free images available, while Google's image search tools have the 'Usage rights' specified to enable the user to select suitable images (helloworld.cc/freeimagetips).

Once pupils have found an image, they should copy the image's URL, such as through using the 'Copy image address' option when right-clicking on an image. This can then be replaced within the `` tag's 'src' attribute using the X-Ray Goggles tool.



NEIL RICKUS

Neil (@computingchamps) is a Senior Learning Manager at the Raspberry Pi Foundation and a Senior Lecturer in Computing Education at the University of Hertfordshire. He is a CAS community leader and a Raspberry Pi, Google, and Microsoft Certified Educator.

ASSESSMENT

Teachers can examine the modified webpages to check whether they included appropriate tags. Pupils could be asked to share their work and describe the tags used within each section.

LEARNING OBJECTIVES

- ✓ Identify HTML text and image tags
- ✓ Edit HTML text and image tags
- ✓ Modify web-based content
- ✓ Evaluate the reliability of web-based content

PUBLISHING TO THE WEB – 15 MINS

Modified webpages can be published online using X-Ray Goggles, although this does require an account; it may therefore be more appropriate to take screenshots of the pages to enable work to be shared. Webpages produced by pupils could be used to facilitate discussions around how to evaluate content. For example, pupils might examine the URL to see if it matches that of the expected organisation / author, or whether the information is verifiable. X-Ray Goggles can also be used as a link to enhance other areas of the curriculum. News stories produced in English could be presented using websites from news organisations, for instance.

DIFFERENTIATION

Support - Pupils could be given paper versions of the relevant HTML tags, which will enable them to produce the required text before entering it into the computer. Pupils could be provided with a selection of images to use, with the image URL explicitly stated.

Stretch and challenge - Pupils could be encouraged to use additional tags and/or tag attributes.

AGE RANGE

7-13

LESSON TYPE

Unplugged (pen and paper)

REQUIREMENTS

- Coloured pens or pencils
- Print out of the worksheet from helloworld.cc/pixelart, or squared paper

PIXEL ART UNPLUGGED

Martin O'Hanlon shows how creating pixel art with pens and paper can help students learn about how digital images are created and stored

Students will work in pairs or threes to make their own pixel art pictures. Creating, encoding, and decoding their own pixel art on paper teaches how computers represent and display images.

The key messages are:

- A computer represents pictures through numbers. Computer screens are made up of a grid of tiny squares called pixels (picture elements). Each square on the grid represents one pixel.

- Everything you see on a computer screen uses many different coloured pixels.
- A pixel can be represented by its coordinates on the grid.
- Pixel coordinates on the grid start from (0,0) in the top left-hand corner. The grid's x-axis goes from left to right, and its y-axis goes from top to bottom.

Unplugged activities allow students to explore computing concepts in a different context, and can help those struggling with a concept to

OBJECTIVES

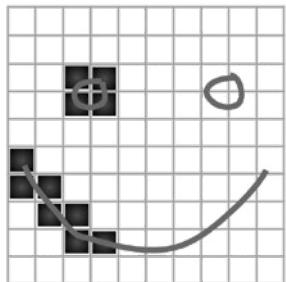
- ✓ Describe how computers create and store images
- ✓ Use (x, y) coordinates

understand. Variants of this lesson plan have been created for Code Club (helloworld.cc/pixelart) and the Scouts digital maker badge (helloworld.cc/scouts). [HW]

ACTIVITY 1: CREATE A SMILEY FACE OUT OF PIXELS

5 minutes

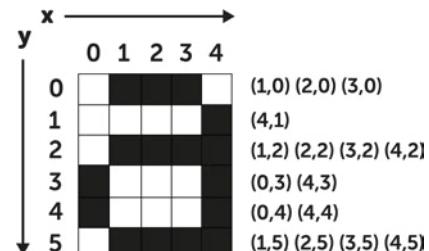
- Tell the students that they are going to draw images the way computers do and explore how they store image as list of numbers.
- Ask the students to draw a smiley (two circles for eyes and a curve for the mouth) on some squared paper.
- The students should then colour in each square which the lines for the eyes or mouth have crossed.
- Explain to the group that this is how computers display images, by turning everything into squares known as pixels.



ACTIVITY 2: IMAGES ARE MADE OF PIXELS

5 minutes

- Describe that computer screens, and the pictures they show, are divided up into grids of very small dots called pixels (picture elements).
- Demonstrate that when you zoom in on a picture, what you can see is its pixels.
- Discuss how computers represent everything, even pictures, with numbers; how this means that when you design a picture for a computer, you need to turn its pixels into numbers.
- Describe that each pixel has an x-y coordinate in a grid of columns and rows. Use the diagram of



the letter 'a' to show how each square on the grid corresponds to a coordinate for each row.

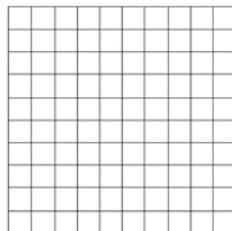
- Interesting fact! The grid numbering starts at (0,0) in the top left-hand corner of the grid. This is because computers use the same (x, y) coordinates as old-fashioned televisions, which display images in horizontal lines from left to right, starting at the top and moving downwards.

ACTIVITY 3: DECODING IMAGES 10 MINUTES

- Explain to the students that they will recreate an image (like the letter 'a') using just the coordinates, and that this is known as decoding.
- Present the students with a numbered grid and the coordinates which, when filled in, will make up an image.
- Student should make a prediction of what they think the image might be by first looking at the coordinates. Then, using pencils or pens, fill in the squares which

coincide with the coordinates to create the image on the grid.

- Let them know they can use any colour they wish.



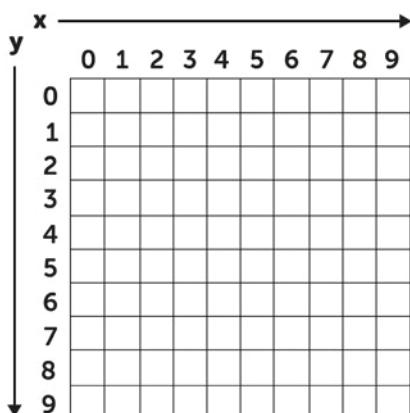
(1,1) (2,1) (6,1) (7,1)
(0,2) (3,2) (5,2) (8,2)
(0,3) (4,3) (8,3)
(0,4) (8,4)
(1,5) (7,5)
(2,6) (6,6)
(3,7) (5,7)
(4,8)

ASSESSMENT

Questions you could ask your students:

- Were there any differences in your images? If so, why?
- Your pixel art pictures only use one colour; how could you create and encode multicolour images?

