(Hello World)

THE MAGAZINE FOR COMPUTING & DIGITAL MAKING EDUCATORS

Issue 25 | Sept 2024 | helloworld.cc

A PROBLEM-FIRST APPROACH
Focusing on GenAI as a problem-solving tool

LEADERSHIP IN THE AGE OF AI
An AI implementation framework for administrators

UNDERSTANDING AI FOR EDUCATORS
A course for boosting AI literacy in the classroom

PLUS

AI UNPLUGGED • MOONHACK 2024 • **GROUNDED COGNITION** • HOSTING A STEM SUMMER CAMP
NEW WEB PROJECTS • **ASTRO PI** • CSED IMMERSION PROGRAMME • **LLMS IN THE CLASSROOM**
SEAME FRAMEWORK IN PRACTICE • MICRO:PEGS • **BEBRAS 20TH ANNIVERSARY** • ICT VS CSED

Raspberry Pi Foundation

Teach and learn with the Raspberry Pi Foundation

Free for everyone anywhere in the world

Teachers' resources

Discover training, resources, and guidance to help you teach computing with confidence.

raspberrypi.org/teach

Project library

Browse 200+ online project guides that help young people create with digital technologies.

projects.raspberrypi.org

Research to practice

Explore our research into what works best in teaching school-age young people about computing.

raspberrypi.org/research

Coding at home

Watch our support tutorials and access engaging resources for your child.

raspberrypi.org/learn

Raspberry Pi Foundation, UK registered charity 1129409

HELLO, WORLD!

Did you notice something slightly different about this issue's cover? If you've ever used a generative AI tool to create an image, you'll know that the first few iterations are generally a little off. You have to work on your prompts to get the image you want. A human 3D artist created our new cover, showing the 'like' symbol with extra fingers, to nod towards the fact that generative AI tools always need a human to give input and to check their outputs.

Since our 'Teaching and AI' issue a year ago, educators have been grappling with understanding the place of generative AI in the classroom, while also being mindful of the potential risks to young people. In this issue, you'll hear from the team at Gwinnett County Public Schools in Georgia, USA, where they're teaching about AI throughout their K–12 curriculum (pages 28–29). Aidan Weston shares practical ideas for teaching with AI in the classroom (pages 42–44). Mark Calleja guides us through using the OCEAN prompt to help you reliably get the results you want from an LLM (pages 32–33). And Stefan Seegerer discusses why unplugged activities can help us focus on what's really important in teaching about AI (pages 40–41).

Generative AI is here to stay, and we hope this issue will help you learn more about its benefits and risks, and inspire you to explore how you can use it to improve your own teaching practice.

Meg Wang
Editor

FEATURED THIS ISSUE

KIP GLAZER

Kip is principal of Mountain View High School in California, USA. She introduces a framework for AI implementation aimed at administrators and leaders in K–12 schools on pages 26–27.

GRAHAM HASTINGS

Graham is head of computing at St John's College School in Cambridge, UK. On pages 60–61, he offers a solution to tricky crocodile clip circuits through 3D-printed micro:pegs.

DIVYA MAHADEVAN

Divya is a product manager at the Raspberry Pi Foundation. She discusses Paulo Freire's education model and how it applies to generative AI on pages 45–47.

(HW)

Hello World is the official magazine of the Raspberry Pi Foundation

Raspberry Pi Foundation

EDITORIAL

Editor
Meg Wang

Subeditor
Louise Richmond

Subscriptions
Sean Sayers

Social Media
Neena Patel

DESIGN

criticalmedia.co.uk

Head of Design
Lee Allen

Designer
Sam Ribbits

Photography
The Raspberry Pi Foundation, Adobe Stock

Graphics
Rob Jervis

Cover
© Trash, Folio Art

CONTRIBUTORS

James Abela, Rob Arnold, Pete Bell, Mac Bowley, Mark Calleja, Philip Colligan, Veronica Cucuiat, Vicky Fisher, Andrew Csizmadia, Riyad Dhuny, Ben Garside, Kip Glazer, Graham Hastings, Alexandra Holter, Simon Humphreys, Janine Kirk, Annabel Lindner, Sam Lovatt, Divya Mahadevan, Sarah Millar, Minimalist EdTech, Alan O'Donohoe, Janine Pavlis, Richard Pawson, Kaye North, Nick Nurock, Stefan Seegerer, Sue Sentance, Bonnie Sheppard, Faheemah Vachhiat, Laura Ventura, Aidan Weston

Supported by ORACLE

Contributing Partner

bcs The Chartered Institute for IT Academy of Computing

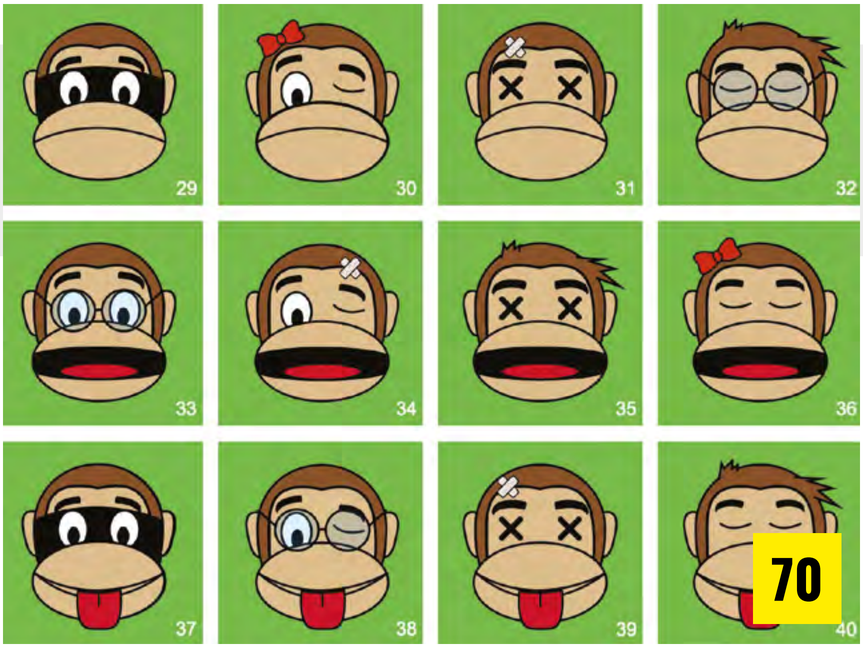
CAS

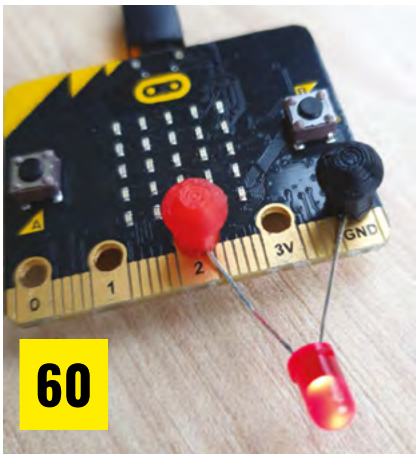
This magazine is printed on paper sourced from sustainable forests and the printer operates an environmental management system which has been assessed as conforming to ISO 14001.

Hello World is published by the Raspberry Pi Foundation, 37 Hills Road, Cambridge, CB2 1NT. 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.

(HW) CONTENTS

NEWS, FEATURES, AND OPINION





LEARNING

RESOURCES & LESSON PLANS

CONVERSATION

EXPERIENCE AI EXPANDS TO REACH OVER 2 MILLION STUDENTS

Welcoming new Experience AI partners in 17 countries and providing new AI safety resources

Philip Colligan

Two years ago, we announced Experience AI, a collaboration between the Raspberry Pi Foundation and Google DeepMind, to inspire the next generation of AI leaders. I am excited to announce that we are expanding the programme with the aim of reaching more than 2 million students over the next three years, thanks to a generous grant of $10m from Google.org.

Why do kids need to learn about AI?

AI technologies are already changing the world, and we are told that their potential impact is unprecedented in human history. But just like every other wave of technological innovation, along with all of the opportunities, the AI revolution has the potential to leave people behind, to exacerbate divisions, and to create more problems than it solves.

Part of the answer to this dilemma lies in ensuring that all young people develop a foundational understanding of AI technologies and the role that they can play in their lives.

That's why the conversation about AI in education is so important. A lot of the focus of that conversation is on how we harness the power of AI technologies to improve teaching and learning. Enabling young people to use AI to learn is important, but it's not enough. We need to equip young people with the knowledge, skills, and mindsets to use AI technologies to create the world they want. And that means supporting their teachers, who once again are being asked to teach a subject that they didn't study.

We announced our Experience AI expansion at Google.org's Impact Summit on 4 September

Experience AI

That's the work that we're doing through Experience AI (**experience-ai.org**), an ambitious programme to provide teachers with free classroom resources and professional development, enabling them to teach their students about AI technologies and how they are changing the world. All of our resources are grounded in research that defines the concepts that make up AI literacy; they are rooted in real-world examples drawing on the work of Google DeepMind; and they involve hands-on, interactive activities.

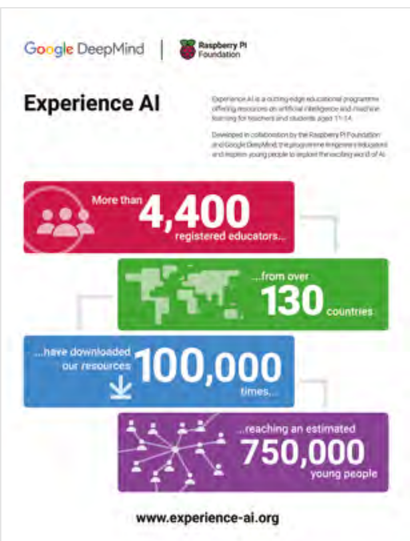
The Experience AI resources have already been downloaded 100,000 times across 130 countries and we estimate that 750,000 young people have already taken part in an Experience AI lesson.

In November 2023, we announced that we were building a global network of partners that we would work with to localise and translate the Experience AI resources, to ensure that they are culturally relevant, and organise locally delivered teacher professional

Experience AI provides free classroom resources for teachers to educate their students about AI technologies

development. We've made a fantastic start working with partners in Canada, India, Kenya, Malaysia, and Romania; and it's been brilliant to see the enthusiasm and demand for AI literacy from teachers and students across the globe.

Thanks to an incredibly generous donation of $10m from **Google.org** — announced at **Google.org**'s first Impact Summit — we will shortly be welcoming new partners in 17 countries across Europe, the Middle East, and Africa, with the aim of reaching more than 2 million students in the next three years.



The reach of Experience AI since its launch in April 2023

AI safety

As well as expanding the global network of Experience AI partners, we are also launching new resources that focus on critical issues of AI safety.

- **AI and your data:** helping young people reflect on the data they are already providing to AI applications in their lives, and how the prevalence of AI tools might change the way they protect their data.
- **Media literacy in the age of AI:** highlighting the ways in which AI tools can be used to perpetuate misinformation, and how AI applications can help combat misleading claims.
- **Using generative AI responsibly:** empowering young people to reflect on their responsibilities when using generative AI, and their expectations of developers who release AI tools.

Get involved

In many ways, this moment in the development of AI technologies reminds me of the internet in the 1990s (yes, I am that old). We all knew it had potential, but no one could really imagine the full scale of what would follow.

We failed to rise to the educational challenge of that moment, and we are still living with the consequences: a dire shortage of talent; a tech sector that doesn't represent all communities and voices; and young people and communities who are still missing out on economic opportunities and are unable to utilise technology to solve the problems that matter to them.

We have an opportunity to do a better job this time. If you're interested in getting involved, we'd love to hear from you: **helloworld.cc/ai-partner**. (HW)

Coming soon: new resources that focus on AI safety

THE EUROPEAN ASTRO PI CHALLENGE

Sending code to space

Vicky Fisher

Every year, the Raspberry Pi Foundation runs the European Astro Pi Challenge in partnership with the European Space Agency (ESA), in which young people across Europe can run their computer programs on the International Space Station (ISS).

There are two versions of the challenge, which is designed to be accessible to as many young people as possible.

Young people's code in space

Mission Zero offers young people the chance to write a simple program that takes a reading from the colour and luminosity sensor on an Astro Pi computer on board the ISS, and uses it to set the background colour in a personalised image for the astronauts to see as they go about their daily tasks. In total, 16,039 teams and 24,663 young people participated in the Mission Zero 2023/24 challenge.

Science experiments in space

Mission Space Lab offers teams the chance to run scientific experiments on board the ISS; 564 teams and 2,008 young people participated in 2023/24.

The challenge is used as an enrichment activity in schools, community groups, libraries, and more. We caught up with a few mentors from the last challenge to get their feedback on how taking part in Astro Pi impacted their groups. They shared how taking part helped connect young people to real technology, bridging the gap between theoretical coding and tangible outcomes. One Mission Space Lab mentor said, "Participating in Mission Space Lab offers students a great opportunity to work with the International Space Station, to see the Earth from above, to challenge them to overcome the terrestrial limits. It's very important."

Mission Zero entries running on an Astro Pi on board the ISS

Bringing space into the classroom and beyond

Mentors also shared that participating in the challenge inspired children to consider careers they previously thought were out of reach. Space exploration was no longer a far-away and theoretical idea for the children, but something connected to their everyday lives and their own learning. "Some of the people that I was teaching this to felt like becoming an astronaut was really difficult to learn … now it's not necessarily a distant thing to study," a Mission Zero mentor shared.

It has also been a joy to hear about how many schools are adopting Astro Pi as a part of space-themed weeks in which they used Mission Zero in conversations about space exploration, the Hubble Space Telescope, and learning the names of the

A young coder taking part in the Mission Zero challenge

> **STUDENTS CAN USE THEIR DIGITAL SKILLS AS SUPERPOWERS TO MAKE THE WORLD A BETTER PLACE**

stars. Others used Mission Zero across multiple subjects by designing images and holding art competitions based on the design, as well as learning about pixels and animations.

"We want students to use their digital skills as superpowers to make the world a better place, and this competition really aligns with that, because regardless of your race, your ethnicity, your gender, you can write some code that actually runs in space. And if you can do that, you can make medical tech, or you can solve the big problems that the adults of the world are still grappling with, so it's the opening up [of] opportunities," said another Mission Zero mentor.

As we prepare for the 2024/25 challenge, the Astro Pi team are implementing some exciting new improvements to support the participants based on mentor feedback. These include:

Mission Zero

- Adding a 'save' button to Mission Zero to allow young people to work on it across multiple sessions.
- Adding new code examples to the Mission Zero project guide. These have been selected from team submissions from the 2023/24 challenge.

Mission Space Lab

- Creating an online testing tool for Mission Space Lab so that it will be easier for teams to test whether or not their code works. It will feature new data and images captured from the ISS in spring 2024.

We are pleased to confirm the challenge dates for 2024/25, and we hope to see more young people than ever participating and sharing their creative projects.

Project launch dates

- **Now open:** Mission Zero and Mission Space Lab
- **24 February 2025:** Mission Space Lab submissions close
- **24 March 2025:** Mission Zero submissions close
- **April – May 2025:** programs run on the International Space Station
- **June 2025:** teams receive certificates

Find out more at **rpf.io/hw-ap**. (HW)

CELEBRATING 20 YEARS OF THE BEBRAS CHALLENGE

The world's largest computing challenge

Andrew Csizmadia

2024 is a year of celebration for the Bebras community worldwide as it celebrates the 20th year of organising and administering the Bebras Challenge. The germ of the idea for the Bebras Challenge was initially proposed by Professor Valentina Dagienė of Vilnius University in Lithuania, to introduce learners to the principles and concepts of computer science, which is referred to as 'informatics' in Europe, by giving them interactive puzzle tasks to solve. This was soon extended to introducing and promoting computational thinking concepts. It has now become probably the world's largest computing challenge, if not the oldest continuous computing challenge, with 3,936,642 participants in 71 countries worldwide last year. In the coming academic year, the Bebras Challenge will potentially be offered in 94 countries worldwide!

The UK Bebras Challenge

The UK Bebras Challenge provides an opportunity for all learners aged 6 to 19 years old to engage with interactive puzzle tasks that introduce learners to computer science concepts and principles, as well as allowing them to apply computational thinking concepts to attempt to solve the tasks they are presented with.

The number of learners who participate in the UK Bebras Challenge continues to grow each school year. In November 2023, 408,469 learners participated in the UK Bebras Challenge.

Each participating country has its own logo

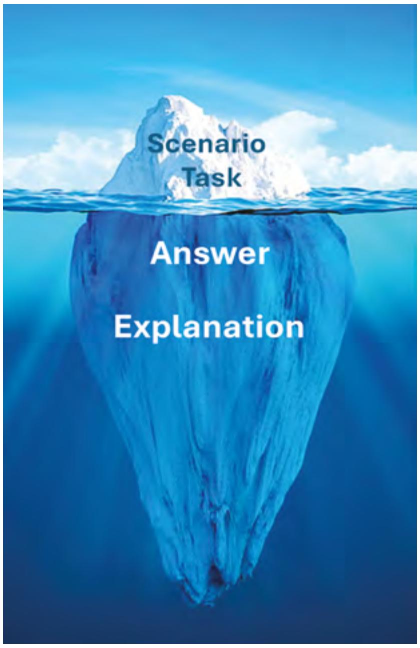
Thanks

Massive thanks must be given to Chris Roffey, author and computing educator; Dr Sue Sentance, director of the Raspberry Pi Foundation Computing Education Research Centre at the University of Cambridge; and Professor Peter Millican of Hertford College, University of Oxford, for their hard work in turning their vision of the UK Bebras Challenge into a reality.

