

# (Hello World)

THE MAGAZINE FOR COMPUTING  
& DIGITAL MAKING EDUCATORS

## CHILDREN'S STORIES AND CODING

How to bring together popular books  
for children and computer science!

Issue 6

Autumn Term 2018 [helloworld.cc](http://helloworld.cc)

## ETHICS, COMPUTING & THE CLASSROOM

How the next generation of  
creators and makers are  
sitting in your classroom

## ESCAPE ROOMS

How a games-based approach can  
help you teach computer science

## INTRODUCING MICROBLOCKS

The Scratch-like language  
for bringing creations to life

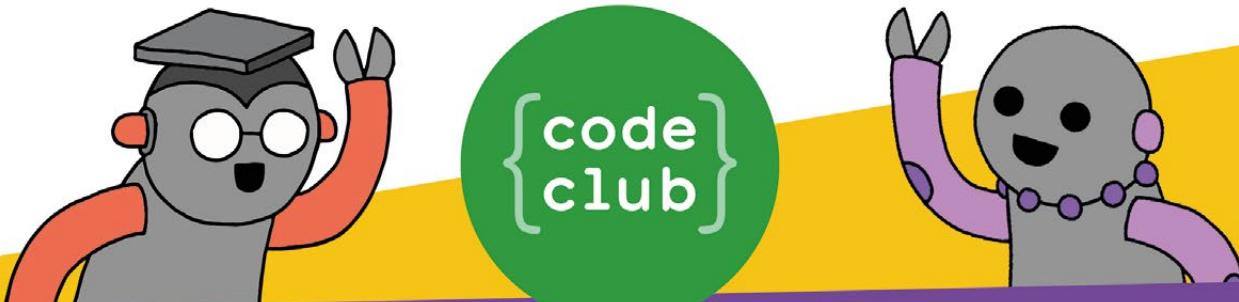
PLUS

CS EDUCATION IN IRELAND • PUTTING COMPUTATIONAL THINKING TO USE • PHYSICAL MANIPULATIVES IN  
COMPUTING • KEY STAGE 3 COMPUTING SPECIAL GUIDE • WHAT'S TURNING GIRLS OFF COMPUTER SCIENCE?

GET ON YOUR  
BIKE WITH  
**MINECRAFT!**

Special guide inside





code  
club

# START A CODE CLUB IN YOUR SCHOOL!

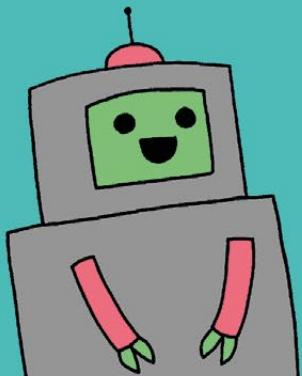
Code Club is a network of volunteers and educators who run free coding clubs for young people aged 9-13.

Our aim is to inspire the next generation to get excited about computer science and digital making.



"We use Code Club's fun educational resources to run a weekly after-school club for Year 7 and Year 8 pupils. The students benefit considerably from the extra challenge!"

**Karen Dadd, Computing Teacher**

- 
- 
- Code Club is free
  - Code Club provides step-by-step guides for Scratch, Python, HTML, and Sonic Pi
  - Code Club helps children develop skills including logical thinking, creativity, and resilience

We have over 6000 clubs across the UK teaching more than 80,000 young people to code—come and join us!

Find out more at [www.codeclub.org.uk](http://www.codeclub.org.uk)





# HELLO, WORLD!

Welcome to the sixth edition of *Hello World*, the free magazine for CS and digital making educators from Raspberry Pi, the BCS, and Computing At School.

We've a focus on ethics in this issue, with some great insights from Peter Millican, Adrian Mee, and Ben Williamson in the UK, and Vicky Sedgwick and Saber Khan in the USA. I'm not alone in worrying that the time spent on Boolean logic and binary arithmetic comes at the expense of considering the wider societal and personal impacts of digital technology. Yeah, it's fun to use machine learning to mine big data sets, but are we always sure that's a good thing?

Game-based learning is something of a theme here too: Mia Chapman links gaming to computational thinking, Alex Clewett's students learn computing through solving escape room puzzles, and Caitlyn Merry introduces the Linux command line through Pac-Man!



SUBSCRIBE IN PRINT FOR FREE  
TURN TO PAGE 32

For those of us in the northern hemisphere, this marks the beginning of a new school year: in England, it's four years since computing became part of the national curriculum (check out Neil Rickus's reflections on this). I'm excited that, at last, government is funding a proper level of support for this through setting up the National Centre for Computing Education (expect more news very soon).

We'd love to share your stories about computing education and digital making with young people: please ping us an email if you'd like to write something for future editions: contact@helloworld.cc

Miles Berry  
**Contributing Editor**



## FEATURED THIS ISSUE



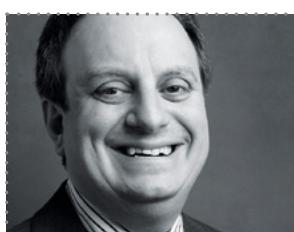
### SIMON HUMPHREYS CAS NATIONAL COORDINATOR

Simon is the National Coordinator for the Computing At School Working Group, which roughly translates to steering the direction and ethos of CAS, and managing the day-to-day activities of the CAS group.



### JANE WAITE CAS LONDON PROJECT MANAGER

Jane is a former industry developer and primary school teacher. She is now a Project Manager with CAS London and works in research at Queen Mary University of London, and King's College London.



### GARY STAGER UNIVERSITY PROFESSOR

Veteran teacher, educator, and speaker, Gary curates the Paper archive at [dailypapert.com](http://dailypapert.com), and is co-author of the highly recommended *Invent To Learn - Making, Tinkering, and Engineering in the Classroom*.

# (HW)

## EDITORIAL

**Publishing Director**  
Russell Barnes  
russell@helloworld.cc

**Contributing Editor**  
Miles Berry  
miles@helloworld.cc

## PRODUCTION

**SBLtd**  
[www.simonbrew.com](http://www.simonbrew.com)

**Managing Editor**  
Simon Brew  
[simon.brew@helloworld.cc](mailto:simon.brew@helloworld.cc)

**Production Editor**  
Rachel Storry

**Production Team**  
Laura Passmore  
John Moore

**Photography**  
Brian O'Halloran/Raspberry Pi Foundation  
Adobe Stock Photo  
BigStockPhoto

## CONTRIBUTORS

Lynne Blair, Julia Briggs, Jody Carter, Mia Chapman, Benson Chong, Alex Clewett, Helen Clothier, Lorna Elkes, Dan Fisher, Lucia Flóriánová, Sway Grantham, Edward Hall, Rosie Hattersley, Simon Humphries, Bob Irving, Saber Khan, Matt Lovegrove, John Maloney, Beverley McCormick, Adrian Mee, Caitlyn Merry, Greg Michaelson, Peter Millican, Stephen Murphy, Alan O'Donohoe, Oliver Quinlan, Neil Rickus, Rowan Roberts, Jez Rogers, Emmanuel Schanzer, Amy Schumacher, Sam Shallcross, Julia Skinner, Sarah J Twigg, Beverley Vallancourt, Jane Waite, Steven Weir, Ben Williamson, Emily Winter

• • • • •

Hello World is a joint collaboration:



**DCS**  
Academy  
of Computing

**COMPUTING AT SCHOOL**  
EDUCATE - ENGAGE - EXCITE

Raspberry Pi  
The Official Partner of DCS

• • • • •  
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., 30 Station Road, Cambridge, CB1 2JH. 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

**COVER FEATURE**

**INTRODUCING ETHICS**  
Technological changes bring new moral questions

## DOING THE RIGHT THING

Can computing help make 'good citizens'?



**16**

**14**

**ETHICS IN THE CURRICULUM**  
How to introduce them to students

## 18 THE NEED TO TALK ETHICS

Teaching future generations



## NEWS AND FEATURES

### 6 NEWS

The latest from the world of CS education

### 22 INSIGHTS

Looking into the community of digital making educators

### 26 GAME-BASED LEARNING

Bringing videogames into the classroom!

### 29 REVIEWING OUR APPROACH TO COMPUTING

A call for a review...

### 30 WHAT'S TURNING GIRLS OFF COMPUTER SCIENCE?

What can be done about it?

### 34 CHILDREN'S LITERATURE

Bringing books to coding!

### 36 ESCAPE ROOMS

Using them to improve problem-solving skills

### 38 PACKET TRACER

Making the principles of networking clear

### 40 PI AS A MANIPULATIVE

Getting hands on with computers – wires and all!

### 42 HACKING MINECRAFT

Using text-based code to change your Minecraft world!

### 44 NETWORK OF EXCELLENCE

The programme aimed at improving teachers' confidence

### 55 USING SCRATCH & MAKEY MAKEY

Tools to help study motion

### 56 PROGRAMS & PARADOXES

Finding paradoxes in self-referential uses of programming languages

### 58 THE 100 WORD CHALLENGE

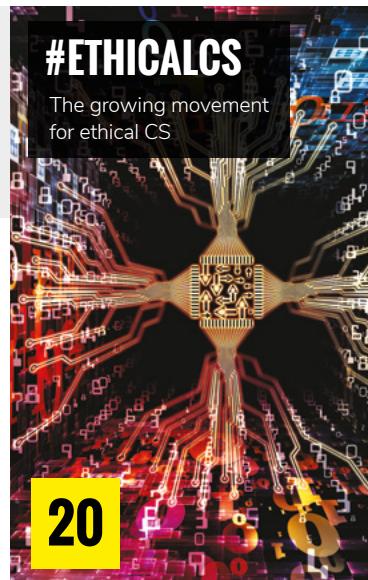
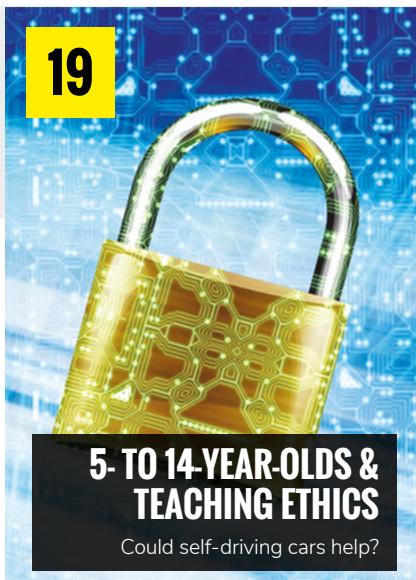
A free resource to unlock creative writing

### 60 CS IN IRELAND

Irish education to introduce CS certificate

### 64 MICROBLOCKS

Bringing electronic creations to life!



- 66 RIDE YOUR BICYCLE TOUR - IN MINECRAFT**  
Turning your bike into an input device!
- 69 FRACTALS**  
More mathematical musings, this time with the Chaos Game!
- 70 APPS FOR GOOD**  
The importance of machine learning
- 72 THERAPEUTIC BENEFITS OF DIGITAL MAKING**  
A Sheffield special school uncovers added benefits from its teaching
- 74 REUSING FAMILIAR TECHNIQUES**  
Example programs are invaluable in the classroom

- 76 TECHATHON**  
Raising the profile of women in tech
- 78 CYBERCRIME**  
The growing global demand for cybersecurity experts
- 88 WHAT HAVE WE LEARNED FROM TEACHING COMPUTING?**  
Four years on from a new curriculum, just what's changed?
- 91 PROFESSIONAL DEVELOPMENT**  
How do you work out what quality CS professional development is?
- 97 GCSE AND A LEVEL CS**  
The highs and lows of this year's exam results

## CONVERSATION

- 62 NEW ONLINE COURSES FROM RASPBERRY PI**  
Four new courses in focus...
- 80 BLUFFER'S GUIDE**  
Hitchhiker's guide to KS3 computing!
- 84 FAQS**  
Your questions answered by the Raspberry Pi team!
- 87 LETTERS**  
Get in touch...

## REVIEWS

- 94 BOOKS ROUND-UP**  
New resources read and reviewed!

## LEARNING

### TUTORIALS & LESSON PLANS

- 46 TURNING UP THE HEAT**  
Using a micro:bit to introduce conditionals
- 48 PAC-MAN TREASURE HUNTING**  
Hacking the Pi's terminal and learning basic cybersecurity skills
- 50 SINGING PIANO - IN SCRATCH!**  
Get children excited about creating programs
- 52 SET A CHALLENGE FOR LEARNERS AND TEACHERS**  
Developing computational thinking

# CODE CLUB AT AGRIHACK

Solving farming problems with a Raspberry Pi

**A**grihack 2018 brought together over 35 organisations to create seven different solutions to three tricky problems.

Code Club Australia's national programme manager Nicola Curnow explains, "We help out on the first day of the event, the kids' day. And the second day and third day is a Hackathon for adults."

"The kids visit a farm in the morning," Nicola continues, "do a design thinking challenge with the sponsor of the event, and then participate in our coding workshops."

Nicola also explains that "in Australia,



Code Club Australia brought kids and farmers together to see how tech can support food production. This year's challenge was to find sustainable ways to reuse milk containers

connectivity is a big issue, [so] almost everything was offline." The Code Club day mainly used Scratch, but "a small group of advanced kids... did a Raspberry Pi / Sense HAT program about weather."

The Hackathon challenges were: reducing stored grain spoilage from high moisture; measuring the live weight of meat chickens to prevent overfeeding; and assisting the handover of farms and fisheries from the older to the younger generation. You can see the winning solutions at [helloworld.cc/2NHdb6x](http://helloworld.cc/2NHdb6x). (HW)

# COOLEST PROJECTS UK

First Coolest Projects show in UK "mind-blowing"

**C**oolest Projects UK was "the same level of mind-blowing" as any Coolest Projects International show, says Rosa Langhammer, CoderDojo's general manager, outreach and engagement.

Held on Saturday 28 April in London, the event "had over 40 projects," says Rosa, "presented with enthusiasm and confidence by each of the project presenters... In total, we had over 500 people attend."

It's these presenters that make Coolest Projects events so special – they're CoderDojo Ninjas, aged 7–17, showing off their self-made projects.

Among some brilliant projects and presentations, Rosa particularly loved

10-year-old Ayve's 'Voice O'Tronik', which "responded to voice commands such as 'move arms' or 'roll eyes'."

Rosa was equally impressed by Ayve's "confidence and depth of knowledge" when presenting Voice O'Tronik, which was written in Python and uses the Google Cloud Speech API.

"Coolest Projects is absolutely not possible to run without project presenters (Ninjas), parents, and volunteers," Rosa acknowledges. "So to each and every person who contributed: thank you!"

Coolest Projects North America takes place in Santa Ana, California on



Hardware category winner Ayve demonstrates her creation: the Voice O'Tronik

Sunday 23 September. Head to [helloworld.cc/2wAwA2O](http://helloworld.cc/2wAwA2O) to register your project or book your ticket. (HW)



Astronaut Tim Peake using a Raspberry Pi aboard the International Space Station. Image credit: ESA (European Space Agency)

# ASTRO PI RETURNS TO THE STARS

Take your code to the International Space Station and beyond, urges Rosie Hattersley

The European Astro Pi Challenge, an ESA Education project run in collaboration with the Raspberry Pi Foundation, is gearing up for its next mission. From 12 September until 26 October, Astro Pi will be accepting applications from space fans keen to impress ISS astronauts with their ideas and coding prowess. Part of the Astro Pi Challenge, Mission Space Lab is a four-phase experiment aimed at secondary schools and older primary school pupils.

The European Astro Pi Challenge is an annual science and programming challenge where student-written programs are run on the International Space Station. Last year, it had 6,800 participants for its two missions: 1,500 for Mission Space Lab and 5,300 for Mission Zero, the majority from secondary schools. This year, the challenge aims to attract at least 7,500 participants.

## Get your code into space

The aim of the European Astro Pi Challenge is to encourage schoolchildren to conduct scientific investigations in space by writing

“**ASTRO PI WILL BE ACCEPTING APPLICATIONS FROM SPACE FANS KEEN TO IMPRESS ISS ASTRONAUTS**

computer programs that will be run on Raspberry Pi computers aboard the ISS. For Mission Space Lab, teams first submit their experiment ideas – no code is required yet.

The teams with the best ideas are then sent an Astro Pi kit including a Raspberry Pi with both an infrared camera and a Sense HAT, plus a microSD card and instructions on (optionally) 3D-printing your own case.

Once your team's code is approved, it's sent up to the International Space Station where your experiment is run on the ISS Astro Pi computers for three hours.

Participating teams receive a download of the data for their experiment and a photo of where the ISS was when their experiment was taking place. Finally, students write up a scientific report about the experiment they conducted.

The teams submitting the 10 best entries will receive a special certificate.

The Astro Pi Challenge is open to entrants from ESA member and associate member states. For more details, visit [astro-pi.org](http://astro-pi.org). (HW)



# BBC Computer Literacy Project 1980-1989

Home Programmes ▾ Explore History Timeline BBC Micro Software

Search



The Computer Literacy Project Archive brings together all of the original 146 CLP shows, plus 121-related programmes

In the 1980s, the BBC explored the world of computing in **The Computer Literacy Project**. They commissioned a home computer (the BBC Micro) and taught viewers how to program.

The Computer Literacy Project chronicled a decade of information technology and was a **milestone** in the history of computing in Britain, helping to inspire a generation of coders.

This site contains all 146 of the **original** Computer Literacy Project programmes plus 121 **related** programmes, broken down into 2,509 categorised, searchable clips.

- [Watch](#) any of the 267 programmes
- [Explore](#) 2,509 programme clips by topic or text search
- [Find out](#) how the BBC Computer Literacy Project came about
- [Run](#) 166 BBC Micro programs that were used on-screen

# BBC COMPUTER LITERACY ARCHIVE

The 1980s was a golden era for home computing. Let's explore it all over again, says Rosie Hattersley

If you've ever wanted to know more about the history of British home computing, a whole treasure trove of interviews, TV shows, programming code, and photos has been assembled for your delectation and inspiration. The BBC has launched an archive of its Computer Literacy Project (CLP).

This delightful archive of home computing and information technology ([helloworld.cc/2Py6FQ7](http://helloworld.cc/2Py6FQ7)) includes all 146 TV programmes from a 10-year endeavour that began 36 years ago.

Interviews with heroes of technology and personal computing – including Microsoft co-founder Bill Gates, Apple's

Steve Jobs and Steve Wozniak, and Apricot's Roger Foster – are all now accessible from the BBC's Computer Literacy Project Archive.

Launched back in 1982, the BBC's Computer Literacy Project was intended to inspire and encourage a whole generation of coders and home computing enthusiasts. The impetus for its launch was a critical *Horizon* documentary in 1978, which suggested a "lack of awareness and competitiveness" in the UK meant that Britain was likely to miss out on the social and economic benefits of microelectronics.

Such was the success of BBC Education's CLP and the BBC Micro

home computer (among others) that, says Hermann Hauser, co-founder of Acorn Computers, "Britain [was] the most computer literate nation on Earth at the time and with the BBC computer created a generation of UK programmers who have become leaders in their field."

The BBC Micro (top right) was a major influence on the creation of the Raspberry Pi, which was designed to capture the coding heyday of the 1980s and increase the number (and quality) of students applying to study computing at Cambridge University.

Eben Upton, Raspberry Pi co-founder, told The Centre for Computing History

([helloworld.cc/2wFP55L](http://helloworld.cc/2wFP55L)): “The first

computer I owned was a BBC Micro.”

The initial idea for the Raspberry

Pi arose from talking about redoing

the BBC Micro (as a response to MIT

planning an Apple II clone).

## Computing history

The original BBC TV shows are bound to fascinate anyone investigating the UK personal computing evolution for the first time, while evoking nostalgia in those who remember the 1980s BBC Computer Literacy Project.

The archive came about due to the difficulty for academics and technology historians to easily access treasures from the CLP. While the TV footage and interviews have been recovered, radio shows broadcast under the CLP banner appear to have been lost.

Many of the TV shows make for interesting viewing from a present-day perspective. In May 1980, *The Silicon Factor* ([helloworld.cc/2oABFU8](http://helloworld.cc/2oABFU8)) getting a paint-spraying robot in a car factory to memorise the actions involved in writing a name on a sheet of paper was ground-breaking. “The memory is recording every single unsteady movement of the pen,” explained presenter Bernard Falk before



showing the robot accurately reproducing his writing.

Meanwhile 1988's *Electric Avenue* issued rather prescient warnings about data sharing and theft: [helloworld.cc/2PyOABv](http://helloworld.cc/2PyOABv).

Mainly, however, the archive is about showing the possibilities and ideas of

digital pioneers. As with the original CLP resources, BBC chief technology and product officer Matthew Postgate says the Computer Literacy Project Archive is “a unique resource for teaching and learning that will hopefully encourage a new generation of computer users.” [\(HW\)](#)

■ The BBC Micro project was instrumental in teaching computing in British schools. (Credit: Marcin Wichary, Wikimedia Commons)



# NEW COMPUTER SCIENCE COURSE FOR IRISH SECONDARY SCHOOLS

Computer Science Teachers' Association of Ireland helps schools prepare

**T**he third element of the Irish government's computer science curriculum will be trialled in September, and the Computer Science Teachers' Association of Ireland (CSTAI) is ready with resources to help teach the new Leaving Certificate, Computer Science course.

Following the previous Junior Certificate short courses in Coding and Digital Media Literacy, the new course aims to "develop an appreciation of the diverse roles of computing technology in society and the environment in which [students] live," according to CSTAI President Stephen Murphy.

The course specification outlines that "embedded systems (such as the Raspberry Pi) will play a central role in the practical component of this subject," says Stephen.

He tells us that "over 200 schools applied for the Leaving Certificate, Computer Science pilot scheme," of which 40 were selected. The Leaving Certificate, Computer Science "will open as an elective subject for all schools to offer if they so wish" in September 2020.



The CSTAI collaborates with other organisations, including CoderDojo; here, CSTAI President Stephen Murphy meets CoderDojo's Tara McGrath and the team from the Cork Institute of Technology CoderDojo

## Supporting teachers

While the Irish government has provided "excellent" support for teachers wishing to teach any of the three courses, the

CSTAI supplements that with "hundreds of teaching resources (PowerPoints, worksheets, videos, and so on) for coding, computer science, digital media, and general computing courses," Stephen reveals.

Access to these resources is free for CSTAI members; to become a member, email [president.cstai@gmail.com](mailto:president.cstai@gmail.com), or visit [helloworld.cc/2MyIEZ](http://helloworld.cc/2MyIEZ).

The CSTAI was founded in November 2017 and has "over 440 members from Ireland, UK, New Zealand, and Dubai," says Stephen. (HW)



**Computer Science Teachers'  
Association of Ireland**



**Ríomheol Oidí na hÉireann**



The Pi Wars 2018 obstacle course drew a crowd of roboeteers and supporters

# PI WARS 2019

Applications open, challenges announced

**T**he application process for Pi Wars 2019 is now open, and this year sees the first Pi Wars theme: space exploration!

Pi Wars co-organiser Mike Horne confirms: "This is the first time we have themed Pi Wars, in honour of the 50th year since the first moon landing." The idea came from previous winner Dave Pride.

The theme means cosmetic changes to both the challenges and the venue, which remains as the William Gates Building of the Cambridge Computer Laboratory. "Lots of painting of the courses is going to occur over the summer," Mike tells us.

## Challenger

The space theme introduces new challenges, but Mike says that "the rules

are pretty similar to before, with a few tightened up and few loosened."

For example, the new Hubble Telescope Challenge is based on 2018's Over the Rainbow, where robots needed to identify coloured balls and drive to them in sequence. "This was the hardest course by a long way," Mike reveals. "We will be making the targets larger, but have not yet finalised that."

Space Invaders is based on the Duck Hunt challenge, "with the same skills required to score." The Spirit of Curiosity challenge involves driving your robot over an obstacle course, collecting a sample (which "will vary in size, but not by much") and driving back.

## Re-entry

Pi Wars 2018 received "over 150 entrants," Mike confirms, "which had to be cut down to just 76, plus reserves." This involves a "very hard couple of days", as Mike tells us that "we cannot just choose previous entrants as we want to make space for noobs."

Mike has this advice for your application form: "Ensure the entrants show their enthusiasm [and] we like detail."

Pi Wars 2019 will take place on 30 and 31 March, so head over to [piwars.org](http://piwars.org) to read about rules, challenges, and to apply. Keep up to date with news via the mailing list or Twitter @PiWarsRobotics. (HW)



The students from Ysgol Deganwy school receiving their £1,000 prize for winning the Primary School Award category

# SUSTAINABILITY COMPETITION WINNERS ANNOUNCED

PA Consulting's annual Raspberry Pi challenge round-up

**T**he winners of PA Consulting Group's annual Raspberry Pi Competition were announced at the grand final in late April, with three schools each winning £1,000.

This year's theme was sustainability: 'to invent something that will help save the planet'. With more than 100 schools applying, there was plenty for the judges – including BBC's Rory Cellan-Jones – to consider.

PA Consulting's Raspberry Pi Competition co-ordinator John O'Neill explains that the projects "are aligned to curriculum activities, so they can help [students] achieve educational goals" as well as being "collaborative and fun".

For example, students of Ysgol Deganwy school, Conwy, made 'Recycle Michael', which reads the barcode of

a piece of rubbish and tells you which recycling bin to place it in.

John confirms that the judges "could imagine Recycle Michael being scaled up to appear in offices and home across the country."

Next year's Raspberry Pi Competition will be revealed in September 2018 – you can register your interest now at [helloworld.cc/2PvpyU0](http://helloworld.cc/2PvpyU0). (HW)

# INTRODUCING ETHICS

As new technologies impact on our lives, we must be equipped to answer new moral questions

STORY BY Peter Millican

**C**ooperation has always been key to our success as a species, as witnessed by our language skills and our large brains that evolved (perhaps primarily) to negotiate a complex social landscape. Ethics is in turn crucial to cooperation, oiling social wheels and avoiding the strife and inefficiency of amoral competition. As our societies grew and developed, we had to move beyond intuitive tribal ethics to face new questions and find novel answers – for example, concerning the artificial institution of property rights – that have in turn become part of our common-sense morality.

Technological changes played a significant role in these developments, but in recent history the most prominent of these (for example, nuclear weapons, innovative medicine) have impacted only at the margin, in extreme contexts of life and death. Computing, by contrast, is now ubiquitous in our lives, mediating our social, employment, economic, educational, informational, and even political relationships, and potentially disrupting all of these (from online bullying, to mass unemployment, to the death of the high street, to plagiarism, fake news, and voter manipulation). AI even impacts on our view of ourselves, challenging our supremacy as rational actors. Yet it also offers hope of a wonderful new ally in solving our complex problems, if only we can learn to harness it safely.

This hope inspired the development in Oxford of our new degree programme in Computer Science and Philosophy (which started in 2012). When designing the curriculum, some choices of appropriate “core” Philosophy courses were obvious: Philosophy of Mind, Cognitive Science, Logic and Language. Others required more deliberation, and we devised a fairly long list to enhance students’ flexibility, but Ethics wasn’t included. Over the intervening years, I have given hundreds of outreach talks

robotics have become regular staples of our press and other media. Ethical issues are now amongst the first to come to mind in discussions about Computing and Philosophy, and prompt many questions at our outreach events. Within academia too, the perspective has changed, and the relevant committee was entirely in accord when, a couple of years ago, Ethics was added as a “core” course to our new degree. Increasingly, education at all levels is appreciating how fundamentally new technologies impact on our lives, raising new

**“ETHICAL ISSUES ARE NOW AMONGST THE FIRST TO COME TO MIND IN DISCUSSIONS ABOUT COMPUTING AND PHILOSOPHY”**

to raise awareness of the degree, always citing Logic, AI, and Ethics as the clearest areas of intersection between the disciplines (amongst many others). Initially, audiences would take the first two as indeed obvious, but making the case for Ethics seemed harder, because security, privacy, and other related issues were far less prominent then, and the idea of robots in our lives seemed, to many, implausibly futuristic.

Over the last few years, however, all this has changed radically, signalling a massive change in public attitudes as the various impacts of big data, social media, and

moral questions which will require innovative answers, and which future citizens must therefore be equipped to address. This need is already obvious to many, and I predict that it will very soon become increasingly obvious to everyone. (HW)

**Peter** is Gilbert Ryle Fellow and Professor of Philosophy at Hertford College, University of Oxford. He is also a member of the Computer Science Faculty, and has developed several free software systems to introduce students to programming, AI and digital humanities.

# ETHICS IN THE COMPUTING CURRICULUM?

There are many ethical issues around digital technology that teachers could explore in the computing classroom, but how do we introduce them to our pupils?

STORY BY Miles Berry

**W**hen we talk about online safety with our trainee teachers at Roehampton, I start by asking my students what sort of qualities they'd like to see in their pupils. We get some great answers, covering things like kindness, courage, self-confidence, curiosity, courtesy, integrity, fairness, and diligence.

It's hard to argue against any of these, but, on the other hand, it's far from clear how we might go about developing these qualities through the taught curriculum in general, and computing lessons in particular. Nevertheless, I'm convinced that if we can get character education right, then so much of what worries us about online safety gets addressed along the way: if young people are honest, they won't lie about their age to get social

media accounts; if young people are kind, they won't bully one another online; if young people have courage, they're perhaps less vulnerable to online grooming.

The English computing curriculum places a lot of emphasis on personal morality, but has little to say about the broader sphere of ethical issues around digital technology. This wasn't the intention of the BCS/Royal Academy of Engineering-led drafting group, which included as an aim for computing education that pupils would: "Develop awareness of the individual and societal opportunities, challenges and risks raised by digital technology, and know how to maximise opportunities and manage risks appropriately."

At the time, ministers decided that we didn't need the ethics bits of the draft

programmes of study, and that pupils would be better prepared for the opportunities, roles, and responsibilities of life through learning about binary arithmetic and Boolean logic. Four years on, the House of Lords AI select committee now recommends "that the ethical design and use of technology becomes an integral part of the curriculum". Quite.

Thankfully, the US CS K-12 framework and their implementation in CSTA's K12 CS standards avoided this sort of short-sighted political interference: fostering an inclusive computing culture is one of the underpinning practices in the former, and the latter has 22 standards specifically addressing the wider impact of computing.

US psychologist Lawrence Kohlberg worked on the stages of children's moral development, seeing progress from an orientation to

obedience and avoidance of punishment and self-interest, via authority and social contracts, to one based on universal ethical principles. If we're to take children's moral development seriously, then perhaps it's worth stepping beyond safety, responsibility, and legality to consider broader ethical principles and practices. Without this broader focus in computing education, it's arguable whether we'll have properly prepared our pupils for a world in which technology seems likely to play an even more dominant role than it does today.

There are many ethical issues around digital technology that teachers and pupils might explore together in the computing classroom. Here are three that I think could make good starting points for pupils' independent research and a reasoned debate between those willing to take different perspectives:

**Reliance on technology:** have we as a society in general, or perhaps young people in particular, become too reliant on digital technology? In what ways are lives better as a result? In what ways have they got worse? Have we consciously chosen to allow technology into our lives in this way, or have we been cynically manipulated by big businesses, motivated by profit? Are social media or gaming harmful addictions?

**Surveillance:** Is there a right to privacy in the digital age? How much personal information is it appropriate to share with those outside our circles of trust? How much information does your school, internet service provider or government have about you? Under what circumstances is it right for schools, service providers, and governments to monitor use of the internet? Is it ever right for individuals to circumvent this monitoring?

**Rules for AI:** As machine learning impacts more aspects of our lives, what ethical safeguards should society build in, if any? If so, is this a price worth paying? How can bias or prejudice in algorithms and training data be reduced or eliminated? The GDPR demands that humans be kept 'in the loop' for decisions that have significant effect on human beings: is it right to do so? How should an AI make ethical decisions? What rules and principles should an AI be programmed to apply?

Beyond the specific details of these topics, there's a case for providing young people with a framework to think ethical problems through for themselves. Both the BCS and ACM have codes of ethics for those working in computing, and an analysis of these might provide some insight into the underpinning principles: acting for the benefit of society, avoiding harm, equality, honesty, respect for the law.

The ethical implications of big data and AI are already huge, and as a society I think we've a responsibility to think these through together and establish the frameworks which govern these: the GDPR is a serious attempt to do this for big data. Beyond this, I think we also have a responsibility to help the next generation wrestle with the as yet unimagined issues they'll face



together: teaching ethics in the computing curriculum is one way of ensuring that we do.

In the rest of this issue's cover features, UCL's Adrian Mee argues that ethics should be a theme that runs through the computing curriculum, and hints that computational thinking might help students think through ethical issues. Primary teacher and online safety expert Matt Lovegrove explores the links between children's safety and responsibility, and the broader realm of technological ethics. Los Angeles teacher Vicky Sedgwick gives some practical suggestions for introducing ethics in primary schools, including discussing privacy through school password policies and Google Forms to programming self-driving cars in Scratch! New York teacher and #ethicalCS organiser Saber Khan introduces a card game to get students thinking about privacy and data. To conclude, the University of Edinburgh's Ben Williamson considers the ethical issues of big data and AI in education. (HW)



## RESOURCES

BCS code of conduct: [helloworld.cc/2wtqS1B](http://helloworld.cc/2wtqS1B)

ACM Code of Ethics and Professional Conduct: [helloworld.cc/2weYvoO](http://helloworld.cc/2weYvoO)

European Group on Ethics in Science and New Technologies statement on AI: [helloworld.cc/2PHIKQs](http://helloworld.cc/2PHIKQs)

Report on European civil law rules in robotics: [helloworld.cc/2ltU3XY](http://helloworld.cc/2ltU3XY)

# DOING THE RIGHT THING IN A DIGITAL WORLD

The education of 'good citizens' has been a core curriculum objective for centuries. Here, we take a look at how computing might make a contribution

STORY BY Adrian Mee

## Computing and ethics in schools

A primary function of schooling continues to be the development of good citizens who can contribute to governance in a democratic society. As society became ever more challenged by the opportunities and risks posed by scientific discoveries, science teachers took on the challenge of constructing a curriculum for 'citizen science'. While this curriculum aims to support the development of scientists and engineers, its primary function is to ensure we have a scientifically literate population capable of making scientifically informed choices on a range of social and ethical issues. Pupils learning physics are required to develop their understanding of 'hard science' and then apply this in the context of debates about nuclear energy, global warming, and so on.