ACTIVITY 4: ENCODING IMAGES 20 MINUTES



- Explain to the students that it's now their turn to create a pixel art image and encode it by creating a list of the x-y coordinates.
- Give the students a blank grid, and ask them to draw an image by filling individual squares just like the pixel art they decoded. The image can be anything they like, but should be created in one colour.
- The pixel art images should then be encoded by writing down the individual coordinates for each pixel in a table.

DIFFERENTIATION

Support: Get students to draw simple shapes first (e.g. triangles, squares, rectangles) to get used to the concept. Once they are more confident, they can draw more complex shapes (e.g. insects, trees).

Students who are unable to use pen and paper could use Piskel (piskelapp.com) to create the images digitally.

Stretch and challenge: Can students figure out how to create animations using pixel art? Can they draw a sequence of pictures that can be animated? How might they make the animation more efficient by only storing the coordinates which have changed?

ACTIVITY 5: SWAP ENCODED IMAGES 10 MINUTES

Students will swap the coordinates for their pixel art with a friend who will try to decode the image

- Ask the students to swap their table of coordinates with a friend.
- They should attempt to decode (recreate) the images from the coordinates.
- When completed, compare the original and reproduced images. If there are differences, the students should attempt to discover why.



MARTIN O'HANLON

Martin (@martinohanlon) works in the learning team at the Raspberry Pi Foundation, creating projects, learning resources, and online training courses.

FURTHER READING

- Raspberry Pi project unplugged pixel art helloworld.cc/pixelartproject
- Code Club pixel art unplugged worksheet helloworld.cc/pixelart

AGE RANGE

13-16 years

LESSON TYPE

- Text-based programming
- Project-based learning

REQUIREMENTS

- Python 3
- IDLE/other appropriate IDE
- Code at: helloworld.cc/database

MY FIRST DATABASE: PYTHON AND TEXT FILES

Mac Bowley shares a lesson that introduces students to databases using Python and text files

In this lesson, students create a library app for their books. This will store information about their book collection and allow them to display, manipulate, and search their collection. You will show students how to use text files in their programs that act as a database.

The project will give your students practical examples of database terminology and hands-on experience working with persistent data. It gives opportunities for students to define and gain concrete experience with key database concepts using a language they are familiar with.

The script that accompanies this activity can be adapted to suit your students' experience and competency. This ready-to-go software project can be used alongside PRIMM, pair programming, or as a worked example to engage your students in programming with persistent data. **(HW)**

WHAT MAKES A DATABASE? 10 MINS

Start by asking the students why we need databases and what they are: do they ever feel unorganised? Life can get complicated and there is so much to keep track of, the raw data required can be overwhelming. How can we use computing to solve this problem? If only there was a way of organising and accessing data that would let us get it out of our head. Databases are a way of organising the data we care about, so that we can easily access it and use it to make our lives easier.

Then explain that in this lesson the students will create a database, using Python and a text file. The example I show students is a personal library app that keeps track of which books I own and where I keep them. I have also run this lesson and allowed the students pick their own items to keep track of – it just involves a little more planning time at the end.

Split the class up into pairs; have each of them discuss and select five pieces of data about a book (or their own item) they would like to track in a database. They should also consider which type of data each of them is. Give them five minutes to discuss and select some data to track.

“ THE PROJECT WILL GIVE YOUR STUDENTS PRACTICAL EXAMPLES OF DATABASE TERMINOLOGY

DATABASE TERMINOLOGY

Databases are organised collections of data, this allows them to be **displayed, maintained, and searched easily**. Our database will have one **table**, effectively just like a spreadsheet table. The headings on each of the columns are the **fields**, the individual pieces of data we want to store about the books in our collection. The information about a single book are called its **attributes** and are stored together in one **record**, which would be a single **row** in our database table. To make it easier to search and sort our database, we should also select a **primary key**, one field that will be unique for each book. Sometimes one of the fields we are already storing works for this purpose: if not, then the database will create an **ID** number that it uses to uniquely identify each record.

CREATE A LIBRARY APPLICATION 45 MINS

Make our first record

Pull the class back together and ask a few groups about the data they selected to track. Make sure they have chosen appropriate data types. Ask some if they can find any of the fields that would be a primary key; the answer will most likely be no. The ISBN could work, but for our simple application, having to type in a 10- or 13-digit number just to use for an ID would be overkill. In our database we are going to generate our own IDs.

The requirements for our database are that it can do the following things: save data to a file, read data from that file, create new books, display our full database, allow the user to enter a search term, and display a list of relevant results based on that term.

We can decompose the problem into the following steps:

- Set up our structures
- Create a record
- Save the data to the database file
- Read from the database file
- Display the database to the user
- Allow the user to search the database
- Display the results

Have the class log in and power up Python. If they are doing this locally, have them create a new folder to hold this project. We will be interacting with external files and so having them in the same folder avoids confusion with file locations and paths. They should then load up a new Python file.

Setting up structures

To start, download the starter file from the links provided. Each student should make a copy of this.

At first, I have them examine the code, and then get them to run it. Using concepts from PRIMM, I get them to print certain messages when a menu option is selected. This can be a great exemplar for making a menu in any application they are developing.

This will be the skeleton of our database app: giving them a starter file can help ease some cognitive load from students.

Have them examine the variables and make guesses about what they are used for.

- `current_ID` – a variable to count up as we create records, this will be our primary key
- `new_additions` – a list to hold any new records we make while our code is running, before we save them to the file

The skeleton of the database app

```
#imports
import csv

#Variables
current_ID = 1

new_additions = []

filename = "library.csv"

fields = ['ID', 'Title', 'Author', 'Genre', 'Year', 'Location']

data = []

#Code

print("-----Welcome to the Python Library organiser-----")

choice = ""

while choice.lower() != "x":
    print("""What would you like to do?
    1 - Add a book
    2 - Display your Books
    3 - Search for a Book""")
    
    choice = input(">")

    if choice == "1":
        #Code to add a record

    elif choice == "2":
        #Display the data

    elif choice == "3":
        #Search the data

        #Display the results

    elif choice.lower() == "x":
        print("Thank you! Shutting down.")
    else:
        print("Sorry, I didnt recognise that option")
```

LEARNING OBJECTIVES

✓ Describe the basic components of a database (tables, records, and attributes)

✓ Understand the basic principles of database manipulation

✓ Select appropriate data types

```

if choice == "1":
    #Code to add a record
    new_record = {
        "ID": current_ID,
        "Title": input("What is the title of the book? >"),
        "Author": input("What is the author's first name? >"),
        "Genre": input("What genre is the book? >"),
        "Year": input("What year was the book released? >"),
        "Location": input("Where is the book? >")
    }
    new_additions.append(new_record)
    current_ID = current_ID + 1
    print("-"*15)
    print("New Record added successfully!")