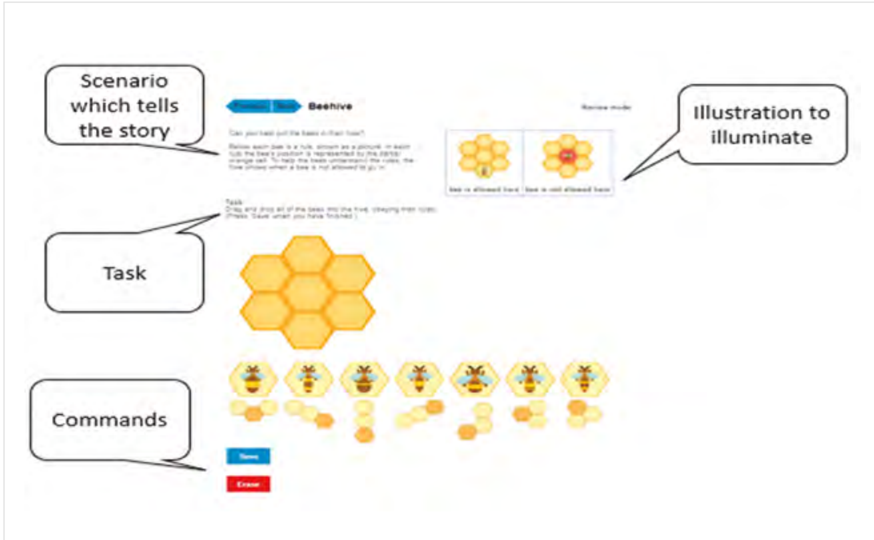
Thanks must also be expressed to the unsung heroes of the Bebras community — the teachers and educators like yourself who engage with and promote the Bebras Challenge, and organise and administer it annually for the learners they teach and support. We really couldn't do it without you.

Anatomy of a Bebras task

Over the last 20 years, the anatomy of a Bebras task has evolved. The tasks have been shaped and sculpted by groups of computing educators and computing education researchers, and more importantly, by feedback from teachers working with learners participating in the challenge. Developments in education technology have also provided greater functionality and usability for interactive tasks. To date, 234 academic publications have been published and can be accessed at: **bebras.org/publications**. These publications have investigated and improved the design of Bebras tasks, and explored the impact of the challenge upon the computing education community.



An iceberg illustrates the components of a Bebras task



The anatomy of a Bebras puzzle has been refined by input from computing educators, computing education researchers, and teachers

The analogy of an iceberg can be used to illustrate the components of a Bebras task. What's above the waterline is what the learner will see and engage with, and what lies below the waterline is what the learner will absorb and build upon. The components are as follows:

- **Scenario:** a carefully crafted story sets the scene, which is normally illustrated by a diagram.
- **Interactive task:** learners are presented with an interactive task to solve, based on the scenario they are presented with.
- **Answer:** a step-by-step worked-through solution is presented, explaining the rationale for the answer. For tasks with multiple-choice options, there are also explanations for wrong answers.
- **Explanation:** there is an explanation of at least one computer science principle and concept that learners engage with in solving the task. Also highlighted are the computational thinking concepts that learners use in solving the task.

REFRESHING THE UK BEBRAS CHALLENGE WEBSITE

As part of its ongoing commitment to supporting the Bebras Challenge, the Raspberry PI Foundation has taken the opportunity to refresh the front end of the UK Bebras Challenge website, as shown opposite.

Please take the opportunity to familiarise yourself with the redesigned site and revised content, and let us know if you have any suggestions or ideas for improvements.

> THE UNSUNG HEROES OF THE BEBRAS COMMUNITY ARE TEACHERS LIKE YOU

Computing educators worldwide work collaboratively to create, refine and refine again individual puzzle tasks that are finally presented to learners to attempt to solve.

Challenge

If you think you have an idea that could be developed into a Bebras task, email Andrew Csizmadia (**andrew.csizmadia@raspberrypi.org**), the Bebras competition manager at the Raspberry Pi Foundation, so you can discuss your idea and work together to develop it into a task that may appear in future Bebras Challenges.

Thank you for participating

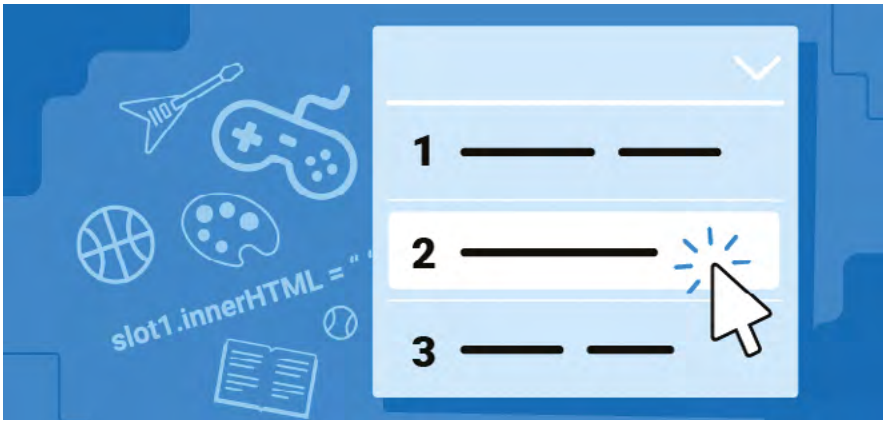
When you promote, organise, and run the Bebras Challenge this year, remember that you are part of a worldwide community that seeks to engage and challenge learners as they attempt to solve interactive tasks. (HW)

NEW WEB DESIGN PROJECTS: HTML, CSS, AND JAVASCRIPT

The Raspberry Pi Foundation's 'More web' project path shows young creators how to unleash the power of JavaScript to build awesome interactive websites

Pete Bell

Modern web design has turned websites from static and boring walls of information into ways of providing fun and engaging experiences to the user. Young creators can get on board with our new project pathway (helloworld.cc/more-web) to extend their web skills.

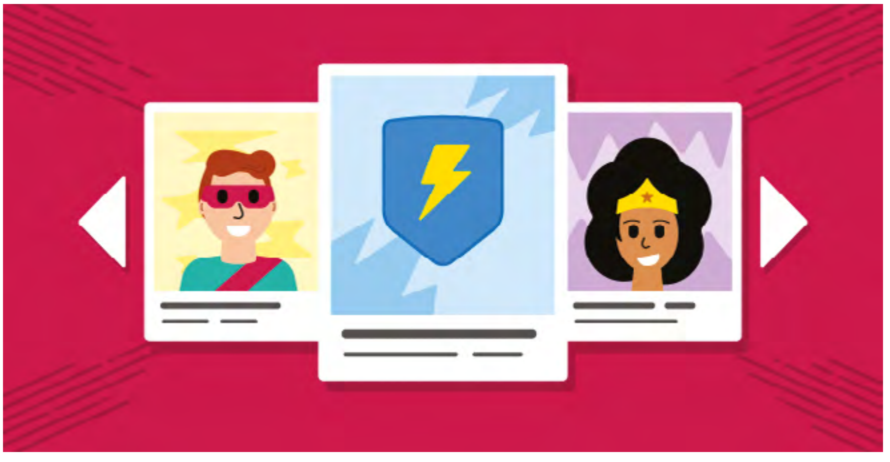


Create a fan website that gives users a choice of content

Portals to new worlds

The web allows us to access experiences that would not otherwise be possible. Want to chat to your mates about the big game, but can't get a ticket? Follow the live stream! Want to keep up with the latest fashion, but can't make it to Paris Fashion Week? Browse the runways from home!

Our free 'More web' project path moves beyond the basics of HTML and CSS seen in 'Introduction to web' to show you how to use JavaScript to take the user on a journey and transport them to somewhere that matters to you. New to JavaScript? No worries! This is your chance to explore a powerful language with plenty of support along the way. The path follows our Digital Making Framework to help you conquer those pesky bugs, proving that coding challenges are just stepping stones to success!



Create a superhero design website

Explore projects

WELCOME TO ANTARCTICA

Use HTML and CSS to create an awesome modern website that lets people discover a place they may never get a chance to visit. Create a navbar, set accessible colours and fonts, and create a responsive grid layout to hold beautiful images and awesome facts about this amazing continent.

COMIC CHARACTER

Build an interactive website where the user can design a superhero character. Use JavaScript to let the user change the text on your website, show and hide elements, and create a hero image slider. Let the user set the colour theme and keep their preferences, even if they reload the page.

ANIMATED STORY

Create an interactive story with animated text and characters that are triggered when the user scrolls. Design for accessibility and improve browser performance by only loading images when they're needed.

Design projects

PICK YOUR FAVOURITE

Practise your skills and bring in your own interests to make a fan website that lets a user make choices that change the web page content.

QUIZ TIME!

Build a personalised web app that lets the user show what they know about a topic of your choosing by answering animated questions and receiving a score.

Invent project

SHARE YOUR WORLD

Use your new coding superpowers to create an interactive website and share a part of your world on the web.

Built for the Code Editor

Projects come with everything you need: step-by-step guides, starter code in the Code Editor, and code snippets to help you build awesome websites. To keep young creators safe, the Editor doesn't allow image uploads, but we've included image banks for you to use. To dive deeper into website creation with the Editor, check out our Getting Started Guide (**rpf.io/editor-guide-html**). (HW)

JAVASCRIPT

JavaScript is a programming language that runs in your browser. As of 2024, it is the most popular programming language in the world!

SKILLS TO LEARN

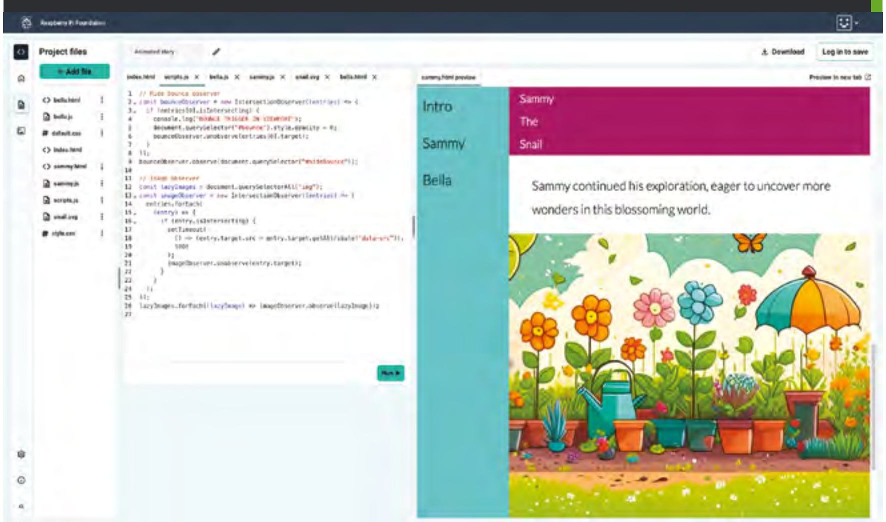
HTML and CSS:

- Navbars
- Grid layouts
- Hero images
- Image sliders
- Form design and handling user input
- Accessibility
- Responsive design
- Sizing elements relative to the viewport or container
- Creating parallax scrolling effects using background-attachment
- Fixing the position of elements
- Using z-index to layer elements

Young creators will be shown how to manipulate the content of a web page by using functions that take advantage of the Document Object Model (DOM) to update elements and the properties of their classes.

JavaScript:

- Local and global variables, and constants
- Selection ('if', 'else if', and 'else')
- Repetition ('for' loops)
- Writing and calling functions to take advantage of the Document Object Model (DOM)
- Using console log
- Concatenation using template literals
- Event listeners
- Use of the intersection observer API to animate elements when they come into view and lazy-load images
- Work with Date() functions
- .innerText and .innerHTML properties
- The localStorage object to retain user preferences
- Use setTimeout() to create time delays



Try out the 'Animated story' project in the Code Editor

THE MOONHACK CODING CHALLENGE

Digital solutions to combat climate change

Kaye North

Moonhack is a free international event, run by Code Club Australia (codeclubau.org), bringing kids aged 8–15 together from across the world for two weeks of coding fun. Over the past eight years, Moonhack has seen over 235,000 children coding Moonhack projects, with each new annual challenge presenting a unique theme about the world or space.

2023 was another record-breaking year for Moonhack, with 44,612 young people registering from 68 countries. Moonhack is set to return on 14 October 2024, to inspire our kids to explore how they can have a positive impact on climate change.

Global engagement to code solutions

This year, the Moonhack coding challenge provides an opportunity for kids in classrooms, code clubs, and at home, to engage in developing digital solutions to help mitigate climate change. This may be gamifying information, creating a call to action, or coding a new innovative solution. Moonhack is for all children, from first-time coders to coding whizz-kids. The projects are designed in such a way that they can either be used to learn new coding skills, or innovated and developed by an individual's own ideas, creativity, coding knowledge, and unique views on the topic.

This year's coding challenge has been developed to put young people at the heart of creating positive change for our environment. Each project has a direct link to UN Sustainable Development Goal 13, Climate Action (**helloworld.cc/climate-action**). This goal calls for us to take urgent action to combat climate change. By participating in Moonhack, we are working towards target 13.3: build knowledge and capacity to meet climate change.

Featured projects

With six projects to choose from (**moonhack.com/projects-new**), there is something for everyone. This year features three Scratch projects of varying levels, a Python project, a project that uses micro:bits, and a design brief (create your own solution to a problem using any coding platform, with or without robotics).

This October's Moonhack challenge features projects about rising sea levels, turtle hatchlings, sugar cane farming, and more

- 'In Deep Water' is a Scratch project for beginners. It addresses the issue of rising sea levels and presents a call to action to switch off appliances and recycle more. These actions will reduce carbon emissions and slow down global warming. The project helps young people create a maze game, and provides challenges to increase the difficulty across more levels.

- 'Sweeten the Crop' is a project collaboration with Canegrowers (**canegrowers.com.au**). Sugar cane is a large agricultural industry in Queensland, Australia. The farmers have some amazing technology, and use data to inform their watering practices. This is showcased in this Scratch project, which creates a clicker game in which the player has to use their knowledge of ground moisture and rain to water the crop correctly and bring it to harvest.

- 'Water Adventure' is a Scratch project aimed at more experienced coders. It creates an animation of the water cycle by using advanced techniques with variables, cloning, and colour blocks. The project encourages young people to further enhance the animation by showing the effects of polluted waterways, demonstrating how the water cycle changes depending on the environment, or suggesting a call to action to keep our waterways clean and healthy.

- 'Turtles and Temperature' is a Python project that uses data from a research

Our Moonhack testing clubs are a vital part of ensuring each project is accurate, successful, and fun

project in Western Australia conducted by Blair Bentley. His research explored the effects of climate change on Western Australian sea turtle populations. Through this project, young people will learn about how the temperature of the sand that the turtle eggs are laid in impacts the sex of the hatchlings. By importing this data into Python, young people will create a program to plot the data on a graph that is then used as the basis of a multiple-choice game.

- 'Talking Rubbish' creates a program with micro:bits that can be used to track what rubbish is placed in a bin. Every time an item is placed in the bin, a button is pressed to record the type of rubbish. After a certain amount of time, the data is exported to a spreadsheet that young people can use to inform a campaign that encourages people to recycle and make better choices about what they place in the bin.

- 'The Great Pacific Garbage Patch' project is a bit different from the others. This project encourages young people to learn about this garbage patch in our ocean, which is larger than Western Australia, and to code their own innovative digital solution. This could mean creating a game in Scratch encouraging people not to litter; an interactive website made with Python that uses data to predict what will happen over time if we clean the patch up or add to it; or creating a prototype rubbish collection device using LEGO WeDo, SPIKE Prime, or any other robotics kit.

Coding in the classroom for Moonhack

Each of the project pages has additional resources to support educators who use the Moonhack projects to teach digital technologies in their classroom. Each project links to a blog that outlines the importance of the project and why it is being featured in the challenge. The blogs also provide curriculum links and feature our Moonhack testing clubs, which have played a vital part in finalising the projects. You can also access resources such as participation certificates, posters, and links to the codealong videos.

As a global community, we want to share our coded solutions and ideas, so Scratch projects with permission to be shared are hosted in our Moonhack Studio (**helloworld.cc/Moonhack2024-studio**), and we also showcase projects on our website (**moonhack.com/showcase**). This year we will also have Moonhack codealong videos, Moonhack meetups, and ways to collaborate online.

To ensure you are kept up to date, subscribe to our Moonhack newsletter (**helloworld.cc/moonhack-newsletter**) and bookmark the Moonhack website (**moonhack.com**). (HW)

IMPORTANT DATES

Make sure to mark these dates on your calendar!

- Registration is now open!
- Moonhack begins 14 October and runs for two weeks
- Project submissions continue until 30 November
- Winners will be announced 8 December

www.moonhack.com

COMPUTING TEACHER RESEARCH PROJECTS 2024

Teacher Inquiry in Computing Education (TICE) and the Teacher Research Network

Sue Sentance and Simon Humphreys

As experienced professionals, teachers develop a deep understanding of their students and the particular context of their school. They have insights into which projects and approaches are effective, what motivates students, and how to facilitate learning. However, teachers rarely have the time or opportunity to investigate their instincts further or conduct their own research to explore why certain interventions succeed and what their students truly think.

To address this, the Teacher Inquiry in Computing Education (TICE) programme is aimed at any teacher, primary or secondary, who would like to investigate something in their own practice.

> TEACHERS RARELY HAVE THE TIME OR OPPORTUNITY TO CONDUCT THEIR OWN RESEARCH ABOUT THEIR PRACTICE

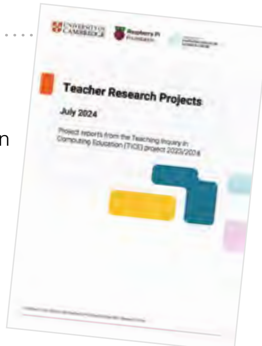
The TICE project was originally conceived in 2015-2016 and run as a Computing At School (CAS) project. It is now being delivered by the Raspberry Pi Computing Education Research Centre in partnership with CAS.

In the TICE project, computing teachers select topics that emerge naturally from issues they are either concerned or curious about in their classrooms. We use an academic partnership model, whereby a group of academic 'helpers' who have experience of conducting research are matched to teachers in order to give them some personalised help with their inquiry projects. The actual investigation or intervention planned takes place in the spring term, and there are webinars and other resources to help.

This year, 22 teachers took part in TICE, of which 11 have chosen to write up the results of their projects in a booklet (**helloworld.cc/TICE-2024**). You can read about the projects undertaken and you may be inspired to carry out something similar yourself. Some of the TICE teachers presented their projects at the CAS conference in July, which we hope to repeat next year.

Interested in taking part in TICE?

From this year, there are two iterations of the TICE project. One is the supported option, which includes a face-to-face day at the end of the (calendar) year, and allocation to a 'helper'. The other is the self-study option, whereby you have access to all our resources and webinars. While applications to the supported option starting this year are already closed we will keep the option to sign up to the self-study option open until the beginning of December. Find out more on our TICE pages (**helloworld.cc/TICE**).