In computing, we've still got a considerable way to go to develop

'citizen computer science'. At a time when AI, big data, surveillance, and digital addiction are constantly in the press, our computing curriculum largely ignores the ethical and social implications of these developments. Where it is included, it's often banished to 'Chapter 10' of the text book, and assessment is frequently little more than listing the key points of data protection legislation.

Just as with the rise of 'citizen science', there are those who wish to avoid ethical issues to concentrate on 'pure computer science' and, in some cases, these voices offer that it's not their role to 'indoctrinate' pupils. Both arguments are underpinned

by profound misunderstandings about what school is for, what 'teaching ethics' means in this context, and the relationship between society and technology.

## Developing the 'ethical dimension'

In developing an ethical dimension to a school curriculum, three issues might be usefully addressed.

Firstly, the ethical and social dimension should not be a 'topic' or a 'unit' but a theme that runs through and complements the computing content, rather like literacy and numeracy.

Secondly, teaching ethics is not about indoctrinating pupils with 'you must

**"TEACHING ETHICS IS NOT ABOUT INDOCTRINATING PUPILS WITH 'YOU MUST ALWAYS' AND 'YOU SHOULD NEVER'"**

## ETHICAL DILEMMAS TO SUPPORT CLASSROOM DISCUSSION

Ethics in the context of computing is about encouraging pupils to analyse problems logically, spot patterns and contradictions, and apply their computational knowledge to suggest solutions that may be technical, social or legal.

You may like to try offering pupils an opportunity to discuss the following, which embody some ancient ethical dilemmas recontextualised in a digital world:

### PROBLEM 1

Your local newsagent might expect a visit from the police if she allowed people to put postcards in her shop window advertising stolen goods. Could she legitimately claim that "my window is just a platform and I'm not responsible for what people place there"?

### PROBLEM 2

Producing a 'compelling and exciting' computer game may be a satisfying and lucrative way for you to use your new coding skills. What if 'compelling and exciting' for some users begins to emerge as addiction? Is this, then, the responsibility of the programmer, the company they work for or the user?

### PROBLEM 3

Some French schools are using digital 'key-fobs' to track and monitor pupils as they move around to help with efficient running of the school and security. Is there a balance to be struck between the need for 'security', 'efficiency', and liberty?

## FURTHER READING ON SOCIAL AND ETHICAL ISSUES

A short bibliography of easy readers :

■ *Radical technologies: the design of everyday life* Adam Greenfield  
ISBN: 978-1-78478-045-6

■ *Irresistible - why you are addicted to technology and how you can set yourself free* Adam Alter ISBN: 978-1784701659

■ *The people vs technology: how the internet is killing democracy (and how we can save it)* Jamie Bartlett ISBN 978-1-78503-906-5

■ *The new digital age: reshaping the future of people, nations and business*  
Eric Schmidt & Jared Cohen  
ISBN 978-1-84854-620-2

Adrian is currently a teacher educator at UCL. His research interests encompass the philosophy of technology and ethics in a digital age.

## “ THE DIGITAL ENVIRONMENT IS NUANCED AND COMPLEX, AND REQUIRES PUPILS TO BE ABLE TO THINK CRITICALLY

'always' and 'you should never'. The digital environment is nuanced and complex, and requires pupils to be able to think critically. Thinking ethically requires us to decompose a situation, identify what is important and what is peripheral, define relationships between one problem element and the next, and construct a solution that is 'the best case' through compromise and consensus. These skills may seem familiar! We're not teaching what to think...but how to think!

Finally, we don't need to stop teaching computer science to teach ethics. Thinking of learning taxonomies, the 'top of the pyramid' requires us to 'analyse, evaluate and create'. So many ethical issues in

computing require the exercise and application of computer science knowledge to construct an informed argument. Where computer science is introduced in the grounded context of issues and problems young people can relate to, we may find a solution to the issue of rejection of the subject by some groups of pupils.

The synergy between ethics and computer science has the capacity to encourage pupils to see the social and personal relevance of what can sometimes seem abstract and detached and to develop pupil's critical awareness of the consequences of the technologies they may go on to create and the ones they'll certainly have to live with. (HW)

# WHY WE NEED TO TALK ABOUT ETHICS IN TECHNOLOGY

How should we teach future generations to adopt an ethical approach to using technology?

STORY BY Matt Lovegrove

**N**ever has there been a more important time to discuss the ethical use of technologies with young people. The impact technology has on our world is colossal and, due to this, we're only just starting to catch up with ourselves.

## What is technological ethics?

Ethics in technology, simply put, refers to moral principles that govern how technologies should be used. These principles include accountability, digital rights, privacy, freedom, data protection, online behaviour, and more. As new technologies become available, ethical issues are raised in an attempt to ensure that these have a positive impact on the human race and the environment.

## Why talk about it?

Last year, the Institute for the Future (IFTF) estimated that 85% of jobs that will exist in 2030 haven't yet been invented. With the implementation of new technology comes an ethical duty, and it's important that the generation who'll be taking up these future jobs have a sound understanding of their ethical responsibilities.

Take hacking, for instance. Hacking has both negative and positive meanings, in that accessing a computer system without

authorisation might be illegal, but it's also a term used for remixing or testing new ways of using computers or programs. It's important that young people have a good understanding of how and when it's acceptable to hack, and when and where the line is crossed.

And then there's copyright. It's easy to download and copy information from the Internet, but is this acceptable behaviour, does it breach someone's rights, and how should someone respond if their work is directly copied or plagiarised? Teaching children about copyright laws and respecting others' work is imperative in a world where more and more content is being made available online.

And what about privacy? With the internet being inherently open, we need to teach young people about their rights and responsibilities within this area: their right to maintain privacy and their responsibility to treat others' information with care. When is it acceptable to share data about others and what does the law say about how we should store this data? A solid understanding of privacy and data protection could help future generations avoid issues currently being experienced by social media companies.

Online behaviour is an area that is particularly important for young people to

## RESPONDING TO CYBER BULLYING

Four easy-to-remember steps can help young people respond to cyber bullying:

- **Don't respond**, as responding can fuel the fire
- **Save the evidence**: this might be needed later
- **Block or report the bully**
- **Seek advice from a parent/carer, teacher or another trusted adult**

spend time reflecting upon, especially as social media technologies are being adopted by a large percentage of children from a very young age. How should people behave online, what is responsible communication, why is it important to respect others online, what are the risks of being able to chat to anyone, and how should cyber bullying be responded to? These questions open up incredibly interesting discussions with children and help promote responsible use of technologies.

## How should we talk about it?

Discussions based around open ethical questions and debates around rights and responsibilities are good avenues to take when introducing technological ethics to young people. Looking at particular case studies, or examples of when ethical considerations weren't taken, might be a useful context to introduce this area of learning. **(HW)**

# TEACHING 5- TO 14-YEAR-OLDS ABOUT ETHICAL COMPUTING

What do self-driving cars have to do with computing for 5- to 14-year-olds? Read on and find out!

STORY BY Vicky Sedgwick

**W**hat should we teach 5- to 14-year-olds about ethical computing? I always start by unpacking the standards that my students need to meet and identifying those that deal with ethics. I also take direction from the ACM Code of Ethics and Professional Conduct ([helloworld.cc/2MDhaow](http://helloworld.cc/2MDhaow)). While my students aren't computing professionals, they're creating with computing and can and should respect privacy, respect the work required to produce computing artefacts, and avoid harm.

## Start with privacy

Some ethical policies may be predetermined by a school's Acceptable Use Policy or Code of Conduct. For example, schools often assign passwords and have a policy that states that passwords shouldn't be shared. This doesn't mean that students understand why we have passwords, why it's important not to share them, and what this looks like in class. This is where teaching ethics in our classrooms comes in. We start the year with lessons emphasising privacy, security, and

ownership. For my youngest students, this might mean unlocking real locks and singing and dancing along to a Password Rap from Netsmartz Kids ([helloworld.cc/2BPifoc](http://helloworld.cc/2BPifoc)). My older students review what makes a password strong, then they create their passwords for the school year.

My pre-teen and teen students love all those online quizzes, but they often don't realise that the quizzes are collecting data about them based on their answers. Having students create one of those quizzes themselves using Code.org's App Lab ([helloworld.cc/2ijEfsh](http://helloworld.cc/2ijEfsh)), Scratch or Google Forms and requiring that they craft and publish a privacy policy detailing how the data from their quiz will be used can be an eye-opener for them.

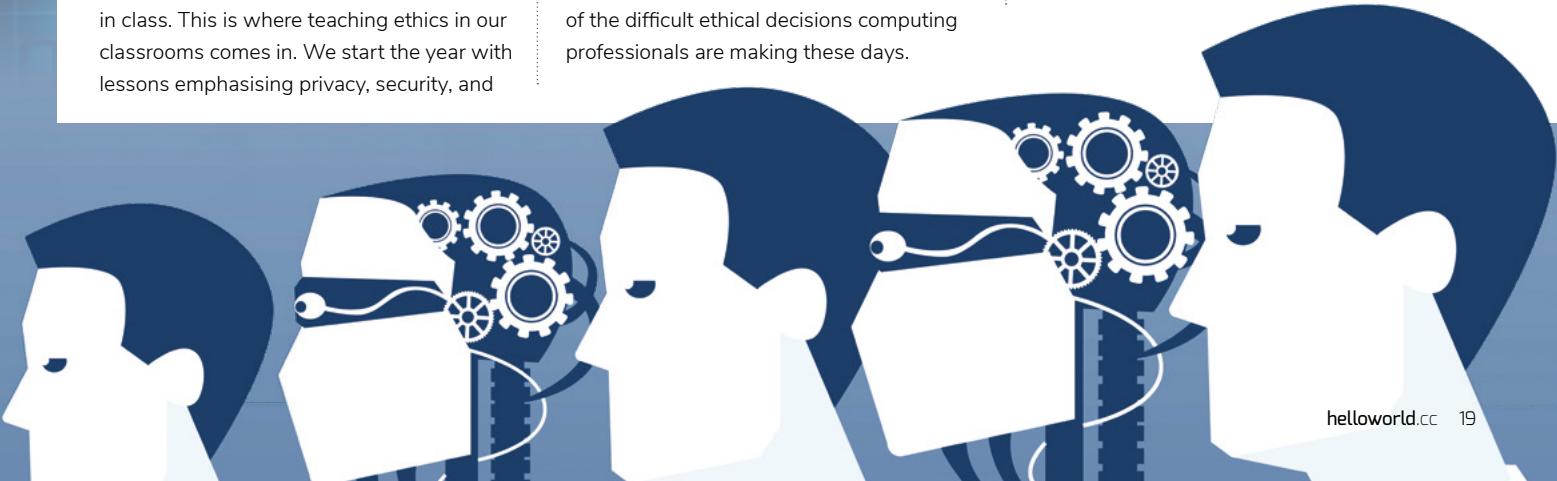
## Engage with current events

Socially relevant topics in the news help spark discussions, and going beyond just a discussion leads to deeper understanding of the difficult ethical decisions computing professionals are making these days.

Discussions about whether it's possible to program a self-driving car to make better decisions than human drivers are even more powerful when students try to create algorithms for an autonomous vehicle. My students aren't going to build an actual self-driving car, but they can create a simulation of one in Scratch or program a robot with line-tracking and distance sensors to navigate a course that includes some hazards to avoid.

We know that autonomous vehicles are based on machine learning, but that may be a new concept for our students. We can introduce machine learning in a hands-on way with The Machine Learning for Kids website ([helloworld.cc/2MDhaow](http://helloworld.cc/2MDhaow)). Students can train machine learning systems and then build Scratch projects around the systems they've trained.

How are you incorporating ethics into your computer science classes for 5- to 14-year-olds? [\(HW\)](#)



# #ETHICALCS - THINKING ABOUT ETHICS, IDENTITY, AND POWER IN COMPUTING CLASSROOMS

Join a growing movement to engage in conversation, share resources, and build lessons for an ethically minded computer science education

STORY BY Saber Khan

**M**ost professional fields have a guild, a clear course of study taught at colleges and universities, and lessons and courses in ethics for students and professionals. Educators can borrow and adapt these curricula into their classroom. But for technology and computing educators, that task is more difficult. The age of the industry, rapid changes, and the secrecy it maintains have all made it hard to build ethical lessons about tech.

Those of us engaged in K-12 CS education have a responsibility to provide our students with ways to engage with these problems and develop their own ethical framework. Not just so that they can react to news, but so they can engage as citizens in a political system that seeks accountability and equity.

Jeannie Crowley and Kenny Graves of Ethical Culture Fieldston School have created an activity that can help students and teachers engage in ethical thinking about technology and computing. The data, privacy, identity card game is an opportunity for students to think about data and how it's used online to identify people, and how different identities may view their privacy differently.

In the game, students are placed in a mixed group and given a bag of cards with different pieces of online data, such as webcam photos or comments on YouTube or Google Search history. They're then asked to sort the cards into five categories differentiated by levels of privacy, from



Sharing the #ethicalCS project with attendees at Mozfest 2017

'data not collected or stored' to 'collected/stored and public'. Sorting the data points leads to discussions among the groups about their personal views on privacy.

The activity is then repeated, but the students are given different identities, such as white teenager in the US, Black Lives Matter activist, or tech company CEO. As the students repeat the activity, they try to take on the perspective of their new identities. Those given ones that are marginalised in society find themselves wanting maximum control over their privacy. Those with identities that profit from commerce on the internet seek less privacy. Discussions among the students often evolve to talking about the advertising and surveillance that make the internet 'free'. Students are now primed to ask more critical questions as they

study CS and imagine a more equitable system in the future.

This is one of the first activities created by a group of educators who are working to create a space where we can share, collaborate, hear from experts, and build momentum for ethically minded CS education. The New York City Department of Education CS4All team was inspired to launch a Twitter chat after training teachers and conversations with industry leaders. Last summer, after Computer Science Teachers Association conference in Baltimore MD, we started the #ethicalCS Twitter edchat. These chats bring together experts and educators on a monthly basis to talk about different aspects of ethics and computing. You can find the activity mentioned and other resources at [ethicalCS.org](http://ethicalCS.org). (HW)



# THE EDUCATIONAL DATA REVOLUTION RAISES BIG ETHICAL CHALLENGES

AI will be able to data-mine student performance and 'personalise' education continuously, but should robo-teachers be allowed in the classroom?

STORY BY Ben Williamson

**R**obots will replace teachers within a decade. That is the prediction of Sir Anthony Seldon, former head of Wellington College and current Vice Chancellor of the University of Buckingham. Many teachers probably don't expect their job to be robotised any time soon. Seldon, though, forecasts artificial intelligence machines that will analyse data from the classroom and then 'personalise' the educational experience around every individual student.

He isn't alone. Companies are already building data-crunching 'learning analytics' that can adapt to the unique performance, progress, and problems faced by learners. ClassAlexa uses Amazon's automated voice assistant as a prototypical robo-teacher. Education business Pearson predicts future teachers working with AI teaching assistants.

These won't be android teachers out to take over the teaching profession. They'll be robot co-workers with which human teachers will need to collaborate, and they'll need access to huge volumes of student information – data that is both 'big' and 'intimate'.

## Big and intimate data

Collecting big data means huge numbers of students will need monitoring continuously

across a variety of tasks. Computer-adaptive testing is an existing example of a technology that compares an individual's performance with a huge database of other students to determine the difficulty of the test. Adaptive learning analytics platforms are the next iteration, constantly analysing each student's keystrokes, mouse clicks, task performance, and progress to personalise the next task.

AI will also collect more intimate data. 'Emotional AI' uses facial recognition and wearable sensors to detect student engagement, attention, anxiety, and confusion. According to designers of 'emotional learning analytics', these data can provide insights into the emotional states that affect student learning. Together, big and intimate data will provide robo-teachers with the AI to personalise education.

## Big ethical challenges

Powerful companies, influential educators, and researchers seem to believe the robot future of education is coming, raising enormous ethical challenges for teachers:

**Privacy and security** Data protection laws mean access to student data by robo-teachers will require full informed consent

from students and their parents. With rising public awareness of personal data misuse, some may be reluctant.

**Surveillance** Constant classroom surveillance may create an atmosphere of anxiety, with students nervous that every stroke of the keyboard, facial movement, or brainwave adds to a numerical profile of their performance.

**Data quality** Whether the data is good enough will always depend on what the designers programmed their platform to collect. If the data is bad, and has negative consequences for students, then who – or what – is accountable? Does the human teacher carry the blame if the robot co-worker makes a bad decision?

**Over-simplification** The dream of AI classroom assistants assumes the data are objective accounts of a student. But the complexity of classrooms can't be reduced to simplified numerical scores or calculations.

Robo-teachers may have merits, but as the AI future unfolds, these ethical challenges will require urgent attention. **HW**

## #INSIGHTS

# TRACER REPORT

STORY BY Jonathan Dickins



A recent report on computing education in the UK from the University of Roehampton has provided some fascinating insights into how the uptake of Computer Science GCSE and A Level qualifications has grown, and into the characteristics of the students who choose to take the subject. The TRACER

report, which explored DfE statistics on education providers and pupils from 2014 to 2017, found that while participation in Computer Science qualifications continues to grow, there are certain demographic groups that are still less likely to choose the subject. The report also warns that the removal of ICT qualifications will not only

result in fewer students studying computing at this level, but will disproportionately affect girls, who are more likely to study ICT than Computer Science.

#### Growth in availability

One headline finding from the report is that students seeking to take Computer Science

are far more likely to find it on offer at their school than they were in 2014, as the subject continues to be rolled out across more and more education providers. At GCSE, Computer Science was available in 52.5% of schools in 2017, up from below 20% in 2014, with comprehensives and grammar schools more likely to be offering the qualification than independent schools. More than three-quarters (76.3%) of all GCSE students are in schools where the Computer Science GCSE is taught, suggesting opportunities for students to study Computer Science have greatly increased. Growth at A Level has been somewhat slower, with 36.2% of schools offering the qualification, versus around 20% in 2014.

### Girls remain underrepresented

Computer Science students are overwhelmingly more likely to be male. At GCSE, 20% of total entries come from girls, with this figure dropping even lower to 10% at A Level. Differences are reported in the proportions of boys and girls studying Computer Science across local authorities, but there are 382 mixed schools across the country where the entire GCSE Computer Science cohort is male, and 25 A Level local authorities with no female participation at all.

The girls that enter at GCSE are, however, more likely to pass, with 64% achieving an A\*-C grade compared to 58.4% of boys. At A Level, boys are now outperforming girls at A\* and A grades, a reverse from 2015.

With the removal of GCSE ICT in 2018, alongside the news last year that the ECDL (an IT skills qualification) is to be dropped from Progress 8 from 2019, the number of girls studying computing in total looks likely to fall this year. Compared to Computer Science, the proportions of boys and girls taking these two qualifications was far more balanced: the ECDL and GCSE ICT cohorts were 45.3% and 38% female in 2017. The report warns that the removal of these two qualifications, while heavily impacting the total number of GCSE students studying computing in some form, will also lead to a much

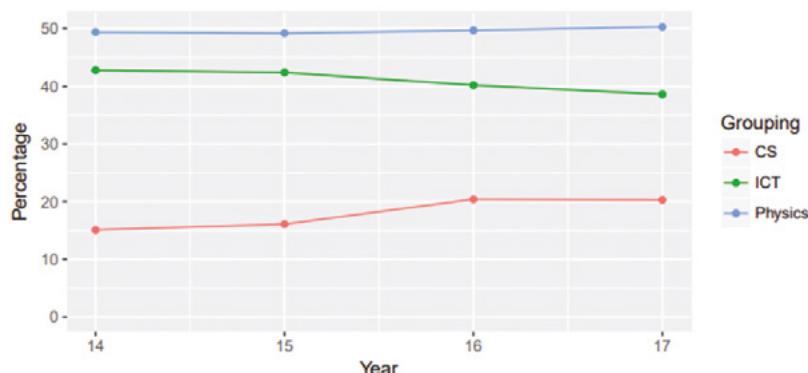


Figure 65: Longitudinal: GCSE female uptake by subject. 2014-17

greater gender imbalance in computing education, effectively leaving Computer Science as the only computing option open to many students.

### Student characteristics

The researchers also took a look at the characteristics of the students who choose to study Computer Science, and whether certain ethnic groups or students taking other specific subjects were more or less likely to be taking Computer Science.

grade. It may be that the subject appeals more to mathematically able students.

- Computer Science students are more likely to be taking Physics, Chemistry, and Maths, than subjects such as History and Geography. Interestingly, this differs from ICT students, who are more likely to take Business Studies, Film/Media, and Design & Technology.

## “ COMPUTER SCIENCE STUDENTS ARE MORE LIKELY TO BE TAKING PHYSICS, CHEMISTRY, AND MATHS

The proportion of students who actually take the subject remains relatively low at 11.9% at GCSE, and just 2.7% at A Level, but some interesting common themes emerged:

- Ethnicity patterns among Computer Science students are different to the patterns among the overall student population, with some groups more and some less likely to take Computer Science. Black and mixed race students are less likely to choose the subject, whereas Asian and Chinese students appear more likely.
- At GCSE and A Level, the typical Computer Science student is likely to also have a relatively high GCSE maths

These imbalances will be of concern to educators who are seeking to improve diversity in computing education. The demographic imbalances in Computer Science may reflect student attitudes that affect whether they choose to study the subject. A perception that a high ability in Maths is essential, or that Computer Science is a difficult subject, for example, may lead to students picking subjects that require less underlying knowledge or that they perceive as easier. With the options for studying computing in school reducing, educators may need to consider how Computer Science is perceived by their students and how they can sell the subject.

You can read the full TRACER report here: [www.bcs.org/category/19331](http://www.bcs.org/category/19331) [HW]

# DIGITAL SKILLS: WHICH MATTER FOR WORK?

STORY BY Oliver Quinlan

**I**t increasingly looks like digital skills are vitally important in the future of work, but which skills are the most vital? Researchers at Nesta have been looking into the skills that are likely to be needed in the workplaces of the future and recently turned their attention to digital skills.

They found that the types of digital skills most likely to be in demand are those involved in non-routine tasks, problem solving, and creating digital assets. This is similar to their earlier findings about the risk of automation for certain jobs; it's the non-routine and creative work that is much less likely to be automated and therefore to be more in demand.

Example skills given as promising for the future include highly creative ones, with elements of artistic skills such as animation and multimedia production. They also include design in engineering, involving tackling problems and coming up with creative solutions to them using technology. The increasing digital activity of work means that the skills of building and maintaining IT systems and networks are predicted to be some of the



**“IT'S NOT JUST ANY DIGITAL SKILLS THAT ARE IMPORTANT, BUT CERTAIN SETS OF THEM”**

## NESTA'S FIVE MOST PROMISING DIGITAL SKILLS

1. Animation
2. Multimedia production
3. Design in engineering
4. Building and maintaining IT systems and networks
5. Research and quantitative data analysis

most in demand. Rounding off the top five most promising skills are research and quantitative data analysis skills. As we amass more and more data about our work, we will need to understand it, and making sense of huge datasets requires non-routine analysis and creative approaches.

These predictions were formed from analysis of the skills asked for in vast numbers of online job adverts. Forty-one million were scraped to form the dataset that led to these conclusions. One of the key findings is that the future of work is not just about digital skills. In fact, some of

the occupations that are not categorised as needing a high degree of digital skills, such as chefs and teachers, are thought likely to grow in the next 10 years. Some of the jobs with poor prospects for the future are very likely to need digital skills, which shows that it's not just any digital skills that are important, but certain sets of them. Those young people developing digital skills involved in non-routine tasks, problem solving, and creating digital assets are predicted to be much better prepared for the future of work.

Read more about this research here: [helloworld.cc/2xs74fK](http://helloworld.cc/2xs74fK) (HW)



# DISCOVERING STUDENTS' UNDERLYING COMPUTATIONAL SKILLS

STORY BY Oliver Quinlan

The exact definition of computational thinking is hotly debated. However, it's more widely accepted that some kind of computational problem-solving skills are important to succeed in computing. Researchers from three universities in Madrid have been working for some time on a Computational Thinking test to assess this ability. Román-González, Pérez-González, Moreno-León, and Robles have found that their test can identify different levels of computational talent, even before learning any programming.

In their latest paper, they've demonstrated that this test is able to detect students who are 'computationally talented' before they have any programming experience. This could allow talented students to be identified and appropriately challenged in the classroom. It also raises the possibility of identifying the computational talent level of all students and using this to put in place the teaching needed for all to succeed.

The Computational Thinking test (CTt) is a 45-minute multiple-choice quiz.

Students either choose appropriate instructions to navigate a PacMan out of a maze, or answer a question about a piece of block-based code. Computational concepts based on the CSTA Computer Science Standards are gradually introduced. For this project, they also looked at the academic grades students achieved in school, and learning analytics about their activity using Code.org and Khan Academy.

would start learning programming on a relatively level playing field. This research suggests that is not the case. Different students are likely to start with different capacities in terms of the underlying thinking skills that programming builds on. It's worth educators exploring these underlying skills to support the success of all students, as it seems likely that even from the start different students will need different levels of support.

## THE TEST CAN IDENTIFY DIFFERENT LEVELS OF COMPUTATIONAL TALENT, EVEN BEFORE LEARNING PROGRAMMING

Students took the CTt before starting to learn programming, and their results for this test were found to predict their grades and their success in the online programming courses. There were large differences in the computational ability of students even before they learn to code. It might be logical to assume that students

The CTt is still the subject of research, but this work suggests that there are underlying skills that computing educators should consider when designing the learning for their students.

You can read more about this research here: [helloworld.cc/2xitj8w](http://helloworld.cc/2xitj8w) (HW)

# GAMES-BASED LEARNING FOR COMPUTATIONAL THINKING

Games can be a great vehicle for improving computational thinking skills.

So why not let learners play in the classroom?

STORY BY Mia Chapman

In 2011, Nesta published a report that arguably catalysed the transformation of the primary computing curriculum in England, by highlighting the skills gap faced by the UK gaming industry and calling for a more rigorous computer science curriculum. Seven years on, programming is a fixture in primary schools, and many young people in classrooms and Code Clubs across the country are using applications such as Scratch and Kodu to make their very own games. At the same time, games-based learning is gaining traction as a means of teaching a range of curriculum subjects, yet the amount of time young people spend playing video games

and the types of games they play continues to attract media criticism.

So, if young people are making games in schools or clubs and playing games at home, why don't we bring the games they're playing into the computing classroom? Aside from providing a meaningful context for learning, solving the problems we face in a game world often requires skills comparable to computational thinking. Looking for patterns, identifying important information, breaking down problems, and finding solutions are key features of many good games. But which games are popular in the current market? And how can we ensure that they're a valuable use of time?

## What is computational thinking?

Computational thinking is often misunderstood as 'thinking like a computer'. However, it helps us to solve problems in a way that can be carried out effectively by both a human or a computer. The primary computing curriculum states the necessity of computational thinking for pupils to understand and change the world. To do this they must understand the four key concepts involved (Figure 1). Although we talk about computational thinking in direct relation to computing, the strategies pupils develop are beneficial to solving all kinds of problems with or without a computer, making computational thinking vital to the whole curriculum and life beyond school, including when playing games.

## Why games-based learning?

Games are a great vehicle for helping learners to solve problems in an environment that is safe and has minimal risk. This can be useful in many curriculum subjects to explore difficult topics or complex concepts. In his book, *What Video Games Have to Teach us About Learning and Literacy*, James Gee advocates the use of games to enhance learning due to the strong learning principles at play when we learn how to play a video game. He likens the learning principles in good games to learning theories of cognitive science, arguing that games offer an active, motivating, and challenging environment to learn in. Often, like real life, games require the use of strategies comparable to those we use for computational thinking.

Also, making mistakes and debugging are important parts of programming. Challenge and failure are central to games and, as

## TRY SOME OF THESE



### The Room

A mystery puzzle game requiring decomposition, abstraction, and logical thinking.

### Monument Valley

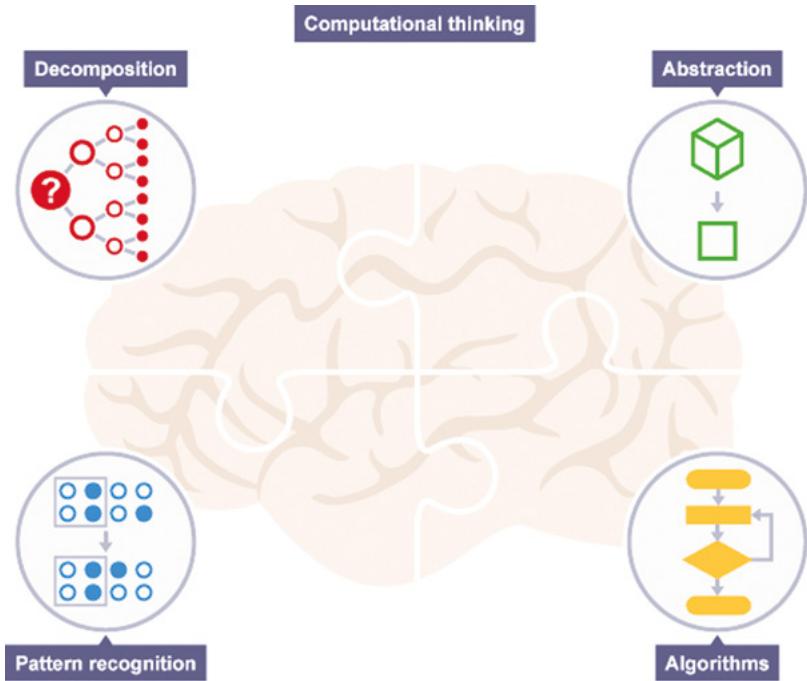
Another puzzler involving mazes and illusions, again making use of decomposition and finding step-by-step solutions.

### Scribblenauts

An action puzzler where players battle in various minigame problems, allowing them to practise computational thinking skills.

### Rollercoaster Tycoon

A simulation game requiring strategy and careful planning.



■ Figure 1: The four key concepts used in computational thinking

players, we expect a challenge to get the most out of it. Often, we find that students are reluctant to fail, yet are motivated to continue playing a game despite failure. Use these experiences to open a discussion on failure in different aspects of life. Ask pupils how they respond when they can't complete a level in a game versus when there's a problem with their code.

### Computational thinking in games

Lots of games, particularly puzzles, require players to use pattern recognition. Tetris or Candy Crush Saga both require players to look for patterns in shapes, colours or quantities to succeed. Abstraction is also important in puzzlers like Candy Crush, with players making decisions about which sweets to match in a row while disregarding others. Being up against time is useful here as it forces players to think quickly about what is or isn't important.

We use decomposition in numerous tasks to break down a problem into more manageable chunks. Playing a game is no different: classic adventure games and escape rooms involve decomposition of bigger problems by looking out for clues and analysing how they depend on each other. This is, of course, a useful skill when programming, but also prepares pupils

or step-by-step solution. This is a useful skill in puzzles and strategy games. Plants vs. Zombies requires players to carefully plan where they'll place their plants to defend their homes. This helps them to think logically and methodically, ensuring they consider how different actions correlate with each other.

Of course, many games aren't limited to just one computational thinking strategy, and there are often links to other curriculum subjects too. Inventioneers requires players to analyse a problem such as rescuing a cat stuck up a tree or delivering a meal to a customer and to solve it by building a machine using the mechanisms provided. This game requires players to recognise patterns in how mechanisms work together, disregard mechanisms they don't need in their machine, break down the problem, and create a step-by-step solution, not to mention the strong links with maths, science, and design and technology (DT).

Similarly, Angry Birds requires players to

analyse a structure and plan their best use of birds to bring down all the pigs, helping to develop an understanding of structures and angles, beneficial, again, to maths, science, and DT.

Try some of these games in the computing classroom and discuss with pupils how they problem solved. Get them to write some algorithms to teach their peers how they completed a particular level, before addressing how these strategies can be used elsewhere, including when they're programming. (HW)

■ Plants vs. Zombies and Angry Birds are so popular that there are Code.org programming activities based on their characters

for the world of work when to manage large projects they'll need to break them down.

When a problem has been recognised and broken down, we need to create an algorithm

**Mia** is a qualified Primary School Teacher and 2018 Code Club volunteer champion, working towards a Masters in Digital Education at the University of Edinburgh. Find her on Twitter @ItsMias Tweets

# THE WEB: THERE'S MORE TO TEACHING PROGRAMMING THAN ALGORITHMS

Current primary computer science education focuses primarily on algorithms, but not on creating a 'product', of which most are now on the web

STORY BY Pal Kerecsenyi, Geza Kerecsenyi, Theodore Tucker

## Introduction

CS students are often rather disillusioned about 'programming', seeing it as writing rather mundane and pointless algorithms executed in a terminal, unlike anything they see and use in the real world. While a drastic improvement has been seen in algorithmic programming education in the last five years, we've neglected the missing link: the creation of a usable 'product'.

## What's the problem?

In today's curriculum, there are huge provisions for algorithmic development. Especially when one looks at the improvement in the last five years, since the rebranding of ICT as Computing, they can see that this field is one that has been a large field of emphasis and effort in most schools. However, all this work may not be as pragmatic as it seems. With algorithmic and software development, students rarely receive a sense of how their knowledge can be applied in the real world. Learning CS is a great tool for their skill set, but it's not much more than that, offering almost no practical experience or knowledge that could be applied past GCSE.

HTML and CSS are usually introduced only at GCSE level (coming as a surprise to most students). However, these skills could be just as important. While the 2013 National Curriculum does partially mention this in their Key Stage 2 agenda, requiring students to "understand computer networks including the internet", teaching algorithmic

thinking isn't enough to achieve this. Web development can't be learned by reshaping the way you think - but actually HTML and CSS focus on it perfectly.

Algorithmic development is an area to which many students are introduced from an early age with tools such as Scratch, but the curriculum makes no provisions for anything like this with the web, although in many ways this is a very dramatic topic to miss. And the most startling part of it all? Teaching the logic behind web development and the DOM will help students massively in understanding computational thinking. But how?

## Welcome to the web

Well, for programmers, the next step from an algorithm is a product. We can draw a comparison to art education here – a student is taught the techniques (an algorithm, in our case), and then makes a painting (the product, incorporating the techniques). The artist must be given the tools to create the piece (paint, brushes, etc), which in our case are web languages, such as HTML and CSS. In art, these can be handed out, but in computing, these must be taught. By combining the skills and the tools, the artist creates a painting. By combining algorithms and a frontend, a programmer creates a product – an application of the skills students are taught, and a result of which they can be proud.