```

■ Dictionaries are used to store records

“ THIS IS A USER-FOCUSED APPLICATION

ASSESSMENT

Students should be able to correctly predict the outcome of their modifications, or describe how this code implements the key database terms covered in the introduction.

- `filename` – the name of the database file we will be using
- `fields` – a list of our fields, so that our dictionaries can be aligned with our text file
- `data` – a list that will hold all of the data from the database, so that we can search and display it without having to read the file every time.

Create the first record

We are going to use dictionaries to store our records. They reference their elements using keys instead of indices, which fit our database fields nicely. We are going to generate our own IDs. Each of these must be unique, so a variable is needed that we can add to as we make our records. This is a user-focused application so let's make it so our user can input the data for the first book. The strings, in quotes, on the left of the colon, are the keys (the names of our fields) and the data on the right is the stored value, in our case whatever the user inputs in response to our appropriate prompts.

We finish this part of by adding the record to the file, incrementing the current ID, and then displaying a useful feedback

message to the user to say their record has been created successfully. Your students should now save their code and run it to make sure there aren't any syntax errors.

How to use the code in your class

To complete the project, your students can add functionality to save their data to a CSV file, read from a database file, and allow users to search the database. The code for the whole project is available at helloworld.cc/database.

You may want to give your students the entire piece of code. They can investigate and modify it to their own purpose. You can also lead them through it, having them follow you as you demonstrate how an expert constructs a piece of software. I have done both to great effect. Let me know how your classes get on! Get in touch at contact@helloworld.cc.

DIFFERENTIATION

You could make use of pair programming, with carefully selected pairs taking it in turns in the driver and navigator roles. You could also offer differing levels of scaffolding: providing some of the code and asking them to modify it based on given requirements.



MAC BOWLEY

Mac Bowley (@Mac_Bowley) loves digital making, video games, audiobooks, and baked goods. He's a Learning Manager at the Raspberry Pi Foundation, and is passionate about computer science education. He has taught in a range of different environments, from summer camps to GCSE classes.

RECURSION AND ALGORITHMIC ART

Paul Curzon, Jane Waite, and Ged Gast show that it's possible to doodle algorithmically

Recursion is a fundamental idea of computing. Recursive processes are also a fundamental of the way many organisms grow. Recursive algorithms are a way to create drawings and paintings of nature, copying the processes that nature itself follows.

A recursive algorithm is just one that applies itself to a simpler version of the problem. What does that mean in practice? To solve a problem, a recursive algorithm breaks it into one or more simpler versions of the same problem. These are called step cases. It solves those step cases in the same way, repeatedly using the same algorithm. It then combines the solutions from those smaller problems back into an answer to the original problem.

Problem solved? Not quite – without a base case, the algorithm wouldn't have

an end point. Because the sub-problems are always simpler, eventually the simpler problems become so simple that they can be solved trivially without recursion. To stop the algorithm continuing indefinitely when you hit one of these trivial base cases, you just solve them, passing the solutions back to be combined to solve the harder problems.

To sort a pile of numbered exam scripts, for example, split the pile into two halves, and sort those smaller half piles recursively (these are step cases) by splitting them in half in the same way. Eventually, the pile you are sorting is only one script long (a base case). You have the solution trivially – a pile with only one thing in it is sorted – so just return it. Merge it with the other half it was split from, which also has only a single script in it, also trivially sorted, to give a sorted pile of size 2. That is merged with the other half it

was split from to give a sorted pile of size 4, and so on. As all the sorted parts are merged and passed back, you end up with ever larger sorted parts of the original, until that whole original list is sorted.

Recursion in nature

This recursive process mirrors self-similar structures in nature. Look at a tree, a fern, or a tall stalk of grass. Each tree branch is just a smaller version of the whole. Those branches are made of smaller versions still. Nature uses recursion as an easy way to build complex things. To solve the problem of creating a tree, nature just grows smaller tree structures off the main trunk. How are those smaller branches made? In the same way, from still smaller 'trees'. The base cases of a tree are its leaves! Trees grow following a recursive process.

GRASSES FROM RECURSIVE ALGORITHMS

This recursive algorithm draws curvy grass. Experiment with variations to create different kinds of grass:

1. Draw a single slightly curved vertical line (see image)
2. DoodleDrawGrass from that line

To DoodleDrawGrass from a given line:

IF the image feels finished

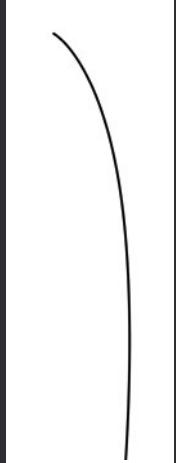
THEN STOP

ELSE

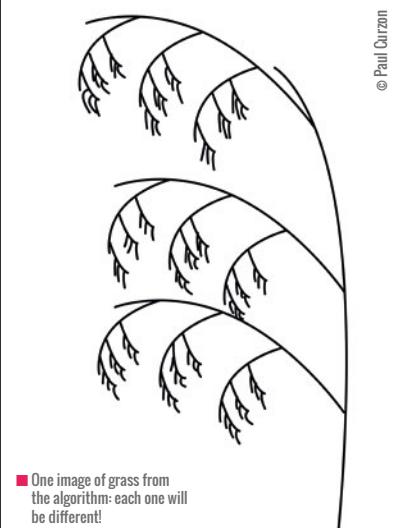
1. Draw three shorter *curving* lines from the *top two-thirds* of the line on its *lower side*

2. Choose a new existing line that hasn't been drawn from and DoodleDrawGrass from that line

Here, the base case is when that the image feels finished and you can stop. The step case involves picking any line and doodle drawing recursively from that line.



■ The starting point for a doodle art picture of grass



■ One image of grass from the algorithm: each one will be different!