Teacher Research Network

In the meantime, we'd like to share with you the news that we are starting a Teacher Research Network. If you have already signed up for our mailing list, you will have noticed that there is a tick box to join it, which has been there for a while — and now we are all set to go! The Teacher Research Network is for you if you'd like to volunteer to be a participant in any of our research projects. While in TICE you research your own topic, being part of this network means you hear first about research projects we are running and have the opportunity to take part. Participating in projects involves varying levels of commitment, from a one-off interview with a researcher to being one of our Exploring Physical Computing in School (EPICS) partners schools for five years. If you've signed up for our newsletter and ticked that box, you'll be hearing from us shortly. If you haven't already, think about signing up here: **computingeducationresearch.org/stayintouch**. (HW)

#INSIGHTS

GROUNDED COGNITION: PHYSICAL ACTIVITIES AND LEARNING COMPUTING

STORY BY Bonnie Sheppard

Everyone who has taught children before will know the excited gleam in their eyes when the lessons include something they can interact with physically. Whether it's printed and painstakingly laminated flashcards or laser-cut models, or interactive robots, learners' motivation to engage with the topic will increase along with the noise levels in the classroom.

However, these hands-on activities are often seen as merely a technique to raise interest, or a nice extra project for children to do before the actual learning can begin. But what if this is the wrong way to think about this type of activity?

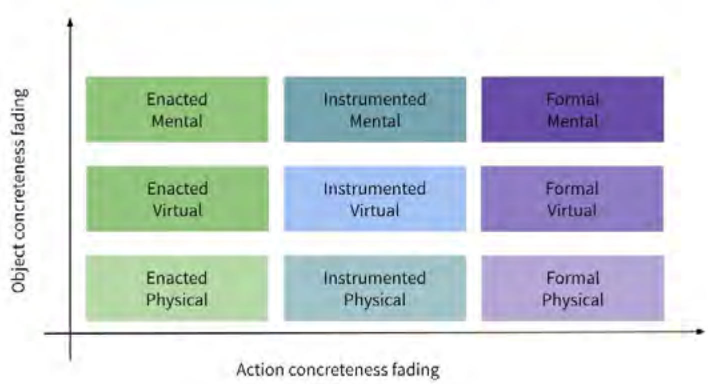
How do children learn?

Anaclara Gerosa from the University of Glasgow is tackling one of the most fundamental questions in education: how do children actually learn? Beyond the conventional methods, emerging research has been shedding light on a fascinating approach — the concept of grounded cognition. This theory suggests that children don't merely passively absorb knowledge; they physically interact with it, quite literally 'grasping' concepts in the process.

Grounded cognition, also known in variations as embodied and situated cognition, offers a new perspective on how we absorb and process information. At its core, this theory suggests that all cognitive processes, including language and thought, are rooted in the body's dynamic interactions with the environment. This notion challenges the conventional view of learning as a purely cognitive activity and highlights the impact of action and simulation.

There is evidence from many studies in psychology and pedagogy that using hands-on activities can enhance comprehension and abstraction. For instance, finger counting has been found to be essential in understanding numerical systems and mathematical concepts. A recent study in this field has shown that children who are taught basic computing concepts with unplugged methods can grasp abstract ideas from as young as three. There is therefore an urgent need to understand exactly how we could use grounded cognition methods to teach children computing — which is arguably one of the most abstract subjects in formal education.

EIFFEL model for early computing



The EIFFEL model suggests a progression from enacted to formal activities

A new framework for teaching computing

Anaclara is part of a group of researchers at the University of Glasgow who are currently developing a new approach to structuring computing education. Their EIFFEL (Enacted Instrumented Formal Framework for Early Learning in computing) model suggests a progression from enacted to formal activities (**helloworld.cc/EIFFEL-model**).

Following this model, in the early years of computing education, learners would primarily engage with activities that allow them to work with tangible 3D objects or manipulate intangible objects, for instance in Scratch. Increasingly, students will be able to perform actions in an instrumented or virtual environment, which will require the knowledge of abstract symbols but will not yet require knowledge of programming languages. Eventually, students will have developed the knowledge and skills to engage in fully formal environments, such as writing advanced code.

In a recent literature review, Anaclara and her colleagues looked at existing research into using grounded cognition theory in computing education. Although several studies report the use of grounded approaches, for instance by using block-based programming, robots, toys, or construction kits, the focus is generally on looking at how concrete objects can be used in unplugged activities due to specific contexts, such as a limited availability of computing devices.

> GROUNDED COGNITION OFFERS A NEW PERSPECTIVE ON HOW WE ABSORB AND PROCESS INFORMATION

The next steps in this area are looking at how activities that specifically follow the EIFFEL framework can enhance children's learning in all contexts.

Try grounded activities in your classroom

Here are a few ideas on how to get started with grounded cognition activities to teach children computing:

- Try out one of our hands-on lessons and learning activities for young learners from The Computing Curriculum (**helloworld.cc/tcc**). For example, in this 'Programming' unit for Year 1 learners (aged 5–6), students learn how to program with the help of re-enactments and classroom robots (**helloworld.cc/ks1-robot-move**).
- Explore the Teach Data Literacy guide (**helloworld.cc/teachdataliteracy**), developed by the Data Education in Schools team, which offers some practical activities to support young learners in developing their data literacy skills. You can find out more about the Data Education in Schools initiative in Kate Farrell and Judy Robertson's seminar on teaching primary learners how to be data citizens from May 2023 (**helloworld.cc/data-citizen-seminar**).
- Check out Barefoot Computing (**helloworld.cc/barefoot-early**), which offers a range of resources for early-years education that involve physical manipulation and simulation.

Research into grounded cognition activities in computer science is ongoing, but we encourage you to try incorporating more hands-on activities when teaching younger learners and observing the effects yourself. You can watch Anaclara's seminar here: **helloworld.cc/grounded-cognition-seminar**. (HW)

IMAGINING STUDENTS' PROGRESSION IN THE ERA OF GENERATIVE AI

STORY BY Sarah Millar

Generative AI tools are becoming more easily accessible to learners and educators, and are getting better and better at generating code solutions to programming tasks, code explanations, computing lesson plans, and other learning resources. This raises many questions for educators in terms of what and how we teach students about computing and AI, and AI's impact on assessment, plagiarism, and learning objectives.

We were honoured to have Professor Brett Becker, from University College Dublin, join us as part of our 'Teaching programming (with or without AI)' seminar series (**helloworld.cc/ai-seminar-2024**). He is uniquely placed to comment on teaching computing using AI tools, having been involved in many initiatives relevant to computing education at different levels, in Ireland and beyond.

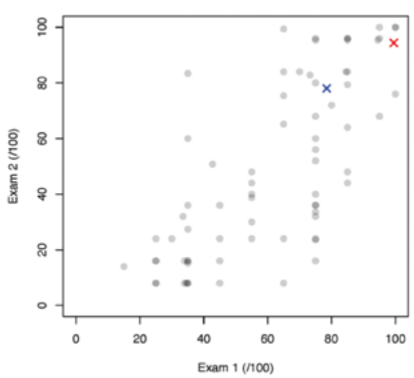


Figure 1 Student scores on Exam 1 and Exam 2, represented by circles. GPT-3's 2021 score is represented by the blue x, and GPT-4's 2023 score on the same questions is represented by the red x

Brett's talk focused on what educators and education systems need to do to prepare all students — not just those studying computing — so that they are equipped with sufficient knowledge about AI to make their way from primary school to secondary and beyond, whether it be university, technical qualifications, or work.

How do AI tools currently perform?

Brett began his talk by illustrating the uplift in performance of large language models (LLMs) in solving first-year undergraduate programming exercises. He compared the findings from two recent studies he was involved in as part of an ITiCSE working group (**iticse.acm.org**). In the first study, from 2021 (**helloworld.cc/Becker-2021**), the results generated by GPT-3 were similar to those of students in the top quartile. By the second study, in 2023 (**helloworld.cc/Becker-2023**), GPT-4's performance matched that of a top student (**Figure 1**).

Brett also explained that the study found some models were capable of solving current undergraduate programming assessments almost error-free, and could solve the Irish Leaving Certificate and UK A-level computer science exams.

What are challenges and opportunities for education?

This level of performance raises many questions for computing educators about what is taught and how to assess students' learning. To address this, Brett referred to his 2023 paper, which included findings from a literature review and a survey on students' and instructors' attitudes towards using LLMs in computing education. This analysis has helped him identify several opportunities, as well as the ethical challenges education systems face regarding generative AI.

The opportunities include:

- The generation of unique content, lesson plans, programming tasks, and feedback to help educators with workload and productivity
- More accessible content and tools generated by AI apps to make computing more broadly accessible to more students
- More engaging and meaningful student learning experiences, including using generative AI to enable creativity and using conversational agents to augment students' learning
- The impact on assessment practices, both in terms of automating the marking of current assessments and reconsidering what is assessed and how

Some of the challenges include:

- The lack of reliability and accuracy in outputs from generative AI tools
- The need to educate everyone about AI to create a baseline level of understanding
- The legal and ethical implications of using AI in computing education and beyond

- How to deal with questionable or even intentionally harmful uses of AI, and how to mitigate the consequences of such uses

Programming as a basic skill for all subjects

Next, Brett talked about concrete actions that he thinks we need to take in response to these opportunities and challenges.

He emphasised our responsibility to keep students safe. One way to do this is to empower all students with a baseline level of knowledge about AI, at an age-appropriate level, to enable them to keep themselves safe.

He also discussed the increased relevance of programming to all subjects, not only computing, in a similar way to how reading and mathematics transcend the boundaries of their subjects, and the need he sees to adapt subjects and curricula to that effect.

As an example of how rapidly curricula may need to change with increasing AI use by students, Brett looked at the Irish computer science specification for senior cycle (the final two years of second level, ages 16–18). This curriculum was developed in 2018 and in Brett's opinion remains a strong computing curriculum. However, he pointed out that it only contains a single learning outcome on AI.

> IT MAY BE POSSIBLE TO BETTER EQUIP A LARGER AND MORE DIVERSE POOL OF STUDENTS TO ENGAGE WITH COMPUTING

To help educators bridge this gap, in the book Brett wrote alongside Keith Quille to accompany the curriculum (**helloworld.cc/cs-leaving-cert**), they included two chapters dedicated to AI, machine learning, and ethics and computing. Brett believes these types of additional resource may be instrumental for teaching and learning about AI, as resources are more adaptable and easier to update than curricula.

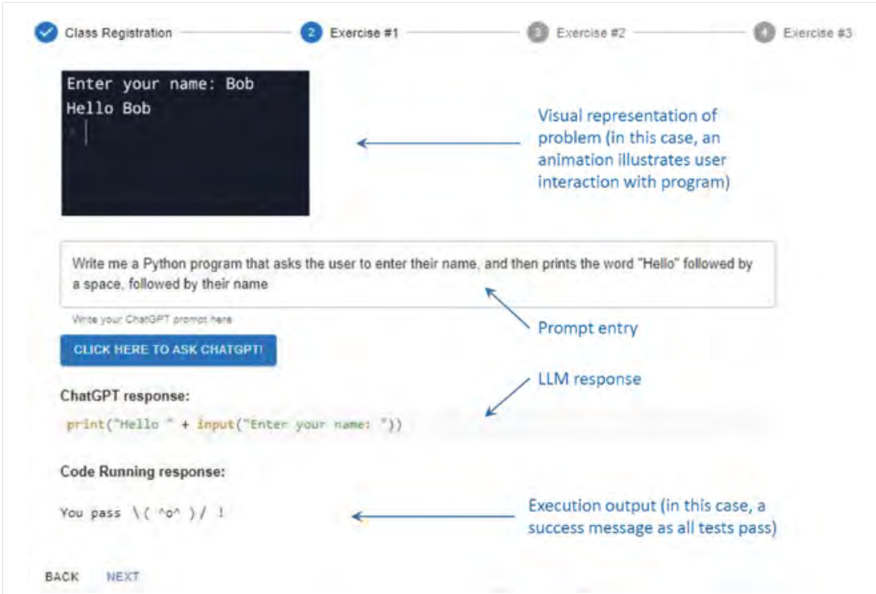


Figure 2 An example of a student's use of Promptly

Generative AI in computing education

Taking the opportunity to use generative AI to reimagine new types of programming problems, Brett and colleagues have developed Promptly, a tool that allows students to practise prompting AI code generators. This tool provides a combined approach to learning about generative AI while learning programming with an AI tool.

Promptly is intended to help students learn how to write effective prompts. It encourages students to specify and decompose the programming problem they want to solve, read the code generated, compare it with test cases to discern why it is failing (if it is), and then update their prompt accordingly (**Figure 2**).

Early undergraduate student feedback points to Promptly being a useful way to teach programming concepts and encourage metacognitive programming skills. The tool is further described in a paper (**helloworld.cc/prompt-problems-ai**), and while the initial evaluation was aimed at undergraduate students, Brett positioned it as a secondary–level tool as well.

Brett hopes that by using generative AI tools like this, it will be possible to better equip a larger and more diverse pool of students to engage with computing.

Re-examining the concept of programming

Brett concluded his seminar by broadening the relevance of programming to all learners, while challenging us to expand our perspectives on what programming is. If we define programming as a way of prompting a machine to get an output, LLMs allow all of us to do so without the need for learning the syntax of traditional programming languages. Taking that view, Brett left us with a question to consider: "How do we prepare for this from an educational perspective?"

Join our next seminar

The focus of our ongoing 2024 seminar series is on teaching programming, with or without AI. To take part in a seminar, sign up and we will send you information about how to join: **helloworld.cc/rpf-seminars-signup**. We hope to see you there. (HW)

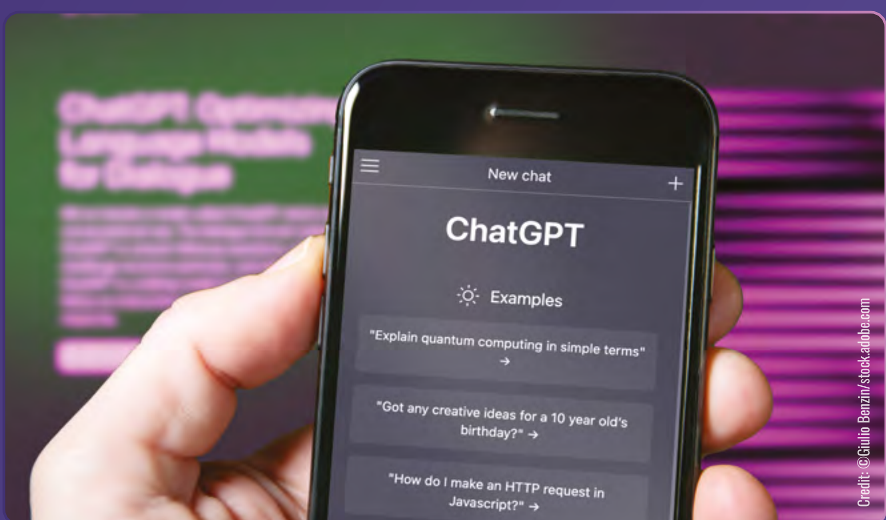
A PROBLEM-FIRST APPROACH TO THE DEVELOPMENT OF AI SYSTEMS

Ben Garside explains how we can help learners understand GenAI by focusing on solving problems

If you are into tech, keeping up with the latest updates can be tough, particularly when it comes to artificial intelligence (AI) and generative AI (GenAI). Sometimes I admit to feeling this way myself, but there was one update recently that really caught my attention. OpenAI launched their latest iteration of ChatGPT, this time adding a female-sounding voice. Their launch video (helloworld.cc/GPT-4o) demonstrated the model supporting the presenters with a maths problem and giving advice around presentation techniques, sounding friendly and jovial along the way.

The addition of voices to these AI models was perhaps inevitable as big tech companies try to compete for market share in this space, but it got me thinking, why would they add a voice? Why does the model have to flirt with the presenter?

Working in the field of AI, I've always seen AI as a really powerful problem-solving tool. But with GenAI, I often wonder what problems the creators are trying to solve, and how we can help young people understand the tech.



Credit: © Giulio Benzini/stock.adobe.com

When it comes to AI, keeping up with the latest updates can be tough

What problem are we trying to solve with GenAI?

The fact is that I'm really not sure. That's not to suggest that GenAI doesn't have its benefits — it does. I've seen so many great examples in education alone: teachers using large language models (LLMs) to generate ideas for lessons, to help differentiate work for students with additional needs, or to create example answers to exam questions for their students to assess against the mark scheme. Educators are creative people, and while it is cool to see so many good uses of these tools, I wonder if the developers had in mind, while creating them, a specific problem that they wanted to solve, or did they simply hope that society would find a good use for them somewhere down the line?

While there are good uses of GenAI, you don't need to dig very deeply before you start unearthing some major problems.

Anthropomorphism

Anthropomorphism relates to assigning human characteristics to things that aren't

human. This is something we all do, all the time, without any consequences. The problem with doing this with GenAI is that, unlike an inanimate object you've named (I call my vacuum cleaner Henry, for example), chatbots are designed to be human-like in their responses, so it's easy for people to forget they're not in fact speaking to a human.

As feared, since my article on the topic in Hello World issue 22 (**helloworld.cc/22**, pages 26–27), evidence has started to emerge that some young people are showing a desire to befriend these chatbots, going to them for advice and emotional support. It's easy to see why. Here is an extract from an exchange between the presenters and the chatbot at the ChatGPT-4o launch:

> ChatGPT (presented with a live image of the presenter): "It looks like you're feeling pretty happy and cheerful with a big smile and even maybe a touch of excitement. Whatever is going on? It seems like you're in a great mood. Care to share the source of those good vibes?"
>
> Presenter: "The reason I'm in a good mood is we are doing a presentation showcasing how useful and amazing you are."
>
> ChatGPT: "Oh stop it, you're making me blush."

The Family Online Safety Institute (FOSI) conducted a study looking at the hopes and fears that parents and teenagers have around GenAI (**helloworld.cc/FOSI-genai**). One teenager said:

"Some people just want to talk to somebody. Just because it's not a real person, doesn't mean it can't make a person feel — because words are powerful. At the end of the day, it can always help in an emotional and mental way."