Software is being shaped by web development; with the emergence of web applications, compared to traditional locally

## ENTER CODEDRAGON

**CodeDragon** is an online platform that allows 7- to 14-year-olds to master the basic and more niche aspects of web development, as well as having a safe, solid platform to share and discuss their work. Using a block-based HTML and CSS editor, students can create their own user experience, and learn arguably the most important languages of the web. [codedragon.org](http://codedragon.org)

executed programs; even to the extent where most job listings for software developers require HTML and CSS experience as a skill!

Consider the last time you purchased software outright and installed it on your machine. Now consider the last time you purchased a subscription to a web-based service. You could see this trend in your own school, moving from desktop-based SIMS.net to web-based iSAMs.

Current students will not have the skills to create the programs of today, and may even consider the lack of a link between what they learn in class, and the code they see in the real world.

## Conclusion

CS education often has a difficult time keeping up with the latest trends and developments. The emphasis now needs to move onto a new type of programming, of the now ubiquitous web app. (HW)



**LORNA ELKES** CAS MASTER TEACHER

# OUR APPROACH TO COMPUTING

Time for a thorough review of our approach to the computing curriculum

**L**ike all schools, we're constantly looking for inspirational ways to engage our learners and be creative about meeting the expectations of the curriculum. Fortunately, those leading the school are fully committed to embracing Technology for Learning, backing this with significant investment. We're a Google school, used across our community including school governors and collaborating schools; we subscribe to additional cloud-based resources; and we have a robust computing curriculum outline devised in-house using CAS guidance.

We've also been pretty lucky over the last few years, winning a 3D printer, robotic arm, and STEM funding. So how do we sustain and continue development for our pupils, ensuring staff are competent and confident in their delivery? We use staff meetings and training days to address new resources; staff collaborate, share experiences; and pupils are Knowledge Ambassadors for items such as 3D printing or Google Glasses. We strive to remain current with technological advances through conferences, tweets, blogs, vlogs, and the computing press.

Discussions with staff and pupils suggest that engagement still relies heavily on 'natural' interest. Staff deliver what is required, pupils complete tasks with varying degrees of support, yet when allowed freer access to resources, their prime focus is playing games. They don't click on the 'see inside', try to emulate, improve or collaborate. Many children seek out games that resemble those of games consoles, choosing to be end users rather than developers or architects. Should we be concerned? Do we need the next generation to have a deeper understanding of algorithms and the underlying principles of logical arguments? We do. Wasn't this part of the aim of the new computing curriculum?

As with all learning, engagement is far greater when given purpose. Pupils are motivated to code and debug with 3D Slash or Beetle Blocks as they can generate real-life models

## MAKING IT REAL – GIVING IT PURPOSE

With our approach to writing, we've used a cross-curricular approach to computing. The children were challenged to recreate or redesign. Over the course of a term, I adjusted our computing planning to link to other areas of the curriculum. The results were superb, and pupils previously reluctant to do more than basic tasks were fully engaged.

fairly quickly. Their skills and knowledge soon exceed those of the teachers – which is a good thing – and the pupil 'experts' support each other, creating ever more complex features.

For the last two years, we've successfully run a Code Club. Numerous pupils have completed activities and developed their programming skills, which supplement those taught as part of our curriculum. Additionally, we have drop-in Chromebook lunchtime clubs for those wanting to access technology for pleasure or complete homework.

By tracking the chosen activities from these, we've found that where learning has been given a clear purpose, pupils have gone on to engage beyond the class lesson and try out new ideas and collaborate. They've started to explore the underlying code and look at possible improvements.

Given the above, we now need to review our approach to coding. It needs purpose – like writing. Pupils need to acquire the main skills, but pedagogical approaches must be broader – pupils learn best when things go wrong. They need to be given tasks related to other areas of the curriculum, not just for computing in isolation. (HW)

**Lorna** is a Deputy Headteacher in Buckinghamshire and CAS Master Teacher.

# WHAT'S TURNING GIRLS OFF COMPUTER SCIENCE?

While many of us teachers have embraced computer science, our female pupils appear to be moving away from it. Why is this and what can we do about it?

STORY BY Rowan Roberts



■ Primary school girls creating their own virtual worlds during London CLC's 2017 Tate Exchange

If you've ever compared the Google image search results for the phrases 'computer for girls' and 'computer for boys', you probably got a nasty shock.

For boys, the images contain pictures of boys using computers, smiling cartoons, and photos of happy children absorbed in whatever is on their screen.

Search 'computer for girls', however, and there are pink toy computers and accessories, a diamante-encrusted mouse, flowery laptop stickers, but scarcely a single girl actually using any of the devices. This kind of imagery surrounds children growing up in a digital age, and it's our responsibility as educators to understand the effect it can have on their development.

In our work with schools at London CLC, we've found the move from ICT to computing to be inspiring and energising, but sadly the same can't be said for all GCSE students. Compared with ICT, computing has been chosen by fewer girls, black pupils

and pupils entitled to free school meals – groups already underrepresented among ICT candidates. And, as more schools have adopted the new GCSE (though, worryingly, still fewer than half), it's actually got worse. An even smaller proportion of 2017 GCSE computing entrants were female than the year before – just 20%.

So why does this change in the curriculum seem to be threatening girls' enthusiasm for the subject? While the ICT GCSE focused on mastering the use of existing tech products, the new curriculum places more emphasis on programming – understanding what's happening underneath the surface and changing it. Sadly, as those Google searches showed, we seem to associate these processes more with boys.

## The gender gap

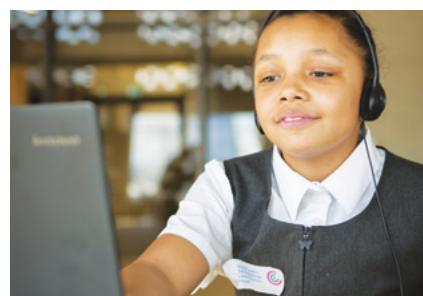
This hasn't always been so. Women have a long history of participating in computer science. Take Ada Lovelace – regarded

by many as the creator of the first ever computer algorithm. Or Pauline Oliveros, a central figure in the development of post-war electronic music. In the 1950s and 60s, NASA employed many female computer scientists to develop and operate software used in the space programme. And as more and more women began studying at universities, computer science enjoyed a steady increase in female participation, right up until the early 1980s.

Then something happened that turned women away from computer science. I believe we can attribute that shift to one factor above all.

## The personal computer

It was around the mid-1980s that personal computers began to find their way into family homes. And they were largely marketed towards men. Adverts would often depict computer use as a father-son activity, and the first few computer games to appear on the scene were generally aimed at boys. As the gaming industry mushroomed, this bias only increased, with the introduction of female characters who always seemed to be either helpless victims or heavily sexualised. As 'geek culture' grew, it became

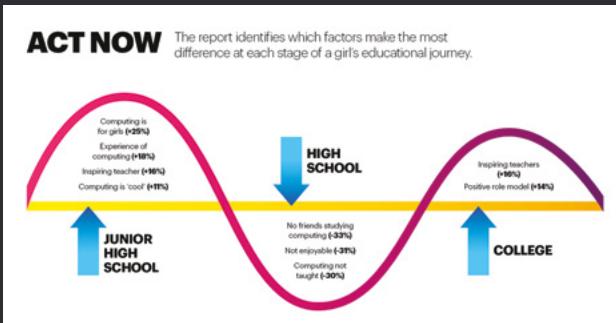


■ Girl creating a digital composition from skyline sketches during London CLC's 2017 Tate Exchange event

# CRITICAL TIMING

Cracking the Gender Code identifies that many girls find themselves in a much more inclusive atmosphere early in their education, with role models and initiatives helping to inspire the

belief that computing can be for girls. But, as girls get older and move up into high school, these influences often become outweighed by factors such as the fact that, for example, fewer of their female friends have an interest in computing. It's not until university that many of these young women find a renewed motivation to pursue computing, and for many it comes too late.



■ Image courtesy of [girlswhocode.com](http://girlswhocode.com)

increasingly apparent that girls weren't really invited to the party, which clearly had an impact on girls' feelings of confidence and self-worth around computing.

## Permission to fail

We know that confidence is enormously important when it comes to pupils' engagement in a subject. A 2015 OECD study found that students who approached a subject with a confidence in their own aptitude felt more free to fail. This allowed them to engage in the trial and error processes that are fundamental to learning in maths and science subjects. It allowed them to take risks.

There's also a difference in the language we use to describe men and women and the fields in which they tend to work. A 2015 study found that people working in male-dominated fields such as maths and physics are far more likely to be described as brilliant or geniuses.

Of course, this filters its way into children's understanding of their place in the world. In another study, children as young as six echoed the idea that brilliance or giftedness is more of a male trait, and that girls must work harder to become good at something. Girls were less likely than boys to participate in an activity when they were told it was for "really, really smart" children, but did show interest in a very similar game

when told it was for children who try "very, very hard".

So, as they grow up, girls learn that it's really important not to get things wrong, but that they probably won't know how to get it right without the help of someone naturally gifted. Given that, is it really surprising they lack the confidence to take risks in computing? Yet it's the very process of figuring out the answers on your own that helps a child to develop as a computational thinker.

If we can nurture our learners to develop a growth mindset, they can become more confident in computing, seeking useful feedback on their progress, eager to learn what they could do differently next time, knowing that mistakes are useful in programming. You can't learn to debug if you do everything right first time! And so they'd understand that there isn't really any such thing as being 'good at' computing. It's just about having enough practice to find it slightly easier than before.

## Developing growth mindsets

So, how can we try to ensure that girls develop a more robust sense of themselves as computational thinkers from an early age? As somebody who teaches programming to thousands of different children, I have a few suggestions to offer:

■ **Model a growth mindset yourself:** catch yourself next time you say something like "I'm a terrible singer". Try to create an environment where children see that skill is plastic and can change, and ask your colleagues to help you.

■ **Challenge learned helplessness:** when a child asks you to help them and you know they don't really need it, fire the question back at them: "I don't know, what does happen when you click that block?"

■ **Visibly enjoy the subject:** I sometimes make Scratch projects to solve simple classroom problems such as allocating different subjects to groups at random. Not because it's easier than drawing names out of a hat, but because it's more fun, and I want the children to know that I think so.

■ **Expose children to a range of role models:** yes, Tim Berners-Lee has made a huge contribution, but so have Ada Lovelace, Grace Hopper, and Annie Easley – and they've all done it in a world that made it harder for them.

■ **Use enquiry-based approaches like ScratchMaths:** this resource scaffolds really well, but it also encourages exploration and experimentation.

Many programming activities focus more on product than process, and I think this puts off a lot of girls. It means there's a single desired outcome, and implies that if you don't achieve it, you've failed – a notion we know tends to be destructive to girls' learning.

It's important for all children to know that, as NASA computer scientist Annie Easley once said, "If you want to... you can do anything you want to, but you have to work at it." You can still think of talent as a gift, but just remember that it's a gift that you have the ability to give to yourself. All you need is a bit of practice. (HW)

**Rowan** is a teaching and learning consultant and local CAS Hub leader at London Connected Learning Centre, an award-winning organisation that helps schools and other settings use digital technologies to improve learning.

# SUBSCRIBE

Sign up today for a year - prices start at **FREE!**



**SUBSCRIBE  
TODAY**

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

# (Hello World)

**FREE  
IN PRINT**  
For UK-based  
educators!



# TO SUBSCRIBE VISIT: [helloworld.cc/sub1year](http://helloworld.cc/sub1year)

**Not a UK-based educator?  
Buy any issue for £6  
Visit: [www.store.rpipress.cc](http://www.store.rpipress.cc)**

# SUBSCRIBE ON APP STORES

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



Available on the  
App Store



GET IT ON  
Google Play



# CHILDREN'S LITERATURE FOR COMPUTER SCIENCE

Ever thought computer science can be costly and overly complex?  
An elitist subject just for 'geeks' and 'nerds'? Well, you're in for a treat!

**STORY BY** Miss Sarah J Twigg, Dr Lynne Blair and Dr Emily Winter

The computer science national curriculum of 2012 caused many teachers a lot of stress and bother. With many having no background in computing, the prospect of teaching five-year-olds how to program was a daunting task.

The curriculum stated that children in Key Stages One and Two needed to learn "The Three Main Programming Constructs", which are sequencing, selection, and repetition.



■ Example function of how to set up and create a story within Scratch



■ Example of how to decompose sections of selection criteria into their own functions, so that each event/meeting has its own function

The document outlines requirements, declaring that by the end of Key Stage One, children should know what an algorithm is, and in Key Stage Two, they should know how to design, write, and debug programs using logical reasoning.

Many of these terms hadn't previously been discussed in a school environment. And teachers had possibly only heard them when listening to computer scientists talk about algorithms on the news with reference to artificial intelligence. These seemingly 'scary' terms branded computer science an 'elitist' subject, when really an algorithm is just 'a set of instructions that must be carried out/followed step by step'. This is where A Programmer's Tale comes in.

## A Programmer's Tale

A Programmer's Tale is an organisation that started off as a research project for an undergraduate dissertation investigating 'The effectiveness of Children's Literature for teaching the basic principles and concepts of programming to Key Stage One and Two students'.

The researchers used the popular book outlet, The Book People, to search for picture books for ages 5-7, with the aim of finding books that were already typically available in classrooms. The initial search returned 75 books, which was reduced to 50 once multiple books/story collections and duplicates were removed. Once coded by the selection criteria, it was time to translate the books to code.



■ A collection of 16 children's books used to teach sequencing, selection and repetition

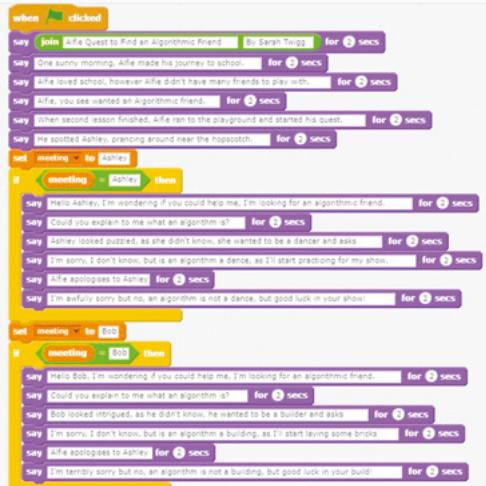
## Translating books to code

A Programmer's Tale researched the meanings of sequencing, selection, and repetition to see how they could be translated in literature, creating a set of criteria for books to be tested against:

■ **Sequencing:** it was anticipated that most, if not all, books would illustrate this construct, since stories generally move through a basic sequence of plot stages.

■ **Selection:** at least one example of a choice of dialogue, actions, environment, and so on. For example, where the dialogue follows a repeated pattern, but changes occur dependent on the context, or where the current context of the storyline is examined to test whether to continue or whether the desired goal has been reached.

■ **Repetition:** at least one example of a pattern of repeated dialogue, actions,



■ A sample of how to incorporate selection without the use of repetition



■ A sample of how to change text in a children's book into Scratch code

environment, and so on. For example, a story may go through different contexts where the same dialogue and/or action is repeated in each different context.

Translating the books in code in a classroom environment takes time and should not be rushed. As the studies have shown, children learn best by translating the books first through unplugged activities, and then on the computers.



■ Further examples of how selection criteria can be used to change the output of a loop using variables

## Computational thinking

Translation from literary text to code also allows for computational thinking methods to be implemented. With many books having lots and lots of text, the programmer is only interested in the change in events and whether there's any repeated text. For example, if a story explains how old a character is, but has no impact on the plot of the story, this can be abstracted out.

Other concepts and methods such as debugging and collaboration are also present in the translation. As shown in testing, the class were playing out the story, often getting confused on where to implement if statements to change the output of the loop, leading to debugging exercises.

## Alfie's Quest to Find an Algorithmic Friend

'Alfie's Quest to Find an Algorithmic Friend' is a draft of a book in production by the researchers to explain the above concepts. Alfie goes on a quest around the school playground at break and lunchtime trying to find a friend who loves algorithms as much as him.

### Sequencing

■ A sample sequence will involve translating the whole book into one long block of code (in Scratch, this will use lots of 'say' blocks).

### Selection

■ In the first instance, Alfie meets a girl called Ashley. Ashley wants to be a dancer, so asks if an algorithm is a dance.

Alfie apologises, saying this wasn't the answer he was looking for, and carries on with his search.

● If loop counter == 1 then

- Say "Hello Ashley, I'm wondering if you could help me, I'm looking for an algorithmic friend. Could you explain to me what an algorithm is?"

- Ashley looked puzzled as she didn't know. She wanted to be a dancer and asks "I'm sorry, I do not know, but is an algorithm a dance, as I'll start practising it for my show?"

## EXTENSION TASKS

Children's literature can also be used to teach pupils the rules of decomposition and functions. Within Scratch, there are two different options for this using either the broadcast function or the make block. The 'make block' setting is great when working with one sprite, and broadcast for multiple sprites. Therefore, if the pupil wants to use multiple sprites, encourage them to do so as they can learn about inheritance. (A Key Stage 4 Topic 'say what!')

- Alfie apologises to Ashley: 'I'm awfully sorry, but no, an algorithm is not a dance, but good luck in your show!'

### Repetition

■ Alfie's quest to find an algorithmic friend has four loops, and each loop has a different output due to the answers of the different characters. On the fourth loop, Alfie finds his algorithmic friend, thus ending the loop. The loop can be created by using a for loop, for example:

- For loop counter < 5, add 1 to loop counter

● If loop counter == 4 then

- Alfie meets Holly, who wants to be a computer scientist. Holly gives Alfie the response he was looking for and Alfie is shocked at first, but then they become the best of friends, creating lots of different algorithms.

As demonstrated in 'Translation (flowchart) – book to code.' (HW)

Graduate researcher/CAS regional coordinator at Lancaster University/founder of A Programmer's Tale. My job is to research new areas, concepts, methodologies, and pedagogies of computer science education, collaborating with teachers in the local area to come up with and test ideas to be brought into a wider market. Using Children's Literature for Computer Science is a research project at Lancaster University in partnership with CAS Lancaster. Our aim is to come up with new pedagogical ideas that create fun and creative classroom environments.

# IMPROVING PROBLEM-SOLVING SKILLS USING ESCAPE GAMES

There's a lot more to computer science than coding. Using a games-based approach can reap rewards in the classroom

STORY BY Alex Clewett

**L**earning how to solve problems is fundamental to computer science. In order to create a computer program to perform a particular task, one must first understand how a human might solve that problem. Furthermore, one must then turn those problem-solving steps into something that a computer can do (algorithm) and then finally into a language that the computer can interpret (code).

There are plenty of resources to teach the fundamentals of computational thinking. Similarly, there's also a comprehensive selection of resources available to teach programming languages. But what about the first step? How do you teach the art of problem solving? Puzzles such as Sudoku, crosswords, and strategy games can all help here, but I've found the immersive quality of escape games to be particularly beneficial. The physical nature of the problems seems to be more engaging and contextual than simply giving students different puzzles to solve. This is perhaps an analog to the recognition of physical computing as an aid to learning in computer science.

## Why use escape games?

The games-based approach is supported by a wealth of evidence and research that shows using games in the classroom can enhance learning in many ways. It improves engagement and motivation, and can encourage learning to continue beyond the classroom. Developmental psychologists and



Solving puzzles collaboratively enables student-led learning

Early Years practitioners have recognised the importance of play in learning, but for some reason with older children this seems to be abandoned in favour of more formal, rote learning methods.

Games, particularly collaborative ones, can also make differentiation easier and more natural. When working in a group to solve a puzzle or complete a task, learners can naturally adopt a role that plays to their own strengths. Channel 4's *The Crystal Maze*, with its mental, mystery, physical, and skill games, is a good example of how this can work. Children learn by doing;

when given the opportunity to actively participate in something that reinforces the learning, they're more likely to be able to recall that information at a later date. This is why these games can be so effective for revision of curriculum content.

## Escape games for free

A number of good-quality resources exist online for using escape games in your teaching. One I've used successfully for teaching computer science content as well as problem-solving skills is by Nichola Wilkin ([helloworld.cc/2PqpFAI](http://helloworld.cc/2PqpFAI)). She also



■ You can purchase ready-made escape kits, but it's much more fun to put one together yourself

has a new game on her blog, which can be downloaded for free if you subscribe to her mailing list. Her games consist of digital and online resources, along with physical printouts and artefacts that the students use to progress through the game and achieve the end goal. The games are well designed and constructed, so a whole class can easily be involved in the same game at the same time, working in small groups.

A quick search on [www.tes.com](http://www.tes.com) will unearth a variety of subject-based escape games developed by teachers. Obviously, much like everything else online, quality will vary, so it's best to look thoroughly through the content before using the resource with your classes. It's also fairly straightforward to develop your own escape games, either physically or by using digital methods.

used Scratch to create an escape game ([helloworld.cc/2OVtRaW](http://helloworld.cc/2OVtRaW)), which is a great way to involve students in creating as well as doing. One class of students can create games for another. It also makes a great basis for transition work between primary and secondary schools.

### Using commercial resources

There are a variety of physical Escape the Room board games on the market, but these can be challenging, so are probably best used with post-16 students or as inspiration for devising your own games. Furthermore,



■ Props and other equipment like UV torches add a 'wow' factor

it's unlikely they'll contain any relevant curriculum content if you're also using them to consolidate subject learning as well as the problem-solving skills themselves.

**www.breakoutedu.com** is a learning games platform that allows you to create your own digital games, as well as hosting a number of physical games in a variety of subjects.

## TIPS FOR SUCCESSFUL GAMES

- Make the first clues accessible to all learners and include hints, otherwise they'll switch off.
- Use a variety of media: printed sheets, artefacts, props, and 'hidden messages' using UV. Be imaginative!
- Let the learners create as well as solve: once they've played some, let them create some.
- Get your colleagues involved: problem solving is a cross-curricular skill.
- Make it fun!



■ Image courtesy of [www.breakoutedu.com](http://www.breakoutedu.com)

### Next steps

The Channel 4 show *The Crystal Maze*, now rebooted with the phenomenal Richard Ayoade, has been a great inspiration for puzzle ideas, and I've recently been experimenting with the idea of props for my games. I've used an inexpensive RFID lock to create a secret drawer for one game and hope to make more use of props in the future.

A game involving puzzles and props using micro:bits and other associated hardware is next on my list of things to try. This is going to fit in nicely with the 'Digital Maker' approach I'm introducing into my practice. I hope you find some of these ideas interesting and try them out with your students, and if you make any game props, please share! **(HW)**

## "GAMES, PARTICULARLY COLLABORATIVE ONES, CAN MAKE DIFFERENTIATION EASIER AND MORE NATURAL"

Designing a game consists of creating a series of clues that are interlinked, where players can't progress until they solve each one. Lots of examples exist online, which can be used as is or modified to suit your particular purpose. The Nowescape blog ([helloworld.cc/2Lcn4aw](http://helloworld.cc/2Lcn4aw)) has plenty of ideas for how to construct your puzzles. Platforms like Google Forms or Office 365 Forms can easily be manipulated to construct an interactive set of questions and puzzles that students need to solve in order to progress to the next step. I've also

The platform works on a subscription-based model, giving you access to the pre-prepared games as well as the game creator. You can also purchase kits that include the physical box to unlock along with a number of different padlocks. Buying the physical kit can be quite expensive, though, as the items are shipped from the US. I put together a kit quite inexpensively by purchasing padlocks and boxes online. All I needed to pay for was the 12-month subscription to access the platform, which is reasonable at under £50 per year.

**Alex** is a Secondary Teacher of ICT and computing, a digital leader, and coordinator of the 'Cracking the Code' programme for North Wales School Improvement Service. He's also a CAS Wales committee member.

# THE PRINCIPLES OF NETWORKING

## MADE EASIER WITH PACKET TRACER

Cisco's powerful simulation tool lets students experiment with network behaviour by building complex networks and encouraging practice, discovery, and troubleshooting – plus it's free to schools!

STORY BY Helen Clothier

**A** reliable network forms the heart of any technology infrastructure, and networking is now at the forefront of technology innovation in our increasingly digital world.

It's therefore vital that teachers are equipped to teach networking in classrooms, particularly given the skills shortages in this area. Yet with many teachers struggling to tackle the principles of networking, purchasing switches, routers, and many other devices might appear to offer the perfect solution.

For most schools, however this isn't a practical option, which is why Cisco has created the next best thing for teaching networking – Packet Tracer.

Packet Tracer simplifies the complexity of teaching networking while giving students a valuable hands-on experience. Cisco developed Packet Tracer to help our Networking Academy students achieve the most optimal learning experience while gaining practical networking technology skills.

Founded in 1997, Cisco Networking Academy is a not-for-profit IT skills and career building programme that connects millions of students, educators, and employers worldwide. As part of this programme, Cisco partners with learning institutions to deliver technical training and problem-

solving experiences to individuals studying networking, security, and IoT technologies.

In 2017, Cisco launched a skills manifesto, focused on digital opportunities for everyone. In partnership with Birmingham City University, we extended our reach to all levels of education through our Computing for Schools programme. It provides support for teachers in the delivery of networking using resources mapped to Computing at School's (CAS) Progressions Pathways. All course content is available free of charge and can be accessed by becoming a Cisco Academy.

### What is Packer Tracer?

Packet Tracer is a powerful simulation tool that allows students to build, explore,

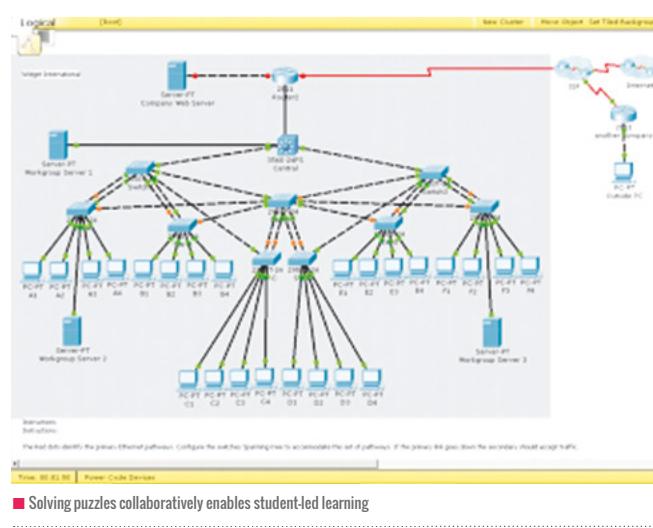
and troubleshoot a variety of network environments as if the hardware was with them in the room. By dragging and dropping routers, switches, and various other types of network devices, virtual 'network worlds' can be developed. This opens the way for teachers and students to explore, experiment, and discover an almost unlimited array of networking concepts and technologies.

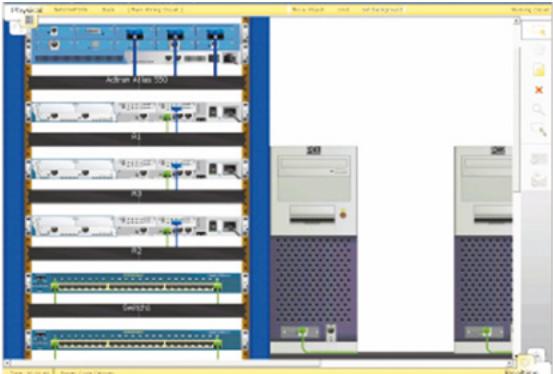
### Key features

Packet Tracer has two workspaces – logical and physical – which can be easily switched at a click of a button. The logical workspace allows users to build coherent network topologies by placing, connecting, and clustering virtual network devices. The

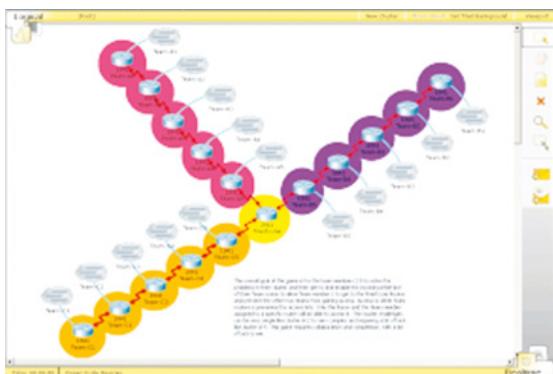
physical workspace provides a graphical physical dimension of the logical network, giving a sense of scale and placement in how network devices such as routers, switches, and hosts would look in a real environment. The physical view also provides geographic representations of networks, including multiple cities, buildings, and wiring closets.

Packet Tracer provides two operating modes to visualise the behaviour of a network: real-time mode and simulation mode. In real-time mode, the network behaves as real devices





■ The physical workspace offers a graphical view of the logical network



■ Multiuser games provide fun learning opportunities for collaboration and competition

do, with immediate real-time responses for all network activities. The real-time mode gives students a viable alternative to real equipment, and allows them to gain configuration practice before working with physical equipment.

### Supports teaching networking

The Packet Tracer tool contains many exciting features that offer an extra dimension to teaching computing. Schools can easily teach and demonstrate complex technical models as well as networking concepts and protocols using an interactive environment.

One key feature, the Activity Wizard, allows teachers to author their own learning activities by setting up different scenarios. They can customise these scenarios with instructional text, while creating initial and final network topologies and predefined packets. The Activity Wizard also includes grading and feedback capabilities. In addition, activities can be saved and shared with other teachers and students.

Packet Tracer aids the teaching of networking by:

- Providing a visual demonstration of complex technologies and configurations
- Letting teachers author customised, guided activities that provide immediate feedback through using the Activity Wizard
- Facilitating numerous learning activities such as individual and group lab activities, homework, assessments, games, network design, troubleshooting, modelling tasks, case studies, and competitions
- Allowing self-paced learning outside the classroom
- Supporting social learning processes by enabling collaboration and competition

### The student experience

Packet Tracer's hands-on

mode of learning means students will be better equipped to apply concepts and configuration fundamentals when exposed to real equipment. Through experimenting with network behaviour and asking 'what if' questions, students will gain a solid understanding of how devices connect and communicate in a live network, and how data flows from one device to another.

The software uses a drag-and-drop user interface, allowing students to add and remove simulated network devices as they wish, and lets students practise using a command-line interface. This 'e-doing' capability is a fundamental component of learning how to configure routers and switches. Just as the physical equipment allows you to modify hardware, Packet Tracer offers the ability to insert interface cards into modular routers and switches, which then become part of the simulation.

Students can also learn how to design complex and large networks, which isn't always possible using physical hardware. From the very basics, such as connecting a PC to a hub, to setting up a server and building a LAN and WAN,

## HOW DOES PACKET TRACER HELP YOU TO TEACH NETWORKING?

Packet Tracer allows users to build and configure a functioning computer network in a simulated environment. Traffic is simulated, with web and email servers being utilised by a variety of desktop applications. Simulations work in both 'real-time' mode, as they would in a physical network environment, or in 'simulated mode' where users can see the actual packets and frames moving through the network and decode the layers structure of Ethernet, IP, and TCP.

*Duncan Maidens, CAS West Midlands Regional Centre Lead @djmaidens*



students can build an almost unlimited number of environments. And as they gain practical experience of configuration, troubleshooting, and other tasks, they become more confident in their abilities. In addition, the simulation-based learning environment helps students develop essential business skills such as decision making, creative and critical thinking, and problem solving.

### How to access Packet Tracer

To access Packet Tracer and explore a range of networking activities suitable for use in the classroom, sign up for the free course at [cs.co/C4S-PT101](https://cs.co/C4S-PT101). If you want to know more about becoming a Cisco Academy, which gives you access to extra learning material/activities and CPD such as Python, C++, C, Cybersecurity, and Linux, visit [cs.co/UKCDASkills](https://cs.co/UKCDASkills).

**Helen** is CDA Skills Programme Manager, Cisco.

# DON'T TOUCH THE PLUGS! YOU'LL BREAK IT!

How can getting hands on with computers (and wires!) help us to break down abstract concepts in primary school computing?

STORY BY Sway Grantham

In my previous article, *Hello World 5*, I asked the question "Is a Beebot a computer?" and considered children's existing perceptions about what a computer is and isn't. However, acknowledging the problem is one thing, tackling new ways to teach it is another... I wanted an exciting, discovery-based learning activity that would allow children to test their assumptions without just being lectured about what makes a computer a computer and what makes a peripheral or an input device.

To do this, I looked to other subjects where we already do this really well: Maths. In Maths, when we get to the tricky concepts, we get out the cubes, the pizza-shaded fraction sets, or the place value number cards, and this allows us to demonstrate these concepts. Children can then be left to explore by moving their manipulatives around and seeing what they can discover. Could we do this in computing?

## Manipulatives in computing

Computing often seems a very 'hands-on' subject because children are 'using' technology. However, we know that this isn't enough for them to understand how it works. Poking mindlessly at a device ultimately just becomes frustrating, and when you do manage to achieve what you wanted, you've often forgotten how you did it so can't repeat it.

And then I saw the wires...

Desktop computers have lots of wires plugged into them... If you're lucky enough

to still have an ICT suite, these wires may even have been cable-tied into the base unit to prevent small fingers from poking them, but this is where the discovery learning happens! This is where you can explore what a computer needs to work and what are just useful extras!

## Now what?

Firstly, my school doesn't have any desktop computers that the children can use (as lovely as our office staff are, they'll probably draw the line at us dismantling their computer for the children to explore in the middle of a work day!). To get around this, I used some Raspberry Pi computers as these are cheaper to get hold of (and a good sized for KS2 children to explore). However, if you have desktop computers for children, use those!

I started by showing the children the Raspberry Pi under the visualiser – no

case, so you could see everything inside (you can take the side off of a regular desktop computer) and said "This is a computer". This may seem counterintuitive for discovery learning, as I told the children the answer, but there's much research that suggests children will often conclude incorrectly based on their own discoveries. I then got them to explore in their groups and discover for themselves what made it a computer. They had the answer, but they had to discover the journey, with each table having their own computer to get up close to. This journey was documented using a 'statements, questions, ideas' model where the children used big paper and in table groups discussed what they could see on a Raspberry Pi that suggested it was a computer (statements), what questions they had or that they would like to explore further to show that the Raspberry Pi is a computer (questions), or any ideas that they

Statements	Questions	Ideas
It has holes on the back	What are the gold stick things for?	Could this be so that you can plug in a keyboard?
There are square things drawn on	Why aren't all the holes the same?	This says power - is it like a phone charger?
Everything is connected with wires	Why is there so much empty space inside?	Is this what's inside things like traffic lights?
This is where you plug in the charger*		

\*While I do know that this isn't factually correct, it's the most common misconception I find at this stage. The children often have never encountered a computer that isn't portable and that doesn't have a battery. This concept often seems like the dark ages to them!

had for what they would like to do with this computer that was a Raspberry Pi (ideas).