© Paul Curzon

“ DIFFERENT ALGORITHMS GIVE YOU DRAWINGS OF DIFFERENT NATURAL PHENOMENA

Doodle art and CGI

The link between recursive algorithms and recursive processes in nature gives us an algorithmic way to draw nature. We can write recursive rules of the process, then draw by following the rules. Different algorithms give you drawings of different natural phenomena. With a little room for randomness in the algorithm, parts not exactly tied down, every drawing will be different. This idea is how computer-generated imagery (CGI) of nature is created for games and films. An artist doesn't draw every tree and blade of grass: whole forests and fields can be generated automatically.

You don't need a computer to draw recursive art, you can do it by hand, following the algorithms yourself, in whatever medium you prefer. With a little artistic skill and creativity providing the randomness, you can create beautiful imagery just by doodling, following an algorithm.

Doodle art inspired by William Morris

In nature, some recursive processes just keep going. They have no base case. Ivy, for example, splits off shoots recursively that grow in the same way, splitting off further shoots. William Morris, grandfather of the Arts and Craft movement, is famous for his patterns inspired by nature and many involve creepers and climbers filling the page. These form the inspiration of the recursive algorithms we have shared here.

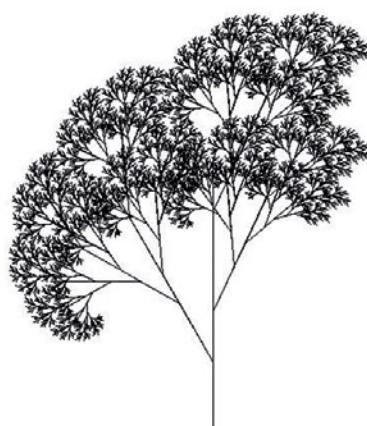
We have doodled algorithmic art inspired by his imagery, but following recursive rules. One of our examples is shown on page 89.

Doodle art in the classroom

Algorithmic doodle art provides excellent examples of algorithms which, when implemented, can create beautiful visualisations. In primary classrooms, pupils can use the activity to follow one of the algorithms physically on a huge piece of paper as they learn what an algorithm is. In cross-curricular art topics, when learning

about artists or looking at natural forms, students can scan and search for recursive patterns and follow the algorithm in their sketchbooks, while at the same time consolidating the understanding of the selection ('if then') concept. Learners can also explore algorithmic thinking, by changing the algorithm to create patterns of their own.

In secondary classes, a similar consolidation of understanding of functions can be incorporated, along with a discussion of how visually creative parameters might be modified to change the colour, line thickness, number of lines, or the basic element (motif) being drawn.



■ A tree created using the Logo programming language and relying heavily on recursion. Each branch can be seen as a smaller version of a tree

At A-level (ages 16–18), recursion itself is formally introduced and doodle art provides an ideal starter or flipped learning activity to introduce the concept of recursion with the opportunity to then implement the algorithm as code. PRIMM (predict, run, investigate, modify, make) can be used to explore the implementation of the algorithm and develop more complex algorithms which use recursion in programming languages such as Processing (see helloworld.cc/tree). This could be incorporated in A-level art and design, as learners can use programming to explore the form of recursive patterns before implementing their design in diverse media including textile, fine art, print, 3D, or digital forms.

Engaging learners in computer science through digital art not only gives learners an opportunity to be creative, it also helps them develop a deep understanding of the underlying concepts as they explore and take ownership for the personally meaningful artefacts that they develop. There are more ideas for algorithmic art at teachinglondoncomputing.org, and we'd love to know if you use these doodle art activities – let us, and *Hello World*, know what you make of them! [\(HW\)](#)

PAUL CURZON, JANE WAITE AND GED GAST

Paul Curzon has been a member of CAS since its foundation. He is a Professor of Computer Science at Queen Mary University of London and co-founded both cs4fn and Teaching London Computing. He is author of the book *The Power of Computational Thinking*.

Jane Waite has worked both as a primary teacher and in industry as a developer. She worked on the Barefoot programme, was CAS London manager, and runs the CAS London Community of Practice. Jane is now researching how to teach primary programming, and works at Queen Mary University of London on a number of teacher training and support programmes.

Ged Gast is a visual arts specialist, education consultant, and past President of the National Society for Education in Art and Design.

WILLIAM MORRIS-INSPIRED DOODLE ART

This recursive algorithm creates William Morris-style imagery. Experiment with your own variations of the algorithm and initial stem designs.

1. Draw a Stem following the instructions below from any point on the page
2. DoodleDrawMorris from that stem
3. Add extra berries, birds, etc. in space that is left

To Draw a Stem: (see image)

1. Draw a green stem that curves into space on the page
2. Add three green buds to the end of the stem
3. Draw a short side stem ending in a blue-red flower-head
4. Draw two green leaves off opposite sides of the stem

To DoodleDrawMorris from a given stem:

IF the image feels finished

THEN STOP

ELSE

1. Draw a Stem from its leftmost end bud
2. Draw a Stem from its rightmost end bud
3. Pick a stem with empty buds and DoodleDrawMorris from that stem



■ The starting image for our William Morris-inspired image



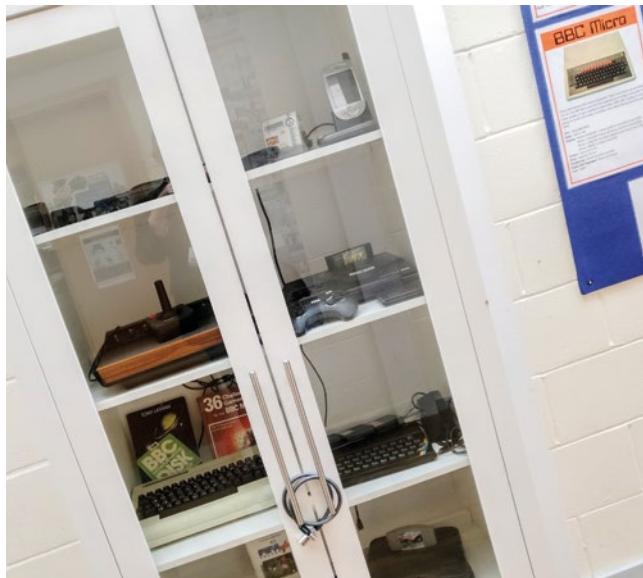
COMPUTING CLASSROOM DISPLAYS

A space for inspiration or a drain on teachers' time? **Alan O'Donohoe** shares ideas for displays that you and your students will love

A privilege of my role as an educational consultant is that it regularly affords me opportunities to visit classrooms in primary and secondary schools. This has allowed me to see many different approaches to classroom displays and observe a range of approaches in action.