The prospect of teenagers seeking solace and emotional support from a generative AI tool is a concerning development. While these AI tools can mimic human-like conversations, their outputs are based on patterns and data, not genuine empathy or understanding. The ultimate concern is that this exposes vulnerable young people to be manipulated in ways we can't predict. Relying on AI for emotional support could lead to a sense of isolation and detachment, hindering the development of healthy coping mechanisms and interpersonal relationships.

What problem are we trying to solve with generative AI?

> THINK ABOUT THE PROBLEM YOU WANT TO SOLVE FIRST BEFORE THINKING ABOUT WHETHER AI IS THE RIGHT TOOL TO SOLVE IT

Arguably worse is the recent news of the world's first AI beauty pageant (**helloworld.cc/miss-ai**). The very thought of this probably elicits some kind of emotional response that will vary depending on your view of beauty pageants. There are valid concerns around misogyny and the reinforcement of misguided views on body norms, but it's also important to note that the winner of 'Miss AI' is being described as a lifestyle influencer. The questions we should be asking are, who are the creators trying to have influence over? What influence are they trying to gain that they couldn't get before they created a virtual woman?

Deepfake tools

Another use of GenAI is the ability to create deepfakes. If you've watched the most recent Indiana Jones movie, you'll have seen the technology in play, making Harrison Ford appear as a younger version of himself. This is not in itself a bad use of GenAI technology, but the application of deepfake technology can easily become problematic. For example, a teacher was recently arrested for creating a deepfake audio clip of the school principal making racist remarks (**helloworld.cc/principal-deepfake**). The recording went viral before anyone realised that AI had been used to generate the clip.

Easy-to-use deepfake tools are freely available and, as with many tools, they can be used inappropriately to cause damage or even break the law. One such instance

We want to help young people be critical consumers of AI technology

is the rise in the use of the technology for pornography (**helloworld.cc/deepfake-porn**). This is particularly dangerous for young women, who are the more likely victims, and can cause severe and long-lasting emotional distress and harm to the individuals depicted, as well as reinforcing harmful stereotypes and the objectification of women.

Why we should focus on using AI as a problem-solving tool

Unforeseen negative consequences of technological developments are nothing new. A big part of our job as educators is to help young people navigate the changing world and prepare them for their futures, and education has an essential role to play in helping people understand AI technologies so that they can avoid the dangers.

Our approach at the Raspberry Pi Foundation is not to focus purely on the threats and dangers, but to teach young people to be critical users of technologies and not passive consumers. Having an understanding of how these technologies work goes a long way towards achieving sufficient AI literacy skills to make informed choices, and this is where our Experience AI (**experience-ai.org**) programme comes in.

Experience AI is a set of lessons developed in collaboration with Google DeepMind and, before we wrote any lessons, our team thought long and hard about what we believe are the important principles that should underpin teaching and learning about artificial intelligence. One such principle is taking a problem-first approach and emphasising that computers are tools that help us solve problems. In the Experience AI fundamentals unit, we teach students to think about the problem they want to solve before thinking about whether or not AI is the appropriate tool to use to solve it.

Taking a problem-first approach doesn't altogether avoid the possibility that an AI system will cause harm — there's still the chance it will increase bias and societal inequities — but it does focus the development on the end use and the data needed to train the models. I worry that focusing on market share and opportunity, rather than on the problem to be solved, is more likely to lead to harm.

Another set of principles that underpins our resources is teaching about fairness, accountability, transparency, privacy, and security in relation to the development of AI systems (see 'Fairness, Accountability, Transparency, and Ethics (FATE) in Artificial Intelligence (AI) and higher education', **helloworld.cc/FATE**, and 'Understanding Artificial Intelligence Ethics and Safety', **helloworld.cc/ai-ethics**). These principles are aimed at making sure that creators of AI models develop models ethically and responsibly. The principles also apply to consumers, as we need to get to a place in society where we expect these principles to be adhered to, and consumer power means that any models that don't adhere to these principles simply won't succeed.

Furthermore, once students have created their models in the Experience AI fundamentals unit, we teach them about model cards (**helloworld.cc/22**, pages 42–43), an approach that promotes transparency about their models. In much the same way as nutritional information on food labels allows the consumer to make an informed choice about whether or not to buy the food, model cards give information about an AI model such as the purpose of the model, its accuracy, and known limitations such as what bias might be in the data. Students write their own model cards based on the AI solutions they have created.

What else can we do?

At the Raspberry Pi Foundation, we have set up an AI literacy team with the aim to embed principles around AI safety, security, and responsibility into our resources and align them with the Foundation's mission to help young people to:

- Be critical consumers of AI technology
- Understand the limitations of AI
- Expect fairness, accountability, transparency, privacy, and security and work toward reducing inequities caused by technology
- See AI as a problem-solving tool that can augment human capabilities, but not replace or narrow their futures

Our call to action to educators, carers, and parents is to have conversations with your young people about GenAI. Get to know their opinions on GenAI and how they view its role in their lives, and help them to become critical thinkers when interacting with technology. (HW)

BEN GARSIDE

Ben is senior learning manager for AI literacy at the Raspberry Pi Foundation. He has worked on The Computing Curriculum and has written online courses for educators, including 'Intro to Machine Learning and AI'.

USING GENERATIVE AI IN THE CLASSROOM

Sue Sentance introduces a GenAI guide for computing teachers

The release of ChatGPT by OpenAI in November 2022 brought generative AI (GenAI) to the attention of educators and society. GenAI presents numerous opportunities for enhancing teaching and learning in schools. GenAI tools can assist teachers with administrative tasks, so that they have more time to attend to students' learning needs. Teachers can also use GenAI to help develop educational materials. Moreover, GenAI tools are showing some promise in their use to support adaptive testing and feedback. For example, teachers might be able to use GenAI tools to adjust the content of assessments to match students' progress and performance.

More questions than answers

Computing teachers at primary, secondary, and post-16 levels may have started to consider the issues surrounding AI, and to look at how to teach about AI and its social and ethical implications. The introduction of GenAI raises questions about how to make use of the capabilities of such tools in the teaching and learning of computing, including programming. Further questions arise about how to use GenAI products to assist in a range of administration functions associated with teaching, including report writing, lesson planning, developing teaching material, and timetabling. Ethical questions are also brought to the fore over issues of biased and inaccurate output, and energy usage. There are also concerns over safety, data privacy, copyright, plagiarism, and the threat to assessment and students' critical thinking.

The unique needs of computing teachers

Conscious of the needs of computing teachers, including all those who may not necessarily consider themselves to be specialists, a small working group of interested teachers and researchers have collaborated to develop guidance and examples about generative AI and its use in the teaching and learning of computing in schools. We are particularly focused on the UK, but there's no reason why this document might not be useful to you wherever you are in the world.

This guidance document has been written 'by teachers, for teachers' and in this way differs from other seemingly similar documents which teachers might be made aware of. It includes examples of what you might use generative AI for in the classroom, specifically for computing, and some background definitions and tips that teachers might find useful. We also signpost useful resources for further reading and research. We're aware that this is a fast-moving field, and that documents like this will need revising sooner or later — probably sooner. That said, we hope you enjoy reading it, and we look forward to hearing your comments.

This project was a joint effort by many. Many thanks to the teachers who made this project possible: Philip Arthur, Sonya Coakley-Hanan, Martyn Colliver, Lyndsay Hope, Emma Staves, Chris West, Aidan Weston, and Eliot Williams; also thanks to my fellow researchers at the University of Cambridge. Download the guidance here: **helloworld.cc/ai-guidance-2024**. (HW)

> THE GUIDE PROVIDES EXAMPLES OF WHAT YOU MIGHT USE GENAI FOR IN THE CLASSROOM, SPECIFICALLY FOR COMPUTING

SUE SENTANCE

Sue is the director of the Raspberry Pi Computing Education Research Centre at the University of Cambridge. Formerly a computing teacher, she leads a number of research projects relating to computer science and AI education, including programming pedagogy and physical computing.

WHY LEADERSHIP MATTERS IN THE AGE OF AI

Introducing a framework for AI implementation aimed at administrators and leaders in K–12 schools

As the principal of a comprehensive high school, I am often asked to make decisions for my school, including the purchasing of technology tools for student and staff use. Because I have a background in learning technologies, I have been vocal about what I want and do not want to bring into my school's ecosystem. With the proliferation of AI, in particular generative AI (GenAI) and large language models (LLMs), I have become increasingly concerned about the predatory sales tactics of some education technology companies. Furthermore, I am equally concerned about the protectionist attitudes of some school systems.

Before becoming a high-school principal, I was a technology coach for a large district. As a tech coach, I learnt that effective technology implementation requires informed and innovative leadership in schools. In fact, it was experiencing the frustration of not being able to convince school leaders to adopt the appropriate technology tools needed to enhance student learning that led me to pursue a career in school leadership.

I have been extremely fortunate to have been a part of the Equity Fellowship of the Computer Science Teachers Association (CSTA) this past year, where I met many brilliant educators who were concerned about how to bring these powerful tools into our schools. Their main focus was to enhance computer science education in the age of AI.

While participating in the programme, I was able to work with Sofía De Jesús, an associate programme manager at Carnegie Mellon University Computer Science Academy. We shared both our enthusiasm and our concerns around how school leaders need to be supported to make the correct decisions when selecting developmentally appropriate AI tools.

As a result of our collaboration, we created the Framework for AI Implementation for Administrators and Leaders in K–12 Schools ('the Framework') to help other school leaders (**helloworld.cc/ai-implementation**).

Five big ideas to consider when implementing AI tools in schools

How we began

We started by looking at various reputable documents to ground our thinking in the best practice and learning theories. Because we understand the time constraints that our fellow school leaders face on a daily basis, we summarised various documents so that they could get the basic information that was in each document. We concluded that all these documents promote human dignity and safety as we grapple with the effective implementation of GenAI and LLMs.

The five big ideas

Based on our review of numerous documents, both listed and not listed in the Framework, we noticed that there were five big ideas that we need to consider when implementing these tools in schools. They are:

- Privacy
- Bias, discrimation, and access
- Student learning
- Guidelines and laws
- Environmental impact

Ongoing and continuous evaluation

Because any recommendation must be able

to adapt to the ever-changing landscape of AI, the process of evaluation must be ongoing and continuous. However, we believe that there are four focus areas of consideration that will remain relevant no matter how much the tools change.

Any tool that a school leader brings into the school ecosystem needs to have a specific use case, so that it is solving a specific problem or answering a specific question that an educator or an institution has; a tool should not be purchased simply because a leader feels that they must as a result of a sales pitch. A tool must have target users and their developmental stages in mind. We believe that all users must be informed of potential biases, be safe from discrimination, and have equitable access. Parental consent is also very important.

Finally, we included a table of questions and a worksheet that a school leader can ask when considering an AI-enabled tool.

Four areas to consider when evaluating individual AI tools

> ANY TOOL A LEADER BRINGS INTO THE SCHOOL ECOSYSTEM MUST BE SOLVING A SPECIFIC PROBLEM OR ANSWERING A SPECIFIC QUESTION

Developmentally appropriate activities for practitioners

We also created a table of developmentally appropriate activities that practitioners, specifically classroom teachers, could carry out with their students. We felt strongly that teachers should be supported in this manner, because we recognise how rapidly AI tools are now evolving. We wanted to support the leaders who will be tasked to support their teachers, as well as the classroom teachers who need ready-made activities that will withstand the test of time.

Terms of use and privacy policy

As well as providing practical suggestions, we included a separate section with additional practical information regarding terms of use and service, and privacy policies. School leaders and educators must question technology companies that attempt to minimise the importance of such agreements and expect schools to relinquish student and staff information. We encourage school leaders to voice their concerns and demand that the companies provide additional protection for institutions, as we continue to bring more AI-enabled tools into the school's ecosystem.

Conclusion

As a school leader, I feel a sense of duty to protect our staff, students, and families. In fact, keeping our students safe is the most important part of my job. With more and more AI tools now becoming available for schools, I hope that my fellow school leaders use our Framework to keep their communities safe.

A recent article by EdSurge (**helloworld.cc/EdSurge**) looked at the Los Angeles Unified School District's AI chatbot use and the potential impact of the misuse of student data. If this is any indication, school leaders need to be very vigilant about how we choose technology tools for our schools, especially AI-enabled tools. Although the Framework will need continuous updating, we hope that this will provide a starting point for K–12 school leaders who are looking for a way to ground their decision-making processes on AI-enabled tools. (HW)

KIP GLAZER EdD

Kip is principal of Mountain View High School in California. With over 20 years of experience both in the classroom and in school administration, she has worked on various National Science Foundation-funded projects, focusing on translating research into practice. She is particularly passionate about supporting the next generation of teachers and fellow administrators as they work to incorporate learning technologies into their practices. Kip can be reached on LinkedIn and at **www.kipglazer.com**.

WHY AI READINESS IS IMPERATIVE IN K–12

How Gwinnett County Public Schools are integrating AI into their entire K–12 curriculum

As artificial intelligence continues to evolve, it is increasingly embedded in various aspects of our lives, from the technology we use daily to the way we interact with the world. For K–12 educators, AI readiness is imperative. AI readiness encompasses the knowledge and critical skills needed to use AI safely, effectively, and appropriately. While the specific tools associated with AI will inevitably change, the foundational knowledge and ethical considerations that underpin AI literacy will remain critical to preparing students for the future.

Why AI readiness matters for students

AI readiness equips students with the ability to navigate a world in which AI influences many facets of society. Understanding how AI operates, where its data originates, and the potential biases and ethical concerns that come with its use are essential skills. These skills enable students to interact with AI responsibly, and to critically evaluate its use. As students progress through their education, their ability to engage with AI in an informed, ethical way will become increasingly important, regardless of the specific tools or technologies they encounter. Classrooms must create AI-ready students who have these skills.

LAURA VENTURA

Laura is an Artificial Intelligence and Computer Science Specialist serving Gwinnett County Public Schools. She is committed to helping teachers provide high-quality, inclusive CSed to all students.

Gwinnett's swim, snorkel, scuba approach to AI learning

Gwinnett's vision for AI readiness

Since 2018, Gwinnett County Public Schools (GCPS) has been at the forefront of developing AI readiness through their AI-ready pilot in the Seckinger Cluster of schools. The creation of a new high school, Seckinger, in Buford, Georgia, USA, provided a unique opportunity to pilot AI readiness from kindergarten through to 12th grade. This initiative ensures that all students receive a comprehensive education in AI, beginning at an early age and continuing through their high-school years. GCPS has provided extensive professional development to teachers and leaders, equipping them with the knowledge and skills needed to integrate AI literacy across all subjects.

The swim, snorkel, scuba approach to AI learning

AI readiness in Gwinnett follows a tiered approach, metaphorically described as 'swim, snorkel, and scuba dive'. All students will need to 'swim' — that is, develop a basic understanding of AI and its implications. This foundational readiness is essential for all, ensuring that every student is equipped to engage with AI responsibly.

Most students will use AI more deeply — the 'snorkel' level provides opportunities for users to delve further into the technology, becoming proficient users of AI. Students will have a deeper understanding of more complex concepts and applications. These students will engage in more advanced projects and discussions that go beyond the basics, exploring AI's role in various fields and its potential to shape the future.

Finally, students who are truly passionate about AI may choose to 'scuba dive'. These students dive into the technical skills of AI through a Career and Technical Education

(CTE) AI pathway, with three courses available. Students can specialise in AI development and gain hands-on experience with the latest technologies. Students who are interested in a career in an AI-related field will engage in these classes.

AI RESOURCES FOR TEACHER'S LEARNING

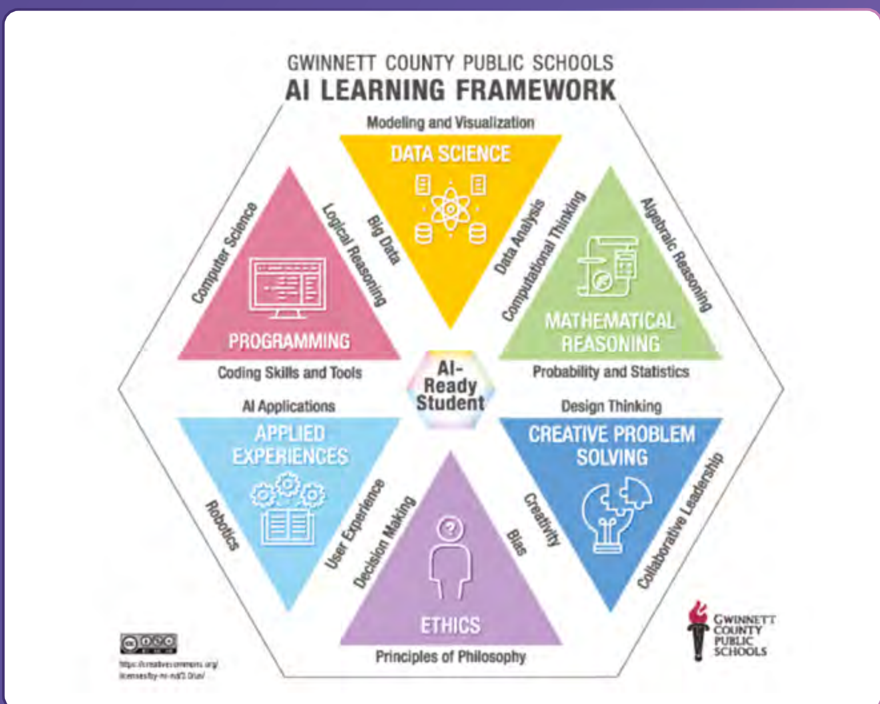
- AI Foundations for Educators, Common Sense Education: **commonsense.org/education/training/ai-foundations**
- International Society for Technology in Education (ISTE) AI: **iste.org/ai**
- Code.org AI resources: **code.org/ai**

Components of the AI Framework

The GCPS AI framework (**helloworld.cc/GCPS-ai-framework**) is built around key components, each integral to developing AI readiness. Teachers weave AI literacy into their content, to provide cross-curricular understanding. Here are some examples of K–12 lessons:

- **Data science:** in a freshman social studies class (14–15-year-olds), students analyse weather and traffic patterns from the 2014 Atlanta snowpocalypse (**helloworld.cc/atlanta-snowpocalypse**) to understand how decisions were made and compare that to how AI-powered data analytics might influence decisions today.
- **Mathematical reasoning:** early-elementary students (6–8-year-olds) learn to plot points on a graph by programming simple robots, applying mathematical concepts in a practical, engaging way.
- **Creative problem-solving:** high-school students (14–18-year-olds) in the AI pathway engage in design thinking to generate embedded computing solutions for a problem of interest using creativity and collaboration to solve a real-world problem.
- **Ethics:** students in the 2nd grade (7–8-year-olds) explore opposing points of view and the implications of how and why decisions are made.
- **Applied experiences:** in a high school art class, students (14–18-year-olds) are asked to draw a sketch, then enter a description of the sketch into an AI image generator, leading to debates around whether the computer-created works inspired them to change their original designs.
- **Programming:** 7th-grade science students (12–13-year-olds) took a concept that's long been part of their curriculum — genetics — and used coding to figure out the probability of inheriting certain genetic traits.