We share this discussion as a class and then I plug everything in and we prove that it is a computer and that it really does work!

This is a very initial exploration, but it's the beginnings of the children recognising inputs and outputs as well as which parts of a computer are required and which part are extra peripherals. With seven- to 11-year-olds, if I jumped straight to asking the children to plug in the Raspberry Pi and turn it on, they would miss a lot of the dialogue which helps them to assimilate their new understanding.

## Playing with plugs

The next lesson, I begin by getting the children to connect the peripherals to the computer and giving them the chance to 'turn it on' in groups. There's not much that can go wrong here: it's like a jigsaw puzzle where each wire only connects in



■ Use Raspberry Pis as a manipulative in computing

one place. For example, if you tried to plug a HDMI cable into the USB port, it simply wouldn't fit. The children are given time to plug everything into an extension lead (which isn't connected to power, until I've checked it!). This is something the children have probably never been allowed to do, and it's both exciting and scary at the same time. They've often been told that electricity is dangerous and you shouldn't touch things as you'll hurt yourself/break them. While safety messages and lessons about electricity are 100% important in children's upbringing, there's very little damage the children could do to a computer, particularly when it's not even plugged into the mains.

After they've achieved this, I start to challenge their misconceptions further. As a class, they watch me with my nicely

connected computer, using the board as my monitor. I ask "what would happen if I unplug the keyboard?" This is met by mostly horrified faces who think the computer will crash and never turn back on again. So I do it. We hold our breath. Nothing happens. It's all very anti-climatic. I ask again, what happens? This time, we conclude that you can't type anything. This doesn't seem that important to most of the children who would rather be clicking on things anyway. I then repeat this process with the mouse, the monitor, and then suggest it with the power supply.

I don't actually disconnect the power, as this is more likely to confuse/corrupt the computer, but do the children recognise that a computer needs a power source? This is not a peripheral, it's a vital component. Now you can discuss whether the computer is still a computer if it doesn't have a keyboard and mouse? Yes, you just can't use the keyboard or mouse, how else could you control it? What about voice control such as Alexa, Siri or Google Assistant? Once again, you're back to exploring what are input/outputs and what parts are necessary for the computer to actually work, and the children are refining their own definition of what a computer is.

Using Raspberry Pis as a manipulative in computing has helped the students to grasp concepts that are of an abstract nature. By starting the exploration with the device unplugged, the children were able to recognise their own understanding, which led to a more focused testing of these beliefs in subsequent lessons.

When I first asked the question "What is a computer?", these were the definition I was given:

- A keyboard and a screen
- A machine with a keyboard
- A search engine
- A machine used for work

After these lessons, I was told:

- A computer has lots of switches and plugs to plug things into – it doesn't have to have a screen

## THE NEXT STEP

For older children, it can be fun to give them working projects that don't use a keyboard and mouse to get them to figure out how they work. Using simple online tutorials, in the past I've set up:

- A motion-sensor camera to take photos when you're nearby
- A warning system for when a plant needs watering using a micro-bit
- A simple robot using a controller to make motors turn

Again, this is a chance for the children to explore what has a computer inside of it and the various inputs and outputs that are fit for purpose.

- A computer needs code on a microchip to make it work – without that, pressing a letter would make nothing happen
- Not all computers look like a computer, they have different shapes, designs and are used for different needs

While these answers still show some conceptual misunderstandings, I think there's noticeable progress in the complexity of their answers and the larger breadth with which they're trying to draw their conclusions. Their reliability on a computer's functionality to define it is much less, and their explanations of what a computer needs to be defined is more dense.

Have you ever asked your class "what is a computer?" – you might be surprised at their answers! (HW)

**Sway** teaches computing at Giffard Park Primary School in Milton Keynes as well as being a Digital Technologies consultant. She was part of the first Picademy to become a Raspberry Pi Certified Educator and loves engaging in conversations about the pedagogy of computing. Find her on Twitter @SwayGrantham



■ Students show off their Minecraft code and builds

# HACKING MINECRAFT WITH PYTHON

Change your Minecraft world with text-based code! And learn some Python!

Here's how to get started.

STORY BY Bob Irving

**H**ow do students make the leap from block-based programming (like Scratch) to text-based coding? I searched for an 'on-ramp' that would be somewhat gentle but also engaging. Being a Minecraft fan, I knew the engagement factor was off the charts and so looked for a way to incorporate coding into Minecraft. I found it in Minecraft Pi, a mod of Minecraft that allows users to code the Minecraft world with Python. I'll explain what I've learned on this journey, tips, tricks, and gotchas, and how to set up and install your Minecraft and coding environment.

Check the Setup and Platforms box for the next step. Once you're installed and connected, check back with me here. Oh, and if you've got a Raspberry Pi, all you've got to do is launch Minecraft and Python.

Now let's get started! Every great coding exercise MUST start with a 'hello world' (hence the name of the magazine you're holding!). That's a time-honoured tradition

amongst coders that demonstrates that you've entered some code somewhere, and when you run it, it outputs the phrase 'hello world' somewhere – a monitor, usually. We're going to have it output in your Minecraft world in the chat window. Regular Minecrafters will know that the chat is where you get messages from Minecraft or enter special codes to make stuff happen.

Open up a file-editing window in Python. Note that this is different from the 'shell'. The shell is kind of like Minecraft chat; it's where Python sends you messages. You can tell you're in the shell because it says 'Python 2.7.11 Shell' at the top. It also shows three angle brackets at the beginning of the bottom line, like so: >>>

If you're in the shell, you can open an editing window by selecting File | New File from the top of the window. This will open a blank window for you to enter your code. If you're keeping score, that's four windows (three if you're using the Raspberry Pi).

You need to have them all open at all times: Minecraft, the Bukkit server, the Python shell, and your editing window. Make sure you leave them all open, as they perform important functions.

In the editing window, type the following:

```
frommcpip.minecraft import Minecraft
mc = Minecraft.create()
mc.postToChat("HelloMinecraftWorld!")
```

Just those three lines! The first line grabs the Minecraft functions that Python needs for giving instructions, the second creates a connection between your code, and the third one, well... that's your Hello World!

Two more things need doing before we see the awesome results of our first program: save it in the correct place and run it. To save it, select File | Save, then navigate to the Adventures in Minecraft folder, and

## SETUP AND PLATFORMS

Minecraft Pi can run on Windows, Mac, Linux, and Raspberry Pi. For the first three, you'll need a copy of Java-based Minecraft plus a personal licence and server software, so you can run your instance of Minecraft on your machine, and the latest version of Python. If you have a Raspberry Pi, you're in luck! Minecraft and Python are installed by default on Raspbian, and you'll need no server software and no personal Minecraft licence!

If you have Minecraft on your machine, go to [helloworld.cc/2Buge0w](http://helloworld.cc/2Buge0w). You can download the Starter Kit for your platform and follow the instructions. There's a set of super helpful videos as well.

then to the MyAdventures subfolder. Save it there. You can't just save it anywhere, because Python needs to find the Minecraft commands, which are in an adjacent folder. Unless, of course, you're using a Raspberry Pi, in which case you can just save it where Python wants to save it. Did I mention how much easier all of this is on a Pi?

Now to run it, go to the top of your editing window and choose Run | Run Module. This will launch your code. Get ready to jump right over to the Minecraft window, because you probably know that chat messages last only about eight seconds before they fade away. And you'd hate to miss this exciting first 'hello world'! The easiest way to do that is to hit the ESC key while you're in any other program, then click on the Minecraft window. If you've done everything right, you'll see that 'Hello

Minecraft World' shining forth with all its might in your very own Minecraft world!

Or, alternatively, you haven't. If that's the case, welcome to the world of coding, where things can (and often do) go amiss! First place to check is the Python shell. Remember we said that's where Python sends you messages? That includes error messages. If there are some words there that weren't there before, usually with the word 'error', then Python tried to do what you wanted but couldn't. It's doing its best to let you know what you did wrong.

A word about error messages: they're your friends. Really. They may not make total sense at first, but they will eventually, and they're usually in a form of English that makes some sense. In addition, error messages will tell you what line the error occurred on, which is super helpful. You should know that the majority of errors are caused by the coder (you). You either misspelled something, made a capitalisation error, or a punctuation error. And here's a special coder tip for finding what the problem is, though it does require you to open another window. If you have a browser window handy, you can select the error with your cursor, copy it, then paste it into the search bar in your browser. There's a good chance the answer you seek is in the first three search results!

Every coder – beginner or veteran – makes multiple mistakes all the time. Don't take it as a sign of failure or an indication of your skill as a coder. Just know that our fingers don't always do what our brains tell them to do. And sometimes our brains forget things. Python is kind enough to help us out. I tell my students that Python is simply training them to speak to it in the way it understands.



Hacking Minecraft with some Python code. Awesomeness awaits!

## RESOURCES

The best resource is the book *Adventures in Minecraft*, published by Wiley, and co-authored by Martin O'Hanlon and David Whale. It's full of clear instructions, fun projects, and further challenges.

The resources page at the publisher's site ([helloworld.cc/2vWhpk3](http://helloworld.cc/2vWhpk3)) has everything you'll need, including code downloads, software starter kits, and helpful videos. They also run a website ([www.stuffaboutcode.com](http://www.stuffaboutcode.com)) which has a forum dedicated to this, and check regularly and offer help!

The authors are on Twitter (@martinohanlon and @whalegeek).

Another great resource is Chris Penn, a teacher in the UK who does loads of awesome stuff in this space. He's @ChrisPenn84.

David Whale's blog, which has great resources for teachers, including Python flashcards: [helloworld.cc/2N6fuQt](http://helloworld.cc/2N6fuQt)

Awesome intermediate and advanced projects blog: [helloworld.cc/2LcLxd](http://helloworld.cc/2LcLxd)

API for Minecraft Pi, including commands and block IDs: [helloworld.cc/2nUA8rR](http://helloworld.cc/2nUA8rR)

So now you've set up Minecraft, got it running in a local server, and written your very own working 'hello world'. Congratulations! You've so many exciting adventures ahead of you. Soon you'll be teleporting, building structures, changing your world with code as you walk through it, and dozens of other cool things. And everything you learn about Python can help you in other coding projects. And learning Python will make it easier when you attempt any other language, like Java or JavaScript. For more exciting projects, check out the resources. There's no end to the awesome things you can do with Python and Minecraft! (HW)

**Bob** teaches middle school computer science at the Porter-Gaud School in Charleston, SC. He's a Minecraft Global Mentor and a Raspberry Pi Certified Educator. More importantly, he provides 'hard fun' for his students.

# NETWORK OF EXCELLENCE

Through the Network of Excellence programme, Computing At School has improved teachers' confidence, their subject knowledge expertise, and their pedagogical expertise

STORY BY Simon Humphries



**B**etween April 2016 and December 2018, Computing At School's Network of Excellence programme has provided over 80,000 teacher hours of CPD and support to nearly 25,000 teachers in over 9,400 schools. In secondary schools offering GCSE Computer Science supported by the programme, there are larger cohort sizes taking GCSE Computer Science and they obtain higher grades than schools not supported.

Simon Humphries, National Coordinator for Computing At School, explains why

the programme has been so successful. "Computing At School first established the Network of Excellence (NoE) programme in 2012. Its dual purpose: to enable teachers in England to become confident, effective and enthusiastic teachers of computing, and to develop and articulate a vision for the subject of computing at the national level. The approach taken by the NoE was in line with DfE guidance on professional development and it built on the Computing At School (CAS) model of CPD. Using an integrated approach to meet the teachers through intensive, face to face, ongoing,

peer-led, local support connected to their practice. It focused on the teaching and learning to maximise impact on pupils and to build strong relationships between the teachers themselves. These continuing relationships are an essential means of support as teachers implement and develop the knowledge rich computing curriculum in their schools.

The NoE established a network of ten Regional Centres based in leading universities who were essential in

My school has benefited hugely from the CAS project. Being a Lead School has given us enormous kudos; parents and teachers recognise our commitment to computer science

CAS MASTER TEACHER

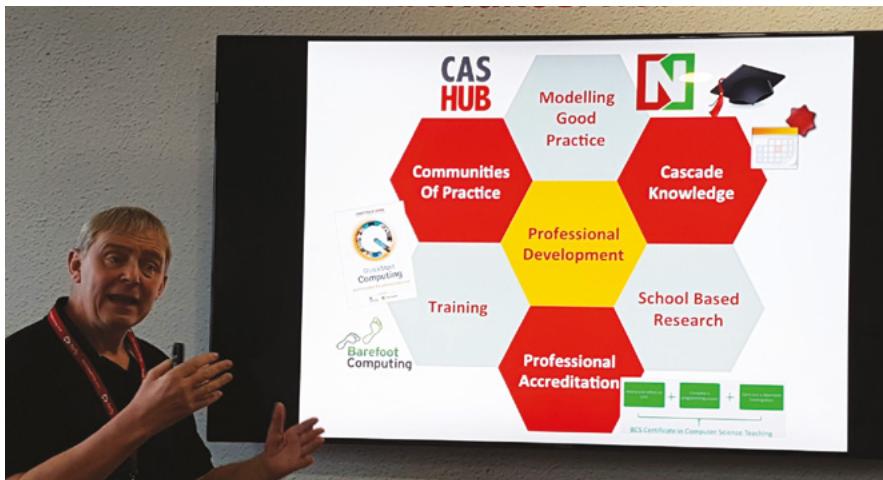
catalysing activity and support in their regions, providing far more than was contracted, or expected, as they connected with local schools and teachers, recruiting CAS Master Teachers to provide training and resources in their regions. The CAS Master Teachers have been the backbone of CPD provision for their



Six years ago, Cathryn couldn't program at all, now she runs courses on it, which she puts down to her involvement in CAS. The CPD and networking opportunities have been life changing, and Cathryn is now a Master Teacher and helps to run a new Primary CAS Hub in Warwick

CAS MASTER TEACHER





local communities. They've championed computer science in schools and the wider teaching profession, acted as specialist lead computing teachers for their areas, provided training, mentoring, and coaching to teachers in their local communities, and fostered collaboration between the schools

curriculum changes for computing. It's vital that every child has access to a high-quality computing education and for that you need teachers with the subject knowledge, classroom expertise, and confidence. The NoE has shown how this can be achieved and we look forward to

Over four years ago, pushed by an email forwarded from my head teacher, I decided to enter myself to become a Master Teacher. As an ICT leader for many years, I had a good working knowledge of IT and had a flare for creative use of technology, but as an Early Years trained teacher, I often found it hard to interact with other IT leaders or offer training due to having a lack of KS1/KS2 teaching. By becoming part of CAS, I was offered five days' training at University College London, which opened my eyes to how much I had to share with other teachers and how my views and insights were valued, from that training, I became more confident to share my good practice with others as well. This led to me training other teachers within my school, to training in my local borough and then all over London.

#### CAS MASTER TEACHER

## “ COMPUTING AT SCHOOL’S NOE PROGRAMME HAS PROVIDED OVER 80,000 TEACHER HOURS OF CPD AND SUPPORT

and universities. The NoE recruited over 500 CAS Master Teachers, exceeding the target given by the DfE.

We, CAS, remain convinced that the model espoused by the NoE works and delivers real change for the many hundreds of teachers struggling to keep abreast of

seeing this develop through the National Centre for Computing.

On 31st March 2018, the contract for the NoE, funded by the DfE, came to an end. CAS and BCS are hugely grateful to all those in the regional centres for making this possible with their host universities.



Each centre has had a significant impact in enabling and supporting activity in its region. They've covered a lot of miles and have supported a lot of teachers! The NoE has been effective, impactful, scalable, and has exceeded all targets set by the DfE for the programme. More importantly, the NoE has improved teacher confidence, subject knowledge, and pedagogical expertise.

The CAS community is thriving, with lots of local meetings right across the country aimed at supporting teachers of computing. If you're a teacher and haven't already joined your local CAS community, why not take a look at what's going on in your area? 



**AGE RANGE**

7-10 years

**LESSON TYPE**

Physical computing

**REQUIREMENTS**

- PC/laptop/tablet/mobile device
- Micro:bit
- Internet access

# TURNING UP THE HEAT

Use a micro:bit to introduce conditionals while creating an interactive thermometer to monitor your classroom's temperature levels using numbers and emoji

STORY BY Beverley McCormick

**B** BC micro:bits have been around for a few years now, but sadly aren't that common in the primary classroom. This is unfortunate, as they're simple to use and, when employed in conjunction with the online emulator, provide an excellent introduction to physical computing.

It's good to note that while engaging students in basic coding, they'll also be developing

essential 21st century skills as they collaborate with partners and communicate by talking through the process. They'll also develop critical thinking skills while they debug and stretch their creativity by considering other ways to solve the problem or use their newly developed skills.

**Anyone got a thermometer?**

Start by asking students to help you solve a real-life problem. This gives them focus and puts their learning into meaningful context. Ask if they knew that they could read the temperature without a traditional thermometer and only a micro:bit. Brainstorm their suggestions for how and where this would be useful beyond the classroom or in project-based learning.

**Forming a plan**

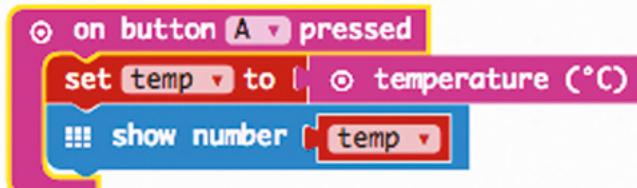
Begin by creating a pseudocode of what exactly they want the micro:bit to do – read the temperature. As

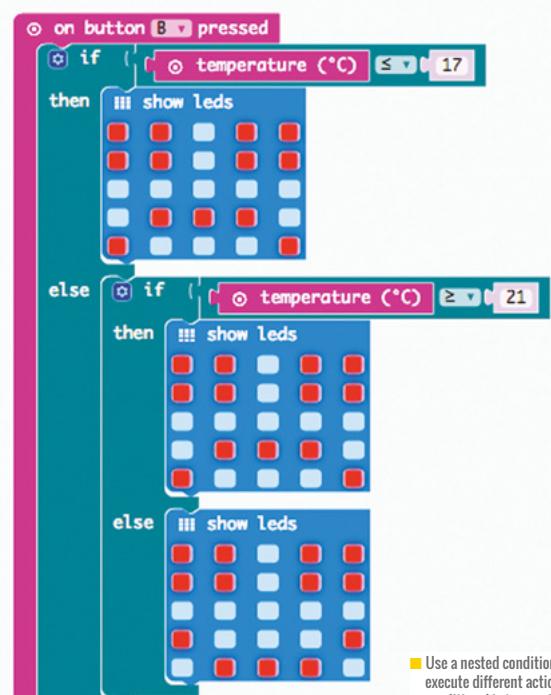
students talk through the process, they'll gain a better understanding of what each block is asking the micro:bit to do. For example, knowing that a variable is a value that can change depending on the condition or instruction will help students to understand the function in this code of calling and showing the temperature as a number.

**Getting creative**

Now that you've created a basic thermometer, it's time to add conditionals. Explain that conditional statements are used in programming to determine which actions to perform based on the evaluated condition. Using conditionals with the programmable LEDs enables the code to become more interactive, showing an image for set limits. Pupils can get creative by creating a smiley face emoji or something else. Using greater than and less than symbols supports mathematical concepts in a meaningful real-life context.

>Create a variable that will change to show the current temperature. The name doesn't matter, but it helps if you make it simple to remember. I used 'temp'.





Use a nested conditional to help the micro:bit execute different actions depending on whether the conditional is true or false.

## ASSESSMENT

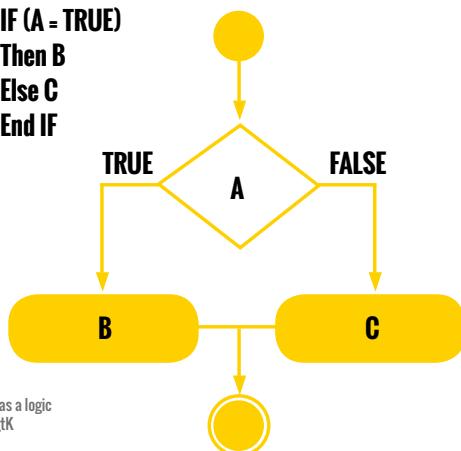
- ✓ Can you 'talk through' how the program works?
- ✓ Explain the purpose of a conditional.
- ✓ How could you modify your code to include an alert if the temperature is too hot?
- ✓ How could you improve your code so that the temperature reading runs constantly?
- ✓ Could you modify the code to use it for a classroom noise monitor?

### Exploring the lesson plan in detail

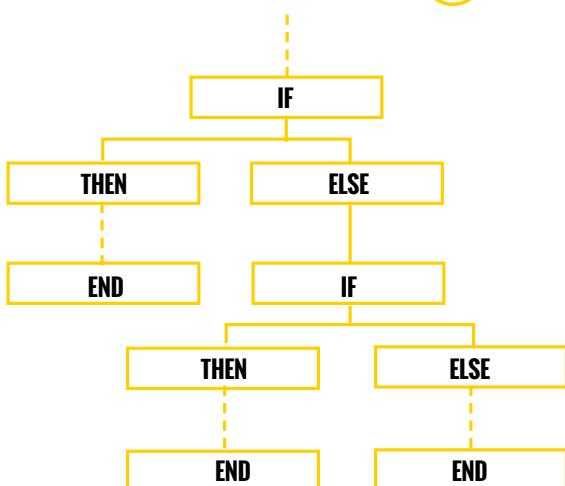
Integrating coding in the primary school across the curriculum in a meaningful way is essential to ensuring that students assign value to this skill. Creating code that can be used in project work to collect data is a great way to develop transferable skills. Can students think of situations beyond the classroom where this code could be useful? Could they give suggestions on how to enhance this code with sound? (HW)

**IF (A = TRUE)**

Then B  
Else C  
End IF



Conditional statements visualised as a logic decision tree: helloworld.cc/2MPYgtK



### ALTERNATIVE ACTIVITY IDEAS

This lesson could be adapted for use with students of other ages and abilities.

#### 5-6 years - Unplugged

- ✓ Support students to understand conditionals through simple scenarios. For example, if it's raining, put on a coat, else don't put on a coat.
- ✓ Find conditionals in stories such as 'Goldilocks and the Three Bears' – if the porridge is too hot, don't eat, else eat it!

#### 7-10 years - Unplugged

- ✓ Encourage students to have a go at writing pseudocode, then talk through their code, explaining the process and what they want it to do to a partner.

#### 11-13 years - Programming

- ✓ Older learners can use a text-based programming language such as Python or JavaScript to code.

## FURTHER READING

More micro:bit ideas: [www.microbit.org/ideas](http://www.microbit.org/ideas)

[helloworld.cc/2wgEpKC](http://helloworld.cc/2wgEpKC) [www.barefootcas.org.uk](http://www.barefootcas.org.uk)

**AGE RANGE**

8-13 years

**LESSON TYPE**Text-based  
programming**REQUIREMENTS**

- Raspberry Pi 3 or 3B+
- Project instructions, which you can find at [helloworld.cc/2Mz96F5](http://helloworld.cc/2Mz96F5)

# PAC-MAN TREASURE HUNT ON THE RASPBERRY PI'S TERMINAL

Hack the Pi's terminal and learn basic cybersecurity skills through a Pac-Man-themed treasure hunt

STORY BY Caitlyn Merry

**W**hile I was Head of Computer Science at a secondary school in London, I created this lesson for a Year 8 hackathon to give them a taste of GCSE Computer Science. It helped me launch the subject, to later become the most popular GCSE option.

Using a computer's Terminal or Shell is as useful as it is appealing. Learning how to navigate the Terminal is often the first step towards a real understanding of the workings of a computer, and I found that the cybersecurity gamification of this resource motivates learners of all ages.

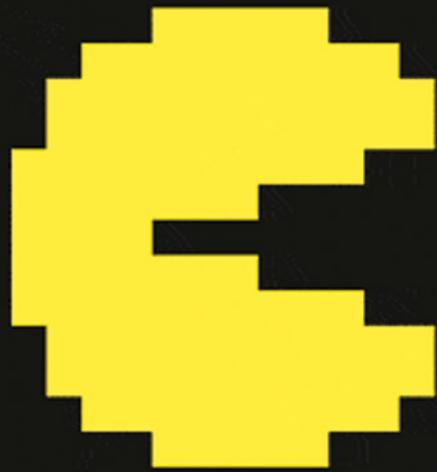
Acting as ethical hackers, students navigate the Terminal using Bash commands in a race to find all the Pac-Man 'ghosts'. As these ghosts files are viruses, they create a quarantine folder which they'll copy the ghost files into before removing them. Each ghost contains treasure, and they learn how to use Nano to find out how much treasure they've found.

## THE CHALLENGE

- ✓ Be introduced to the mission: hack the terminal to find all the Pac-Man 'ghosts'
- ✓ Learn how to run a script from the command line to start the treasure hunt
- ✓ Use Bash commands like 'ls', 'pwd', and 'cd' to navigate the terminal
- ✓ Find the ghost 'virus' files
- ✓ Create a quarantine directory and copy the files into it for safety
- ✓ Open the quarantined files and collect your treasure!
- ✓ Safely remove the quarantine folder and all 'infected' files

File Edit Tabs Help

pi@raspberrypi:~ \$ cd



## Exploring the lesson plan in detail

There are a few concepts in teaching pedagogy that inspired me to create this resource.

Learning commands can be very repetitive, so I used gamification to make practising commands fun.

Using Pac-Man as a model also gets across the idea that the command line is like a maze, and makes it clear that viruses are challenges, like the Pac-Man ghosts, that need to be eliminated.

This resource is also designed to give students a taste of what it is

like to be an ethical hacker, a career that is increasingly important yet suffers a huge skills shortage. Thus, it mimics in a simplified manner the style of real hacking competitions and challenges such as the popular 'capture the flag'. [\(HW\)](#)

## ALTERNATIVE ACTIVITY IDEAS

There's plenty of room for extending this activity. It could be easily adapted for use in any Linux, Unix or Windows terminal or emulator.

### Explore and make

There are numerous ways you could allow learners to implement this project-based learning, on a variety of levels. Here are some ideas of varying levels of difficulty:

- ✓ For younger or less experienced learners, have them simply find and delete the ghost virus files
- ✓ For more experienced learners, get them to plant their own ghost virus in the system and have another group or learner find it
- ✓ Make it a competition or race to find all the ghosts

## ASSESSMENT

- ✓ How do hackers make it difficult to find viruses?
- ✓ How do ethical hackers help keep computers safe?
- ✓ What are some differences between using the command line and using the regular interface?

## FURTHER READING

### Cybersecurity lesson resources:

[helloworld.cc/2LsYSk8](http://helloworld.cc/2LsYSk8)

### Ethical hacking:

[helloworld.cc/2MUFdyf](http://helloworld.cc/2MUFdyf)

This is just an example of some of the resources you can find that give an overview of ethical hacking

## AGE RANGE

7-10 years

## LESSON TYPE

Visual / block-based coding

## REQUIREMENTS

Computers running Scratch (1.4 or 2.0) for each group

# MAKE A SINGING PIANO IN SCRATCH

In this lesson, you'll create a piano in Scratch that plays recorded singing notes and lyrics, much to the amusement or consternation of your teacher!

STORY BY Jody Carter

The record function in Scratch gets children excited about creating programs with the thrill of hearing their own voices being played, and is the focus of the Singing Piano. This can be split into a series of lessons, where children create a musical instrument through recording and playing their own singing voices. The first lesson introduces sounds with only a few coding blocks, so children can quickly create a functioning piano. In the next lesson, children record their own voices to replace

the electronic notes with their own singing voice. Finally, using a variable, children make a piano that sings the lyrics to songs they've recorded. Each lesson progresses from the next by using similar code blocks, moving from playing a simple note to playing a recorded note to playing a recorded lyric with the addition of a variable.

## A basic piano

First, we create a piano with a range of notes to play Frère Jacques ([helloworld.cc/2nXq9BO](http://helloworld.cc/2nXq9BO)). The piano has basic functionality: when a key is clicked, it will change colour, play a note, then change colour back again.

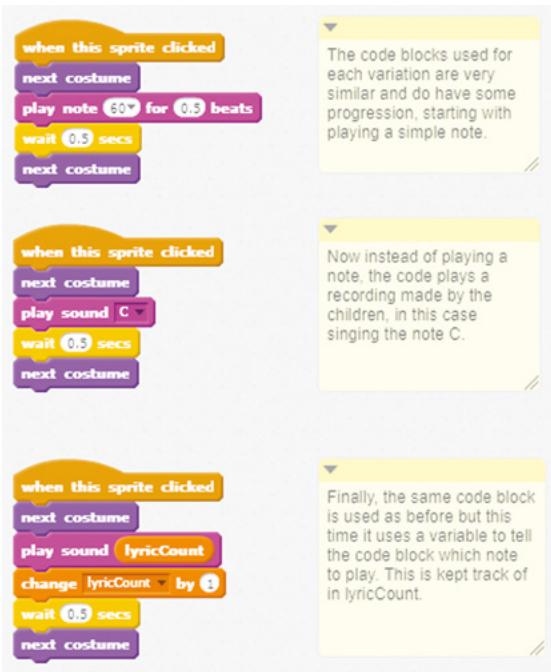
Demo the working piano to the children, but rather than have them simply copy the code, get them to think through the functionality of the piano. What do they want it to do? What would this look like and how would it work? Creating annotated design plans is great so they can see the piano on paper before they start coding, and it also allows for a reflection at the end. When introducing the code, challenge the children to try putting the blocks back together or spot the error in the code. Can they find

a different way of achieving the same goal using different code blocks? Does the code work the same way if you put the blocks in a different order? All of this will hopefully reinforce a deeper understanding.

Start by demonstrating how to create a sprite to represent a piano key, a costume for the 'on' state and adding in code blocks to add the functionality. Each key then needs to be duplicated and the note played changed. For Frère Jacques, we only need five notes, C-G (values 60, 62, 64, 65, and 67), and I always make sure the keys are different colours. Let the children get on with this part themselves and give them some time to experiment with playing different notes. Once they have the basic piano setup, give them some tasks based around these questions: how many different notes are there? Is there a maximum or minimum value you can enter for a note? What happens if you type in a letter or word instead of a note? How can you change how long the note is played for and how loud/quiet it's played?

## Recording your voice

We now need to swap the electronic piano sound with a recorded version where the note is sung instead. This is fairly straightforward, but the children



The code looks simple enough, but can you set challenges to provide the depth of learning for each child?

will need to practise in order to not have a lot of dead air. Again, give them a few minutes to get used to hearing their own recorded voices out of their system after it has been introduced. Once done, you can use a real piano or other musical instrument so that each note is sung more accurately. I find it best to let the children record in their own time rather than all at once to avoid a chorus of voices, and make sure to give the recordings a suitable name, for example the name of the note. Let the children experiment with editing their recordings and adding effects. Remember to set small tasks to guide them with their learning, for example, how can you make it fade in/out? Can sections of the recording be moved or duplicated? At this point, the functionality of the piano is essentially the same as before, but uses the children's

## MUSIC AND CODING

Using music and coding is a brilliant way to engage children and to make those all-important cross-curricular links. Sonic Pi is popular and fairly straightforward to get going. The added advantage is that you can create all sorts of different sounds, including drum beats and different instruments all layered on top of each other. See [helloworld.cc/2nV8w5H](http://helloworld.cc/2nV8w5H) for more.



**New sound:** F

1 F 00:01.08

Microphone volume: 0

■ Give the children some time to play around with recording their own voice

voices to sing the notes instead, so swap the play note block with the play sound block.

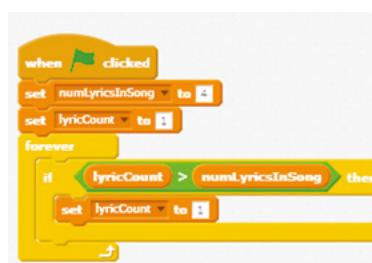
### Singing a song

The next step is to record lyrics to a song that is played when pressing the piano keys. This uses a variable to keep track of the lyric count. Variables are

used to store values that can be accessed from sprites in Scratch. Instead of recording a note being sung, the words to a song can be recorded and played when pressing a key. See a working version at [helloworld.cc/2whamlm](http://helloworld.cc/2whamlm) for a better idea of what I'm talking about (but please excuse my singing!).

In essence, whichever piano key is pressed, it will play the next lyric of the song in that note. So for 'Happy Birthday', you would record each word: happy, birthday, to and you, for each of the four notes used in the first part of the song. Each lyric is then named 1, 2, 3 and 4 to keep the correct order. When a key is pressed, it checks the variable `lyricCount` and plays the correct lyric (if `lyricCount = 1`, it sings happy; `lyricCount = 2`, it sings birthday, and so on). To get this working, we need to use the play sound block, but instead of selecting a sound to play from the dropdown, add the `lyricCount` variable. Now add in the code block to increase `lyricCount` by 1. This needs to be replicated for each of the keys.

For fun, you can further adapt this to turn it into a singing insult generator (you'll be the best judge as to whether this is appropriate for your class) using the tune to 'Happy Birthday', but to sing 'You are a nincompoop' instead. The children can then come up with their own insults and make the recordings themselves. It will also give children a further opportunity to use their learning, and you can challenge them in any of the ways mentioned. (HW)



■ Initialise the variable at the start and to keep tabs on how many words have been sung

This initialises the code, setting `lyricCount` to 1.

It also keeps track of `lyricCount` (the number of times the keys have been pressed) and this is where you tell it how many lyrics there are to sing, in this case set in variable `numLyricsInSong` as 4. If `lyricCount` is greater it is reset to the beginning. If there are 5 words to be sung, change the 4 to a 5 etc.