While the research on the links between classroom displays and learning are few and varied in their findings, there are several arguments for investing time into creating effective displays:

■ **Enjoyment** – Some teachers claim to derive a great sense of fulfilment and joy from creating classroom displays. That may also be perceived as a welcome distraction from marking books and other mundane aspects of the role. This may contribute to a more positive sense of well-being.

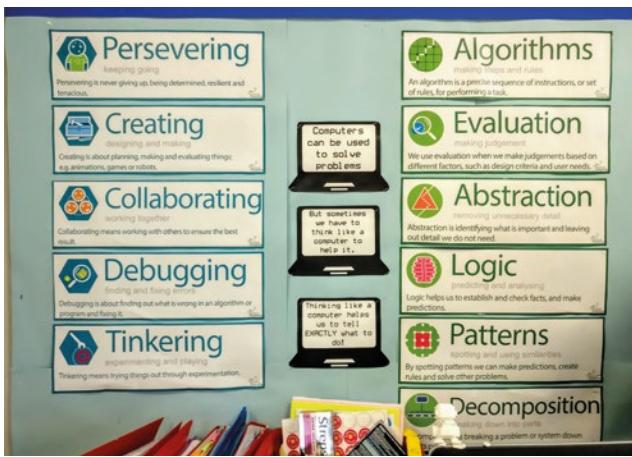


■ A cabinet displaying retro games consoles

- **Exhibits of enthusiasm** – They may also be used by teachers who are incredibly passionate about computing education to practically demonstrate their love of the subject to their classes, and hope in turn that some of the same enthusiasm rubs off onto their pupils. These teachers want to be proud of their classroom and they want to show that.
- **Environmental improvement** – Bright, busy, highly decorated environments might lead learners to feel inspired, motivated, and welcomed.
- **School culture** – In your school, there may be long-held traditions and expectations that all staff regularly refresh their wall displays as part of their professional duty.

There are some arguments to the contrary, however, suggesting we avoid classroom displays altogether. These include:

- **Workload** – The National Workload Agreement in 2003 listed “Preparing, setting up, and taking down classroom displays” as an admin task that should not be expected of teachers. This is not to say that teachers cannot plan and prepare classroom displays, but that they should not be expected to.
- **Procrastination** – It can be argued that creating wall displays is a wasteful distraction for teachers, as there are far more important activities that they should be working on than merely making their classrooms look more presentable.
- **Distractions** – It's difficult to find any proof that classroom displays contribute to improvements in learning, and evidence suggests that highly decorated classrooms can be detrimental to learning, particularly for autistic pupils. Perhaps we shouldn't be rushing to clutter our classroom walls with unnecessary decorations all vying for the short attention spans of our learners, and instead aim to keep the classroom environment more business-like and tidy. Displays can quickly deteriorate and make a room look shabby or tired.



■ Computing vocabulary from CAS Barefoot

Displays to inspire

Here are some ideas that I've seen when visiting schools – most are suited to the traditional 8 x 4 ft fibreboard display boards found in many UK classrooms. Why not experiment with a few of these and discover which ones you feel add the most value to your classroom and teaching style?

Display competition - One year, I invited some of my students to put forward design proposals for an engaging classroom display on a range of themes I suggested. I selected three winning entries and offered them the opportunity to create the display using a range of materials available to them. We invited the education reporter from the local paper to interview the winning students about their designs, and the students were thrilled to see themselves and their work feature in the paper.

“ STUDENTS WERE THRILLED TO SEE THEMSELVES AND THEIR WORK FEATURE IN THE PAPER

Focus - You could use a display as part of a strategy to raise the profile of an important or emerging issue, e.g. the contributions of women to computing, diversity in STEM, cybersecurity threats, or careers in computing.

Expectations and habits - Displays provide an opportunity to remind your students of your expectations for behaviour, or ways they should work. For example, you could design a display that highlights good programming habits such as 'try to first identify what the problem is you're trying to solve', or you could highlight the characteristics of effective drivers and navigators in pair programming.

Gallery - You could showcase the best examples of pupil work, to award achievements throughout the year, 'work of the week', or to celebrate your 'Class Computing Champions'. This can be built up

PRACTICAL TIPS

- **Seek inspiration** - Before you start on your display project, use photo sharing communities, like Pinterest, to help provide you with some inspiration. Some teachers have shared their displays on Twitter and Facebook to support others. You might consider sharing yours afterwards.
- **Interactivity** - To seek deeper engagement with a display, seek ways to interact with the viewer. This might include a series of questions with the correct answers hidden below, or you could even follow the examples of teachers like Rebecca Lofthouse, Matt Moore, and Allen Tsui who have designed Raspberry Pi and micro:bit-controlled wall displays to interact with.
- **To laminate or not?** - If you are already taking steps to reduce your consumption of single-use plastics, rather than laminating everything in sight you might choose to be more selective about what you choose to laminate. This could be used to protect some components of a display, or to add a 'dry-wipe' functionality so that some of the content can be easily updated and refreshed as required.
- **Protective coverings** - Wall displays on corridors, supervision 'blind spots', or other areas of high traffic are likely to deteriorate rapidly due to wear and tear. Heavy-duty rolls of film are available that cover the whole display and do a fantastic job of protecting displays from idle hands. While costs of these products vary, protecting an entire display board amounts to about £5.



■ Nicola Looker has decorated her classroom door

DISPLAY TIPS FROM TEACHERS

"Avoid displays that are simply 'wallpaper', that are only there to fill a space. Displays should be thought-provoking and allow teachers to make reference to them with appropriate questions."

- Mike Goobric, Teacher

"I believe good wall displays help to make an interesting, creative, and stimulating learning environment for pupils to thrive in. Displays should be informative and engaging, inviting pupils to interact with them, providing support and inspiration when they need it. They can be used to support language development, and challenge pupils to choose technical language to 'level up' their work. Also, as a teacher, it can support your teaching through covering key concepts, answering questions you are asked repeatedly, and a resource to encourage independence in your classroom"

- Nikie Arthurs, Computing Teacher

- into quite an achievement for a pupil to have their work featured in the gallery that week, and you could send a message home to the parents of these pupils as part of the recognition.