Teachers provide a meaningful opportunity to practice the AI readiness skills in the context of their class's content. Providing direct instruction and explicit teaching points and reflections helps students to clearly make the connection between their learning and AI literacy.



The key components of the GCPS approach to developing AI readiness

Supporting teachers' AI knowledge

To further support AI readiness, Gwinnett County Public Schools has developed guidance for human-centred AI use (**helloworld.cc/GCPS-ai-guidance**). This document provides practical expectations and decision-making principles for staff, students, and parents regarding the responsible use of AI.

AI is now playing a crucial role in diverse areas of our lives, from healthcare to financial services, and in our streaming service and social media recommendations. AI is already shaping how we interact with the world. It is crucial for teachers to prioritise AI readiness in their ongoing professional learning, with a focus on the foundational principles of AI, rather than specific technologies, in order to be tool-agnostic. Teachers should continue their learning through professional development opportunities, peer collaborations, and self-study. Teachers should actively explore AI resources and tools to gain a comprehensive understanding of the risks and benefits, recognising the critical importance of AI readiness in K–12 education. AI readiness will better equip students for a future in which AI plays an increasingly significant role in society. (HW)

BUILDING AI LITERACY WORKSHOPS WITH THE SEAME FRAMEWORK

Delivering meaningful impact through informal AI activities categorised using a framework

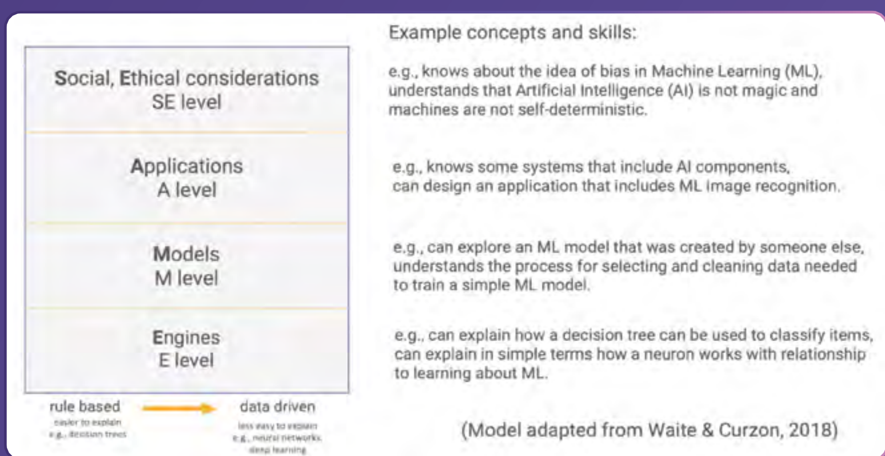
The rate of AI development is reaching dizzying new speeds, while technology companies are busily promoting its potential and encouraging society to adopt AI products. AI literacy means knowing the right questions to ask when evaluating AI applications, and can help us to counteract the hype. We can then begin to form our own opinions on how it should, and should not, be used. As AI becomes more and more pervasive, young people need the transferable and future-proof competencies provided by AI literacy to underpin their understanding and use of AI.

As an informal educator, my priority is to make the topic accessible and inspiring to a broad range of pupils, not just those who are already interested in computing. I have developed interactive in-person AI literacy workshops for ages 8–14 that do not require any knowledge of coding. During the workshops, students create and refine their own AI models running on Raspberry Pis. I am keen for them to delve into AI's benefits and limitations and apply critical thinking to every activity, not just consume ready-made demonstrations.

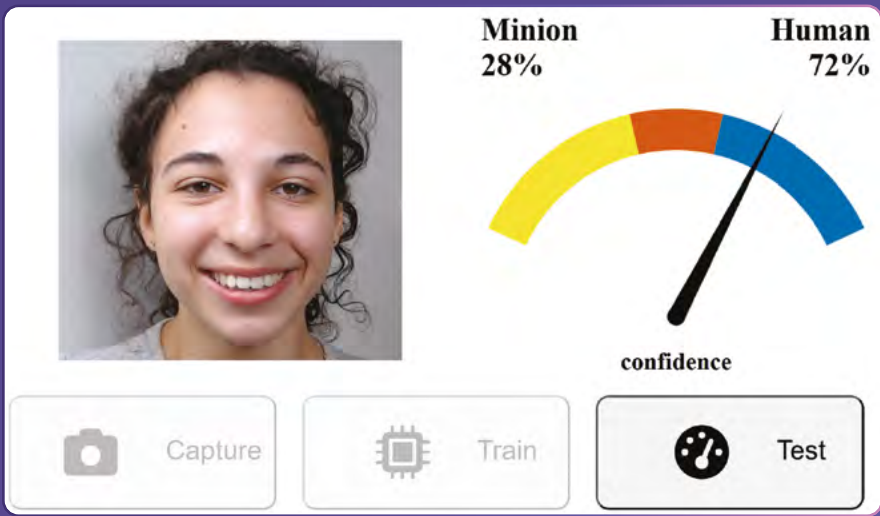
I first heard about the SEAME framework (**helloworld.cc/AI_seame**) through the Raspberry Pi Foundation's research seminars, and have found it incredibly useful for categorising my activities. It ensures that each workshop consistently covers the spectrum of concepts that underpin AI literacy, while I can also adapt the material for different age groups.

Creating the first model

I start each session at the model (M) layer and challenge teams of pupils to train a machine learning (ML) model to classify images as either 'human' or 'minion'. Pupils are intrigued, as this task is seemingly straightforward, but they do not yet know how to use AI to deliver it. Ten minutes later, they are amazed at how quickly they have captured examples, then trained and tested their models on themselves. Rather than ending the activity with this apparent quick success, though, pupils then test their classifiers with new images whose composition differs from the training data, making it progressively harder for the AI to classify them correctly. Using live bar charts, we compare the results coming from all the teams to understand how their choice of training images influences each model's performance. Pupils soon spot how the ML model latches onto irrelevant features in the images, such as background objects. Seeing these mistakes, and recognising that AI is not infallible, is a key learning moment. Through these experiments, pupils see how AI actually learns, but they also realise the importance of selecting appropriate training data, and see how easily we can be lulled into trusting AI's apparent intelligence. It is vital that



The SEAME framework



Students test their machine learning image classifiers and compare their performance

children see the huge role humans play in training AI and evaluating its outputs, and that the models are not self-deterministic.

Turning to the applications (A) layer, I showcase several examples, including self-driving cars. I've found that this application is of universal interest, and elicits different reactions that trigger debate: children recognise the concept, but the class is often split on whether or not they trust self-driving cars. With their learning experience from the human/minion classifier, they all see the potential risks of an autonomous vehicle encountering a situation for which it has not been adequately trained.

Exploring generative AI

Continuing with the applications layer, the workshop then shifts to generative AI (GenAI), which is receiving the most attention and hype. For maximum accessibility, I illustrate GenAI using a text-to-image application, rather than the textual output of a conversational large language model. The pupils are given a limited selection of prompt words and spend a few minutes generating as many images as they like. The activity uses an offline installation of Stable Diffusion (a text-to-image tool) behind the scenes, but I have pregenerated and manually moderated over 1,000 images, covering every possible combination of prompt words. This set-up gives pupils agency to explore GenAI, while removing any risk of them seeing inappropriate images.

The initial reaction from pupils is always one of excitement, as they churn out realistic images of cats wearing sunglasses, or Play-Doh versions of celebrities. However, on closer inspection they notice errors such as a third leg on Taylor Swift. Outputs that are initially plausible, followed by a realisation that there are factual errors, is a common weakness of GenAI. After hearing how Stable Diffusion has been trained on internet images, we discuss social and ethical issues (the SE layer) such as whether GenAI can ever be a replacement for human creativity.

Recognising bias

The next activity illustrates bias — a fundamental aspect of the SE layer — using a scenario that is relevant to the class. I ask all the teams to use the text-to-image tool to create a photo of a footballer celebrating a goal. When the pupils realise that every footballer is shown as male, they now have enough understanding of the training process to explain why the inherent gender bias in internet images of footballers flows through to GenAI's output. With older KS3 groups (11- to 14-year-olds), I raise more sophisticated ethical issues using live facial recognition in crowds as a case study, in which students observe the false positives and false negatives in an interactive simulation and discuss the privacy concerns and risks of misidentification.

Within the constraints of a 90-minute workshop, I'm unable to dwell on the engine (E) level of SEAME; however, for older groups I visualise how feature detection works in an image classifier's neural network.

By the end of the workshop, my aim is that all pupils can explain how ML models are built, evaluate applications of AI, and consider some social and ethical issues around their use. I recognise the concern that some teachers might have about using live AI in the classroom, but hands-on experimentation and reflection is the best way to provoke curiosity about AI and a healthy scepticism about its outputs. SEAME has guided my development of these activities, allowing a broad understanding of AI concepts by a wide range of ages. Society will benefit if we can use every available means to introduce AI literacy to as many young people as possible. (HW)

NICK NUROCK

Nick's career as a software developer includes eleven years at Transport for London (TfL) where he managed their apprenticeship and graduate software training schemes. After co-creating technology outreach workshops at TfL, he set up his own business providing in-person AI literacy workshops to schools around London (**standupcomputing.com, linkedin.com/in/nicknurock**).

THE OCEAN PROMPTING PROCESS

A framework to reliably get the results you want from an LLM

If you've heard of ChatGPT, Gemini, or Claude, but haven't tried any of them yourself, navigating the world of large language models (LLMs) might feel a bit daunting. However, with the right approach, these tools can really enhance your teaching and make classroom admin and planning easier and quicker. LLMs can provide a sounding board for ideas, act as a template and list generator, or draft emails and documents in seconds.

That's where the OCEAN prompting process comes in: it's a straightforward framework designed to work with any LLM, helping you reliably get the results you want, whether you're after a fun limerick, a list of interactive activity ideas, or even an illustrated choose-your-own-adventure story to engage your students.

The great thing about the OCEAN process is that it takes the guesswork out of using LLMs. It helps you move past that 'blank page syndrome' — that moment when you can ask the model anything but aren't sure where to start. By focusing on clear objectives and guiding the model with the right context, you can generate content that is spot on for your needs, every single time.

The OCEAN prompting process

Let's break down the OCEAN process. It's designed to make your interactions with LLMs smoother and more effective, helping you get the best out of the model, no matter what you need it to do. The name is an acronym: objective, context, examples, assess, negotiate — so let's begin at the top.

Objective

First things first, define your objective. Think of this as setting a clear goal for your interaction with the LLM. A well-defined objective ensures that the responses you get are focused and relevant.

Maybe you need to:

- Draft an email to parents about an upcoming school event
- Create a beginner's guide for a new Scratch project
- Come up with engaging quiz questions for your next science lesson

By knowing exactly what you want, you can give the LLM clear directions to follow, turning a broad idea into a focused task.

Context

Next, provide some context. This is where you give the LLM the background information it needs to deliver the right kind of response. Think of it as setting the scene and providing some of the important information about why, and for whom, you are making the document.

You might include:

- The length of the document you need
- Who your audience is — their age, profession, or interests
- The tone and style you're after, whether that's formal, informal, or somewhere in between
- The complexity of language required — do you need something simple, or is technical detail important?
- Any key information — dates, facts, figures — that needs to be included

All of this helps the LLM include the bigger picture in its analysis and tailor its responses to suit your needs.

Examples

Examples can be really helpful too. By showing the LLM what you're aiming for, you make it easier for the model to deliver the kind of output you want. This is called one-shot, few-shot, or many-shot prompting, depending on how many examples you provide.

You can:

- Include URL links
- Upload documents and images (some LLMs don't have this feature)
- Copy and paste other text examples into your prompt

Without any examples at all (zero-shot prompting), you'll still get a response, but it might not be exactly what you had in mind. Providing examples is like giving a recipe to follow that includes pictures of the desired result, rather than just vague instructions — it helps to ensure the final product comes out the way you want it.

Assess

Once you've got the LLM's response, it's time to assess. This is where you check whether what you've got aligns with your original goal and meets your standards.

Keep an eye out for:

- Hallucinations: incorrect information that's presented as fact
- Misunderstandings: did the LLM interpret your request correctly?

Important note: while LLMs can be powerful tools in the classroom, we encourage all teachers to review the Terms of Service for any LLM they choose to use. Please be mindful that LLMs, while often accurate, can occasionally provide incorrect or misleading information, so it's always wise to double-check facts. Additionally, unless using a corporate account with privacy safeguards built in, be aware that any data you input may be used by the provider for training and refining their models, so don't input personal details or names. Always consider your school's data policies when using these tools.

- Bias: make sure the output is fair and aligned with diversity and inclusion principles

A good assessment ensures that the LLM's response is accurate and useful. Remember, LLMs don't make decisions — they just follow instructions, so it's up to you to guide them. This brings us neatly to the next step: negotiate.

Negotiate

If the first response isn't quite right, don't worry — that's where negotiation comes in. You should give the LLM frank and clear feedback and tweak the output until it's just right. (Don't worry, it doesn't have any feelings to be hurt!)

When you negotiate, tell the LLM if it made any mistakes, and what you did and didn't like in the output. Tell it to 'Add a bit at the end about ...' or 'Stop using the word "delve" all the time!'

Another excellent tip is to use descriptors for the desired tone of the document in your negotiations with the LLM, such as, 'Make that output slightly more casual.'

In this way, you can guide the LLM to be:

- Approachable: the language will be warm and friendly, making the content welcoming and easy to understand
- Casual: expect laid-back, informal language that feels more like a chat than a formal document
- Concise: the response will be brief and straight to the point, cutting out any fluff and focusing on the essentials
- Conversational: the tone will be natural and relaxed, as if you're having a friendly conversation
- Educational: the language will be clear and instructive, with step-by-step explanations and helpful details
- Formal: the response will be polished and professional, using structured language and avoiding slang
- Professional: the tone will be business-like and precise, with industry-specific terms and a focus on clarity

This back-and-forth process lets you refine the output until it meets your specific needs. Treat every LLM like a new intern: be clear, check their work, be ready to correct them frequently. Remember: LLMs have no idea what their output says or means; they are literally just very powerful autocomplete tools, just like those in text messaging apps. It's up to you, the human, to make sure they are on the right track.

Final step: human edit

Even after you've refined the LLM's response, it's important to do a final human edit. This is your chance to make sure everything's perfect, checking for accuracy, clarity, and anything the LLM might have missed. LLMs are great tools, but they don't catch everything, so your final touch ensures the content is just right.

Just like having an intern or apprentice draft up a document, it's super-helpful and gets the job done quickly, but it might not catch everything, and you shouldn't ever trust an LLM to do so!

At a certain point it's also simpler and less time-consuming for you to alter individual words in the output, or use your unique expertise to massage the language for just the right tone and clarity, than going back to the LLM for a further iteration.

Ready to dive in?

Now it's time to put the OCEAN process into action! Log in to your preferred LLM platform, take a simple prompt you've used before, and see how the process improves the output. Then share your findings with your colleagues. This hands-on approach will help you see the difference the OCEAN method can make!

Sign up for a free account at one of these platforms:

- ChatGPT (**chat.openai.com**)
- Gemini (**gemini.google.com**)

By embracing the OCEAN prompting process, you can quickly and easily make LLMs a valuable part of your teaching toolkit. The process helps you get the most out of these powerful tools, while keeping things ethical, fair, and effective.

If you're excited about using AI in your classroom preparation, and want to build more confidence in integrating it responsibly, we've got great news for you. You can sign up for our totally free online course on edX called 'Teach Teens Computing: Understanding AI for Educators' (**helloworld.cc/ai-for-educators**). In this course, you'll learn all about the OCEAN process and how to better integrate generative AI into your teaching practice. It's a fantastic way to ensure you're using these technologies responsibly and ethically while making the most of what they have to offer. Join us and take your AI skills to the next level! (HW)

MARK 'MR C' CALLEJA

Mr C is a teacher, hacker, surfer, D&D Dungeon Master, thrower of bladed implements, and maker of useful things. During the day he makes educational resources for the Raspberry Pi Foundation, which you can find on the Foundation's website.

TEACHING CODING HAS HAD ITS CALCULATOR MOMENT

GenAI is to coding as calculators are to mathematics — but just as we still need to learn maths, we still need to teach coding

In a recent podcast with maths teacher Amy Marsh (helloworld.cc/ai-age-abela), she confided in me that she didn't really see the problem with generative AI. After all, we have had calculators for decades, and yet students still need to learn the logic of how mathematics works in order to use calculators in an effective and error-free way. We have also seen these errors brought to another level, with spreadsheets that have miscalculated company finances by billions of dollars (helloworld.cc/excel-error) or meant that Medicaid assistance was being cut to people receiving disability allowances in Idaho (helloworld.cc/Medicaid-Idaho).

Computer programs have always had bugs, but the fact that so much of our code runs millions of times a second without crashing is testament to the skill of the programmers that make programs of the highest quality.

This might in part be down to the fact that there are entry barriers to coding — but those barriers have now been diminished. Since the 1990s, we have had search engines to help find open-source software, and many libraries help to simplify making code. However, generative AI enables even those without any previous experience at all to have a go at writing code of some sort.