# SET A CHALLENGE FOR LEARNERS AND TEACHERS

Developing computational thinking is core to the computing programme of study but also to developing independent learning in all areas of the curriculum

GUIDE BY Julia Briggs

In 2013, action research with a group of schools in Somerset identified benefits to learners through appropriate learning experiences with Scratch. These included 'developing independent learning', 'developing logical thinking', and 'encouraging exploration' (Briggs, 2013). Following active interactions with Scratch software, children spoke about how they learnt and what they learnt. These contributed to a model of computational thinking, which includes attitudes as well as skills, as envisaged by Jeannette Wing.

For children to realise universal benefits from those attitudes and skills, schools have a challenge to harness the computational thinking they're developing, through appropriate programming experiences. Children can be supported to an awareness of its application to other learning and to appreciate the contribution it can make. In this way, they're building knowledge, understanding, and skills for continuous lifelong learning.

In Somerset, we've developed open-ended challenges to use the computational thinking skills, and attitudes gained

through appropriate programming experiences, within all aspects of computing. The challenges are set within contexts of other learning, such as geography, science or mathematics.

This article describes:

- Preparing children to start using computational thinking
- Setting challenges to use the computational thinking developed
- Implementing a challenge
- Teacher characteristics required to develop the use of challenges

## Preparing learners

**1. Plan for children to DO** appropriate programming activities. They need to make mistakes and persevere without too high a level of frustration.

**2. Build metacognition of computational thinking** as you ASK THE QUESTIONS to illicit children's awareness. What have you learned? How did you learn it?

ATTITUDES	SKILLS
Making mistakes	Pattern recognition
Perseverance	Decomposition
Imagination	Algorithm design
Collaboration	Abstraction and generalisation

Self-reflection will need modelling and scaffolding, but this is time well spent at the end of activities.

**3. Provide opportunities for children to REVIEW** how the attitudes and skills support their learning and confidence at other times. This could be playground, class or home situations.

## Setting a challenge

**1. Identify context** A challenge can be set in the context of the learning that is currently a class focus such as a study of a local river. It could be a response to recent news or events, such as helping the headteacher to avoid a security breach of the school network.

**2. Select learning outcomes or assessment focus** The teacher will be clear about the learning outcomes they're planning for but, at the same time, be open to other learning that will take place. The example challenges we've created in Somerset link primarily to computing objectives but also relate to outcomes in other curriculum subjects. The wording for the challenge can enable a teacher to focus on an expected outcome they want to assess or could provide a learning opportunity for new skills and knowledge.

**“ COMPUTATIONAL THINKING REPRESENTS A UNIVERSALLY APPLICABLE ATTITUDE AND SKILL SET EVERYONE, NOT JUST COMPUTER SCIENTISTS, WOULD BE EAGER TO LEARN AND USE. WING, 2006: 33**

## Implementing a challenge

The teacher will need to support children to identify the computational thinking they're using and also to remind them to think about any online safety considerations that must be part of their use of technology during the challenge.

So, what does this look like in a classroom? A challenge can:

- Be used as a whole class activity or assigned to a group or pair of children
- Provide opportunities for children to plan, develop, and evaluate their use of technology
- Allow children to demonstrate their attainment in computing, including developing 'mastery' of aspects of computing once learners are working at age expectations
- Support teachers to differentiate learning experiences to meet the needs of their learners

An individual or pair of children could work on a challenge independently while other children are being supported by the teacher or working with another adult in the class.

## PROCESS OF WORKING THROUGH A CHALLENGE

### EXPLORE AND DEFINE

When children first read the challenge, they need to take time to understand what the challenge requires. Are there words or phrases they're not sure about yet? What do they already know? What will they need to know?

Model or guide children to underline the 'not sure about yet' and to make jottings of what they know and need to know. Support them to recognise what is being asked for.

ABSTRACTION to identify core of the problem.

### IMAGINE AND IDENTIFY PROCESS

Support children to discuss what an outcome to meet the challenge could look like. What will they do to achieve this? What will they need to do first? What comes next?

Initially they may need scaffolding to identify the process and the tools they'll use, particularly where you are working with younger learners. Over time, and for more confident children, they can work independently.

IMAGINATION to consider what the outcome could look like and ways to solve the problem

GENERALISATION and PATTERN RECOGNITION to consider prior experiences which can help to solve this problem.

DECOMPOSITION to break down the problem.

ALGORITHM design to plan the process.

### DO AND REVIEW

Children have time to work through the planned process, reviewing its effectiveness and adapting it as they work together.

COLLABORATION to work together to achieve the required outcome.

PERSEVERANCE to be determined to make the solution work.

Willingness to MAKE MISTAKES and find solutions when plans do not go the way you want.

### EVALUATE AND CELEBRATE

Children evaluate the outcome based on how they envisaged it, and changes they made each time they reviewed their work. They celebrate their successes and plan for improvements for future work.

GENERALISATION AND PATTERN RECOGNITION to consider ways in which this may help solve problems in the future.

## YEAR 1

### PLANTS

The town gardening club want to know whether their plants are getting enough water. How can we help them? Will the plants need watering tomorrow?

## YEAR 3

### DANCE

Our school is running a dance competition. We need to create a program to show examples of dance sequences those entering could use. How can we do this?

## YEAR 5

### FIRE ALARM

The school needs a new fire alarm. What sound will you create? How do you know everyone will hear it?

## YEAR 2

### PATH SHAPE

The Year 3 teacher would like to do a treasure hunt using  $\frac{1}{4}$ ,  $\frac{1}{2}$ ,  $\frac{3}{4}$  and whole turns. What shape path should the treasure hunt follow to show the children understand these?

## YEAR 4

### SWEET STORE

The school governors have told us we must find a new place in school to store chocolates and sweets for the summer fair. You need to prove to them there is a good place, or we might not be able to sell them!

## YEAR 6

### ACTIVITY GUIDE

Our headteacher has asked us to create a guide for families to help them find online activities they can enjoy doing together.

■ Other challenges used by teachers working with different primary ages

► A mixed Year 3 and 4 class were set the challenge: a company is thinking about opening a local shop in our village/part of town. They want advice on whether they would have customers and what should be stocked. What should the shop look like? And how should it be organised?

This was the first time they had experienced a challenge where they were expected to decide the work they would need to do. Consequently, the teacher decided to plan this as a whole class learning experience where she modelled parts of the process and guided children to work in small groups for other parts. What surprised the teacher was the way in which they quickly became independent and provided support for each other.

Following the challenge, the teacher reported:

- “I was able to provide a mix of experiences to the class”
- “Children had the chance to make decisions”
- “I could see what the children had learnt over the year”

Other challenges used by teachers working with different primary ages are shown on the previous page.

### **Challenge for teachers**

The use of a challenge approach enables computational thinking to be applied in all aspects of the computing curriculum and across other curriculum areas. It requires teachers to be resilient and demonstrate characteristics that underpin effective teaching in all subject areas:

- Take risks
- Acquire sufficient subject knowledge
- Recognise knowledge of learners
- Ask probing questions
- Be willing to have a go
- Allow learners to lead the learning

## **PROCESS OF WORKING THROUGH THE EXAMPLE SHOP CHALLENGE**

### **EXPLORE AND DEFINE**

The children discussed what a local shop was and who the customers might be. The teacher recognised the need for them to understand what a local shop provides. Photos of a shop in a neighbouring village were discussed and a gentleman who remembered a shop being in their village was interviewed.

### **IMAGINE AND IDENTIFY PROCESS**

The teacher guided the children to agree together the information a company might require. This included a customer questionnaire, a suggested floor plan, design for the outside of the shop, colour scheme for the shop including uniforms for shop assistants, possible plots of land in the best position in the village, and an overall report for the company. The teacher modelled the decomposition of this into tasks and establishing an algorithm for each task. Children were put into a small group and each assigned a task using either laptops or iPads.

### **DO AND REVIEW**

Most children worked in pairs within their group, making use of skills they had gained earlier in the year. One pair worked on the questionnaire and initially asked for support to make sure they understood the software they were using, but one of the girls quickly gained confidence and led that task. The group working on the overall report independently decided to create a presentation. Without intervention from the teacher they agreed a layout and then began to collect outcomes from the other group. A clear leader emerged, who was guided to equip others in the group to complete parts of the task. In other groups, those with skills provided the input other children needed to achieve the outcome they wanted without any prompting from an adult. The children used PowerPoint, 2Investigate, PicCollage, Book Creator, and Google Earth. The teacher provided QR codes for additional information such as population numbers.

### **EVALUATE AND CELEBRATE**

Most of the work was completed within one day which allowed time for the children to celebrate their outcomes via the presentation that had been created. This was revisited a few days later for children to evaluate the thinking they had done and effectiveness of the information they could present to the company.

- Set appropriate challenge
- Achieve balance of child initiated and teacher modelled
- Allow time for thinking, changing mind, and discovering new ways of doing things
- Recognise there is more than one way to achieve something
- Scaffolded (guided by adult to achieve appropriate outcomes)
- Independent (working by themselves with appropriate interventions from an adult to achieve appropriate outcomes)
- Interdependent (working independently with others to achieve appropriate outcomes)

The teacher will need to know their learners, identifying those:

- Requiring support (working with an adult to achieve appropriate outcomes)

This is an approach that brings excitement for the teacher and children, and helps make sense of the purpose of the computing programme of study. **(HW)**

# USING SCRATCH AND MAKEY MAKEY TO HELP STUDY MOTION

A computer-based method to engage students in measurement of average speed and acceleration

STORY BY Steven Weir

## Introduction

The study of motion of objects, including average speed, velocity, and acceleration, is encountered at Key Stage 3 and GCSE. The calculation of average speed can be easily achieved by measuring the time (using a stopwatch) for an object to travel a known distance, then dividing the latter by the former. The computation of acceleration from experimental measurements requires equipment such as light gates attached to a data logger. This facilitates the determination of initial velocity and final velocity, and thus the calculation of change in velocity, which can then be divided by the total journey time to calculate acceleration. This article shows an approach to determining motion properties of a trolley on a slope using a simple Scratch program and a Makey Makey. **Figure 1** shows the finished experimental setup. The slope is a length of plastic guttering.

## Motion detection

As the trolley travels down the slope, motion at certain points is detected by a reed switch activated by a magnet located inside the trolley (see **Figure 2**). As the reed switch closes, it completes a circuit, which the Makey Makey senses and captures a time for. The time is captured by wiring reed switches to the up, right, down and left inputs of the Makey Makey and earth. The associated Scratch

programming and a schematic showing the position of four ball sprite characters to help visualise the position of time readings from the slope is shown in **Figure 3** along with example generated data.

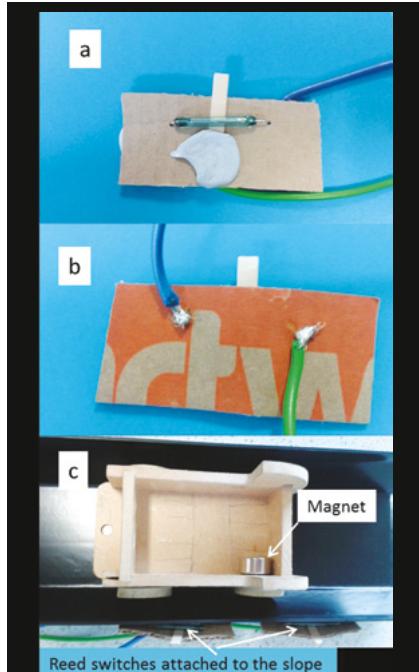
## Calculations

Using the time data generated from the Makey Makey and the measurement of distances between the reed switches, it's possible to calculate average speed at several points on the slope using the equation:  $\text{Average speed} = \frac{\text{Distance}}{\text{Time}}$ . If the distance is short between switches, this will minimise the effect of acceleration on the average speed or velocity during that part of the trolley's journey. Thus, values for initial velocity and final velocity can be calculated using:  $\frac{\text{Distance}}{\text{Time}}$ . By measurement of the total journey, time acceleration can be calculated using:  $\text{Acceleration} = \frac{(\text{Final velocity} - \text{Initial velocity})}{\text{Total journey time}}$ .

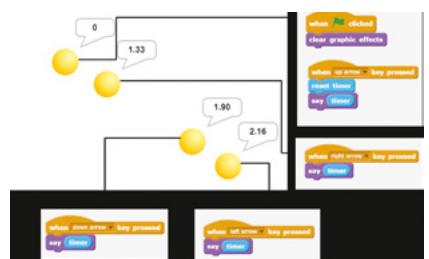
Total journey time



**Figure 1.** Finished experimental setup of a trolley on a slope attached to a Makey Makey and laptop running a Scratch program



**Figure 2.** a. A reed switch attached to a piece of card, a wooden stick so its location is readily visible when in-situ and Blu Tack to attach the unit to the slope. b. Reverse side of image a showing the wires attached to the reed switch. c. A trolley fitted with a magnet and two reed switches attached to the slope



**Figure 3.** Scratch programming associated with four ball sprite characters arranged to help visualise where readings were taken on the slope



GREG MICHAELSON HERIOT-WATT UNIVERSITY

# PROGRAMS AND PARADOXES

We can find paradoxes in self-referential uses of programming languages

**A** paradox is a statement that is self-contradictory; that is, it asserts its own truth and falsehood at the same time. As we shall see, we can find paradoxes in self-referential uses of programming languages, even in something as innocuous as the Linux or Windows filesystem.

We take it for granted that we use programs to make other programs. In particular, we use language processors, like compilers and interpreters, to check and run our code. And we tend to treat such language processors as the standards for whether or not our programs satisfy the requirements of the language.

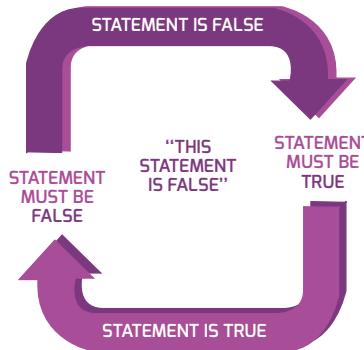
We have to write a language processor in some programming language. In effect, we're using that language to describe the language to be processed. We call a language for describing other languages a metalanguage, from the Greek 'meta' for 'over'. For example, if we write a Python interpreter in Java, we're using Java as a metalanguage for Python.

We can use a programming language to write language

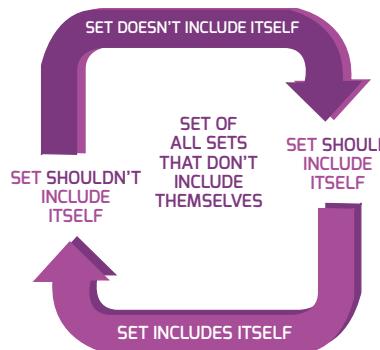
processors for that language itself. For example, suppose we have a working Python interpreter written in Java. We could then use it to run any Python program, including a compiler for Python written in Python. For our compiler, we've then used Python as its own metalanguage. Our compiler can be used to compile any Python program, including itself. So we could compile our compiler through itself running on the original Java interpreter.

Now, an intractable problem with languages that can be used to talk about themselves is they give rise to paradoxes. One of the oldest known paradoxes, the Liar Paradox, has variants going back as far as 600 BCE. A stripped-back modern version is simply: "This statement is false". If the statement is false, it must be true. And if it's true then it must be false (see **Figure 1**). This arises because we can use a human language, in this case English, as its own metalanguage.

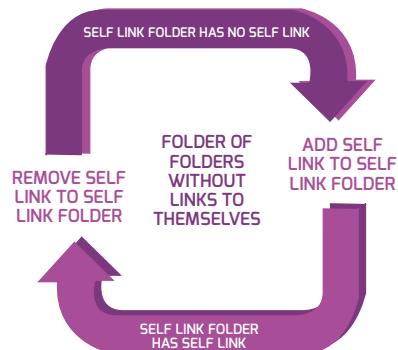
At the start of the 20th century, the British philosopher Bertrand Russell blew a hole through the international project of using mathematics to describe itself with what



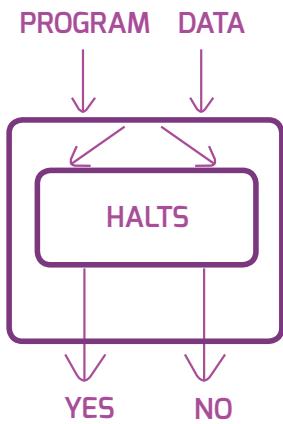
■ Figure 1. Liar paradox



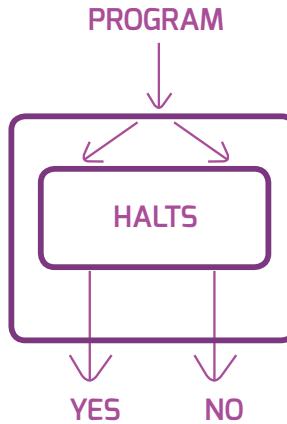
■ Figure 2. Russell's paradox



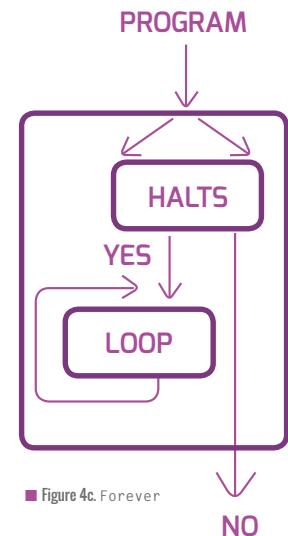
■ Figure 3. Folder paradox



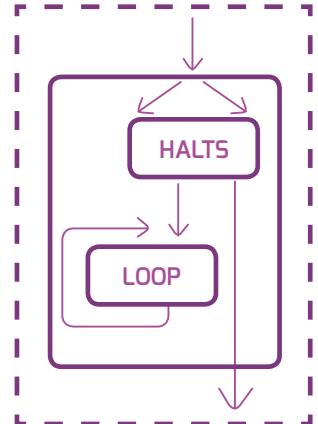
■ Figure 4a. Halts



■ Figure 4b. SelfHalts



■ Figure 4c. Forever



■ Figure 4d. Forever self-applied

is now known as Russell's Paradox. Russell was exploring sets that have other sets as members. For example, we might have a set of dogs and a set of cats and put them both in a set of animals.

Russell wanted to know whether a set could include itself as a member. That doesn't seem like a very useful thing to do. Still, given a whole load of sets, suppose we try to construct a set of all the sets that don't include themselves and see if there are any left over. Now we have a problem with where to put that set of sets. If it isn't in itself, then it should be in itself. And if it is in itself, then it shouldn't be in itself (see **Figure 2**).

Russell's paradox may seem pretty arcane, but it's implicit in computer filesystems. It's very convenient to be able to make shortcut links to folders and put them in other folders, especially on the desktop, which is just a folder that is always displayed. And it's easy, if eccentric, to put a shortcut to a folder in that folder itself. So we could divide up all our folders into two groups: those that do have links to themselves and those that don't.

Given a language library that exposes the filesystem, it's straightforward to write a program that repeatedly checks for folders that don't have links to themselves and keeps track of them. The program begins by making an empty `NoSelfLink` folder. Then it starts at the root and works down through the file hierarchy checking each folder. If a folder doesn't have a link to itself, and it isn't already in `NoSelfLink`, then the program adds a link for that folder into `NoSelfLink`. And if a folder is in `NoSelfLink`, but has acquired a link to itself since the last check, then the link in `NoSelfLink` is removed.

Note that `NoSelfLink` is just another folder and that it starts out empty, without a link to itself. Then, at the end of the first check, `NoSelfLink` will have a link to itself. And, during the second check, the program will find that `NoSelfLink` now has a link to itself, so that link is removed (see **Figure 3**).

Alan Turing was the first person to recognise the possibility of writing programs that could manipulate their own representations. In his seminal 1936 paper, he showed how to define very simple computations, now known as Turing machines, based on a tape of symbols manipulated by rules to inspect and change them. Turing then defined a universal Turing machine that could execute a symbolic representation of any other Turing machine, including a universal Turing machine.

In a fundamental result, Turing showed that it's not possible to tell if an arbitrary Turing machine ever stops, by constructing an elegant paradox of self-reference. This is a simplified account.

Let's suppose we can write a language processor called `Halts` to check whether or not an input program terminates for given data. Note that `Halts` always terminates (see **Figure 4a**). We can make a variant of `Halts` called `SelfHalts`, which halts if its input program terminates with itself as input. That seems like a reasonable thing to consider: for example, we expect a compiler compiling itself to terminate (see **Figure 4b**).

Now for the paradox. How about a program called `Forever`, that runs forever if an input program terminates on itself (see **Figure 4c**)? Finally, we can apply `Forever` to itself (see **Figure 4d**). Alas, if `Forever` terminates with itself as input then it runs forever, and if it doesn't terminate with itself, it terminates.

Turing concluded that it wasn't possible to write a language processor like `Halts` for Turing machines; that is, the Halting Problem is undecidable. In highly practical terms, this means that we can't write a language processor to check how much memory or time or energy an arbitrary program requires.

At least there will always be lots of jobs for people with strong software analysis skills! (HW)

# HOW TO IMPROVE CREATIVE WRITING USING A SIMPLE BLOG

Are you frustrated that you have no time for creative writing? 100 Word Challenge is a free resource that provides great motivation for turning your writers into authors

STORY BY Julia Skinner

**T**he old adage of 'practice makes perfect' is all well and good, but time is often the enemy in a busy classroom. Do we get time to embed skills when there are so many areas to cover? For teachers of older students, the problem may not be as acute, as you'll usually be a specialist and therefore concentrate on that subject, and the timetable will include 'practice' sessions. For the primary age practitioner, however, it can be more difficult because the list of subjects to cover grows with each year.

In recent years, the emergence of grammar lessons has not only made news headlines, but added to the time problem. In the UK, primary colleagues have the dreaded SPaG to deal with. In case you haven't come across this initiative, it stands for Spelling, Punctuation and Grammar. It's taught quite formally and the children are tested on it externally at the end of Year 6. The test is a series of questions along the lines of 'comprehension'. Writing is also assessed, but internally by teachers using a checklist linked to the elements of SPaG.

For me, that is a lost opportunity. Surely, the intention of teaching SPaG is to promote the technical skills of writing, so that when students write, it's grammatically correct. But when do they actually 'write'? Writing isn't grammar. Writing is a creative act, where ideas are communicated via the written word. Writing is to engage a reader, to share with an audience:

'The unread story is not a story; it is little black marks on wood pulp. The reader, reading it makes it live; a live thing, a story.'

Ursula K Le Guin

■ Students sharing their work via Skype at St. Ignatius of Antioch, Yardley, Pennsylvania, USA



## 5 SENTENCE CHALLENGE

For those emerging writers who may find writing 100 words using a specific prompt too much, 5 Sentence Challenge ([fivesc.net](http://fivesc.net)) provides a similar experience but with more freedom. It runs every two weeks and the prompt is usually a busy picture or a theme.

Teachers of both young writers and EAL students have found it an ideal resource.



### What is 100 Word Challenge?

Writing creatively can be daunting for pupils. They don't see examples of it in their everyday lives. Apart from teachers marking books or peers doing the same exercise, when do they see people writing? A series of questions can often accompany the request to write:

- What am I going to write about?
- How long does it have to be?
- What is the point?

100 Word Challenge ([100wc.net](http://100wc.net)) provides an answer to all those questions and more, and provides an excellent opportunity to hone those SPaG skills.

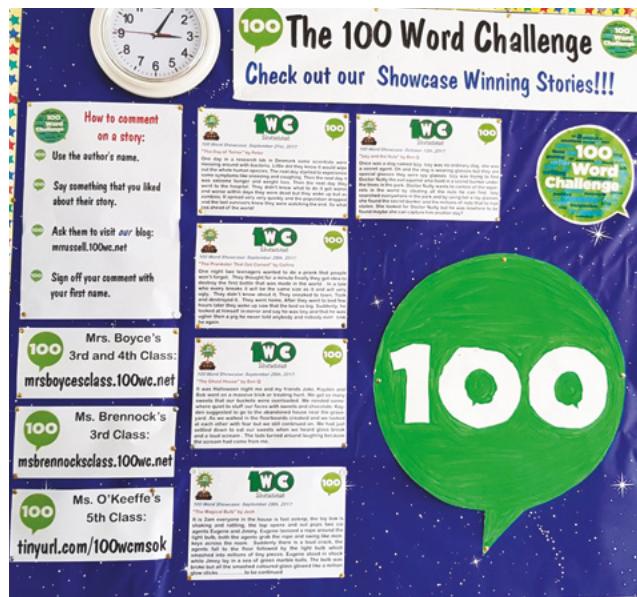
100WC is a simple, free writing project for young writers up to 16 years. A weekly prompt of a picture, a few individual words or part of a sentence is set, and the pupils respond by writing a creative piece that is around 100 words in length. The restricted wordage appeals to the full ability range. Those who find writing difficult have 'only 100 words' to write, while those who write freely have to restrict their ideas to around a couple of paragraphs.

One of the key factors in getting students to write creatively is not only giving them time, but providing them with a purpose for doing it beyond keeping their teacher happy. Unless you're the keeper of a secret diary, as a writer you want an audience. Having someone read your words not only to give you a mark but to engage with your thoughts is the motivation that all authors start with. 100 Word Challenge provides

that purpose by extending the audience way beyond the walls of the classroom. But where do you find this audience? Actually, the audience is the whole world and you reach them by blogging.

### Blogging

For some in education, blogs are like the devil incarnate. They allow the big, bad world into schools with trolls and inappropriate comments.



■ Posts can be selected for the showcase. Here's the celebration wall belonging to 6th Class at St Stephen's De La Salle, Waterford, Ireland

In actual fact, a class blog is one of the safest places for students to share their work. When blogs are created, there's a series of boxes that can be ticked, giving full moderating control to the teacher. This means that nothing is published

by the students or visitors without the teacher's agreement.

So, back to the audience. Once the 100-word piece is written, it's published on a simple blog and linked to [100wc.net](http://100wc.net). By publishing writing on a blog, the URL (address at the top of the webpage) can be copied and pasted into the simple form on [100wc.net](http://100wc.net). This will add the piece to a list of hyperlinks. Once linked, a team of volunteers (Team 100) from across the world can click on this link to see the blog post, and then read it and leave a constructive comment. This is where the magic happens! It's very simple and it means that the writing is appreciated without any of the baggage that can influence 'marking' in a school. Team 100 only know the writing, not the history of the author. For the writer, the world has visited!

### From writers to authors

To know they're free to write whatever the prompt ignites in them, to know there's no wrong answer and that someone other than their class teacher will be engaging

with their work can be very powerful to young writers. It makes them want to take part. When you point out that they've gone from being a writer to an author, their feeling about writing changes completely. They realise that having an audience for their work puts them in the same category as Roald Dahl, J.K. Rowling, David Walliams, and other authors who make up their reading diet. (HW)

**Julia** is a retired headteacher whose passion for creative writing led her to develop 100 Word Challenge. She has spoken at international conferences and visits classrooms both in person and via Skype to model the impact the 100WC can have. Follow her @theheadsoffice.

# COMPUTER SCIENCE IN IRISH SECOND-LEVEL EDUCATION

After many decades of false dawns, Ireland is introducing a rigorous and stand-alone computer science subject to senior second-level education

STORY BY Stephen Murphy



Stephen Murphy of the CSTAI meets with the CoderDojo Team from Cork Institute of Technology. Teaching resources and ideas were exchanged at this meeting. The two organisations are looking to work together more in the future

## Leaving Certificate Computer Science

Computer science is coming into Irish education as a rigorous STEM subject. In September 2018, 40 secondary schools around the country will pilot the programme to Leaving Certificate Level (A Level equivalent).

The Leaving Certificate Computer Science (LCCS) specification was written based on an international standard of best practice in computer science education in secondary schools in Ontario, New Zealand, England, Scotland and Israel.

The LCCS specification is composed of three strands:

1. Practices and Principles
2. Core Concepts
3. Computer Science in Practice

The Learning Outcomes for these three strands are interwoven and can be taught in any order. A brief description taken from the specification for each strand follows:

### Strand 1: Practices and Principles

The overarching practices and principles of computer science are the behaviours and ways of thinking that computer scientists use. This strand underpins the specification and is fundamental to all learning activities.

### Strand 2: Core Concepts

The core concepts of computer science represent the major areas in the field of computer science: abstraction, data, computer systems, algorithms, and evaluation/testing. Students engage with the core concepts theoretically and practically in this strand.

### Strand 3: Computer Science in Practice

Computer science in practice provides multiple opportunities for students to apply the practices and principles, and the core concepts. Students work in teams to carry out four applied learning tasks over the duration of the course, each of which results in the creation of a real or virtual computational artefact. These artefacts should relate to the students' lives and interests. Where possible, the artefacts should be beneficial to the community and society in general.

The main challenges of computer science education are:

- Equipment/logistics
- Teacher availability

STRAND 1	STRAND 2	STRAND 3
Computers and society	Abstraction	Interactive information systems
Computational thinking	Algorithms	Analytics
Design and development	Computer systems	Modelling and simulation
	Data	Embedded systems
	Evaluation and testing	

■ Embracing the pedagogy of the specification

■ Second/third wave of schools will be problematic due to lack of proper pedagogy

The key messages to take away from the LCCS specification are:

■ LCCS is for everyone

■ There are many ways to use the specification

■ The Learning Outcomes can be taught in any order

■ LCCS should be taught in a constructivist manner

■ LCCS is a practical subject like music, not physics

■ Digital technologies will enhance collaboration, learning, and reflection

■ Both the students and teachers must have a growth mindset

## The Computer Science Teachers' Association of Ireland

The Computer Science Teachers' Association of Ireland (CSTAI) was founded in November 2017. To date, we have over 450 members, and 36 of the 40 LCCS pilot schools have joined.

The CSTAI offers a free collection of resources for computer science, coding, digital media, computing, and IT/ICT on Google Drive, where the members can access, download, and modify for their classes needs. Examples of such resources are PowerPoints, Notes, worksheets, and videos.

These resources cover from the early primary level to Junior Certificate Coding/Digital Media to LCCS. The LCCS resources are currently being created and aligned with our specification. In order to fill the gap that currently exists, the CSTAI has made links with other computer science teacher organisations in England, Northern Ireland, America, Australia and New Zealand, and the

## LCCS PILOT SCHEME Q&A

■ **How many schools applied to do the LCCS pilot scheme?**

Over 200 schools applied in December 2017.

■ **How many schools were selected?**

Forty schools were selected based on their gender intake, size, geographical location, school status, and the language the students are taught in (Irish or English).

■ **When will the pilot scheme start?**

September 2018.

■ **When will it first be examined?**

June 2020.

■ **How will the LCCS be assessed?**

In January 2020, the State Examinations Commission in Ireland will release a coursework project to be completed over 6-8 weeks. This coursework will be based on Strand 3 of the LCCS specification above. This project will involve the creation of a computational artefact and report. These items will be sent to the State Examinations Commission for marking and standardisation.

It will contribute 30% of the student's final marks. The remaining 70% will be a final computer-based summative examination in June 2020.

■ **What computer languages must be used?**

The National Council for Curriculum and Assessment (NCCA) have said that only Python or JavaScript may be used during the pilot scheme.

■ **What happens after June 2020?**

Once the pilot scheme is finished and evaluated, LCCS will be available to all schools that wish to offer it as an elective subject.

In September 2019, the pilot schools may offer LCCS to the new cohort of senior students to be examined in June 2021.

■ **How are teachers being supported?**

The government provides a Professional Development Service for Teachers (PDST) specifically for computer science to help the teachers through online support and CPD days. The PDST works with teachers and organisations to provide the best foundation and support during and after the pilot scheme.

The State Examinations Commission will release sample LCCS exam papers in March 2020.

■ **What else is going on in Ireland?**

Coding and digital media courses are offered to Junior Level. These are only examined through coursework and have no final, summative examinations.

■ **Do I have to be from Ireland/a teacher to join the CSTAI?**

Membership is open to any person who wishes to join. We have people from America to the UAE joining, computer teachers and non-computer teachers, as well as people from further education and third-level institutions. I am of the belief that the more diverse the membership pool of an organisation, the better.

UAE to provide resources for Irish teachers.

The CSTAI places a strong emphasis on the Irish language and offers a set of hundreds of technical computing terminology translated into Irish.

One of the main reasons the CSTAI was established was to promote sharing of ideas and resources between teachers. Joining and accessing the CSTAI is completely free. The resources are free to take, with no obligation to contribute, but if you have something appropriate to give, do share it with us to help the CSTAI grow.

If you would like to get involved in

this exchange of teaching resources, please email: [president.cstai@gmail.com](mailto:president.cstai@gmail.com).

We are also on Facebook ([helloworld.cc/2Pry3QA](https://www.facebook.com/helloworld.cc/2Pry3QA)) and Twitter ([helloworld.cc/2MtQdDB](https://twitter.com/helloworld.cc/2MtQdDB)). Please look at our information page: [helloworld.cc/2nVmsN1](https://helloworld.cc/2nVmsN1) (HW)

**Stephen** is a second-level Teacher of coding, digital media, and computer science in Gaelcholáiste Mhuire A.G., Cork City, Ireland. He is also Founder and President of the Computer Science Teachers' Association of Ireland (CSTAI).



■ Dan Fisher



■ Martin O'Hanlon



■ James Robinson

# Q & A: NEW ONLINE COURSES BY RASPBERRY PI

Dan Fisher talks to the course developers as they prepare to unleash their creations on an unsuspecting world...

STORY BY Dan Fisher

**T**here's always a flurry of activity at the start of the new academic year, and the Raspberry Pi Foundation are getting in on the act by announcing four new online courses that will be released in autumn/winter. They're completely free and aim to give educators a grounding in the concepts and practical applications of computing. I caught up with course developers Marc Scott (MS), Caitlyn Merry (CM), James Robinson (JR), and Martin O'Hanlon (MOH) to find out what's in store...

**DF:** Hi everyone, can you give me a rundown of what your courses are called and what the motivation was for creating them?

**MOH:** Sure, so my course is called Programming 101: An Introduction to Python for Educators. We wanted to create an 'introduction to programming' course which anyone could follow while ensuring that learners get to understand concepts as well

as practise coding. They'll leave with a really good understanding of why programming is so useful and how it works.