WAGOLL wall - Quite different from the Gallery, this display is used to highlight 'What A Good One Looks Like'. This way, pupils working on a current piece of work that may be unfamiliar can benefit from annotated examples and snapshot views previously prepared by the teacher.

Resource bank - These displays share information sources likely to support pupils in independent learning activities. These often make use of posters available from organisations and suppliers and may include things like posters on computational thinking, Python cheat sheets, and examples of Scratch programs. You could encourage pupils to seek information here first rather than asking the teacher.

Big picture - I've used these when pupils have been working on projects with lengthy timescales, to help them see the learning journey, marking out milestones and potential obstacles. In one example where a class were creating a podcast episode, I used a display to help them relate their current position on the project to the end point of the project. I've found that without an oversight of longer projects, pupils can become confused, leading to lower motivation, lack of direction, and a loss of purpose.

Vocabulary - A long list of technical words related to computing, highlighted in different colours, may not make for the most inspiring or imaginative classroom display ever seen. However, it may just prove to be the most useful one. I've tried to keep the number of words visible at any one time to twelve or less, but having the words laminated separately with hook-and-loop tape on the back enabled me to use these for rapid assessments, quizzes, plenaries, lesson starters. Remove three from the wall display before the class arrives and the students can try to guess which words are missing as they enter the room.

Progress chart - Personally, I've preferred to maintain electronic progress charts hosted online so that I can update them when and

A computing timeline



“

CURATE A GALLERY OF HIGH-QUALITY PHOTOGRAPHS OF STUDENTS WORKING IN YOUR CLASSROOM

where I want. Each lesson, the pupils I taught would pester me to display the progress chart and update their positions, I found it a massive motivator for all pupils. Some of my colleagues preferred to have paper versions of these permanently visible in the classroom. I would advocate that these charts only ever displayed how far students were along a trajectory of completing a project and never grades or assessment scores – these can have a demotivating effect.

Wonder wall - This type of display can be used to arouse interest and curiosity in a particular topic of computing. Some teachers have used these to display internal components from computers, game consoles, tablet devices, and mobile phones and they often include prompts and questions for students to try to answer, e.g. ‘Where might the CPU be?’, ‘How many input devices can you identify?’

Aside from the traditional display boards, there are other ways you could decorate your classroom to support and celebrate students:

Photos - Curate a gallery of high-quality photographs of students working in your classroom, the sort you might expect to find in a school marketing brochure. If you, or someone in your school community, has an interest in photography, over time you can build up a high-quality collection for the purpose.



■ Research has shown that making more students feel welcome in classrooms would help recruit more girls into computer science

Artefacts - Teachers Kay Sawbridge and David Bate have created their own mini ‘Computing Museums’ using display cabinets. If you appeal to your community through the school for donations or loans to the ‘Computing Museum’, you’ll be amazed at some of the things that people have held onto, e.g. vintage computer game consoles. To seek wider engagement, you could ask staff to write captions about their own recollections of the devices, and turn it into a quiz where students have to match the memory with the staff member. Ensure you risk-assess the use of these artefacts in the classroom.

A working wall - These are becoming popular in schools as an alternative to wall displays, and might be seen as an extension to your whiteboard. Teacher Esther Devonport often prints children’s work and annotates them with key features. Depending on levels of trust and how much space you have available, it may be that some areas are reserved for pupils during group work. A spray bottle of tap water and some cloths help keep your working walls pristine.

No displays - Perhaps a controversial approach, but you could request that all display boards are removed to keep walls free of clutter. This will require a certain amount of bravery, as you leave yourself open to criticism from others when you challenge the status quo.

Alan's gallery of classroom displays is available at helloworld.cc/displaygallery. If you'd like to add yours, email alan@exafoundation.org.



ALAN O'DONOHUE

Alan (@teknoteacher) has more than 20 years' experience teaching and leading Technology, ICT and Computing in schools in England. He runs exa.foundation, delivering professional development to engage digital makers, support the teaching of computing, and promote the appropriate use of technology.

FURTHER READING

The computing education community on Twitter have these tips for classroom displays:

- Neil Rickus (@computingchamps) pointed out that research has shown that classroom environments can transmit stereotypes to high school students about who belongs in computer science - helloworld.cc/inclusivedisplays.
- CAS London (@cas_london_crc) and cs4fn (@cs4fn) have produced a free set of posters celebrating diversity in computing - helloworld.cc/diversityposters (see pages 8-9 of *Hello World* issue 8 for further information).
- David Whale (@whaleygeek) suggested the free sets of posters for primary (helloworld.cc/primaryjet) and secondary (helloworld.cc/secondaryjet) schools from IET Education.
- Jo Hodge (@hodge_jo) suggests using the CAS Barefoot posters (helloworld.cc/barefootposters), which are useful references for key vocabulary when teaching.

YOUR QUESTIONS

If you have a question you'd like the Learning Team at the Raspberry Pi Foundation to answer, contact us on Twitter via [@HelloWorld_Edu](#). Alternatively, email us with 'Question' in the subject line (contact@helloworld.cc).

Q DO YOU HAVE ANY RECOMMENDATIONS FOR ACTIVITIES FOR A LESSON ON REPRESENTING SOUNDS IN BINARY?

SUZY ROWE, COMPUTER SCIENCE TEACHER



A You can really bring this topic to life by including actual audio samples in your lesson. Dave Marshall's website (helloworld.cc/binarysounds) has examples of sounds at a variety of different sampling rates and resolutions, which you can play to learners to illustrate exactly what effect changing the sampling can have. You can also link

this topic to material they may have covered in physics lessons on waves, and sometimes students will initiate this cross-curricular link themselves.

LAURA SACH (@CODEBOOM)
SENIOR LEARNING MANAGER, POST 16
RASPBERRY PI FOUNDATION

Q I WANT TO TEACH A TEXT-BASED PROGRAMMING LANGUAGE TO YEAR 6. WHICH WOULD BE BETTER, PYTHON OR JAVA?

A Any discussion about what language to teach is always coloured by personal preference, so I will try to outline the benefits and drawbacks of each for young learners. The friendly syntax of Python means that getting a program working is quick and does not require much 'boilerplate' (long and routine) code. For example, to create a program where you type something in, you just need to use the function `input()`, whereas in Java multiple lines of code are needed. Python has good beginner-friendly editors such as Mu and Thonny, but

Java has the Greenfoot environment which boasts graphical representations of objects on the screen. Greenfoot is also well supported in terms of resources and tutorials, but due to its popularity there is a wider range of existing material for Python.