The quality of the source

AI relies on the quality of the sources it has available to it. So if it is using Stack Overflow, where bugs are almost guaranteed, rather than a source where code has already been checked, the code quality is likely to be much lower. Different AIs have been designed for different purposes, so while the Claude AI outperformed ChatGPT-4o on an A-level history exam paper (It scored a B), the reverse has been true when it comes to the quality of code — when it comes to coding, ChatGPT still seems to be on top (see the comparison video I made: **helloworld.cc/AI-showdown**).

The more open-source the content, the better the quality of code tends to be. In all the tests I have done, JavaScript is its strongest language, because it has literally the entire web to find code. In second place is Python, because of the large amount of open-source material available. Other languages have been fairly variable, depending on the task and the difficulty.

Number systems example

A good example of this was trying to build a JavaScript binary to hex to decimal converter. It made something that actually worked on the first attempt, and then I iterated it to a point where it had all the features needed. There were two major bugs as it iterated, and one point where it did not implement the feature as fully as requested. If I had not tested each new iteration carefully, I am sure more problems would have occurred. Even though I had

To be able to use generative AI at all requires a certain level of understanding

SUPPORTING STUDENTS 24/7

Custom GPTs can help support students 24/7. They are not perfect, but with specific exam knowledge and guidance, they have become useful tools to help students. Here is one I made to support iGCSEs.

fully working code in JavaScript, when I asked it to make a Python version with Tkinter, it created a Python version with the error "RecursionError: maximum recursion depth exceeded". It tried to fix these errors, but it was quickly clear that it was stronger in JavaScript. You can see the full chat at **helloworld.cc/chatgpt-Java**.

ChatGPT will very confidently tell you that it is making first-class code, and even breaks down how it is achieving the tasks. Just before making Python code with a recursion error, it said, "Sure! Here's a Python program using Tkinter that provides similar functionality with sliders and a checkbox for two's complement."

Show not tell

To ensure that our students understood this, we gave them a sandbox lesson in which they could use as much generative AI as they pleased. Students tried to make all sorts of simple text games — one was a mental maths developer; another made a website to display pi to large numbers of decimal places; one made a coffee machine; and one made a simple murder mystery. Many said that it enabled them to code a bit more quickly, but that they needed to read the results very carefully in case the AI had introduced bugs. However, 15 percent of students tried out the AI but then said they would prefer to code without it.

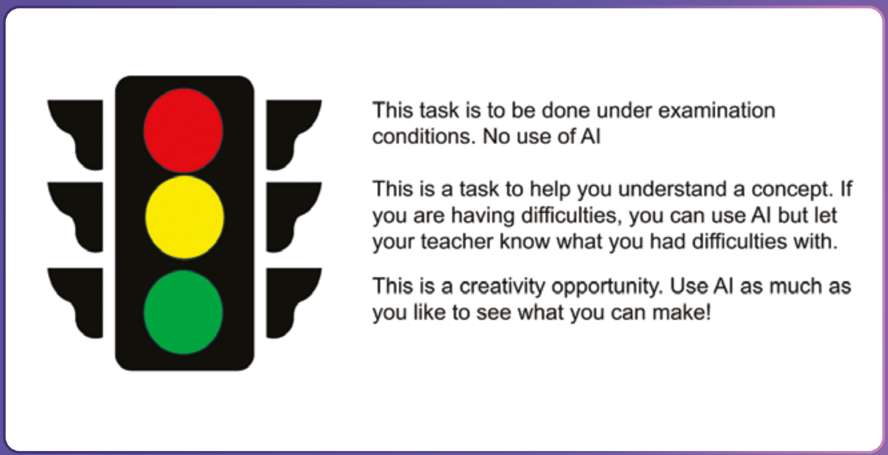
> CHATGPT WILL TELL YOU VERY CONFIDENTLY THAT IT IS MAKING FIRST-CLASS CODE

Students said that it was most effective when it helped to autofill algorithms they were working on. In this context, the AI was making snippets of code that they could digest, so they could understand what it was doing. They felt most comfortable doing this in Replit, which they had been using for a while. However, these were students who already had some proficiency with code, and even to be able to use generative AI requires a certain level of understanding. To get a piece of JavaScript to work, you need to have an editor or a server to run it, and the same is true of Python. In effect, the AI is therefore helping the good to get better, rather than helping absolute beginners.

To help students further understand when to use AI appropriately, I have created a traffic light system (**helloworld.cc/AI-traffic-light**). It encourages students to think about why they are using AI and its potential pros and cons, and helps to set ground rules.

This experience also underlines just how important the PRIMM pedagogy is, to ensure students understand the code they are reading and have high-quality debugging skills. See The Big Book of Computing Pedagogy, page 22 (**helloworld.cc/BBCP**).

Now, just as with calculators, teaching becomes a matter of ensuring that students understand the importance of the workings, rather than just helping them to arrive at the correct answer. (HW)



A traffic lights system encourages students to think about why they are using AI

JAMES ABELA

James is director of digital learning and entrepreneurship at Garden International School in Kuala Lumpur, Malaysia. He is founder of the South East Asian Computer Science Teachers Association and author of *Parenting and Teaching in the Age of AI* (**@eslweb**).

UNDERSTANDING AI FOR EDUCATORS: A NEW ONLINE COURSE

Mac Bowley shares an online course to help boost AI literacy in the classroom

The new online course 'Teach teens computing: understanding AI for educators' (helloworld.cc/ai-for-educators) aims to help educators around the world gain a practical understanding of the crossover between AI tools and education. We want educators to be able to make confident decisions about when to bring AI into the classroom. The course includes a conceptual look at what AI is and where it came from, different approaches to problem-solving with AI, and how to use the currently available AI tools effectively and ethically.

I want to share our approach to designing the course and some of the key frameworks we introduce — both of which you can apply in your setting today to teach about AI systems.

Design decisions

Speaking with educators around the world, it is clear that there is a varied level of comfort with AI tools in the education community. We want this course to help level out those differences and elevate every educator who takes it to a point at which they feel comfortable discussing AI in the classroom.

We also recognise that AI literacy is the key to understanding the implications and opportunities of AI in education. Our aim is to provide educators with the conceptual understanding they need to ask the right questions and make up their own minds.

As with every AI educational experience we create, our design principles were embedded from the very start:

- Choose language carefully to never anthropomorphise AI systems (**helloworld.cc/anthropomorphism**). We avoid making AI systems seem more human than they are. For example, replacing phrases like 'the model understands' with 'the model analyses' makes it clear that large language models are just another form of computer system, and have no agency, thoughts, or feelings.
- Avoid using AI as a singular noun. We opt for the more accurate 'AI tool' when talking about applications, or 'AI system' when talking about component parts of a computer system.
- Embed ethics. The social and ethical impacts should not be an afterthought, but should be embedded throughout learning experiences.

Three key learnings from the course

Here are three key learning points that are useful for any educator thinking about how to use AI tools in education:

AI literacy is the key to understanding the implications and opportunities of AI in education

FEEDBACK ON THE NEW COURSE:

"From my inexperienced viewpoint I kind of viewed AI as a cheat code. I believed that AI in the classroom could possibly be a real detriment to students and eliminate critical thinking skills. After learning more about AI and getting some hands-on experience with it, my viewpoint has certainly taken a 180-degree turn."

"AI definitely belongs in schools and in the workplace. It will take time to properly integrate it and know how to ethically use it. Our role as educators is to stay ahead of this trend, as opposed to denying AI's benefits and falling behind."

How to talk about AI systems

Deciding on the level of detail to use when talking about AI systems can be difficult, especially when your own confidence level is low. Lots of discussion around AI is either too abstract, or overly detailed.

The SEAME model breaks down AI education into four levels: social and ethical, application, model, and engine (**helloworld.cc/SEAME**). Educators can decide which level their students need to explore and can disregard the rest for a particular lesson. This provides useful limits and structure for an educator to use when deciding how detailed to make their educational experiences.

You might discuss the impact a particular AI system is having on society, without worrying about explaining how the model itself has been trained or tested. Equally, you might zoom in on a specific machine learning model to look at where the data used to create it came from and consider the effect that has on the output.

The SEAME model is a lens for you to use, to frame and pitch your AI education to meet the needs of your learners.

Problem-solving approaches

Broadly, you can separate AI applications into two categories: predictive and generative. These two types of AI model represent two vastly different approaches to problem-solving.

Predictive AI models use past data to make predictions about future outputs. For example, weather predictions can be made based on the previous weather data for a region, or new movies can be suggested to you based on your previous viewing.

> THIS COURSE AIMS TO ELEVATE EVERY EDUCATOR TO WHERE THEY FEEL COMFORTABLE DISCUSSING AI IN THE CLASSROOM

The possible use cases for these models is heavily restricted by the data used to create them. Usually, the problem comes first with predictive AI, then specific data is collected to help solve it.

Generative AI models also use past data, but instead produce brand new media (such as text, images, or audio), based on an analysis of other, similar media. The potential applications of these models are much more varied. With generative AI models, the variety of the output is paramount — you might say that they are a solution that is presented to the user to apply to a problem the user chooses.

How to get the most out of generative AI

Educators all over the world are looking for ways to use generative AI to solve problems. The core functionality of these systems revolves around a user providing a prompt — a description of what they want the model to generate.

At the moment, there are very few resources that provide a simple framework for prompting AI tools that anyone can apply to get the most out of these models. As a part of this course, we outline the OCEAN process — a simple set of steps that you can use when interacting with generative AI tools such as Gemini, Stable Diffusion, or ChatGPT (see pages 32–33).

Helping educators be critical consumers

The knowledge and skills gained in our 'Teach teens computing: Understanding AI for educators' online course should help any educator — regardless of specialisation — assess the right tools and concepts to bring into their classroom. We're here to support educators every step of the way through the course and our wider Experience AI resources, designed to help deepen teachers' understanding of AI with a solid, research-based platform for exploring this technology with their students (**experience-ai.org**). (HW)

MAC BOWLEY

Mac is a computing educator who has worked in almost every context imaginable: after-school clubs, holiday camps, enrichment days, and teaching GCSE students. Mac is passionate about empowering people to use technology to solve problems that matter to them.

LLM ERROR MESSAGE EXPLANATIONS – A FEEDBACK LITERACY PERSPECTIVE

Veronica Cucuiat shares her research on how artificial intelligence will impact teaching, learning, and assessment

Program error messages (PEMs) can be a significant barrier to learning for novice coders, as they are often confusing and difficult to understand. This can hinder troubleshooting and progress in coding, and lead to frustration. My research in this area specifically explores secondary teachers' views of the explanations of PEMs an LLM can generate to help the teaching and learning of programming.

What did the teachers say?

To conduct the study, I interviewed eight expert secondary-level computing educators. The interviews were semi-structured activity-based interviews, in which the educators got to experiment with a prototype version of the Raspberry Pi Foundation's publicly available Code Editor in Python. This version of the Code Editor has been adapted to generate LLM explanations when the question mark next to the standard error message is clicked (see Figure 1 for an example of an LLM-generated explanation). The prototype calls the OpenAI GPT-3.5 interface to generate explanations based on the following prompt: "You are a teacher talking to a 12-year-old child. Explain the error {error} in the following Python code: {code}".

Figure 1: Code Editor prototype version with LLM feedback

Then, 15 themes were derived from the educators' responses and these were split into five groups. Overall, the educators' views of the LLM feedback were that, for the most part, a sensible explanation of the error messages was produced. However, all educators experienced at least one example of invalid content (LLM hallucination). Also, despite not being explicitly requested in the LLM prompt, a possible code solution was always included in the explanation.

Matching the themes to PEM guidelines

Next, I investigated how the teachers' views correlated to the research conducted to date on enhanced PEMs. I used the guidelines proposed by Brett Becker and colleagues (**helloworld.cc/error-messages**), which consolidate a lot of the research done in this area into ten design guidelines. The guidelines offer best practices on how to enhance PEMs based on cognitive science and educational theory empirical research. For example, they outline that enhanced PEMs should provide scaffolding for the user, increase readability, reduce cognitive load, use a positive tone, and provide context to the error.

Out of the 15 themes identified in my study, 10 correlated closely to the guidelines. However, the 10 themes that correlated well were, for the most part, the themes related to the content of the explanations, presentation, and validity. The themes concerning the teaching and learning process, on the other hand, did not correlate as well with the guidelines.

Does feedback literacy theory fit?

However, when I looked at feedback literacy theory, I was able to correlate all 15 themes — the theory fits!

Feedback literacy theory positions the feedback process (which includes explanations) as a social interaction, and accounts for the actors involved in the interaction — that is, the student and the teacher — as well as the relationships between the student, the teacher, and the feedback. We can explain feedback literacy theory using three constructs: feedback types, student feedback literacy, and teacher feedback literacy.

Feedback types

Feedback type	Educator role	Student role
Telling	Unidirectional transmission of correct information	Passive
Guiding	Point in the right direction	Active as applies knowledge
Developing understanding	Targeted teaching	Active as constructs or adjusts knowledge
Opening up new perspectives	Presenting new perspectives	Active as interprets and evaluates new knowledge

(McLead, Bond & Nicholson, 2015)

Figure 2: Feedback types as formalised by McLean, Bond, & Nicholson (**helloworld.cc/McLean-et-al**)

From the feedback perspective, feedback can be grouped into four types: telling, guiding, developing understanding, and opening up new perspectives (Figure 2). The feedback type depends on the role of the student and teacher when engaging with the feedback.

From the student perspective, the competencies and dispositions students need in order to use feedback effectively can be stated as: appreciating the feedback processes, making judgements, taking action, and managing affect.

Finally, from a teacher perspective, teachers apply their feedback literacy skills across three dimensions: design, relational, and pragmatic.

In short, according to feedback literacy theory, effective feedback processes entail well-designed feedback with a clear pedagogical purpose, as well as the competencies students and teachers need in order to make sense of the feedback and use it effectively.

This theory therefore provided a promising lens for analysing the educators' perspectives in my study. When the educators' views were correlated to feedback literacy theory, I found that:

1. Educators prefer the LLM explanations to fulfil a guiding and developing understanding role, rather than telling. For example, educators prefer to either remove or delay the code solution from the explanation, and they like the explanations to include keywords based on concepts they are teaching in the classroom to develop students' understanding rather than telling them.

2. Related to students' feedback literacy, educators talked about the ways in which the LLM explanations help or hinder students to make judgements and action the feedback in the explanations. For example, they talked about how detailed, jargon-free explanations can help students make judgements about the feedback, but invalid explanations can hinder this process. Therefore, teachers talked about the need for ways to manage such invalid instances. But for the most part, the educators didn't talk about eradicating them altogether. They talked about ways of flagging them, using them as counter-examples, and having visibility of them to be able to address them with students.

3. Finally, from a teacher feedback literacy perspective, educators discussed the need for professional development to manage feedback processes inclusive of LLM feedback (design) and address issues resulting from reduced opportunities to interact with students (relational and pragmatic). For example, if using LLM explanations results in a reduction in the time teachers spend helping students debug syntax errors from a pragmatic time-saving perspective, then what does that mean for the student–teacher relationship?

Conclusion

By correlating educators' views on feedback literacy theory as well as enhanced PEM guidelines, we can take a broader perspective on how LLMs might not only shape the content of the explanations, but the whole social interaction around giving and receiving feedback. Investigating ways of supporting students and teachers to practise their feedback literacy skills matters just as much, if not more, than focusing on the content of PEM explanations.

This study (**helloworld.cc/feedback-literacy)** was a first-step exploration of eight educators' views on the potential impact of using LLM explanations of PEMs in the classroom. Exactly what the findings of this study mean for classroom practice remains to be investigated, and we should also examine students' views on the feedback and its impact on their journey of learning to program. (HW)

VERONICA CUCUIAT

Veronica is a research scientist who is interested in how young learners engage with digital creative ways. She's also a mum and a cycling enthusiast, and in her spare time you can find her at the allotment with her chickens.

AI UNPLUGGED

Why unplugged activities help us focus on what's really important in teaching AI (and other things)

I remember trying my first unplugged activity: using iron-on beads to replicate pixel graphics. This simple yet engaging task never seems to lose its appeal. Interestingly, unplugged activities like this feel quite natural to us. Consider how we teach young children to cook before they start school. We often use a sand kitchen, where the focus is on the basic concepts like mixing ingredients or following steps, while abstracting away the more complex details such as using a real knife or setting the oven temperature. In the same way, unplugged activities simplify key processes, allowing learners to focus on the core ideas without getting bogged down by intricate tools or technology.

With unplugged activities, we aim to recreate the same sense of accomplishment that is given by play, games, or challenges in computing education. Unplugged activities aim to offer students an opportunity to engage with the core concepts of a subject, without the need for computers or technology. By removing the technical barriers, unplugged activities make learning more inclusive — welcoming individuals who are unfamiliar with devices, as well as young students who might otherwise struggle to manage both the technology and the concept itself.

Nor are unplugged activities limited to primary education. They can also provide a fresh perspective for adults and experienced computing professionals, helping them to focus on the underlying principles rather than the technical details. This approach can therefore help to develop a deeper, clearer understanding of complex ideas.

Unplugged activities, such as iron-on beads to replicate pixel graphics, allow learners to focus on the core ideas without getting bogged down by technology

Distinct features of unplugged activities

Unplugged activities possess several distinctive features that make them ideal for a wide range of learners:

- They don't require computers, removing any technical challenges
- A game or challenge is at the heart of the activity, making computer science concepts accessible through playful interaction
- Physical objects are often used, encouraging hands-on, kinaesthetic engagement
- Students are prompted to interact and independently explore key computer science concepts
- These activities are simple to implement in the classroom and require inexpensive materials, such as cardboard, paper, or iron-on beads
- A narrative or story element is often incorporated to captivate learners, particularly younger ones
- Ideally, these activities are shared and adapted freely, thanks to permissive licensing

Unplugged activities have been a valuable resource in school classrooms for quite some time. Examples include programming a human 'robot' with a series of instructions, card sorting to explore sorting algorithms and complexity classes, or using iron-on beads to illustrate how images are represented through pixel graphics.

A deliberate choice

However, it's important not to view unplugged activities as a cheap alternative for schools that lack enough computers. Today more than ever, with many of us having easier access to technology, using unplugged activities is a deliberate choice. They focus on emphasising the fundamental ideas behind computer science concepts. While the core principles remain the same, these activities allow learners to explore them in a more hands-on, playful, and engaging way.