**JR:** Then we'll be releasing Programming 102: Think Like a Computer Scientist as a follow-up to many beginner online programming courses. A lot of courses spend time focusing on the syntax and core elements of a language without much focus on how to plan and construct a program. We feel the skills involved in understanding and breaking down a problem, before representing it in code, are fundamental to computer science. My course is therefore designed to give you the opportunity to explore these problem-solving skills while extending your knowledge of programming.

**MS:** How Computers Work fills in the gaps in people's knowledge about these amazing lumps of silicon and plastic. Computers are very abstract machines. Most people understand that computers can run large, complicated programs, but few understand

how computers are able to perform even the simplest of operations like counting or adding two numbers together. How Computers Work shows people how computers use simple components such as transistors to do incredible things.

**CM:** Mine is called Bringing Data to Life: Data Representation with Digital Media. Data representation is a huge part of the GCSE Computer Science curriculum and we wanted to present some of the more theoretical parts of the subject in a fun, practical, and engaging way. Data is everywhere – it's such an important topic nowadays with real-world impact, so we're making sure it's also useful for anyone else who wants to learn about data through the lens of creative media.

**DF:** So who are the courses for?

**MOH:** 101 is for anyone who wants to learn how to program in Python and gain an understanding of the concepts of computer programming.

**JR:** Programming 102 is for beginners who've already tackled some programming basics and have some experience in writing text-based programs.

**CM:** Bringing Data to Life is great if you want to understand how computers represent data using digital media such as images, sound, text, and video.

**MS:** How Computers Work is for anyone who is interested in learning about how computers work. [laughter from group]

**DF:** Short and to the point as ever, Marc.

**MS:** Okay, if you want a sensible answer, it would most help computer science teachers at secondary/high school level get to grips with the fundamentals and architecture.

### **DF: What will they learn?**

**MOH:** 101 will show you how to set up your computer, how to create computer programs in Python, about the programming concepts of sequencing (running instructions in order), selection (making choices), repetition (doing the same thing more than once), and how to use variables, input, output, ifs, lists, loops, functions, and more.

**JR:** 102 discusses the importance of algorithms and their applications, and shows you how to plan and implement your own algorithms and reflect on their efficiency. Throughout the course, you'll be using functions to structure your code and make your algorithms more versatile.

**MS:** You'll learn some of the historical origins of computers and programming, how computers work with 1s and 0s, how logic gates can be used to perform calculations, and about the basic internals of the CPU.

**CM:** You'll learn how text, images, and sound data is represented and stored by computers, but you'll also be doing your own media computation: creating your own code and programs to manipulate existing text, images, and data.

### **DF: Cool! So what will learners end up taking away from your courses?**

**MOH:** When you've completed the 101 course, you'll be able to create your own computer programs using Python, educate others in the fundamental concepts of computer programming, and take your learning further to understand more advanced concepts.

**JR:** After 102, you'll be able to plan and create structured and versatile programs, and make use of additional programming concepts including functions, lists, and dictionaries.

**MS:** You'll have a solid grounding in how computers actually function, and an appreciation for the underlying simplicity behind complex computing architectures and programs.

**CM:** The takeaway will be an understanding of how computers present to you all the media you view on your phone, screens, and so on, and you'll gain some new skills to

## AT A GLANCE...

Programming 101: An Introduction to Python for Educators (Oct 15)  
How Computers Work (Oct 29)  
Programming 102: Think Like a Computer Scientist (winter 2018/9)  
Bringing Data to Life: Data Representation with Creative Media (winter 2018/9)

Sign up today: [rpf.io/trainonline](http://rpf.io/trainonline)

manipulate and change what you see and hear through computers.

### **DF: How much do you need to know before you start?**

**MOH + MS:** 101 and How Computers Work are suitable for complete beginners with no prior knowledge.

**JR:** For 102, you'll need to have already tackled some programming basics and have a little experience writing text-based programs, but generally speaking the courses are for beginner-level learners who are looking for a place to start.

**CM:** A basic understanding of Python. Taking our Programming 101 course would be sufficient.

**DF:** Final question, for James: will you be doing your signature 'T-Rex arms' in any of the videos of your next course?

**JR:** [laughing] You'll just have to wait and see!

**DF:** Thanks guys!

Programming 101 and How Computers Work are available to sign up for today by visiting the Raspberry Pi homepage on FutureLearn: [rpf.io/trainonline](http://rpf.io/trainonline).

Programming 102 and Bringing Data to Life will be available to sign up for later on this year. [\(HW\)](#)

Dan is Learning Team Coordinator for Raspberry Pi.



Marc Scott



Caitlyn Merry

# MICROBLOCKS: LIVE PROGRAMMING FOR MICROCONTROLLERS

Bring electronic creations to life with this free Scratch-like language for the BBC micro:bit and other microcontrollers

STORY BY John Maloney

**T**he sixth-year class looked on in fascination as the rabbit's ears began to twitch. A few gasped when the rabbit's eyes changed colour. But the class really went wild when the rabbit began to dance with the carrot...

No, this is not a scene from Alice in Wonderland. The rabbit in question is a handmade robotic creation, affectionately called Rosa (after Rózsa Péter, the "founding mother of recursion theory"). The brain of the rabbit is a BBC micro:bit microcontroller programmed in MicroBlocks, a free blocks-based programming language similar to Scratch.

## Sparkling the imagination

From classrooms in Olot, Spain, to teacher workshops in West Virginia, the MicroBlocks team has seen people of all ages get excited about programming microcontrollers like the one in Rosa. What makes microcontroller programming so fascinating? Part of the appeal may be that, compared with complex, packaged consumer electronics, microcontrollers seem refreshingly simple and transparent.

However, the most compelling thing about microcontrollers – the thing that really sparks the imagination – is the feeling of empowerment they create. With a bit of coding, and perhaps a few external components, anything seems possible! MicroBlocks is designed to fan that creative spark.

■ Robotic rabbit programmed with MicroBlocks. Light sensors below the rabbit's face make it turn towards the flashlight mounted on the carrot

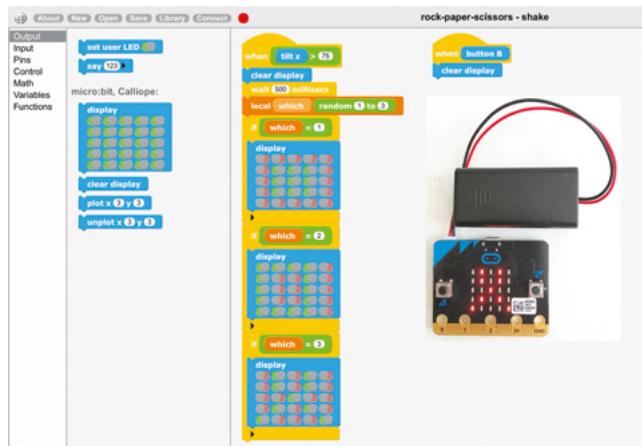


## Live programming and autonomy

Like Scratch, MicroBlocks is a live programming system. When a user clicks on a block or code snippet, the code is instantly downloaded and run on the microcontroller, and the user sees the result right away. LEDs light up, motors spin, and sensor values are displayed. The immediacy of live programming fosters learning by exploration, experimentation, and discovery. It also allows the user to follow their train of thought without interruption, as one creative idea leads on to the next.

Unfortunately, most programming systems for microcontrollers aren't live. They require a compilation and download process that slows experimentation and interrupts the smooth flow of ideas. For example, Microsoft's MakeCode takes 10-15 seconds to compile and download a program to the BBC micro:bit. While that may not seem like much, after a few dozen cycles, the process can feel tedious.

Tethered systems like Snap4Arduino take a different approach. They run the user's program on the host computer and treat the microcontroller as a peripheral. While this does provide a compelling live programming

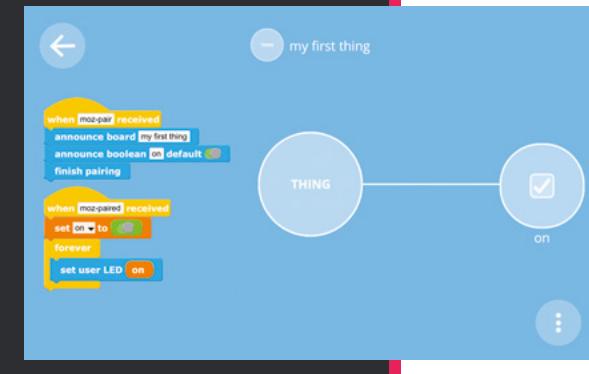


■ Rock-paper-scissors game played by shaking. The inset image shows the game running autonomously on a micro:bit powered by a battery

experience, the microcontroller must remain tethered to the host computer that is running the program; it isn't autonomous. This requirement to remain tethered to the host computer makes it cumbersome to embed the microcontroller in an art project, wear it on a hat, or build it into a mobile creature like our robotic rabbit.

## INTERNET OF THINGS

The Internet of Things refers to the ability to control and monitor devices like lights, thermostats, and home appliances remotely through a browser interface. Thanks to Mozilla's open IoT project ([iot.mozilla.org](https://iot.mozilla.org)), users can program their own "Thing" with MicroBlocks, plug it into a Raspberry Pi running Mozilla's free IoT Gateway software, and control it securely from anywhere on the internet.



MicroBlocks supports both live programming and autonomous operation. Users can see and test code changes immediately, yet their code continues to run when the microcontroller is untethered from the host computer. MicroBlocks gives the user the best of both worlds.

### Plays well alone and with others

The BBC micro:bit has a carefully chosen set of I/O devices built in, including an unexpectedly versatile 5x5 LED matrix, buttons, accelerometer, temperature sensor, and light sensor. As a result, the micro:bit is fun all by itself, without constructing circuits or adding extra components.

For those who wish to go beyond those built-in I/O devices, there are many options.

Rosa, the robotic rabbit, was built around the new micro:bit-powered Hummingbird Bit board from BirdBrain Technologies ([helloworld.cc/2LmLa2a](https://helloworld.cc/2LmLa2a)). Rosa's ears are hobby servos, her legs are gear motors, and her eyes are RGB LEDs with googly eyes from a craft store stuck on front. Rosa also has two light sensors, and she is programmed to turn towards a bright light source, such as a flashlight mounted on a cardboard carrot. Thus, the user can make Rosa dance just by moving the carrot.

### Get MicroBlocks

MicroBlocks ([microblocks.fun](https://microblocks.fun)) is free, runs on Windows, Mac OS, Linux, the Raspberry Pi, and Chrome OS, and may soon run in the browser as well. In addition, it will be integrated into Snap! ([snap.berkeley.edu](https://snap.berkeley.edu)), allowing the microcontroller and Snap! sprites to interact.

If you have a BBC micro:bit, Calliope mini, or AdaFruit Circuit

Playground Express, setup is easy: just download the MicroBlocks application, use it to install the MicroBlocks runtime onto the microcontroller, and start coding. MicroBlocks also supports other microcontrollers, such as the NodeMCU and many 32-bit Arduinos, but those require additional setup.

While MicroBlocks makes it easy to get started, it also supports more ambitious activities like generating music, collecting sensor data, working with digital signals, animating strips of colourful LEDs, and controlling robotic creatures. Whether you're a beginner or an advanced user, we hope you'll agree: MicroBlocks is fun! (HW)

**John** was the lead developer for Scratch for its first 11 years. For the past 18 months, he has been creating MicroBlocks with Bernat Romagosa and Jens Mönig. John has a PhD in computer science from the University of Washington and has been working on live, beginner-friendly programming systems for over 25 years.

# RIDE YOUR BICYCLE TOUR - IN MINECRAFT

Turn your bicycle into an input device. Pedal your way around your Minecraft world using a handy computer – the Raspberry Pi

STORY BY Benson Chong

**M**ost of us will have used a keyboard or joystick to control the movement of a character in a computer game. But have you ever thought of creating your own dedicated game controller? We'll show you how to use the pedals of a bicycle as an input to interact with the Raspberry Pi to control the objects in a Minecraft game.

Minecraft Pi is pre-installed in the latest version of Raspbian, which also comes with plenty of other educational tools, such as Python, Scratch, Sonic Pi, and Mathematica, for all your learning needs.

Minecraft Pi is a kind of mini version of the original game. In addition to being free, it brings lots of surprises to the user as you can use the Python language to interact with Minecraft. For example, you can create a big mountain or building with just a few lines of script, or use a sensor to take a numerical value in the real world and then use that value to control objects in your Minecraft world.

## Project fact check

**Time:** 15 hours

### Materials:

- Raspberry Pi 3, 8Gb SD card, power cable, HDMI cable
- Monitor (to display Minecraft)
- MCP3009 (analogue-digital converter)
- Bicycle
- Bicycle stand
- Bicycle energy generator
- 1k $\Omega$  resistor
- Wood, 1x2 inch
- Breadboard
- Screw and nut

### Tools:

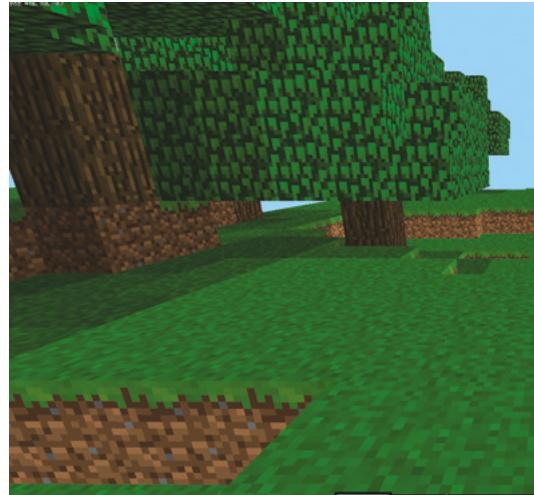
- Multimeter
- Portable jigsaw
- Cordless screwdriver
- Wire stripper

### The first step to creating your Minecraft world

To start the project, you need to create your own world in Minecraft. After launching the game, simply press New and the system will take you to a whole new world. This is



■ Overview of the project



■ The player is moving in Minecraft Pi



■ The startup menu of Minecraft Pi

where you'll build your tunnel for players to walk inside. Also, when a player arrives at a specific location, a message will be displayed on screen to indicate where they are at that time. Of course, you could also put different scenes in the tunnel to enhance the gaming experience.

After starting the game, you use the keyboard buttons W, A, S, D or Space to move the character forwards, backwards, left and right, and E to select different equipment or bricks. You press the left

mouse button to clear the bricks in front of you, or click the right mouse button to build with selected bricks. Although you can build your world slowly, brick by brick (and some people really enjoy working this way), it is possible to write Python scripts to build your world quickly.

### Using Python scripts to create a tunnel

As mentioned earlier, Minecraft Pi is specially designed for the Raspberry Pi, and you can use Python to create programs to interact with the game. A

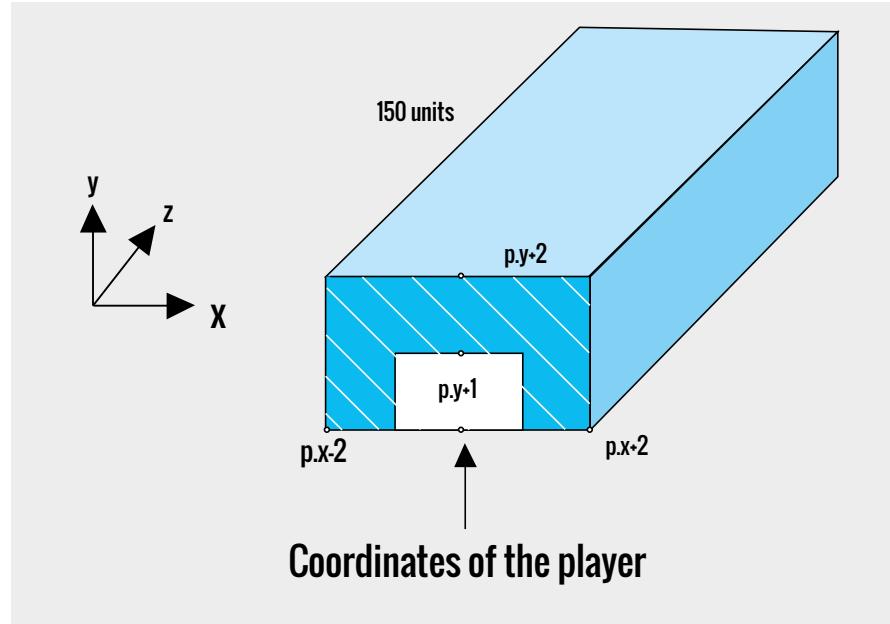


tunnel can be created in just a few seconds using scripts listed below.

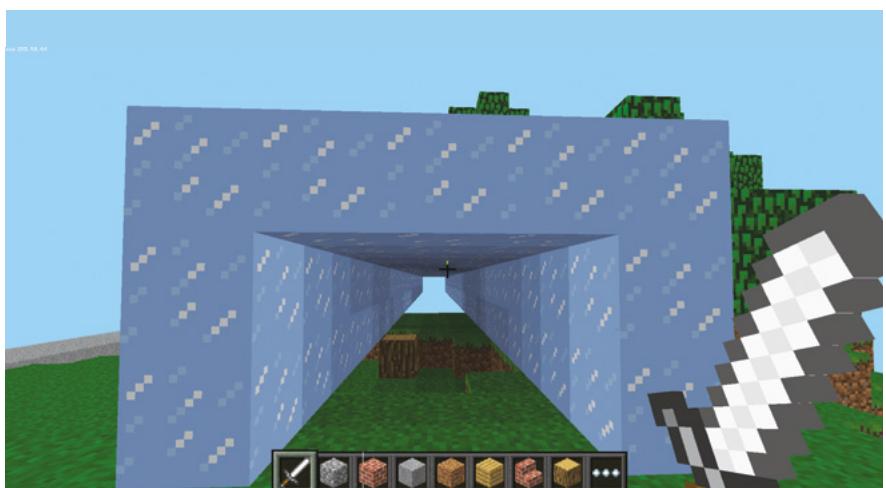
To create objects in your Minecraft world, you must first know your position in the game. When the player enters the game, the coordinates in the horizon direction of the space in which the character is located are represented by the X axis and the Z axis, while the vertical direction is represented by the Y axis.

The principle of creating a tunnel is to first create a rectangular and then remove the bricks inside to make it hollow. The concept behind this construction skill is the SUBTRACT operation in 3D modelling: select the main body first, then select the overlapping entities to be subtracted.

The details of the scripts are given in the following Python program (createTunnel.py). After running the scripts, it will create 150 unit long, one unit thick tunnel based on the current position of the player.



■ Design of the tunnel



■ Tunnels created by Python scripts

### createTunnel.py

#### LINE

#### PYTHON SCRIPTS

```

1   from mcpi.minecraft import Minecraft
2   import mcpi.block as block
3   mc = Minecraft.create()
4   p = mc.player.getPos()
5   mc.setBlocks(p.x, p.y, p.z-2, p.x+150,
6   p.y-2, p.z+2, block.STONE)
7   mc.setBlocks(p.x, p.y, p.z-1, p.x+150,
8   p.y+1, p.z+1, block.AIR)

```

#### Explanation of createTunnel.py:

- **Line 1-2:** Import the necessary Library into the program
- **Line 3:** Create the player in the Minecraft game
- **Line 4:** Get the coordinates of the player
- **Line 5:** Create a rectangular object
- **Line 6:** Remove the bricks inside the rectangular to make it hollow. It becomes a tunnel.

## Display the preset message

To enhance the interactivity of the program, you can write Python scripts that will display relevant information when the player reaches a specific location. For example, when the X coordinate value of the player is within a certain range, such as 32 to 33, the preset message “Welcome to this zone” will be displayed on screen.

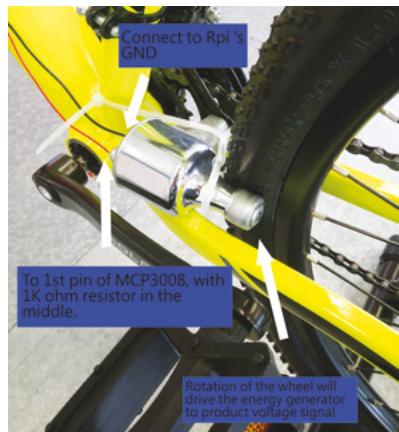
## Control the movement of the player

The last step is to turn the pedals of the bicycle connected to the controller that controls the movement of the player in Minecraft. We installed an energy generator on the rear rim of the bicycle. When a player pushes the pedals, the wheels rotate, which drives the energy generator. The generated voltage is used as the input to the Raspberry Pi. The logical operation of the program is as follows: when the generated voltage is larger than 0.5V, the screen of the game moves forward as if you’re riding in a virtual world.

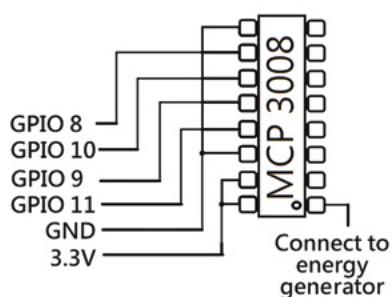
Because the voltage is an analogue signal, you need to convert the signal using an analogue-digital converter (ADC) MCP3008 and feed the signal to the Raspberry Pi. The connection circuit is shown opposite.

The final step is to write the Python script shown above. Just put the code in a file called stepForward.py, and the above devices will be ready to use.

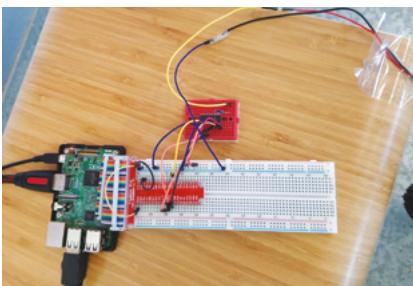
■ Showing the message to the player



■ Attach the energy generator to the bicycle



■ How to connect the ADC MCP3008 to the Raspberry Pi



■ How to input the voltage signal to the Raspberry Pi

## LINE

```

1 from gpiozero import MCP3008
2 from mcpi.minecraft import Minecraft
3 import time
4 voltageGen = MCP3008(0)
5 mc = Minecraft.create()
6 while True:
7     if voltageGen.value >0.5:
8         playerPos=mc.player.getPos()
9         mc.player.setPos(playerPos.
x+10,playerPos.y,playerPos.z)
10    if playerPos.x >32.0 and playerPos.
x<32.6:
11        mc.postToChat("Welcome to this
Zone")
12        time.sleep(0.1)

```

## PYTHON SCRIPTS

### stepForward.py

#### Explanation of stepForward.py:

- **Line 1-3:** Import the necessary Library into the program
- **Line 4:** Create a variable that will receive the signal from the MCP3008, which is connected to the first pin of the ADC
- **Line 5:** Create the player in Minecraft
- **Line 6-9:** Create a while True loop in the script. When the signal from the MCP3008 is larger than 0.5V, the screen of the game will move forward along the X axis(playerPos.x+10).
- **Line 10-11:** When the X coordinate of the player is within the range of 32.0 to 32.6, a message “Welcome to this Zone” will be shown on screen.

## Further development

By building up different landscapes in Minecraft, you can use this game controller for different activities. For example, you could create a historical scene for students to travel through, which would definitely develop their interest in learning history. In a geography class, a mountain’s topography could be simulated to allow students to walk in the middle of it. I firmly believe that students should experience these different ways of learning and will benefit from them. **(HW)**

Benson teaches at the Caritas Wu Cheng-chung Secondary School in Hong Kong.



MARK THORNBER TEACHER

# MATHEMATICAL MUSINGS

Want to make a fractal by accident? Play the Chaos Game

**M**ichael Barnsley introduced the Chaos Game in 1988 to illustrate a technique for image compression using fractals that he had developed. The rules are very simple:

- Choose a number of fixed points in the plane
- Pick a fraction between 0 and 1
- Choose a random starting point
- Repeat the following as many times as you like:
  - Choose one of your fixed points at random
  - Move your current point in a straight line to the chosen point until the distance is reduced by your chosen fraction
  - Plot a point

```
import math,random,turtle
numPts = 3
frac = 0.5
numDots = 30000
angle = 2*math.pi/numPts
points = [[300*math.sin(angle*i),300*math.cos(angle*i)] for i
          in range(numPts)]
pos = [0,300]
t = turtle.Turtle()
t.penup()
for i in range(numDots):
    point = random.choice(points)
    pos=[frac*pos[i]+(1-frac)*point[i] for i in
         range(2)]
    t.goto(pos[0],pos[1])
    t.dot(2)
input("Hit return to quit\n")
import math,random,turtle
numPts = 3
frac = 0.5
```

```
numDots = 30000
angle = 2*math.pi/numPts
points=[[300*math.sin(angle*i),300*math.cos(angle*i)] for i
        in range(numPts)]
pos = [0,300]
t = turtle.Turtle()
t.penup()
for i in range(numDots):
    point = random.choice(points)
    pos=[frac*pos[i]+(1-frac)*point[i] for i in
         range(2)]
    t.goto(pos[0],pos[1])
    t.dot(2)
input("Hit return to quit\n")
```

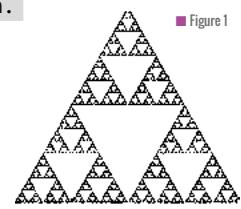


Figure 1

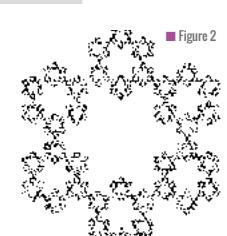


Figure 2

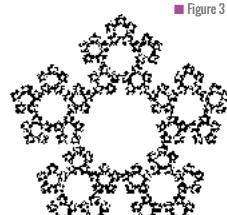


Figure 3

**Figure 1** was made using this procedure with three points at the vertices of an equilateral triangle and a fraction value of  $\frac{1}{2}$ . Can you see how the image is made up of a set of smaller images, each  $\frac{1}{2}$  the size of the next larger? This version was made by running the program in the box. It plots 30 000 points and this takes a very long time with Python Turtle graphics. I left it running overnight!

**Figure 2** was made using six dots and a fraction value of  $\frac{1}{3}$ . Notice how this makes the smaller hexagons fit together exactly. Can you figure out what fraction value I used for **Figure 3**?

You can find lots of information on the web by searching for 'Chaos Game'. If you want to plot lots of points quickly, investigate the use of more powerful Python libraries such as matplotlib. It's just the turtle part that runs slowly; Python with appropriate libraries does calculations quickly and is a popular tool used by professional mathematicians. [\(HW\)](#)

# MACHINE LEARNING WITH APPS FOR GOOD

We explore what machine learning is, why it's important for students, and how you can bring it into your classroom

STORY BY Amy Schumacher

■ By becoming confident with new technologies like machine learning, students gain skills as innovators, problem solvers, and creators for the future



**T**he world young people are facing is increasingly volatile, uncertain, and complex. All around us, there's a growing awareness of both the positive and negative consequences of emerging technologies like machine learning. By gaining hands-on experience and applying machine learning technologies for good, students can gain new skills and be empowered to harness the opportunities of the digital age.

Machine learning is becoming an increasingly important part of life with

technology. While there are some great tools that exist to teach machine learning concepts, there's currently a lack of accessible resources to teach the subject in Key Stage 3. To address this gap and help teachers get ahead, we've created a machine learning course alongside subject experts.

## What is machine learning?

For simplicity's sake, we'll describe machine learning as a system where – rather

than a computer programmer deciding the best way to sort, organise, classify or use information – a computer program develops its own set of instructions based on information that users feed it. Machine learning algorithms are all around us. They're powering customer service chatbots, making personalised recommendations for us on Netflix, and helping our iPhones to identify the phrase 'Hey Siri'.

Machine learning is a topic of interest

for many at the moment, as people are understandably excited to unleash the potential of this technology. Recent advances have rapidly improved the performance of machine learning algorithms, and within just a few years they've become much more capable. Machines can now learn at a mind-blowing pace and can handle growing amounts of data in shortening amounts of time. Concepts like face detection and image recognition may have been around for a while, but information can now be processed very quickly and at enormous volumes. Machine learning is also being applied in innovative areas, like predicting which medication could be used to fight cancer.

Of course, there's a current debate about potential issues with the application of machine learning. For instance, if you're training a driverless car, is it more important to protect passengers than pedestrians? When making a decision in an emergency, is it more important to save young people or old people?

## Why teach machine learning?

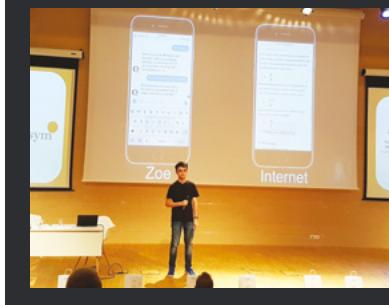
It's important for students to explore these implications of emerging technologies, as well as be aware of how this technology is impacting on them personally. When students are able to consider ethical questions as they directly apply new technologies to create solutions for real-world problems they care about, they're gaining new skills as well as nuanced perspectives as future creators of tech for good.

By becoming fluent and confident with new technologies like machine learning, students gain important skills as innovators, problem solvers, and creators. They're preparing to tackle the problems of the future, very likely tackling problems we're not even aware of yet. With machine learning skills in their repertoire, students will be able to solve bigger and bigger problems faster and faster.

By exploring machine learning, students are also preparing for the newest jobs in the technology market. Nearly every technology company is considering how machine learning can transform business, and these companies are on the lookout

## CHATBOTS FOR GOOD

The ZOE bot is an example of Apps for Good students creating machine learning solutions for problems they care about. ZOE is a Facebook messenger chatbot that helps Polish students revise for physics classes. ZOE was trained with all the physics notes the student creators had collected together. Since winning at our Poland showcase in 2017, the students have continued to refine and promote ZOE.



for skilled professionals to help them. With the current skills gap, there aren't enough people to fill these roles. Students exposed to machine learning concepts are gaining a head start when it comes to future employability, gaining the technical skills to help them succeed in this changing professional landscape.

Let's look briefly, for instance, at SAP, the market leader in enterprise software. We've partnered with SAP for the past five years, and most recently collaborated with them to create a new machine learning course for teachers. As an organisation, SAP often incorporates machine learning solutions into their products. They place a lot of importance on developing future technology talent and closing the digital skills gap.

## Getting started

If you're interested in teaching your students about machine learning, there are a couple of useful resources out there to help you get started in a fun and easy way. Machine Learning for Kids is a tool created by Dale Lane, a machine learning expert from IBM who previously worked on Watson, a question-answering computer system capable of answering questions,

that actually went on to win first place on the quiz show Jeopardy.

Machine Learning for Kids is an interactive and practical tool, which provides a fun way for kids to learn about machine learning by making things with it. Students can use it to train different types of models, create games, and work on other interactive projects. It gives students a blank canvas where they can apply machine learning in fun ways. The tool builds on existing efforts to introduce and teach coding to children, by adding these models to Scratch, allowing students to create projects with the machine learning models they've trained.

Machine Learning for Kids has been carefully pulled together with other helpful resources into our free machine learning course for teachers in the UK. The course includes sessions exploring some of the most common machine learning algorithms such as decision trees, regression, and clustering. There's also an optional Python programming session where students can apply these algorithms to imported datasets.

No previous knowledge of the subject is required. If you're feeling like you want some backup, we even have optional mentoring sessions available with industry experts. Within just a few course hours, your students will be creating machine learning solutions for good. If you're interested, all you need to do is head to our website and sign up. After that, you can immediately begin reviewing the free lesson plans and workbooks.

Through learning about machine learning, we can empower students all across the UK to unlock their confidence, skills, and talent, and to see the difference they can make both in their own lives and the communities around them using emerging new technologies for good. If you asked us, when armed with these tools, the future is looking pretty bright in the hands of these young tech-minded change makers. **(HW)**

Apps for Good helps teachers to develop entrepreneurial, creative, and resilient students who are prepared to face an ever-changing future. Across the UK, our courses have impacted 100,000+ students.



■ 'Digital making' is a new addition to the timetable at Phoenix School and has had a surprisingly therapeutic effect on students who suffer from mental health problems

# DIGITAL MAKING AND MENTAL HEALTH

A special school in Sheffield for students with complex mental health needs has unearthed some of the wider benefits that digital making has to offer

STORY BY Edward Hall

**A**ll Cygnet schools are located within tier 4 CAMHs inpatient hospitals, providing services from acute (general) to low secure. Young people who are suffering from severe and/or complex mental health conditions are admitted to our hospitals either on an informal basis or through a mental health section. The hospitals provide a safe, therapeutic environment, allowing the young people to stabilise while various clinical assessments are completed to provide community teams with the necessary information to run a longer term treatment programme post discharge. Alongside this, the school team liaises with the young person's home

school to collect academic information and develop an individual education plan so that they can continue being educated during their stay. School forms part of the young person's daily therapeutic timetable, which also includes occupational therapy and psychology sessions while helping to maintain valuable structure and routine. The average length of stay is 6-10 weeks.

A challenge for the hospital schools is managing risk while still allowing equal access to educational opportunities. Quite a high proportion of students have a history of self-harm, so any risk items must be managed properly, and some activities may need adapting to prevent a mental

health incident occurring (for example, using plastic microscope slides in science rather than glass).

## Picademy inspiration

Having completed a Picademy training course in Manchester, teachers at Phoenix School in Sheffield were inspired to develop a 'digital making' scheme of work to deliver as part of the school timetable. The aim of the scheme was to help develop digital skills required in the modern workplace, build on wider transferable skills (problem solving, team working, creativity, and so on), and allow young people to experience the benefits of finding enjoyment through

## TEACHER Q&A – PHILIP CROWSON



**Q: Please introduce yourself!**

**A:** Philip Crowson, science and digital making teacher at Phoenix School in Sheffield.

**Q: What prior programming/Raspberry Pi experience did you have?**

**A:** None! However, my son had the first Raspberry Pi model given to him from school and it inspired him to take A Level computing, so I jumped at the chance to get involved.

**Q: How did you learn the skills so that you could teach and support the young people in digital making sessions?**

**A:** My colleague went on a PiCademy course and has imparted his knowledge to the rest of the school. There are many books and magazines available as support (e.g. *The MagPi*), and it's all about prior planning.