LAURA SACH (@CODEBOOM)
SENIOR LEARNING MANAGER, POST 16
RASPBERRY PI FOUNDATION

YOUR LETTERS



Our letters page is a place for you to join our conversation. If you've got a comment, a question, or an announcement to share, contact us on Twitter via @HelloWorld_Edu. Alternatively, email us with 'Teacher Letter' in the subject line (contact@helloworld.cc).

A focus on inclusion and accessibility

Dear Hello World,

I have been working in a secondary school and sixth form centre for 23 years. I support students of all abilities, but I give particular focus to children with learning difficulties and visual impairment. I often work one-to-one with children who struggle to access the computing curriculum. I spend a lot of my time finding and creating solutions to inaccessible tasks or looking for alternatives to try to ensure that our most-hard-to-teach children can meet their computing objectives.

I am very happy that *Hello World* is available and I am an enthusiastic reader. I would like to see a greater amount of focus on inclusion and accessibility. A lot is said about gender equality in computing, but not enough of about inclusion of learners with disabilities. I see the lack of inclusion and accessibility in computing classrooms as a national crisis as there's no national strategy. Individual schools and institutions seem to do their 'own thing'. Some do nothing, leaving many disadvantaged children behind.

Ilija Jovanovic, SEN IT Learning and Technical Support at St Benedict Catholic Voluntary Academy, Derby

Inclusion and accessibility in computing is something we are always looking to highlight in *Hello World* – all young people should have the opportunity to harness the power of digital

technologies. With this in mind, for example, we recently featured a piece from Emma Bennett, Head of Computing at a school for students with special needs, which focused on strategies that are transferable to all educators.

There is, of course, more that we can do to highlight the issue around the challenges of accessibility of students with visual impairments and learning difficulties. This is something that I am aware of when commissioning and reviewing submissions for the magazine. Alongside this ongoing effort, issue 11 of *Hello World* (published in November 2019) will have the theme of diversity and inclusion.

In terms of wider support for educators of students with disabilities or additional needs, I would also recommend that readers take a look at CAS #include (helloworld.cc/CASinclud) which features a toolkit on inclusion, as well as an active discussion forum with details of relevant events and workshops.

The CoderDojo Accessibility Guide (helloworld.cc/dojoaccess) also has clear suggestions to make computing more accessible, many of which are applicable in a range of educational settings.

It would be fantastic to hear more about the strategies you and other educators have developed in supporting your students. I would encourage anyone interested in contributing a feature to the magazine to get in touch at contact@helloworld.cc.

Sian Williams Page, Editor

Plastic packaging

Dear Hello World,

I am a subscriber and fan of the magazine. However, the packaging it comes in is plastic. I am sure your readers would appreciate a move to a more sustainable option.

Paul Sutton

Thank you for raising this very important point. Discarded plastic packaging can take up to a thousand years to decompose, is toxic to wildlife, and blocks the airways of animals. The manufacture of plastics also exacerbates the climate crisis.

This is at the forefront of my mind when sending out each issue of *Hello World*. The team at Raspberry Pi Press is investigating environmentally-friendly packaging options that are compatible with our machinery. We will keep you posted on progress.

Sian Williams Page, Editor

CORRECTIONS

- Dave Gibbs's feature on initiatives to increase the diversity of students of computer science (issue 8, pages 26-27) used figures from the Roehampton Annual Computing Education Report without reference.
- The lesson plan *Getting started with unity* (issue 7, pages 52-53) was misattributed. Its authors were Philip Harney and Robert McGregor.

We apologise for these errors, both of which are now corrected in the online versions of the magazine.

THE ART OF DIGITAL FABRICATION:

STEAM PROJECTS FOR THE MAKERSPACE AND ART STUDIO

INFOBY Erin E. Riley | PUBLISHER Constructing Modern Knowledge Press | PRICE £29.95 | ISBN 978-0997554335 | URL artofdigitalfabrication.com

It's difficult to open this brilliant book on any page without learning something (ever heard of a 'living hinge') or being inspired to start creating something (interactive gingerbread anyone?)

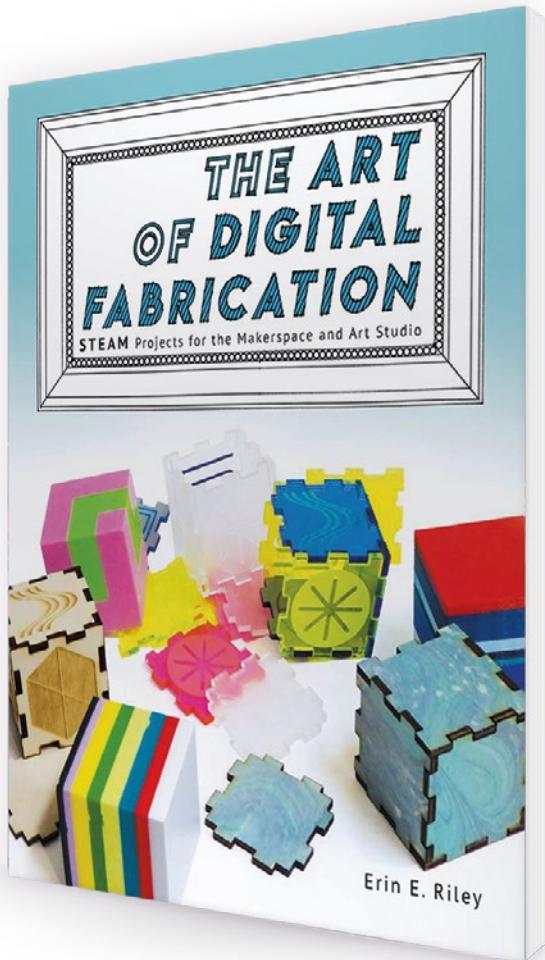
Riley presents 24 projects which bring together software and hardware from the digital world, with ideas and inspirations from the art world.

Before diving into the details of the projects, Riley begins with an introduction on learning, making, and art, and discusses the influences on new media art from fine and modern arts.

Casting with chocolate or concrete?

To make Riley's projects you'll need access to a computer numerical control (CNC) machine, 3D printer, vinyl cutter, digital embroidery machine, and a laser cutter. If you don't have all of these (and who does?), your DT department or local makerspace may be able to help. If this isn't possible, you'll also get ideas for creations that can be made using free software such as Inkscape and the (2D) printers you already own.