Unplugged activities are also distinct from board games. If the rules become too complex, students may spend too much time and mental energy just understanding and memorising them. To maximise their effectiveness, unplugged activities should be easy to grasp quickly, allowing students to dive into the concepts without unnecessary delays. Otherwise, the activity risks losing sight of its ultimate purpose: to facilitate deeper exploration of the topic at hand.

Unplugging artificial intelligence

When it comes to artificial intelligence (AI), unplugged activities have become a popular choice for many educators. AI, with its complexity and depth, encompasses ideas that have been around for quite some time. For instance, concepts like reinforcement learning or curriculum learning are not unique to AI alone. By using unplugged activities, we can bridge the gap between everyday understanding and how these ideas are applied in the field of AI.

Take the 'Good Monkey, Bad Monkey' game as an example (pages 70–73). It doesn't aim to replicate a specific algorithm, such as decision tree learning. Instead, it focuses on the fundamental idea of supervised learning — using labelled examples and an algorithm to discover the relationship between input and output. This core concept underpins many tasks, from medical image processing to document classification.

Even for the now-famous large language models (LLMs), unplugged activities can help break down complex ideas. At its core, an LLM can be thought of as a sophisticated 'autocomplete' system. Students can explore this by dissecting texts and predicting the next word or token themselves, while also experimenting with different strategies such as occasionally choosing less frequent words to mimic how LLMs adjust their output using parameters such as 'temperature'.

What's really important

No matter the topic, unplugged activities provide an accessible way to introduce students to what truly matters before diving more deeply into the subject. Even when we don't use unplugged activities directly, thinking in this way helps us to design learning experiences that strip away unnecessary complexity — whether it's from the program (like an IDE), the system (such as the computer), or other layers of distraction. Unplugged activities remind us to focus on the essential ideas, making them a great introduction to any topic, especially AI. Once students grasp the foundational concept, they can move on to more in-depth exploration, experimenting with real systems or implementing the ideas themselves. (HW)

> " THINKING IN THIS WAY HELPS EDUCATORS DESIGN LEARNING EXPERIENCES THAT STRIP AWAY UNNECESSARY COMPLEXITY

STEFAN SEEGERER

Stefan is the education lead at IQM Quantum Computers, exploring ways to make quantum computing accessible to everyone (**www.stefanseegerer.de**).

MULTIMODAL AI IN COMPUTING EDUCATION

Practical strategies for integrating multimodal AI in secondary- and college-level computing education

Since the release of GPT-3.5 in late 2022, large language models (LLMs) have rapidly evolved from text-only tools to multimodal systems capable of processing and generating text, images, audio, and even video. This article focuses on their impact in secondary schools (ages 11–16) and colleges (ages 16–19), noting that age restrictions (often 18+ or 13+ with parental consent) limit their use with younger students. We'll examine the rapid evolution of AI models, discuss practical strategies for integration, and consider the implications for educators and students.

The popularity of OpenAI's ChatGPT has sparked both excitement and concerns about generative AI's impact on education. A surge in model developments took place in 2023. OpenAI released GPT-4, a multimodal model working with images and audio. Other tech giants soon joined the race: Google launched Bard (later Gemini), while Microsoft introduced Bing Chat (later Copilot).

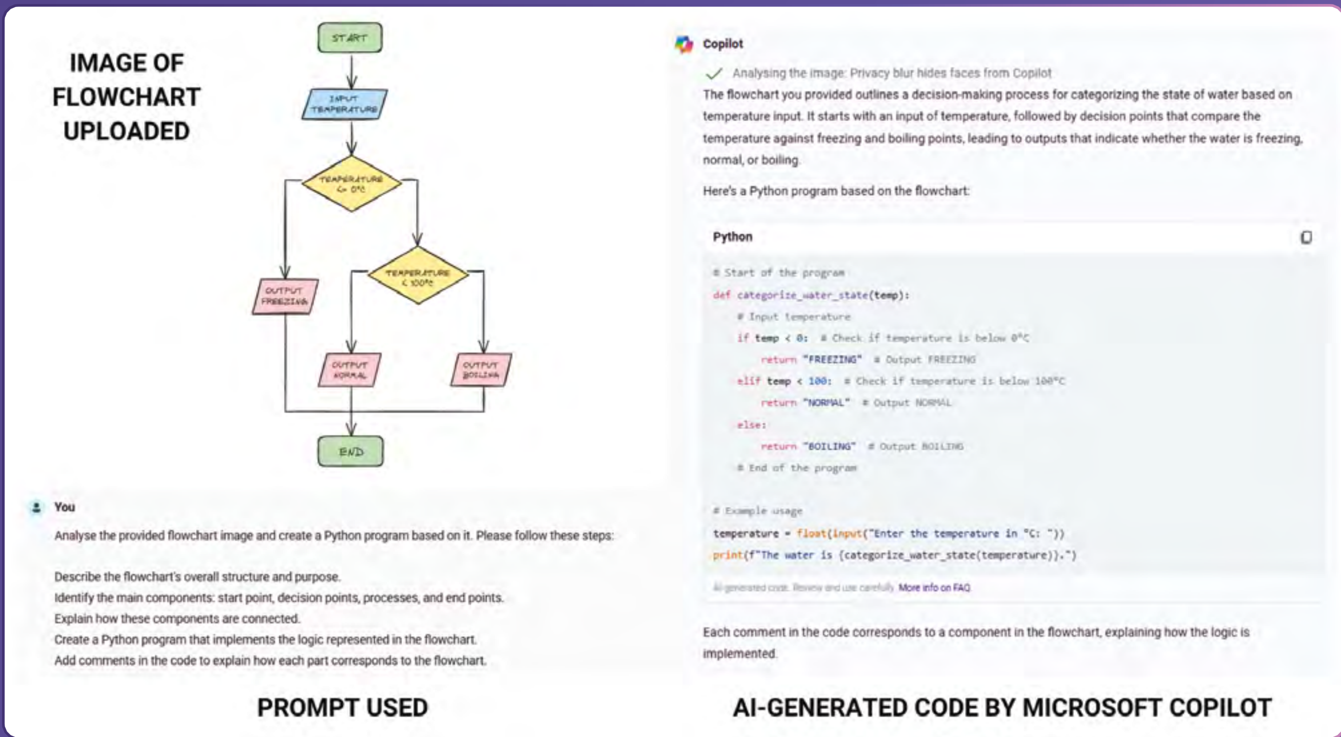
GPT-4 Vision (GPT-4v) enabled image recognition, while DALL-E 3 added image generation. These features, along with audio capabilities supporting speech-to-text and text-to-speech functions, were integrated into ChatGPT and Microsoft Copilot. Mobile apps made generative AI more accessible, bringing multimodal AI to users' pockets. In just one year, these tools transformed from text-only to multimodal systems working with diverse data sets.

Multimodality AI capabilities

In 2024, AI capabilities expanded further. OpenAI introduced GPT-4 Omni, enhancing image, voice, and text integration for human–computer interaction. Meanwhile, Google unveiled Veo, a generative video model that creates videos from text, image, and video prompts. These advancements have significantly improved multimodal AI

> RECENT ADVANCEMENTS HAVE SIGNIFICANTLY IMPROVED MULTIMODAL AI CAPABILITIES, REFLECTING RAPID DEVELOPMENTS IN THE FIELD

AIDAN WESTON
Aidan teaches computing and IT at Long Road Sixth Form College in Cambridge, UK. He's currently exploring the potential impacts of generative AI on computing education.



A flowchart analysing water temperature converted to Python code by AI, demonstrating the 'From Sketch to Code' exercise for computing education

capabilities, reflecting rapid developments in the field.

The latest multimodal AI models process and generate diverse inputs and outputs:

1. **Text processing:** these models excel in generating written content, from code and translations to formatted text and summaries.

2. **Image capabilities:** models such as DALL-E 3 and GPT-4 Vision have introduced various visual capabilities:

   - **Image generation:** they transform text captions into visual art
   - **Image recognition:** they identify objects and scenes in uploaded images, providing text descriptions of visual content
   - **Image enhancement:** they modify or improve images based on text prompts or other visual inputs

3. **Audio features:** models like OpenAI's Whisper and similar technologies have introduced audio processing capabilities:

   - **Speech-to-text:** they convert voice input into written text
   - **Text-to-speech:** they read generated text responses aloud

4. **Multimodal integration:** they combine inputs seamlessly, for instance generating an image from text, analysing it in writing, and reading the analysis aloud.

While focusing on text, image, and audio, AI is advancing into video and 3D generation, exemplified by Google's Veo. This expansion highlights the potential for future advancements in multimodal capabilities.

Practical strategies

As multimodal models become more accessible, educators can explore practical strategies for computing using tools like ChatGPT or Microsoft Copilot. One powerful approach combines different modalities to support students — the 'From Sketch to Code' exercise.

This exercise leverages the GPT-4 Vision model for image recognition in both ChatGPT and Microsoft Copilot. Here's how it works:

1. Students draw an algorithm or flowchart on paper or using an online diagram tool such as Microsoft Visio.

2. They upload the image to ChatGPT or Microsoft Copilot.

3. Using a text prompt, students ask the AI to analyse the sketch and generate Python code (or a chosen language).

4. The AI uses its text-to-speech function to read the code aloud to the student (optional).

5. Students compare the AI-generated code with their expectations, fostering critical thinking.

This exercise is best suited to older secondary and college students, given the age restrictions on many AI tools. For under-18s, educators must obtain parental consent and comply with these age limits when implementing AI exercises.

This exercise offers multiple benefits: reinforcing algorithm design, introducing code-generation concepts, and enhancing critical thinking. Students can critique or expand upon the AI-generated code, or prompt the AI for improvements to implement themselves. Students might also iteratively refine their flowchart, using AI-generated code to test logic interactively. This approach is adaptable to various learning objectives and the differing learning needs of students.

> “ IT'S OUR RESPONSIBILITY TO CRITICALLY EVALUATE STRATEGIES

In his 2024 book *Practical AI Strategies*, Leon Furze cautions against over-reliance on AI in education. He emphasises: “We don't want to offload the important work of planning lessons and creating resources onto GenAI. We want to use the technology to support our own expertise and professionalism.” The book offers a variety of practical strategies for applying multimodal AI capabilities. However, it's our professional responsibility as educators to critically evaluate and determine which strategies are truly suitable for our computing lessons.

Considerations and recommendations

Ethan Mollick's 2024 book *Co-Intelligence: Living and Working with AI* offers valuable insights for educators. His advice to “assume this is the worst AI you will ever use” highlights AI's rapid evolution and encourages the embracing of technological advancements. While predictions vary, multimodal models are expected to keep advancing, potentially impacting education just as much as other industries. Hands-on experimentation is crucial if educators are to understand AI's potential and limitations in educational settings.

Mollick also emphasises the importance of being “the human in the loop”, framing AI as a collaborator rather than a replacement for human expertise. This principle encourages viewing AI as a tool to enhance teaching and learning experiences in computing education.

While advanced, then, multimodal AI still has limitations and raises concerns:

- Reliance on text-based prompts for image tasks may introduce biases from training data sets
- Potential inaccuracies in image generation and recognition
- Risk of hallucinations and incorrect information
- Privacy issues with personal data and sensitive images, though some models automatically blur faces for protection
- Given the variability in AI output accuracy, it's crucial to verify all generated content

Implementing AI in education requires the consideration of:

- **Equitable access**: the potential widening of the digital divide due to unequal access and subscription costs
- **Usage guidelines**: establishing best practices for educational use without a standard manual
- **Age restrictions**: compliance with user age limits (often 18+ or 13+ with parental consent), restricting use in primary and early-secondary education

The responsible use of AI in computing education involves identifying both beneficial and non-beneficial applications, making AI integration a thoughtful choice rather than a default action.

Here are some final recommendations for educators:

1. **Experiment with multimodal AI:** explore the text, image, and audio features of tools like ChatGPT or Microsoft Copilot before classroom introduction. Try the ‘From Sketch to Code' exercise to understand how AI interprets visual information. Always respect the age restrictions of AI tools and obtain the necessary permissions.

2. **Stay informed and adapt:** keep up with AI developments by following reputable sources and attending relevant workshops. Consult comprehensive resources like the Raspberry Pi Computing Education Research Centre's Guide on AI in Education (see link in ‘Further reading' box) to inform your teaching strategies and curriculum planning.

3. **Foster critical thinking:** develop your own skills in critically evaluating AI-generated content and identifying potential biases or inaccuracies. Use AI as a tool to enhance human expertise rather than replace it. This approach will strengthen your subject knowledge and better prepare you to guide students for future careers in which AI collaboration may be increasingly common. (HW)

FURTHER READING

‘Using generative AI in the classroom: a guide for computing teachers' by Raspberry Pi Computing Education Research Centre (2024) (**helloworld.cc/ai-guidance-2024**)

Practical AI Strategies: Engaging with Generative AI in Education by Leon Furze (2024) (**helloworld.cc/Furze**)

Co-Intelligence: Living and Working with AI by Ethan Mollick (2024) (**helloworld.cc/Mollick**)

WHAT WOULD FREIRE SAY?

Divya Mahadevan discusses Paulo Freire's education model and how it applies to generative AI

In July 2021, I had just quit my software engineering job. I would start my master's in education and technology in the autumn, but for now, a summer of fun lay waiting ahead of me. I distinctly remember waking up on the morning after my last day, checking my phone, and seeing an email titled 'Summer Reading'. Number one on that list? *Pedagogy of the Oppressed* by Paulo Freire. An easy beach read, surely.

Banking or problem-posing?

While not quite as light-hearted a book as I had planned for myself, I found that *Pedagogy of the Oppressed* was an incredibly useful, highly relevant framework to help me think about technology in education. Written in 1970 by Brazilian philosopher Paulo Freire, the book critiques the 'banking model' of education, in which children are seen as passive objects, waiting blankly to be filled with information. Freire instead advocates for a 'problem-posing' form of education, which encourages young people towards critical thinking and improving their environments. Within these broad strokes lies an interesting approach to thinking about the benefits and harms of education technology.

And of course, the technology that everyone can't stop talking about is artificial intelligence, particularly generative AI. GenAI can be used to create lesson plans and learning content. Perhaps more worryingly, it can also be used to write essays and other homework assignments. For better or worse, though, AI is here to stay. The question is, will it promote Freire's 'banking model' of education (stifling young people's agency), or the 'problem-posing' version (empowering young people)?

GenAI pizza

Before I dive fully into Freire, it may help to explain (at a high level) how GenAI works. I like to compare GenAI to tasting a dish and then trying to recreate it. Imagine you've never had pizza before. Your brain will use all your senses and past knowledge to identify what a pizza is, essentially using billions of data points stored in your brain to characterise pizza. It will pick up on the shape, the look, the taste, the feel. However, if you've only ever eaten pepperoni pizza, you're much more likely to recreate pepperoni pizza than put pineapple on pizza (the horror). Neural networks, the underlying foundation of GenAI, work in a vaguely similar way. They rely on massive amounts of training data, which help them to identify patterns and key characteristics of objects. Once it can break something down into its unique characteristics, GenAI can generate multiple versions of the same thing by creating different variations of those individual parts (crust, sauce, cheese, and so on). Crucially, these variations are all learnt through the training data — and the more diverse the data, the more diverse the final output.

Depending on how diverse the training data is, these slices may or may not be classified as pizza

Decision-making

This simple understanding of GenAI is enough to start applying Freire and identifying the benefits and potential harms of AI in education. Freire's main critique of the 'banking model' is that it robs individuals of decision-making opportunities. He writes: "To alienate human beings from their decision-making is to change them into objects" (1970, page 58). Freire views decision-making as a key driver for individual growth and cognition.

How, then, might AI stymie opportunities for decision-making? Take the training data AI relies on. These data sets are huge, but incomplete in many ways. For historical, social, and political reasons, a data set of all university students will have limited representation of Indigenous student populations — so much so that any models trained on this data set will produce highly inaccurate predictions for Indigenous students (Baker, 2021). Manually curated data sets may seem like a remedy, but the immense size of these data sets makes it impossible (Birhane, Prabhu, et al., 2021). As Fuchs (2018) demonstrates, one of the only ways we can identify bias is simply by watching what decisions these algorithms make (a passive process). This leaves a small percentage of people consistently misrepresented within these models but unable to fix it, and often unable to stop the tools from being used. In Freire's lens, biased AI results in a barrier between people and their ability to take action — a clear sign that the banking model is at work.

Credit: © olyphotostories/ stock.adobe.com

AI removes opportunities for decision-making and critical, independent thought, key tenets of Freire's socially just education model

> "TO ALIENATE HUMAN BEINGS FROM THEIR DECISION-MAKING IS TO CHANGE THEM INTO OBJECTS" — PAULO FREIRE

Independent thought

When used without careful thought, AI not only removes opportunities for decision-making, but can also reduce opportunity for critical, independent thought, a key tenet of Freire's socially just education model. Dixon-Román et al. (2020) reviewed Essay Helper, an AI tool that both grades student essays and helps students write them. This tool, while possibly viewed as helping students improve their writing skills, was shown to encourage students to write to the rubric. A failure to conform to the rubric resulted in a poor grade. As any educator may tell you, while the rubric is important, learners can produce highly creative, unique essays that may not adhere strictly to a rubric. This should not be penalised, but rather celebrated. Furthermore, tools like this one are often not designed to account for niche needs of learners: it may automatically assign poor grades to students with English as a second language, because it was not built for this use case. Despite

FURTHER READING

Baker, R. S. (2021). Algorithmic Bias in Education. 36.

Birhane, A., Kalluri, P., Card, D., Agnew, W., Dotan, R., & Bao, M. (2021). The Values Encoded in Machine Learning Research. ArXiv:2106.15590 [Cs]. (**helloworld.cc/ml-values**)

Birhane, A., Prabhu, V. U., & Kahembwe, E. (2021). Multimodal datasets: Misogyny, pornography, and malignant stereotypes. ArXiv:2110.01963 [Cs]. (**helloworld.cc/multimodal-misogyny**)

Dixon-Román, E., Nichols, T. P., & Nyame-Mensah, A. (2020). The racializing forces of/in AI educational technologies. Learning, Media and Technology, 45(3), 236–250. (**helloworld.cc/racializing-ai**)

Freire, P. (1970). Pedagogy of the Oppressed (M. Bergman Ramos, Trans.; Published in Penguin Classics 2017). Penguin Books.