**Q: Why do you think it has been a success in your setting?**

**A:** The young people enjoy the sense of achievement and empowerment derived from programming and physical modelling.

**Q: What transferable skills do the students pick up from the sessions?**

**A:** One student wanted to be an IT technician and felt the course had supported him in this ambition. He felt he had learned collaborative skills with peers acting as a project manager and troubleshooting problems for other students.

a new activity. Key Stages 3 and 4 students are now timetabled one session per week, and the course is also offered to some Key Stage 5 students. During education sessions, a record is always kept of what equipment has been provided to

each student, and if necessary lower risk items can be used: for example, PiStop (a programmable set of plastic LED traffic lights) could be used as a safer alternative to regular metal LEDs and resistors. The course was linked with existing AQA unit awards, allowing students to recognise their achievements and validate their work. Following this, Phoenix School had five new digital making units successfully approved at level 2 so that students work is accredited accordingly.

### A surprise

The success of digital making in our setting has come as no surprise to us, however the noticeable therapeutic benefit of the sessions has. Young people have treated

certainly make an interesting academic study for someone in the future). However, on current reflection, there is something in 'that moment' the instructors at Picademy talk about when somebody cracks/ debugs a problem and the satisfaction it evokes. On a biological level, the rush of neurotransmitters stimulated by this type of experience surely must have a positive effect on any young person, particularly those experiencing poor mental health. Another observation is the obvious enjoyment they receive from 'seeing things happen', and these visual outcomes do allow students to link programming code to real changes in the physical world. Digital making is also a subject that lends itself to the development of 'soft skills' (such as, peer

## IT HAS INSPIRED SOME YOUNG PEOPLE TO EXPLORE RELATED COLLEGE AND CAREER OPPORTUNITIES

all equipment respectfully, sessions are calm, and a good level of engagement has been maintained with minimal interference from poor mental health. Hard to reach students have been engaged, and it has inspired some young people to explore related college and career opportunities.

OFSTED recognised this success when they inspected Phoenix School in July 2017, with the report stating "in an information and communication technology lesson, teachers tap into pupils' computer interests to enable the skilful programming of robotic technology" when referencing a digital making session on Explorer HAT Pro. Lesson observations carried out by the headteacher have also echoed this sentiment, stating that sessions "promote independent learning, resilience and a culture of young people trying technology for themselves" and that "one particular student was happy with the outcome (flashing light)" and that this "gives young people a real sense of achievement".

### Why?

The reasons for this are a little less obvious, and to explore in depth would be beyond the scope of this feature (but it would

discussion, interaction, and collaboration) which are central to any successful Social, Emotional & Mental health curriculum offer. The students also clearly benefit from the sense of achievement/reward during both the construction and completion stage, and this has been capitalised on by the offer of AQA unit awards. There's evidence to show the lessons help enhance learner/educator relationships, which can sometimes be a barrier in this setting. A final thought.... the definition of recovery (from [mentalhealth.org.uk](http://mentalhealth.org.uk)):

"Recovery can be a voyage of self-discovery and personal growth. Experiences of mental illness can provide opportunities for change, reflection and discovery of new values, skills and interests." (HW)

**Edward** is the Head of Education for Cygnet Health Care, one of the UK's largest independent providers of mental health care. Cygnet currently operates three independent schools, providing education to young people aged 13-17 who are placed in their care. He became a Raspberry Pi certified educator following a battle to engage the young people with IT.

# RE-USING FAMILIAR TECHNIQUES

There are lots of techniques we can use to teach primary pupils to program, including using example programs. Perhaps this is an opportunity for teachers to re-use familiar techniques across subjects

STORY BY Jane Waite

**A**s part of the work I did last year (2017) reviewing research in teaching computing for the Royal Society Report on Computer Science Education, I produced a very long list of strategies to teach programming. I gathered the list from research studies from the last 10 years. As soon as I had created it, I knew I needed to share it in a more 'primary teacher format'. I created the 'Continuum of Scaffolding' (a name I dislike, but I simply couldn't think of a better one) and started to use it in teacher CPD (continuing professional development). The continuum grouped techniques from the standpoint of how supported the tasks were. It has been very popular in training events as it uses familiar vocabulary to help teachers talk about how they teach computing. I wrote a *Hello World* article on it called 'Copy Code, Shared Coding, Tinkering and other techniques for teaching programming'.

Within the original list of strategies to teach programming, I noticed that many of the techniques included the use of example programs. I hadn't particularly drawn this out in the continuum as the focus there was on the degree of support (or scaffolding) that different techniques incorporated.

More recently, I've been helping Dr Sue Sentance with her research on teaching programming in secondary schools. The research is investigating an approach that Sue has developed called PRIMM. The starting point of the technique is to provide learners with an example program for them to read. For further information, read Sue's

super blog or read Oliver Quinlan's review of the approach.

At the same time, I've been doing my own research, looking at the synergies between using planning when teaching writing, and design when teaching programming.

When teaching writing, we always start by reading gorgeous high-quality examples of the genre that our learners will be writing. We include example texts in guided reading, use examples in drama for speaking and listening, familiarise learners with the patterns in the text, and exemplify 'what we are looking for'.

When we start to write, we draw upon this rich experience of a now familiar genre, and carefully model the process of writing by creating class examples and gradually fading support (often through approaches such as Talk for Writing Imitate/Innovate/Invent stages).

In early focus groups in my research, teachers have been saying that this modelling of how to design and write programs is something they think they need help with and feel they should be spending time doing.

But what has struck me is that we don't have great examples of programs. We don't have those gorgeous high-quality programs that learners can use, or that we can 'read' to them or for them to read. The work I've been doing on Smelly Code links to this, as I'm not sure we've quite agreed yet what high-quality code looks like for different points in learners' progression.

However, to try and bring this all together, I've cross-referenced techniques for teaching programming with speaking and listening, reading and writing, and included Pie Corbett's approach to fading support.

I've grouped the list by Use, Modify, Create (Lee et al, 2011), a learning progression suggested by a research team in the United States that has been around for quite a while and is an underpinning framework contributing to PRIMM, and I have cross-references to PRIMM.

As always, the order of my new list (as shown in the diagram and listed below) isn't suggesting the order of use of techniques – this is just a list to give ideas, start debate, and/or maybe to help a review of planning. The list is based on the use of example programs, where a program includes the task, the design (including the algorithm), and the code itself.

## Using example programs: Unplug it

Teachers can link the example program using unplugged techniques to real-world scenarios or act out the algorithm in the design. For example, if a program will draw a square, then talk about this and physically enact the algorithm by walking a square. If the program is a quiz, then do a quiz, with the learners being the quiz master rather than the computer. If the program has a selection statement, use a real-life example, such as if it's raining put on a coat. Act out the program, act out the programming

constructs and talk about it while you do, just like drama, and speaking and listening.

## Play it (and identify the features)

Learners can simply run the example program and play the animation, quiz, game, and so on. So if it's an animation, watch it. If it's a quiz, answer the questions. If it's a game or simulation, play it, try it out. If it's a physical product, use it. This is a bit like being read to. So the learners are the recipient of the effect of a program or a book. Building on this, learners can be more active with the program, they can identify its main features. In doing this they're starting to identify the design. This can be a great way to 'reverse engineer' a design. I've done this with physical computing products, as learners use the example product I've already made (in *Blue Peter* style). Lots of other teachers have suggested this idea. I've seen it in Phil Bagge's planning and in others.

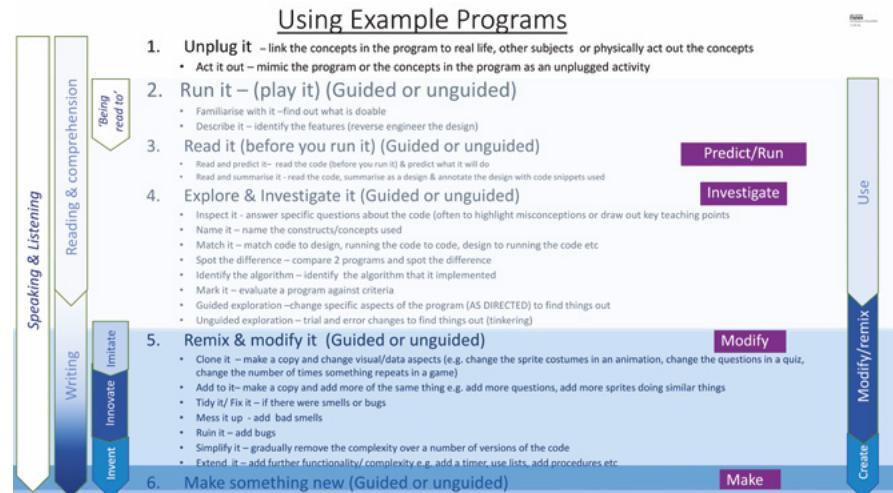
## Read it (before you run it)

Rather than being read to by someone else, example programs can be read by learners themselves. Before they run the code, they predict what the program might do. This could be a summary of what they think the code will do, or it could be that they look at each line and 'trace' the program with example data. This ties in directly with the program of study in England, which requires learners in KS1 (ages 5-7) to be able to "use logical reasoning to predict the behaviour of simple programs". Once they've predicted the program, learners can run the program to check out their prediction and confirm or adjust their thinking, so developing their logical reasoning. This is the equivalent to the PR in PRIMM. P = Predict, R=Run.

As well as reverse engineering the overall design from running a program, learners can look at the code and spot which sections of code implement the different features that they identified. These code snippets could be added to their design as annotations (reminders) to use when creating their own version once they get to writing.

## Explore and investigate it

There's a whole raft of different ways to explore and investigate example programs.



■ Image courtesy of Jane Waite, PRIMM courtesy of Sue Sentance, reference to Imitate, Innovate, Invent from Pie Corbett's *Talk for Writing*

Teachers might carefully decide on questions here, addressing misconceptions, highlighting key teaching points, and drawing out useful ideas for later project work. Learners can be prompted to answer questions about the code such as being asked to name constructs used, so spot the use of a repeat or notice where variables are used. Or learners might explore the code by starting to change it, either in response to guidance or through tinkering with no direction. At this point, learners are moving from using the code to modifying and remixing. In literacy, this is moving from reading and comprehension into writing. But at this stage, perhaps still with a view to finding things out rather than creating something of their own. This is the I in PRIMM. I = investigate.

## Remix and modify it (guided or unguided)

As well as investigate, there are a whole set of different modifying techniques. These can be highly scaffolded with lots of teacher (or online tutor, or worksheet) guidance, or very unguided. Learners might clone the example program and change surface aspects, or they might add more of the same code, such as extra questions or new sprites/objects doing similar things. Or they might tidy up smelly code, or fix buggy code. Learners can mess up code, sabotage example code. Or simplify or extend it. At this point in English lessons, this might be the equivalent of learners moving from an imitation of the original text to an innovation. This is the first M of PRIMM.

M= Modify. Here, learners are starting to move from using someone else's product to having some ownership of a new version.

## Make something new (guided or unguided)

Finally, we're at the point where learners are make something new, something that is truly 'theirs'. The example program is no longer leading the teaching experience. However, example programs will have provided learners with a repertoire of common designs and how they're implemented (design patterns), which they're now 'remixing' or reusing. 'Make activities' may be tightly scaffolded tasks or free and open activities. Here, we're akin to 'invention' in writing and the final M in PRIMM. M= make.

I would be really interested in having feedback on the list, as I'm sure there are techniques I've missed, and I bet there are some better collective terms for some of the sub-techniques. Also, in creating this list, I haven't incorporated the different ways example programs can be used for assessment, nor how example programs can be used to provide progression - through exemplifying common design patterns.

I hope that by cross-referencing some of the different approaches for teaching programming to the approaches we use in teaching writing, a degree of familiarity may be provided, and that this might help teachers who are less experienced in teaching programming feel more confident as they re-use their experience from teaching another subject. (HW)

# TECHATHON HELPS GIRLS FIND THEIR FORTE

Interested in raising the profile of women in tech? Hosting an event for students to challenge, innovate, and risk take hits the high note!

STORY BY Sam Shallcross

**0**n 9th March 2018, the Girls' Day School Trust (GDST, the UK's leading network of independent girls' schools) hosted a Techathon for over 200 girls from 22 of their schools at South Hampstead High School. Now in its fourth year, the Techathon challenges, encourages, and inspires budding female digital leaders in our schools. On hand to help with this mission were 60 staff and industry mentors from organisations such as Discovery Education, Capita SIMS, and Firefly alongside mentors from start-ups and GDST alumnae.

## Setting the tone

The focus of this year's event was music and digital technology. The action-packed day began with an inspiring keynote speech from Elizabeth Davis, a classical music journalist and content editor, on the role technology has played throughout history in the production and sharing of music. From the first female composer Kassia, a 9th Century Abbess, to Lady Gaga today, Elizabeth encouraged our students to become experts in the newest tech, to embrace the fast pace of change in these industries, and to use their skills creatively in their future career decisions. Despite associating technology with male employees, Elizabeth challenged this stereotype, arguing, "Look at the cutting edge of technology through history and that is where you will find the women carving out their spaces to have their voices heard"

Suitably inspired, the girls then set about their challenge for the day – to produce



The focus of this year's event was music and digital technology

a video and a prototype of a product or service for use in the music industry. Working as a team, the girls used their digital skills to create presentations, use green screen for promotional videos, build websites, write code, and manipulate materials into, for example, VR headsets and interactive wrist bands! As the teams worked on their task, they were supported by industry mentors, building skills for the 21st century such as teamwork, problem solving, and leadership. It is these skills,

quite apart from the digital experience and expertise they develop, which allow students to build their resilience, think critically, and explore their intellectual and creative curiosities.

## Hitting the high note

Alongside working on their project, girls were invited to attend a workshop. Showcasing the latest ideas, the workshops gave the girls the opportunity to 'have a go' with innovative tech.



■ Pupils explored the iOS app Samplr, creating their own sounds with voice

Juniors explored the iOS app Samplr, creating their own sounds with voice. For seniors, the Algoravers, Coral and Jo, were on hand to demonstrate the manipulation of Sine waves to create interesting tones and pulses. Using live coding techniques, the students composed dance music created from algorithms. Coral and Jo also shared their experiences of travelling around Europe, sharing their passion for algoraving, and highlighted that coding is an accessible platform for anyone interested in making electronic beats of any genre, and is an exciting and unpredictable way of producing audio.

### The finale

As the day drew to a close, teams pitched to industry judges. All of the teams showed tremendous creativity,

ingenuity, and resilience as they put together their ideas and shared them at the product showcase. Our mentors were in awe of the achievements of the students. The winning team came from Sheffield High School, who were commended for their concept, which provided a very practical way of using the power of music to reach people with Alzheimer's. Harnessing images and music, this idea connected people who have often stopped communicating in other ways.

Reflecting on the day, Liane Katz, founder of MAMA.codes, commented, "It was a fantastic event, congrats on organising a very impressive and uplifting gathering. Count me in for next year if you need!" And Melenie Schatynski, Web Developer at Primo Toys commented, "I thoroughly enjoyed the day and it was fantastic to see so many great ideas!"

### Composing your own

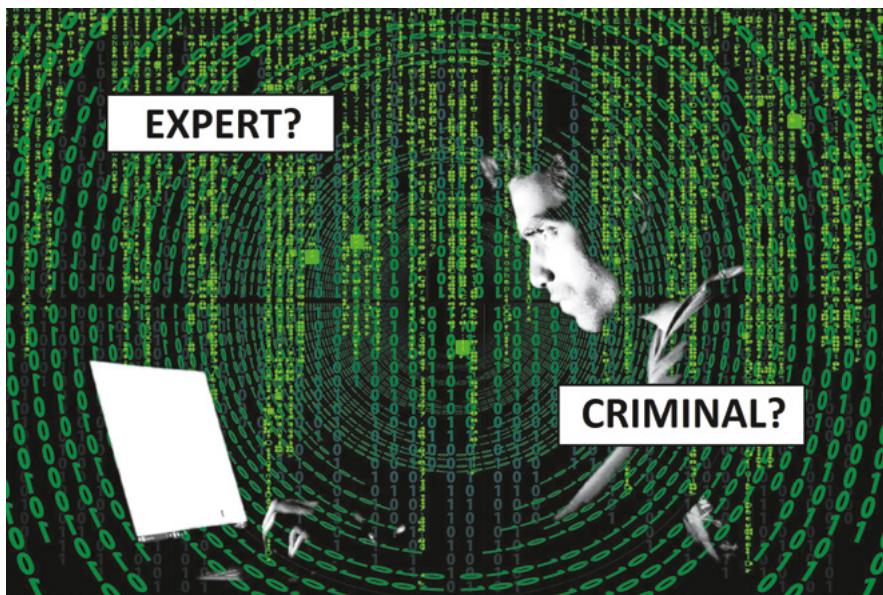
Hosting an event like this provides an important counter narrative to those that focus on the harmful effect of technology on young people, and instead encourages and supports social entrepreneurship, digital creativity, and technological experimentation in a supportive and innovative environment. We would really encourage you to consider hosting collaborative events with clusters of schools and planning your own Techathon. If you want any more information, please contact Sam Shallcross, [s.shallcross@bro.gdst.net](mailto:s.shallcross@bro.gdst.net). (HW)

**Sam** is a Computing Teacher at Bromley High Junior School, GDST and a Trust Consultant Teacher for Digital Learning at the GDST. **Amy Icke** is the Digital Learning Platform Manager at the GDST.

# CYBER PREVENT: YOUR COUNTRY NEEDS YOU

There's a global demand for cybersecurity experts, which the UK is in a prime position to fill. YOU are in a position to help!

STORY BY DC Jez Rogers



Now is the time to enable young people to become experts and not criminals

**A**s an educator of young people, you'll no doubt have come across a number of talented individuals who've demonstrated strong cyber skills. There has never been a better time to signpost young people toward becoming a cyber professional. The cyber world has evolved dramatically and, although the traditional academic route still has its place, industry has recognised that those individuals with the necessary skills may not perform within an academic environment and yet they're eminently employable.

I'm no expert, but I agree with the anecdotal suggestion that there's a strong link between neuro-diverse young people and cyber skills. We're awaiting the publication of academic research from Bath University. Industry has recognised this and understands that an individual

on 'a spectrum' may not achieve well academically but that they have strong cyber skills that make them a suitable candidate for employment.

The readers here will know far better than I that resources for cyber-gifted individuals and those who educate them vary greatly. Schools and colleges often don't have the capacity to allocate the required resources to support these gifted and talented individuals or their teachers.

The world is crying out for cybersecurity professionals, with a current global shortfall of 350,000 and the prediction that this will rise to 1.8 million in the next three years. There's a massive diversity of roles which would suit all types of young people, from the outgoing to the shy and retiring, but all of whom will possess the necessary skill set. Globally, the UK remains the fifth most

cyber-attacked country, so we need these experts. We have an opportunity to make the UK a worldwide hub of cyber excellence, thereby driving the UK economy.

## The dark side

This lack of curricular focus and understanding may be causing a slide by young people into committing offences against the Computer Misuse Act. One in four teenagers in the UK have committed a CMA offence, and the average age of those being arrested is 17 years.

Although ignorance isn't a defence in law, it's a lack of education, awareness, and support that often causes these young people to stray into criminality. There are various misconceptions by young people that their online activity isn't against the law, or that there's little or no law enforcement monitoring of the internet, and so there's little chance of being caught. The other damaging misconception is that cybercrime is victimless.

It's a sad fact that around 50% of small-to-medium businesses that suffer a cyber-attack will close within six months.

The wrong route often starts in the gaming world with 'mods' and cheats. If they're a particularly motivated gamer, then 'DoS-ing'\* another gamer from the gaming platform may follow. The process is cheap and relatively untaxed. This success may lead to peer appreciation and the beginnings of a cyber 'rep'. For many young people, this will provide a level of acceptance and kudos that they haven't received before.

\*DoS: Denial of Service attack – overwhelming an internet connected device with small data packets

## RESOURCE FOR EDUCATORS

### Cyber Discovery

Annual competition run by HM Government's Cyber Schools programme for students in years 10-13.  
[helloworld.cc/2Mwnr50](http://helloworld.cc/2Mwnr50)



With a little research and time spent on relevant online forums, our gifted and talented young person has learnt how to completely bypass and/or DoS-attack the school website which they may then turn into a service for hire. Again, this increases the 'lulz'.

Their cyber reputation increases and they become even more motivated to improve their knowledge. They now purchase or create 'stressor' tools or 'trojans' that they deploy for themselves or others.

### Cyber Security Challenge UK

The CSCUK offer a number of teaching resources including lesson plans.  
[helloworld.cc/2NbVhZk](http://helloworld.cc/2NbVhZk)

### Cyber Centurion

Annual competition run by CSCUK for ages 12-18.  
[helloworld.cc/2BMihx8](http://helloworld.cc/2BMihx8)

### Tech Partnership

Resources for coding, cybersecurity, and game and app design. Also has a section for teacher CPD.  
[helloworld.cc/2o4Dl8g](http://helloworld.cc/2o4Dl8g)

### Digital Cyber Academy

A set of browser-based learning labs including challenges.  
[helloworld.cc/2N9kxzn](http://helloworld.cc/2N9kxzn)

on positive diversion. We're reaching out to educators and volunteers to help identify this young talent so that they can receive suitable education about the law and then identify positive pathways to upskill them ready for a professional cyber career. Teachers are the key to identify and support these people at the earliest opportunity, whether they're the gifted and talented or those at risk of the slide into cybercrime. As such, schools and colleges must ensure that teachers have the correct tools within the classroom to enable this required level of interaction and positive signposting.

Cyber Prevent teams aim to support teachers by providing relevant information to assist them with dealing with cyber-talented individuals: lesson plans; online capture the flag platforms; online coding and hacking 'labs'. Multi-school workshops for identified gifted and talented pupils are being created, which will offer access to industry and future careers, motivational talks from reformed 'hackers', and guidance on how to stay on the correct side of the law.

As a new national programme, many of these projects are a work in progress and we continue to seek talented partners from academia and industry to assist with our diversionary projects, including:

- Extra-curricular education
- Work experience opportunities
- Apprenticeships
- Mentoring
- Employment

Should you wish to assist, find out more or discuss a young person, please email the team: [CyberPrevent@serocu.pnn.police.uk](mailto:CyberPrevent@serocu.pnn.police.uk). If you're outside of the South East region, we can refer you to our counterparts nationwide. You can find out more or download our resource pack at:

[serocu.police.uk](http://serocu.police.uk)

(HW)

## THERE WILL BE A WORLDWIDE SHORTFALL OF 1.8 MILLION CYBER SECURITY PROFESSIONALS BY 2021

They're now fully on the wrong side of the Computer Misuse Act. Before long, they've come to the attention of an Organised Crime Group that wants to use these talents for more serious criminality. OCGs trawl the forums looking for exploitable young people for all manner of reasons. Cyber skills are a way to make a great deal of money and using a teenager as the actor is typical OCG manipulation and modus operandi. The

### The solution

The national Cyber Prevent strategy aims to deter and divert those who may be on the cusp of cybercrime into a legitimate cybersecurity career with the reputation and pay to match.

The Cyber Prevent ethos is not to criminalise young people unnecessarily. The team and I are cops and so have a legal duty to discharge, but this role concentrates

Jez has been a Police Officer for over 27 years and worked in a number of roles within two county police forces: uniform response; a full-time public order unit; the management of high-risk sex offenders and dangerous offenders; intelligence and now within Cyber Prevent.

# DON'T PANIC!

# THE HITCHHIKER'S GUIDE TO KEY STAGE 3 COMPUTING

To ensure the most highly engaging and rewarding experiences for all students aged 11-14 years, computing teachers may be tempted to cram too many learning activities and content into the precious few timetabled hours available. In this guide, Alan O'Donohoe (Computing At School Master Teacher and Leader of exa.foundation) suggests some strategies and approaches to curriculum planning to ensure your students are engaged and inspired

**T**he following strategies and approaches are based on the principle that Key Stage 3 (KS3) shouldn't only serve as a solid foundation for your Key Stage 4 (KS4) courses, but should also offer as realistic an experience as possible of KS4, to encourage your students to make more informed choices later on.

## Less is more

I'll admit that I myself have been guilty in the past of trying to squeeze too many different topics into our KS3 curriculum. For example, one academic year, we had planned eight separate topics for students to study, which included online safety, computer systems, photo editing, data modelling, games development, text-based programming, and sound editing. You know, the very fact that I can't remember what the eighth topic was shows just how foolish this move was!

At the end of that year, we teachers conducted a 'pupil voice survey', in which we asked students to tell us about the topics they had enjoyed learning the most and the topics they had found most useful and relevant. Apart from the most recent they had studied, very few could even remember the titles of the topics from earlier in the year, let alone remember how they felt about them or what they had learned. That served as the wake-up call I needed, persuading me to plan far fewer topics, with more time invested in each.

■ Don't squeeze too many topics into your curriculum



Quite apart from anything else, it's very difficult to plan and measure learning progress if every five lessons you're starting a new topic that doesn't seek to build on students' learning from the previous five lessons. Anyone who has spent time learning to play a musical instrument or taken driving lessons, will know the true value of repetition and practice to improve competency, confidence, and deepen one's understanding.

### Three topic model

My recommendation would be to plan a computing curriculum around three key topics that broadly reflect the options available to your students at KS4 and then repeat these same three topics every year. But instead of merely repeating everything again, each year go as deeper and far with each topic as you can. The key is to carefully select three broad topics that you know and your teaching colleagues will feel confident delivering, which have clear pathways into KS4. I've suggested three topics in the example below, but your topics may be completely different from these:

### Suggested curriculum plan

**Year 7:** Autumn Term - Games Development

Spring Term - Computer Networks

Summer Term - The Web

**Year 8:** Autumn Term - Games Development

Spring Term - Computer Networks

Summer Term - The Web

**Year 9:** Autumn Term - Games Development

Spring Term - Computer Networks

Summer Term - The Web



■ Keep topics relevant to your students to ensure they're engaged

### Keep topics relevant

When deciding which topics to focus on, it's worth trying to select those that are most directly relevant to your students and avoid any content that is not. When your students see how relevant the learning content is to them, it will make much more sense to them why they're studying these topics and hopefully persuade them of the value of choosing to study the subject further at KS4.

For example, if you chose to plan a term on the topic of 'Computer Networks', you would elect to focus much of the learning on cybersecurity threats, and how to protect networks and individuals against them. This may seem much more relevant to your students than, for example, teaching them about token-ring networks or other network topologies that are required on a GCSE specification. This would also provide highly relevant opportunities for students to understand what has happened when security alerts are reported in the news, like the WannaCry attack that affected much of the NHS.

### Keep learning progress highly visible

I believe it's extremely important for students to have a real grasp of their own sense of achievement and progress throughout the year. There are an endless variety of tools available to teachers that enable progress data to be shared with students, but the trick is making sure that this is truly accessible. One mistake we made was that the student progress data, although accessible 24/7 through our learning platform, was actually quite difficult for students to access quickly and easily. We found that by having a physical copy of the same progress data readily available at a glance in an exercise book, it meant that students had instant access at any point during lessons, which they could refer to.

The abundance of Multiple Choice Questions already available through CAS Quantum diagnostic questions provides a great





■ Most students enjoy hands-on creative, digital making activities and experiences

► starting point for teachers to create their own assessment tool that can be used regularly to evidence learning progress from lesson to lesson, rather than waiting until the end of a topic to inform a student how much progress they've made.

### **Emulate other successful subjects**

In my last school, the geography department had achieved success on many different levels. The GCSE results were among the highest in the region and the subject remained over-subscribed at KS4.

To emulate some of the geography department's success, I invested some time in talking to our teachers of geography as well as casual conversations with students to ascertain what made the subject so popular at GCSE. The majority of students responded that while they didn't find geography particularly stimulating, they experienced a high degree of success and achievement, and felt that choosing GCSE geography was a safe choice that would guarantee them desirable GCSE grades. They could also see why they needed to learn certain topics to understand the effects of climate change, tourism, and international trade.

### **Don't serve other subjects**

One mistake I feel we made at our school was in being too willing to serve other subjects in our school. For example, the maths department (and it wasn't the only example) asked if we would

teach a certain set of data-handling skills to enable them to teach data-handling more successfully in maths. While we embraced this request early on, I don't feel that we did ourselves any favours, as it soon became clear from student perceptions that our subject existed purely to meet the needs of other subjects. This in turn meant that students didn't value the unique status of our subject, computing.



■ Allow students to work on extended projects lasting a term or longer

As our department made the transition from teaching ICT to computing, we realised it was better to leave the maths department (and other departments) to teach the things the maths department needed for us to be seen as a subject in our own right. I recall how we spent some time reviewing and analysing our subject-specific vocabulary for computing and then went to great lengths to encourage students to use these terms in lessons.

## Celebrate your unique features

Our subject has some unique qualities and characteristics. Teachers should identify these – the ones with maximum student appeal – then capitalise on them as much as possible to ensure the long-term growth and success of their subject, trying to focus more on them when planning learning activities.

It was clear to us in our computing department that the majority of students enjoyed the hands-on creative, digital making activities and experiences. So we sought to make these an even more prominent feature, with much more time being devoted to collaborative problem solving, practical activities, and the creation of digital artefacts. Rather than learn the theory of networking and network security, it would be far more rewarding and stimulating for students to actually build a network using real network gear and Raspberry Pi computers, or model a network using a tool like Cisco Academy's Packet Tracer.



■ Devote more time to practical activities

## Flip the learning of the theory

We employed some strategies to ensure that the more traditional learning activities took place outside of our precious lesson time. We adopted a flipped-learning approach in which students would study the theoretical content for a topic out of lesson and then we would assess these briefly in lessons using a light-touch MCQ so that students could see their progress.

To do this, we identified relevant content that could be learned from books, online, and videos so that students would then improve their own understanding. We prescribed 12 topics per term in each year in advance and then revisited the same 12 topics each term, each time requiring the students to go deeper in their understanding.



■ Allow students to study theoretical content for a topic out of lesson, then assess it briefly in lesson

Alternatively, you might prescribe 39 topics per year in advance and then revisit the same 39 topics each year, each time requiring the students to go deeper in their understanding. You need to signpost students to the locations where they can access the content, and then students use a simple system to demonstrate that they've visited the topics featuring the following:

- Description/title of the topic
- Date
- Minimum of words
- Some images to convey understanding of the topic
- Good use of the page

We still relied on traditional exam-style questions and written responses as a more accurate means for assessment and reporting, but these were limited to just one 10 minutes every five or six lessons to ensure there was still plenty of time for hands-on practical activities. The exam-style assessments we created could be marked by students, which saved valuable teacher time and also taught the students other skills, helping to improve the quality of their responses.

## Extended projects

When we allowed students to work on extended projects lasting a term or longer, we observed that they were much more heavily invested in their own learning, and this led to them suggesting additional areas of study where they felt they would need support from teachers to further their learning.

It's entirely practical for students to work on an extended project in tandem with other whole-class learning in the classroom. You may have one class project serving as a 'core project', with students' own individual projects running alongside. Warning – don't try to assess student projects, only the student learning! You'll only end up creating overly complicated sets of criteria on which to judge their work.

If you'd like some free, friendly advice about planning your computing curriculum, contact author Alan O'Donohoe on [alan@exafoundation](mailto:alan@exafoundation) (HW)

**Q** AT WHAT AGE DO YOU RECOMMEND INTRODUCING ETHICS IN COMPUTING TO STUDENTS? IS IT SOMETHING THAT NEEDS TO WAIT UNTIL SECONDARY LEVEL?

**A** I don't think there is one appropriate age to introduce ethics in computing to students, and I definitely don't think it should wait until secondary level. Ethics should be an integrated part of computing education, especially considering the repercussions in our current digital age. I think the important question here is 'how' – ethics must be introduced at an age-appropriate level and in consideration of what young people understand about computing and the digital world.

As soon as students are using technology, they're exposed to ethical questions about its use. What is it appropriate to say online? How do we use computers responsibly? How does our own access to technology compare to others around the world and what does that mean? I found out someone else's password; what's the right thing to do?

**Q** HOW CAN I CHECK THAT THE RESOURCES I WANT TO USE ARE ETHICALLY SOUND?

**A** Abide by the computer professionals' code of conduct ([helloworld.cc/2wHwc2z](http://helloworld.cc/2wHwc2z)) as well as the code of conduct for teachers and your school's digital use policy.

You might look at how representative of society the resources are in terms of gender, age, and ethnicity – are they inclusive? Consider your own use of the materials – is that ethical? Are you abiding by the terms of the licence on the material?



## **Q** DO YOU HAVE ANY SUGGESTED ASSIGNMENTS I COULD SET MY SECONDARY CLASS TO GET THEM TO BETTER INVESTIGATE THE IMPORTANCE OF ETHICS?

**A** It's a very wide subject, but these ideas should get secondary students involved in some of the debate/questions around ethics and computers:

- Self driving cars: who is liable in case of injury or death during a collision?
- Worker robots: how should we treat them? In the future, should they pay taxes?
- Copyright: how accessible should information on the internet be? Do you agree with the sentencing of the founders of Pirate Bay?
- What should companies be able to do with the data we share with them?

Set them up with some case studies on privacy and copyright or get them to research an important dispute.



## **Q** HOW CAN I ENSURE MY SCHOOL'S OWN POLICIES CONFORM TO THE ETHICS THAT I'M WANTING TO TEACH IN THE CLASSROOM? IS THERE A RECOMMENDED CODE OF PRACTICE, FOR INSTANCE?



**M**any schools share an acceptable use policy (AUP) with their staff and students. In my experience, while this might be based on ethics, it's often presented as a set of rules. This policy, which everyone needs to agree to, is a good place to highlight ethical standards and also a handy reference for the classroom.



**Q** DO YOU HAVE ANY RECOMMENDED MATERIALS FOR ME TO ADDRESS THE SUBJECT OF ETHICAL HACKING, WITHOUT RISKING COMPLAINTS FROM PARENTS?

**A** Look for materials termed 'cybersecurity', rather than mentioning the word 'hacking' if you are concerned about complaints. However, the word 'hacking' is a neutral term, and I believe that the more it is used to refer to ethical hacking and in a positive light, the more informed parents and other members of society will become.

**Q** CAN YOU RECOMMEND ANY GOOD BOOKS ON COMPUTER ETHICS PLEASE?

**A** Here are a few to get you started:

- *Computer Ethics*, Deborah G Johnson
- Historical: *The Human Use of Human*

*Beings: Cybernetics And Society*  
■ *Computer Science Education*,  
Sue Sentence, features a nice chapter on  
interaction between society and computers.