The book includes a 'Process Inventory' that has clear instructions for many creative techniques that work well with digital fabrication



machines used in the art studio. As an example, Riley explains how it's possible print a 3D shape, make a mould from that, and use plaster, wax, concrete, or even chocolate to create sculptures.

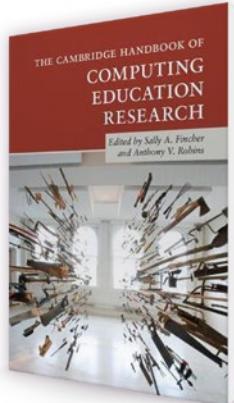
A useful 'Resources' section includes checklists for working with machines, software, and materials, along with tips to explore common creative software programs such as Tinkercad and Photoshop. **(HW)**

THE CAMBRIDGE HANDBOOK OF COMPUTING EDUCATION RESEARCH

INFO

BY Sally A. Fincher and Anthony V. Robins (eds.) **PUBLISHER** Cambridge University Press
PRICE £44.99 | **ISBN** 978-1-108-49673-5 | **URL** helloworld.cc/handbook

With diverse chapters from world-leading researchers in computing education, *The Cambridge Handbook of Computing Education Research* provides a comprehensive overview of the state of the field today. It places research in context, explains tools, summarises results, and provides links to the literature.



Research' provides insight into how teaching programming has evolved and is benefiting from research. 'Computational Thinking' contains a timely exploration of a controversial topic. And 'Schools (K-12)' presents an overview of current research on issues such as programming, attitudes, and course design.

It includes sections on pedagogy, learner misconceptions, motivation, and inclusion. Further, the authors of Learning Sciences for Computing Education and Cognitive Sciences for Computing Education shed light on how computing education can be improved by incorporating insights from related disciplines. [\(HW\)](#)

While the handbook is aimed at researchers, the editors attribute the expansion of computing education research in recent years to the introduction of computing in schools, and many sections are relevant and useful for teachers. The chapter 'The History of Computing Education

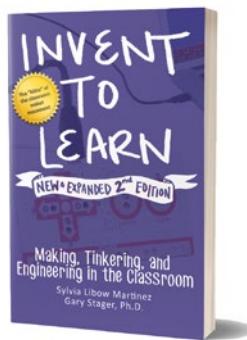
INVENT TO LEARN: MAKING, TINKERING, AND ENGINEERING IN THE CLASSROOM (2ND EDITION)

INFO

BY Sylvia Libow Martinez and Gary Stager | **PUBLISHER** Construction Modern Knowledge Press | **PRICE** £27.37 | **ISBN** 978-0997554373 | **URL** inventtolearn.com

The second edition of the 'bible' of the classroom maker movement includes new coverage of the micro:bit, Scratch 3.0, littleBits, Hummingbird robotics, laser cutters, equity issues, and lessons from schools around the world.

Martinez and Stager provide educators with the how and the why behind making in the classroom, library, or makerspace.



The book features tales of teachers – their successes and failures in fostering a creative spirit in education. It includes thoughts on how to make the case for 'maker-style learning' and a list of resources.

Invent to Learn gives an unparalleled overview of making and learning, placed within the context of existing educational theory and practice. [\(HW\)](#)

ESSENTIAL READING

Three titles that explore aspects of the arts in computer games

VIDEOGAMES: DESIGN/PLAY/DISRUPT

BY Marie Foulston (editor) **PUBLISHER** V&A **PRICE** £19.49 **ISBN** 978-1851779406 **URL** helloworld.cc/foulston

From blockbuster games to quirky indie titles, this thoughtful catalogue from the V&A's recent video game exhibition explores the place of video games in contemporary culture.



DRAWING BASICS AND VIDEO GAME ART

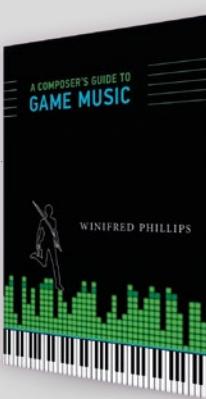
BY Chris Solarski **PUBLISHER** Watson-Guptill **PRICE** £15.74 **ISBN** 978-0823098477 **URL** helloworld.cc/solarski

Solarski covers enough art theory to make a start at creating graphics for games, and helps gamers make more sense of the graphics they see.

A COMPOSER'S GUIDE TO GAME MUSIC

BY Winifred Phillips **PUBLISHER** MIT Press **PRICE** £14.99 **ISBN** 978-0262534499 **URL** helloworld.cc/phillips

More about the creative challenges and opportunities of composing music for games than a technical manual, this places music composition within the broader context of game development.



“HELLO, WORLD!”

Everything you need to know about our computing and digital making magazine for educators

Q WHAT IS HELLO WORLD?

A Hello World magazine is a magazine for computing and digital making educators. Written by educators, for educators, the magazine is designed as a platform to help you find inspiration, share experiences, and learn from each other.



Q WHO MAKES HELLO WORLD?

A The magazine is a joint collaboration between its publisher, Raspberry Pi, and Computing At School (part of BCS, The Chartered Institute for IT). Hello World is supported by Oracle.

Q WHY DID WE MAKE IT?

A There's growing momentum behind the idea of putting computing and digital making at the heart of modern education, and we feel there's a need to do more to connect with and support educators inside and outside the classroom.

Q WHEN IS IT AVAILABLE?

A Your 100-page magazine is available five times per year.

IT'S FREE!

Hello World is free now and forever as a Creative Commons PDF download. You can download every issue from helloworld.cc. Visit the site to see if you're entitled to a free print edition, too.

WANT TO GET INVOLVED?

There are numerous ways for you to get involved with the magazine.

Here are just a handful of ideas to get you started:

● Give us feedback

Help us make your magazine better – your feedback is greatly appreciated.

● Ask us a question

Do you have a question or a bugbear you'd like to share? We'll feature your thoughts and ideas.

● Tell us your story

Have you had a recent success (or failure) you think the wider community would benefit from hearing? We'd like to share it.

● Write for the magazine

Do you have an interesting article idea or lesson plan? We'd love to hear from you.

GET IN TOUCH

FIND US ONLINE

www.helloworld.cc



@HelloWorld_Edu



fb.com/HelloWorldEduMag

Want to talk? You can reach us at:
contact@helloworld.cc

SUBSCRIBE
IN PRINT
TODAY!

PAGES 40-41



(Hello
World)

helloworld.cc