**FURTHER READING**

- [www.cyberaware.gov.uk](http://www.cyberaware.gov.uk)
- [helloworld.cc/2PA8dJz](http://helloworld.cc/2PA8dJz)
- National cyber security centre:[www.ncsc.gov.uk](http://www.ncsc.gov.uk)
- BCS code of conduct:[helloworld.cc/2N9Lkis](http://helloworld.cc/2N9Lkis)



# 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](#) or using the **#HelloWorld** hashtag. Alternatively, email us with 'Teacher Letter' in the subject line ([contact@helloworld.cc](mailto:contact@helloworld.cc)).

## When the student becomes the teacher

**Dear Hello World,**

My club is about coding, where we learn a bit about computing. We alternate each Thursday between Scratch and micro:bit. Micro:bit is a lot of magic in a small package, with an accelerometer, radio signals, a 5-by-5 grid of red LEDs and total control over lots of other robotics.

I was one of 12 Young Coders who participated in the Young Coders Conference in Tate Modern on February 2018. Ever since, I have been invited to run workshops at a few Raspberry Pi jams. This motivated me to do more workshops and share my ideas with my peers and similar aged children. I was stuck for ideas on where to go next, but then it just hit me: 'Well, what better place than to do it at school?' From then on, I started to share my knowledge with the children in my school through the lunchtime club. I am lucky enough to have been given an opportunity to run this. I sincerely thank my headteacher Mrs Lorimer and my class teacher Miss McPeane who have both supported me through this.

It was agreed with my teacher that every Thursday lunchtime we will hold this coding club that includes simple micro:bit and Scratch projects. After some time I



asked the attendees about how they would like their club to go ahead. I prepare the projects accordingly when they do. Sometimes I relate the projects to the week/day the club is held on (for example, during Ramadan we did a Scratch project regarding the Islamic festival of Ramadan).

I have 10 students. When I told Ben – one of my students – I got the micro:bits from London, his jaw hit the floor! Later that day, he told me I was really lucky to get them. It made my heart rise! Savannah, another one of my club attendees, once mistook Wednesday for Thursday and she was upset that she couldn't go to the club that day. Trudy, another attendee, interviewed me for a magazine she runs, known as the Meeply Weekly. I take feedback from them regularly which has shown me how enthusiastic they have been during this time.

In the future I would like to run the same coding club with more participants. I am also interested in running clubs in other schools. I am interested in making learning to code simple and easy.

**Avni, Age 10**

Isn't that just wonderful? Thanks so much for writing in, Avni, and thanks too for the work that you're doing, inspiring young coders! If anyone else has stories of brilliant students they want to tell us about, then please do get in touch.

## More budget demands

**Dear Hello World,**

I greatly enjoyed your most recent issue of the magazine, and have found many of the articles of particular use. My most vigorous head-nodding, though, was saved for the letters page, and the 'name supplied' letter you printed. Isn't it a pity that when discussing matters so pivotal, we find ourselves opting to remove our names, as I've had to do, for fear of comeback?

The letter in question, of course, talked about the growing budget problems that many of us teaching CS in schools are facing, to the extent of asking parents to help fund things beyond a level that I personally believe to be fair. Were it not for elements of the industry so willing to engage with teachers, we'd genuinely struggle to effectively cover everything we need to be talking to our students about.

I can only thank magazines like this for their work, and hope that in the future, someone will decide that education budgets need proper investment, rather than 'just enough'.

**Name supplied**

Thank you for your letter and, as always, we'll try to share as many ideas and resources as we can every issue. Please do send in anything you feel would be helpful to others, and we'll have another batch of ideas for you in our next issue... **(HW)**

# WHAT HAVE WE LEARNED FROM TEACHING COMPUTING?

Four years since the introduction of the new curriculum, what do we now have a better understanding of to enable us to teach computing effectively?

STORY BY Neil Rickus



■ Neil recently shared our experiences of teaching computing in English schools at the US Computer Science Teachers' Association (CSTA) conference

**I**t has been over four years now since the computing curriculum was introduced into English schools. During this time, our understanding has developed significantly in a number of areas, particularly around how to teach the subject, how to engage all learners, and some of the difficulties pupils experience when accessing curriculum content. Drawing on the views of a number of sources, including magazine articles, academic reports, Twitter conversations, and personal experience, we'll examine some of the most important lessons learned.

## Pedagogy

Some of the most extensive research has been around how the subject is taught to children, particularly those of primary age. While there may be a place for copy code activities, as outlined in the last issue of *Hello World*, it's likely that this

approach will have a limited impact on pupils' development of computational thinking ([helloworld.cc/2Py7Mjv](https://helloworld.cc/2Py7Mjv)) and also impact on their motivation in lessons. We therefore need a range of techniques to help develop pupils' understanding, including unplugged activities, which have been recognised in Sentance and

**" WE NEED A RANGE OF TECHNIQUES TO HELP DEVELOP PUPILS' UNDERSTANDING, INCLUDING UNPLUGGED ACTIVITIES**

Waite's research as generating, "a high level of understanding for very young children" ([helloworld.cc/2P2rumF](https://helloworld.cc/2P2rumF)), and inquiry-based approaches, such as 'guided exploration' ([helloworld.cc/2I4TEgl](https://helloworld.cc/2I4TEgl)), where pupils are given a few commands to explore, rather than unstructured tinkering.

Girls in particular may find certain pedagogical approaches disengaging, which brings us on to...

## Gender

The Royal Society's After the Reboot

- Computing education in UK schools report ([helloworld.cc/2jn52px](https://helloworld.cc/2jn52px)) discussed how around 80% of pupils taking GCSE computer science were boys.

To make lessons more engaging, The Girls' Day School Trust ([helloworld.cc/2MvN9Xp](https://helloworld.cc/2MvN9Xp))

highlights the importance of providing collaborative activities and opportunities for talk, while the Institute of Physics ([helloworld.cc/2Lk1vok](https://helloworld.cc/2Lk1vok)) outlines the importance of assigning roles for practical work, which is vital when using approaches such as 'paired programming' ([helloworld.cc/2MsCtbX](https://helloworld.cc/2MsCtbX)).

Katie Vanderpere-Brown, Director of ICT and Area Coordinator Vocational and Computing at Saffron Waldon Count High School, who has successfully encouraged girls to study computer science qualifications, also highlights the

importance of having engaging lessons and making the subject as rigorous as possible ([helloworld.cc/2MNTpZS](https://helloworld.cc/2MNTpZS)). The need for positive role models, as mentioned in Microsoft's recent blog post ([helloworld.cc/2MuWYoD](https://helloworld.cc/2MuWYoD)), appears essential, which is echoed by the CAS #include group

## INTERVIEW WITH CAS MASTER TEACHER MATT MOORE

**Q1 - What is the most important thing you have learned about how to teach computing?**

**A1 -**Teaching the children to fail. Programming can be very unforgiving, which children can find challenging.

**Q2 - What technologies have you used to engage pupils in lessons?**

**A2 -**Makey Makey with Scratch to inspire younger pupils. Older pupils love the micro:bit, and the range of resources available makes it really accessible.



**Q3 - What are your next steps to deliver computing more effectively?**

**A3 -**Include more technical vocabulary during lessons, rather than just focusing on concepts and skills.

([helloworld.cc/2Pxnk7a](http://helloworld.cc/2Pxnk7a)), which outlines the need for technology with a range of outputs, which brings us on to...

### Physical computing

Recent research indicates how the use of physical computing activities, which involve programming components, including LEDs, buzzers, buttons, and motors, is extremely motivating for pupils, as they provide instant feedback during coding activities. Physical computing devices, such as the BBC micro:bit ([microbit.org](http://microbit.org)), Crumble ([helloworld.cc/2o2VHq0](http://helloworld.cc/2o2VHq0)), and the Raspberry Pi ([www.raspberrypi.org](http://www.raspberrypi.org)) are increasingly evident in schools and can be used to develop a range of creative projects linked to other areas of the curriculum. There are also many programming environments available for physical computing devices to facilitate the transition from block-based programming environment (such as Scratch) to text-based languages...

### Text-based programming

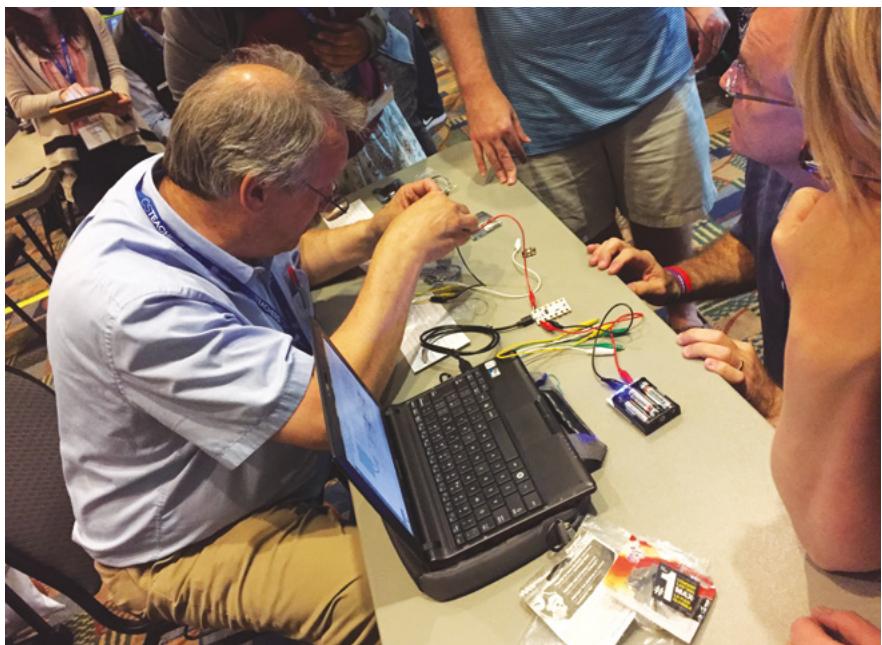
At Key Stage 3 (11-14 years), the English national curriculum states pupils have to be able to use at least one text-based language. Pupils are faced with a multitude of barriers when utilising a text-based language for the first time, including the need to interpret unfamiliar syntax and

([www.alice.org](http://www.alice.org)), GP ([gpblocks.org](http://gpblocks.org)), and Greenfoot ([www.greenfoot.org/door](http://www.greenfoot.org/door)) to help with this transition, particularly for pupils at Key Stage 3...

### Key Stage 3 (11-14 years)

In secondary schools, the old ICT curriculum at Key Stage 3 (KS3) was often delivered by teachers with other subject specialisms. Therefore, the shift towards the inclusion of more computer science-related content within the computing curriculum has meant some staff are lacking in theoretical and technical knowledge ([helloworld.cc/2Ne9eX0](http://helloworld.cc/2Ne9eX0)). KS3 also appears to be the time when many pupils become disengaged with the subject, so it's vital to give learning a purpose and to ensure it's relevant to pupils' interests. While this is now often the focus of many schemes of work, further opportunities can become available through enabling pupils to work towards competition entries. Initiatives such as the Teen Tech Awards, as outlined in the previous issue of Hello World, allow pupils to develop a product to meet a specific need, while other competitions, including the University of Manchester's well-established Animation '18, allow pupils from both secondary and primary schools to demonstrate their knowledge... ➤

error messages, along with the need for well-developed problem-solving skills. Attempts have been made to examine appropriate strategies for pupils to successfully transition from block- to text-based programming, including those proposed by White and Dorling ([helloworld.cc/2wjjeqx](http://helloworld.cc/2wjjeqx)), and Utting et al ([helloworld.cc/2Ne9eX0](http://helloworld.cc/2Ne9eX0)). Various environments can be used, such as Alice



■ Teachers require time and appropriate CPD to develop their pedagogical and technical knowledge

## ■ Key Stages 1 and 2 (5-11 years)

Within primary schools, the quality of provision also varies, with the Royal Society After the Reboot report highlighting how “32% of primary teachers feel more confident teaching the earlier stages of the curriculum than the latter”. However, recent research outlines that there is some excellent practice evident, with pupils particularly enjoying the more creative aspects of the curriculum ([helloworld.cc/2Ne9eX0](#)). Within both UK homes and primary schools, the use of tablet devices is becoming more prevalent ([helloworld.cc/2PyGIWP](#)) and can lead to pupils being unfamiliar with traditional input devices. Basic IT skills may also be lacking, which are essential for children’s future work and education; an issue previously addressed at GCSE level...

## Key Stage 4 (14-16 years)

This summer saw the last cohort of pupils taking a GCSE in ICT, as the qualification has now been replaced with the Computer Science (CS) GCSE. While some ICT qualifications do exist for this age group, they’re often not equivalent to a GCSE and don’t count in league tables. The Roehampton Annual Computing Education Report (TRACER) ([bcs.org/roehampton](#)) outlined, in 2017, only 11.9% of all pupils at KS4 took GCSE CS, which, although a small increase on the previous year, supports the Royal Society’s recommendation that “the qualification



■ Creative projects, linked to children’s interests, have been shown to effectively engage pupils in lessons

academically, closer to a typical mix for ethnicity, and more likely to be female”. TRACER also outlined how pupils

years and, in addition to echoing the information above, many also stated the need for significant, funded CPD ([helloworld.cc/2wikkmz](#)). This academic year sees the UK Government pledging £84 million to establish the National Centre for Computing, which will aim to provide additional CPD, including training 8,000 teachers to teach computing and the production of additional teaching resources ([helloworld.cc/2BD0ulm](#)). I wonder what we will learn in the next four years? (HW)

## “ THE USE OF TABLET DEVICES CAN LEAD TO PUPILS BEING UNFAMILIAR WITH TRADITIONAL INPUT DEVICES

landscape needs urgent attention to ensure the broadest range of pupils become equipped with digital skills”. TRACER provides further details around the students choosing to study CS and ICT, with those opting to study the latter qualification being, “(on average) from less affluent background, weaker

studying GCSE CS typically get half a grade lower than in their other subjects...

### Next steps

So, how can some of the issues above be addressed? I recently asked members of the CAS community forums what they had learned during the past four

**Neil** is a Senior Lecturer in Computing Education at the University of Hertfordshire. He is a CAS Hub Leader and a Raspberry Pi, Google and Microsoft Certified Educator.

# IN SEARCH OF QUALITY PROFESSIONAL DEVELOPMENT

How does one define quality computer science professional development?

**STORY BY** Beverly Vaillancourt and Emmanuel Schanzer

**H**ere in the US, the Computer Science Teacher Association (CSTA) is an independent nonprofit with members representing teachers, advocates, researchers, companies, and CS-Ed providers.

As a neutral party, the CSTA creates standards, convenes members of the community, and plays an active role in computer science (CS) education nationwide.

This year, the members of the CSTA's

PD Steering Committee were tasked with the question of defining quality CS professional development (PD). The intent was to develop a process for evaluating the plethora of CS PD offerings in a way that would allow teachers to make valued and cost-effective choices when choosing which PD to experience.

While there's universal agreement that CS education is core to every child's education, only 40% of schools in the United States and less than 50% in the UK teach CS. How then to engage teachers in building their CS knowledge and teaching practices that translates into opportunities for teaching CS both in structured CS classes as well as integrating CS throughout the curriculum? From the start, the CSTA PD Committee focused on the importance of the impact of PD on students, both in the number of students who benefit from teachers engaged in CS PD, but also on the critical need for reaching all students, no matter the location, income level, gender or racial and ethnic biases, or challenges to learning.

## The process

CS PD comes in many sizes. Some PD efforts focus on a particular curriculum or language, while others focus more broadly on issues of equity or pedagogy. Some are meant to be used in standalone CS classes, while others might integrate CS into other subject areas through informal settings. Some PD is well established and well funded, others



# COMPUTER SCIENCE



are part of start-up companies or small nonprofits. Some require teachers to be physically present, while others are online or a combination of face-to-face and online learning.

This wide variety of CS PD is a bonus for teachers who are searching for teacher training that fits their style of learning, time commitment, financial resources, and interests. It also provides teachers with opportunities to experience new teaching strategies and build their confidence for learning how to code or expand their knowledge of programming languages.

is made up of representatives from across the community, including former and current CS teachers, curriculum designers, and researchers. While quite diverse in experience and background, there was consensus that 'high-quality' PD should:

- Include content that equips teachers to provide students with an understanding of computational thinking and basic programming skills
  - Be interactive in that teachers apply their skills in ways that will help their

- Inspire teachers to reflect and bring new strategies for teaching CS

- Be grounded in research and aligned to standards

## **From values to rubrics**

Our goal was not to 'measure' PD, but rather to define a minimum level of quality and give providers multiple dimensions by which to demonstrate it. Choosing these dimensions was a unique opportunity to formalise the values of the CS-Ed community, highlighting dimensions that are widely recognised as important or lacking from current offerings. For example: if there's a broad desire to focus on the importance of real research and evidence for what works, including a scale for published results in the rubric becomes a subtle way of making it part of the DNA of quality CS PD.

Developing a rubric along these lines sparked many discussions among the members of the CSTA PD Steering Committee. Some of the elements of quality PD were clear cut. There was broad consensus that important elements of quality PDs include (a) alignment to CS standards, (b) outreach

With so much variety in place, the question was how to create an appropriate and useful measure of PD quality.

The CSTA PD Steering Committee first decided on an iterative model for defining quality PD: develop a rubric → test it on a wide range of known PDs → refine the rubric → test again → repeat. The Committee

students engage with CS in fun and exciting projects and activities

- Be collaborative
  - Reach beyond the PD experience with sustained support and coaching, as needed

to underrepresented minority (URM) students, (c) accommodations for differently abled students, and (d) ongoing peer support after the PD is over.

Other conversations, however, were far more nuanced. How does one compare a PD that impacts tens of thousands of students to one that purposefully focused on a small, specific population? Are PDs that have been around for years inherently of higher quality than those that are brand new? How do we compare innovative but untested approaches to long-running approaches with a strong research base?

To resolve these issues, it was decided

with a greater degree of objectivity. For example, a programme either has outcomes or it doesn't, and those outcomes can be published in peer-reviewed journals or not. Rating the actual quality of those outcomes (or the experimental design!) gets into murkier waters. By design, the rubric seeks to measure only what is objectively verifiable, so questions like these are not part of the quality measurement.

### From rubrics to questionnaires

The rubric scales seamlessly translated into a questionnaire, which also includes questions about logistics (for example,



that the rubric would have scales for multiple dimensions of quality, and programmes' total scores would only be used to determine whether or not they pass a certain minimum threshold. Publishing scores would lead to system gaming and the false impression that a programme with a higher score must be inherently better than one with a lower score. By sticking to a 'minimum threshold' system, we allow well-rounded providers to accumulate points on multiple dimensions while also permitting highly targeted PDs to succeed on the strengths of their particular focus area.

We were also careful to choose scales

## “THE GOAL OF ASSESSING PROFESSIONAL DEVELOPMENT OPPORTUNITIES FOR TEACHERS IS AN ITERATIVE PROCESS”

when, where, costs, and so on). All information gathered through the questionnaire is placed on an easily readable PD opportunity chart. With a glance, teachers can see the description of the PD, how long it has been used, the number of students impacted, its mode of delivery, cost, and scholarship opportunities. They also have a link to



the PD registration page. With the chart, teachers can quickly sort through PD options and choose one that best fits their professional needs and educational environment.

While the rubric scores aren't shared with teachers, only PDs that meet the minimum threshold established by the CSTA PD Committee are included on the PD opportunity chart. This is critically important for teachers looking for high-quality and impactful PD and to the CSTA's credibility as an association dedicated to fostering high-quality CS education and teacher development.

### The future

The CSTA recognises that the goal of assessing PD opportunities for teachers certainly is an iterative process. As the questionnaire and rubric are used, changes will be

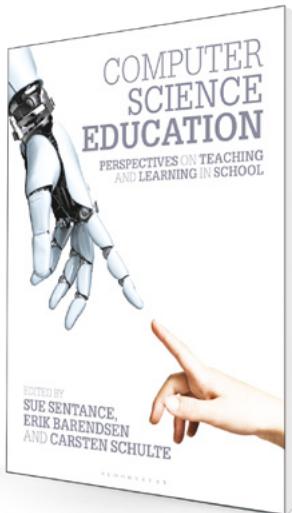
made to enhance the ability to truly determine the impact of PD experiences. The CSTA PD Committee will continue to analyse feedback from teachers and PD providers as it improves the process of evaluating PD for teachers, and its ultimate impact on improving and expanding CS education for all. **(HW)**

# COMPUTER SCIENCE EDUCATION: PERSPECTIVES ON TEACHING AND LEARNING IN SCHOOL

Computer science education researchers give 13 chapters of stimulating advice on computer science: what it is, why we teach it, and how to improve our teaching

## INFO

BY Sue Sentance, Erik Barendsen and Carsten Schulte | PUBLISHER Bloomsbury | PRICE £22.49 | ISBN 978-1-350-05710  
[URL helloworld.cc/2PH4fzp](http://helloworld.cc/2PH4fzp)



**I**t's divided into three parts: Why Teach Computer Science in School, Aspects of Teaching and Learning, and Delving Deeper: Research-led Teaching of Computer Science. I found something of interest in each part and every chapter, starting with the first, The Nature of Computing as a Discipline, which discusses whether computing is a field of engineering and design, a sort of mathematics, or a science. The point is made that which you 'choose' influences how you see the subject and teach it.

After Computational Thinking: A Competency Whose Time Has Come, which looks at computational thinking in the classroom, Investigating Attitudes to Learning Computer Science looks at fixed and growth mindsets and the effect a learner's, and a teacher's, mindset has on learning and teaching. Computer Science, Interaction and the World by the editors themselves discusses the

teaching of CS in terms of three levels of interaction: human/computer, hybrid network, and computing and society.

Part 2 starts with Perspectives on Computing Curricula and introduces a useful distinction between the intended, implemented, and attained curriculum. Teaching of Concepts is from the QMUL group responsible for CS4FN and Teaching London Computing, so it's no surprise their chapter is full of tricks and role playing.

Teaching Programming is useful from a historical perspective, while Teaching Computing in the Primary School is by Tim Bell, of CS Unplugged fame, who discusses the need for concentrating on the 'bigger picture' approach to computing.

This part ends with Assessment in the Computing Classroom, which covers research into assessing, describing ways of assessing, and ends with techniques for assessing projects and two case studies.

Misconceptions and the Beginner Programmer, in the last part of the book, lists 41 common misconceptions that our students may "have about specific programming constructs or about the way programs

work in general". It's really unfair to single out any one chapter, but if I had to keep only one this would be it.

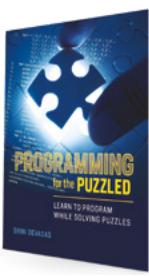
Equity and Inclusion in Computer Science Education covers approaches to our major problem in Computer Science Education in terms of gender, ethnicity and learning differences/disabilities, with five strategies for helping raise equity and inclusion.

Language and Computing considers the language we use to teach, and has 10 recommendations on helping students with variations in the language we use, and making students and teachers more aware of the way language can affect learning.

Finally, Taxonomies and Competence Models is the most theoretical chapter, but has useful practical information on the learning objectives for CS across different countries, taxonomies in general, and for CS in particular, finishing on the idea of competencies.

Key concepts are highlighted in each chapter together with examples and example activities, so even those chapters that might at first glance seem dry and theoretical will give you something to think about and use. (HW)

# PROGRAMMING FOR THE PUZZLED

**INFO**BY Srinivas Devadas | PUBLISHER MIT Press | PRICE £18.99 | ISBN 9780262534307  
URL [helloworld.cc/2MQWRmc](http://helloworld.cc/2MQWRmc)

There are lots of approaches to teaching introductory programming courses, and it's perhaps not reasonable to expect that what works for one group of students will work for another. While teachers may well have had success with animation, games, apps, and physical computing, Devadas's book takes recreational puzzles as its motivating context.

This is computationally rich territory, as these problems are amenable to algorithmic solutions if approached the right way. Devadas's approach doesn't labour the point

about computational thinking, but the book is much more about developing a systematic algorithmic approach to problem solving than Python programming, although worked solutions are presented in Python. A particular strength is the way the author scaffolds the construction of data structures.

It's not an easy text, but would offer much to a well-motivated GCSE or A Level student with an interest in mathematics and puzzles. This is the course text for Devadas's MIT course with the same name, available at: [helloworld.cc/2NmZtWx](http://helloworld.cc/2NmZtWx) (HW)

## COMPUTER SCIENCE K-12: IMAGINING THE POSSIBILITIES

**INFO**BY Doug Bergman | PUBLISHER CreateSpace | PRICE £9.86 | ISBN 978-1986171816  
URL [helloworld.cc/2oepcFD](http://helloworld.cc/2oepcFD)

Doug Bergman is head of CS at a private school in Southern Carolina. He's an enthusiastic, innovative, and effective teacher, and has drawn on his years of experience to write this short guide to planning, teaching, and assessing CS.

Bergman's approach is unashamedly one of project-based learning, encouraging students to take much more responsibility for their learning, and the outcomes of their work, than more traditional approaches. Alongside the advice

for those teaching CS, there's much for those who find themselves leading CS education in their schools, from curriculum design and resources to the construction of assessment criteria.

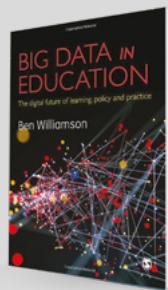
The book's written in a lively style, mirroring Bergman's infectious enthusiasm for CS: teachers looking for a personal and practice-based account of how to teach programming and engage students in digital making will find much to interest them here, not least Bergman's own outline curriculum plans; those looking for references to the latest academic research might be better suited elsewhere. (HW)

## ESSENTIAL READING

A few recent books on big data, machine learning, and AI in education, and some of the ethical issues that these raise

### BIG DATA IN EDUCATION

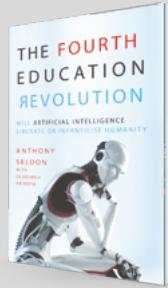
BY Ben Williamson  
PUBLISHER Sage  
PRICE £27.99  
ISBN 9781473948006  
URL [helloworld.cc/2wigtct](http://helloworld.cc/2wigtct)



An insightful and authoritative overview of the state of play of how big data and the applications of algorithms to the classroom affects education and schooling today and in the near future.

### THE FOURTH EDUCATION REVOLUTION: WILL ARTIFICIAL INTELLIGENCE LIBERATE OR INFANTILISE HUMANITY?

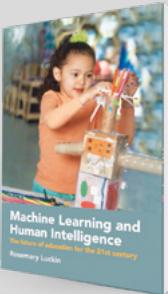
BY Anthony Seldon and Oladimeji Abidoye  
PUBLISHER The University of Buckingham Press  
PRICE £14.99  
ISBN 978-1908684950  
URL [helloworld.cc/2PFdpN5](http://helloworld.cc/2PFdpN5)



The former Master of Wellington College argues that the use of machine learning in class to personalise learning will soon morph into intelligent tutoring systems that could prove more effective (in some sense) than human teachers.

### MACHINE LEARNING AND HUMAN INTELLIGENCE

BY Rosemary Luckin  
PUBLISHER UCL IoE Press  
ISBN 9781782772514  
URL [helloworld.cc/2LvyE0k](http://helloworld.cc/2LvyE0k)

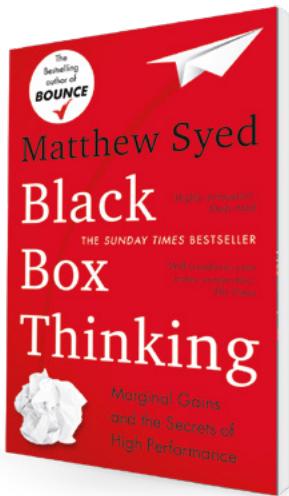


Rose Luckin argues that AI is not all that is claimed for it, and that there is much more to human intelligence than its artificial counterpart, although AI might nevertheless help extend human capabilities

# BLACK BOX THINKING IN COMPUTER SCIENCE

A practical look at encouraging pupils to accept failure and learn from it based upon Matthew Syed's *Black Box Thinking*

STORY BY Phil Jackson



**T**he ability to learn from mistakes is an attribute many computer science teachers will identify as essential to succeed in the subject. The 'growth mindset', learning from mistakes, using failure to make future gains, is high on the agenda in Matthew Syed's book, *Black Box Thinking* (BBT).

For a computer science teacher, this can be considered on two fronts.

How can we ensure we use Black Box Thinking when planning materials and delivery for lessons? But also, and maybe more importantly, how can we inspire our students to be brave and accept failure as part of the learning process?

When training, teachers are encouraged to self-evaluate after each session. Completing lengthy evaluations becomes difficult when teaching more lessons. However, a quick Post-it Note as a reminder to make a change later can be completed while delivering a session. The Post-it Notes can then be actioned at a convenient time.

Anecdotally, promoting Black Box Thinking can be more difficult with higher ability students as they may be more familiar with achieving success first time. The phrase 'right first time' is often used in education, but can conflict with the idea of rapid improvements and reflections. As

budding modern-day computer scientists, students should be ready to release often and patch quickly rather than waiting for that illusive 'perfect' build. Industry has pressures other than simply releasing a product when it's ready, and embedding this attitude early can contribute to those all-important employability skills. This issue is further highlighted during the testing section of a project. Students tend to ignore failed tests and instead record loads of 'as expected' results.

## Reflecting and making improvements

Having limited access to actual beta testers or end users, students also tend to draw their own conclusions on what the end user requires or how successful their system is in achieving its objectives. This is covered in BBT by the phrase 'Cognitive Dissonance', adapting the evidence to fit pre-existing beliefs. Many would argue that the Windows 8 interface was a step backwards in usability, yet Microsoft was brave in attempting to build a mobile first interface that would be future-proof, at least in the short term. They listened, they adapted, and they rebuilt. Failure in the short term leads to success in the long term.

Recent changes in some A Level components have allowed for extended feedback from peers, including the 'Discussion' and 'Post Prototype Refinement' sections in the Eduqas/WJEC specification. This has developed students' ability to

## BLACK BOX THINKING: KEY POINTS

- Culture: learning from and acting on mistakes
- Looking at the evidence and removing barriers
- Testing and assessing results
- Making gains
- Accountability
- Future growth

accept constructive feedback and consider alternative approaches. The final solution is almost always an improvement on the prototype, and students are able to reflect on this during their evaluation.

Finally, for Black Box Thinking to take place, the classroom environment needs to be one where failure is accepted as an opportunity for improvement. Some students may suffer from a lack of confidence and may need encouragement to respond to feedback in a positive manner. But, by persevering and developing a growth mindset, students receive an education that is more than just information in text books; they develop skills that will benefit them throughout life. (HW)

Syed has now released a version of BBT aimed specifically at encouraging children to accept failure as part of growth, titled *You Are Awesome*.





**MILES BERRY** PRINCIPAL LECTURER

# GCSE AND A LEVEL CS

The highs and lows of this year's exam results

**A**ugust saw GCSE and A Level results days in England: as a co-author of the TRACER analysis of pupils' and schools' performance in these exams, I was understandably interested in how things went with the most recent series of exams.

At GCSE, I am, of course, delighted that so many students did so evidently well in their exams: congratulations to them and their very hard-working teachers! That said, I'm still appalled that, due to suspected malpractice, exam regulator Ofqual took the decision to give no marks for anyone's work in the practical programming project (the NEA), while still insisting that schools set 20 hours aside for student to work on this literally pointless task. When the BCS successfully argued for the inclusion of CS in the EBacc performance measure, they said: "Maintaining a significant project based assessment component is one of the best ways to ensure pupils have the chance to solve complex, challenging computational problems that also demonstrates their potential for being innovative and creative, which we feel is a very important aspect of developing computational thinking."

It astonishes me that, as things stand, students could, theoretically, have got an A\* in GCSE CS without being able to, well, code. As we move forward, we must get back to a position where practical programming counts: there's a persuasive case made for mandatory practical endorsement, as we see in GCSE science, but I'd like the boards to look seriously at on-screen programming exams, or day-long, unseen lab work.

This summer saw the last ever GCSE ICT: few would mourn its passing, but the lack of any replacement disturbs me, as it seems to result in fewer students studying any aspects of computing at GCSE, a narrowing of the subject's scope to CS only, and a rather less diverse cohort than previously. If we're serious about CS for all, and indeed IT skills and critical digital literacy for all too,

then perhaps a broader based GCSE in computing (i.e. foundations, applications and implications) might serve our students (and their future employers) better than what's on offer at the moment?

At A Level this year, despite the more demanding specifications, grades are up, the number of girls taking the subject increased by nearly 50%, and the overall numbers are now (just) higher than they were back in 2003. Unlike at GCSE these days, practical programming is a big part of the A Level, and students across the country produce some amazing work, including some fab projects based on the Raspberry Pi, such as Andrew Mulholland's PiNet ([pinet.org.uk](http://pinet.org.uk)).

Even if A Level CS is still not an entry requirement for CS degrees (!), the Russell Group acknowledge A Level CS as being useful preparation for many degree subjects (maths, the sciences, engineering, medicine, economics, and so on), recognising that work in all these domains is increasingly likely to demand at least some coding skills. And yet, only 36% of schools and colleges offered this A Level in 2017, and even in those that do the class size is typically very small and may well be economically unsustainable. Perhaps it would be different if CS, like the other sciences, was recognised as a 'facilitating subject'; in other words, one that leaves open lots of options for degree choices? Latin and Greek both make this list: there's nothing wrong with either of those, but perhaps studying CS at A Level might actually be better preparation for university (and employment) these days? (HW)

**Miles** 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.

# "HELLO, WORLD!"

Everything you need to know about the new computing and digital making magazine for educators

## Q WHAT IS HELLO WORLD?

**A** Hello World magazine is a new 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 sponsored by BT.

## 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 new 100-page magazine will be available three times per year in time for each new term in January, April, and September. Would you like it to be available more frequently? Let us know!

## IT'S FREE!

Hello World is free now and forever as a Creative Commons PDF download. You can download every issue from [helloworld.cc](http://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 for a FAQ section 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? We'd love to hear from you.

## GET IN TOUCH

## FIND US ONLINE

[www.helloworld.cc](http://www.helloworld.cc)

 @HelloWorld\_Edu

 fb.com/HelloWorldEduMag

Want to talk? You can reach us at:  
[contact@helloworld.cc](mailto:contact@helloworld.cc)

SUBSCRIBE  
IN PRINT  
TODAY!

PAGES 28-29



(Hello  
World)

helloworld.cc