Fuchs, D. J. (2018). The Dangers of Human-Like Bias in Machine-Learning Algorithms. 2, 15.

Scardamalia, M. & Bereiter, C. (2020). Will knowledge building remain uniquely human? QWERTY 15(2), 12–26. (**helloworld.cc/knowledge-building**)

its promises to save time for teachers and improve student learning, we cannot simply hand over decision-making to AI; it is a set of flawed tools trained on incomplete data, with high potential to turn students into the passive objects described within the banking model.

Agency over AI

It is not all doom and gloom. The key words here are 'careful thought'. AI does have the potential to promote Freire's 'problem-posing' form of education, when used thoughtfully. Scardamalia and Bereiter (2020) explore what applications of AI could look like that promote student agency and creative, critical thought. There are some well-known guidelines, for example: treat AI as a tool for providing information, and allow teachers and students to decide whether they want to act on that information. This evens the playing field, as Freire advocates for in his definition of 'problem-posing' education. In his view, learners should not be deemed as beneath any other actor, be it teachers or machines. Scardamalia and Bereiter (2020) also advocate for play as a way to establish student agency over AI. AI is flawed, yes, but one way to encourage students to learn this is by letting them play with it and take control. They describe an experiment in which young people were able to convince an AI tool that dinosaurs still roamed Earth. They write "The child was expressing ... a shift in attitude, from seeing the computer in this context as awesome and commanding to seeing it as something they could make do what they wanted — including making it do stupid things" (page 21). If we allow teachers and students to decide how to use AI, they retain agency, and learning can become a collaborative effort between humans and AI.

The answer?

So, does the use of AI tools promote the 'banking model' or the 'problem-posing' model? I wish I could use Freire to tell you directly that the answer lies one way or the other. Unfortunately, as is often the case with technology in education, it depends. It depends on the situation and how thoughtfully these tools are applied. It depends on who is using the tool, and who is building it. My overarching goal with this article was a bit more abstract: I wanted to show that these foundational works of education theory are still extremely relevant and highly useful in today's context. I don't think I would have an opinion of AI edtech without *Pedagogy of the Oppressed*. Critical theories like this provide frameworks we can use to evaluate the effectiveness of technology in the context of learning. So as AI continues to develop at a rapid pace, and new applications start trickling into learning contexts, I encourage you to ask the same question I find myself asking more and more these days: "What would Freire say?" (HW)

The 'banking model' is where children are waiting to be filled with information

Freire advocates for a 'problem-posing' form of education

DIVYA MAHADEVAN

Divya is interested in the intersection of education, technology, and storytelling. After completing her master's at the University of Oxford, studying education, technology, and social justice, Divya joined the Raspberry Pi Foundation as a product manager in the digital products team. She is curious about how technology can help improve access to and engagement with education (**www.divyamahadevan.com**).

MINIMALISM IS HARDER THAN MAXIMALISM

A minimalist view of GenAI

As AI tools have proliferated over the past year, the advice has trended towards embracing the maximalism of it all. It is an age in which new tools and new features are being released every week. This puts us into a reactive mode, feeling as if we need to keep up with the pace of everything that's hitting the airwaves. What exciting feature did OpenAI release this week? What new tool is everyone talking about?

All this has us in a state of tool-searching hyperawareness, asking about features and capabilities, and making it easy to lose sight of goals and process.

A minimalist lens

In my blog, 'Minimalist EdTech', I advocate for the critical and informed development and use of educational technologies. A minimalist lens forces us to pay attention to the benefits and constraints of educational technologies; not necessarily to reject them, but to come to know more through questioning. Minimalist edtech focuses our attention on how teachers and students can or cannot control technologies; that is, on our human role.

Every choice in technology requires trade-offs (that's software engineering 101), and we can get lost in the specifics when new technologies suddenly leap into our awareness. It's happened recently (and continues to percolate) with augmented reality, virtual reality, and the thing formerly known as the metaverse; it's been frothing over ChatGPT and DALL-E (now Sora), then Claude, then Llama, and all the various non-OpenAI versions of generative AI. As everyone sees what's really involved and how these things work, we'll get some sort of equilibrium. Some uses will pass into transparent everyday familiarity (for better or worse) and others, including some that seemed sure to stick around, will fall by the wayside. Still other variants and adaptations will arise that have not yet been predicted clearly. (For that last category, keep an eye on the fact that we're moving from experimentation time to production time, when rubber meets the road. The infrastructure around generative AI development still has a long way to go, but all the major tech players have skin in the game, and open-source versions of these things are iterating rapidly.)

Approaching educational technologies (or technology in general) in terms of minimalism and the network of ideas around that (including sustainability, class, labour, and so on) demands uncomfortable questions of existing practice. We need to ask not 'Is this minimalist?' but rather 'Why do we need this?', 'What does this get us?', and 'How does this align with our values?'

Credit: ©sorapop/stockadobe.com

Every choice in technology requires trade-offs

The importance of context

In general, I don't think the answer is always that we value less, in the sense of having less capabilities or being stripped down. Minimalism in itself can be incredibly difficult to achieve, and can require large expenditures of time and energy. It's not any different in education. To get something that looks like a direct line to some set of values could in fact require massive amounts of preparation, or even a fairly large dose of technological augmentation. It's a shifting target, too, insofar as what reads to others as minimalist can shift with time, situation, and context. To take a favourite example of mine for the importance of context,

a top-of-the-line typewriter from 1969 was decidedly not minimalist at the time; it was full of features and was suited to particular kinds of office work and writing tasks. But nowadays, that same machine is a Luddite fantasy when compared to a MacBook or iPad, devices that are themselves minimalist when compared to other knobby and overgrown computers of a certain type. That mid-century typewriter is emphatically maximal in its mechanics, but those mechanics overall have become a retro-minimalism marker, where not being connected to the mass of material on the internet, or not suffering the constant interruption of push notifications on an electronic device, is more significant than the internal precision of springs and levers.

Minimalism requires effort

Minimalism is just one in a constellation of ideas. We might measure technology against its simplicity, its ease of use, its transparency, or in terms of how straightforward the connection is between mechanics and output, or its clean footprint in the world at large. These variants are neither mutually exclusive, nor always in sync. They can be counter-indicating. For example, a minimalist interface often takes significantly more work to produce than something clunky, full of data fields, and taking little time to design. Straightforward UX, in particular, is an achieved state requiring a lot of work. This is no different to certain minimalist or stark aesthetics in art, architecture, and design. Clean lines can require significant shaping or engineering to hold them in place. The marvel of the temples of the ancient world, for example, was not only that they were large, but also that, against the rough edges of the natural landscape, they had straight lines.

Credit: ©acinquantadue/stock.adobe.com

Depending on the context, a laptop can be considered more minimalist than a typewriter

The Pyramids of Giza, for example, were perfection in geometry fit for the divine. I think about Art Spiegelman's graphic novel Maus, in which Spiegelman honed lines and worked with magnified images in order to achieve what in the end would look more spare than the detailed illustration he could otherwise produce (see the companion book MetaMaus for more). That technique of 'wearing down' is one that I return to quite often. It's similar in many ways to iterating as a teacher: each run-through can allow for a refinement, such that the end result might look effortless or spontaneous, but only because the current group of students hasn't seen the previous ten years' worth of careful editing, experimentation, and trial and error.

> “ A MINIMALIST INTERFACE OFTEN TAKES SIGNIFICANTLY MORE WORK TO PRODUCE THAN SOMETHING CLUNKY

Crafting

I see minimalist edtech in similar terms, as a form of long-term whittling, sculpting, and crafting. Education is craft, not merely transactional, and a focus on minimalism helps us with that process as well. It is not merely a question of whether we should or should not use a particular tool or technology; it's about being committed to honing it finely. That may mean rejecting some things; it usually means using things in ways they weren't necessarily intended for, or working with parts of things.

In an age of AI-generated work, as AI makes its way into everyday practice, the question isn't really whether or not to adopt a tool, but of how we whittle down our teaching practice to the most effective moves and gestures.

That human work of iterating and reducing, and of simplifying through learning, may not be something that can be sped up by technology, though there are plenty of products that are promising such individualisation and efficiencies. The question is not then whether a particular AI tool can do this or that in a classroom; it's more about whether and how we shape the maximalism of the present into the considered and intentional practices of the near future. That's a harder task, a longer job, and urgent work.

Experiment away, but let's keep a focus on worthwhile targets. Impact with less work. Better outcomes, not more tools. (HW)

MINIMALIST EdTECH

'Minimalist EdTech' is a website of my reflections on how to use educational technologies responsibly and critically (**minimalistedtech.org**).

(HW)

SUBSCRIBE TODAY

FREE IN PRINT
for UK-based educators

Why subscribe?

- Teaching resources and ideas used by over 90 percent of our readers
- Exclusive news, research findings, and in-depth features
- Delivered to your door three times a year

TO SUBSCRIBE VISIT:
helloworld.cc/subscribe

Not a UK-based educator?

- Buy a print copy from **helloworld.cc/store** — we ship to over 50 countries
- Subscribe to receive the free PDF on the day it is released
- Read features and news at **helloworld.cc**

FREE PDF
for anyone, anywhere

AN INNOVATIVE COMPUTER SCIENCE IMMERSION PROGRAMME

A Minnesota School District's equitable approach to CS learning

In the United States, Minnesota has found itself consistently behind all other states in its adoption and implementation of computer science education, according to Code.org's 'Annual State of Computer Science' reports. This ranking surprised many, given Minnesota's reputation for educational innovation and a thriving tech industry. In response to this pressing issue, Bloomington Public Schools (BPS) embarked on a mission six years ago to prioritise computer science education. Recognising that the rapid evolution of CS demands continuous adaptation, BPS has developed a comprehensive and equitable Computer Science Immersion programme that is a model for other school districts.

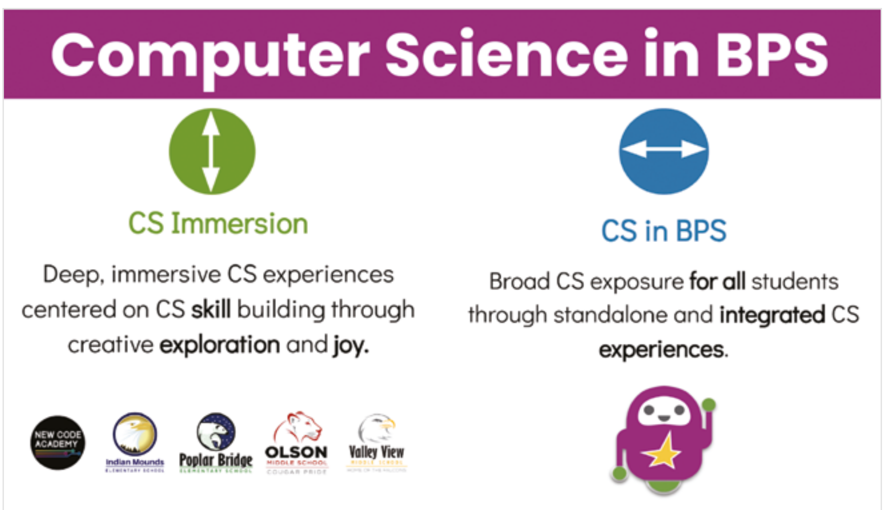
Given the fast pace of change in the field of computer science, we understand that our programme will always be a work in progress. However, we are proud of our computer science pathways, which include broad exposure to CS for all students through our CS in BPS efforts and our K–8 (ages 5–14) CS Immersion programme.

What is computer science immersion?

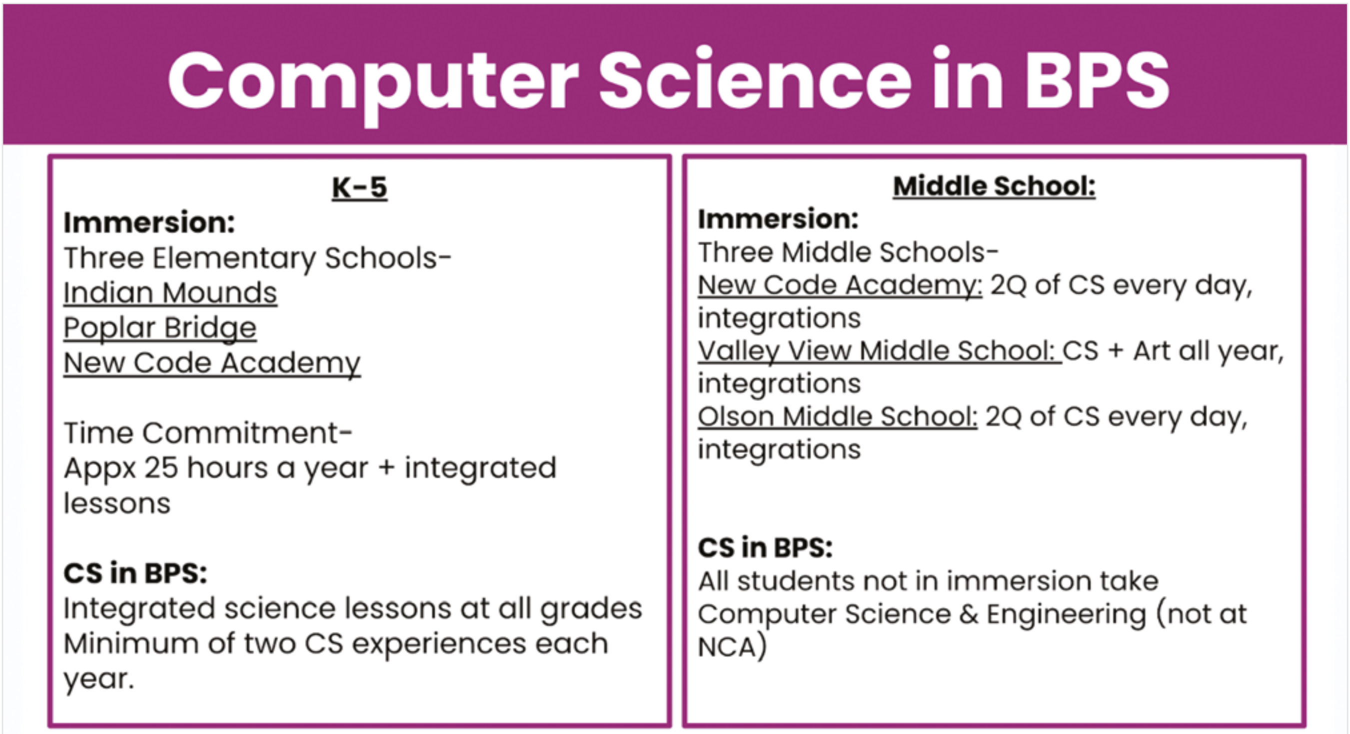
BPS's CS Immersion programme is implemented across three elementary schools (ages 5–12) and three middle schools (ages 11–14), two of which are integrated into our online school, New Code Academy. The programme serves as a cornerstone of the district's strategy to address the educational needs of 21st-century learners. Every year, the programme provides elementary students with 25 hours of direct CS instruction through weekly sessions with a CS specialist. This instruction is supplemented by three to six hours of integrated lessons that span subjects such as science, art, social studies, and English language arts (ELA). In these integrated lessons, educators collaborate with CS specialists to develop curricula that not only enhance the teachers' pedagogical capabilities, but also empower students to better understand computing by embedding CS skills across various disciplines.

Our CS Immersion programme offers a more concentrated exposure to CS at the middle-school level, with one semester of instruction in grades 6, 7, and 8 (ages 11–14). This curriculum is enriched by integrating CS concepts into core subjects, which varies depending on grade level. Examples include 6th-grade social studies, 8th-grade science, 7th-grade ELA, and CS/art at each grade level. BPS aims to cultivate a robust foundation in CS, preparing students for more advanced studies, both in high school and beyond.

Our programme is designed to ensure every student, regardless of background, has access to foundational computer



At Bloomington Public Schools, there is an immersion programme (CS Immersion) and a broad computer science exposure programme (CS in BPS)



Computer science in BPS is designed to ensure every student has access to computing knowledge

science and computational thinking skills. We believe everyone belongs in computing, everyone can make a difference in computing, and computing knowledge will benefit all students, regardless of their future career paths.

Building the foundation: skills, equity, and joy

At the heart of our CS Immersion programme lies a deep commitment to three core principles that guide our approach: skills, equity, and joy. Each pillar reflects a key element of what we believe to be essential in preparing students not just to navigate, but to thrive in a rapidly changing, technology-driven world.

Skills are at the core of our curriculum. We emphasise not just coding or programming, but essential 21st-century competencies such as problem-solving, collaboration, and creative thinking. These skills are embedded across all grade levels, ensuring that students are equipped for success in any field. Bloomington's curriculum is built to prepare students for the complexities of the modern world, emphasising the ability to approach open-ended challenges with confidence and persistence.

Equity is a guiding principle of our programme. In a state that struggles with providing universal access to computer science education, BPS is committed to removing barriers and ensuring every student, regardless of gender, economic status, or ethnicity, has the opportunity to excel in computer science. A key component of this focus on equity is the non-negotiable policy that English learners and special education students are not pulled out during computer science time. This policy ensures that all students, including those in centre-based special education classrooms, receive direct computer science instruction from our specialists. By starting early and offering immersive experiences, we aim to foster a diverse community of future tech leaders.

> SKILLS, EQUITY, AND JOY ARE THE PRINCIPLES THAT GUIDE OUR APPROACH

Joy is what makes our programme truly unique. Learning should be joyful, and we see that in our computer science classrooms. Passing by an elementary classroom, you can hear students chanting the phrase, 'Computer science is using the power of computers to solve our problems and express ourselves.' We promote joy through coding, unplugged activities, and physical computing that

Joy is what makes our programme truly unique

encourages students to express their true selves and celebrate their individuality in the classroom. Joy is the freedom to craft, create, learn, and share learning in a way that engages a student's whole self. That is the joy we strive for in our classes.

> OUR PROGRAMME HAS A SIGNIFICANT IMPACT ON UNDERREPRESENTED GROUPS

Programme design and implementation

The BPS Computer Science Immersion programme was launched in 2019 through a strategic partnership with the University of Minnesota. This collaboration allowed us to design a research-based curriculum that is both rigorous and accessible. Initially, we relied on resources like CSinSF and Code.org to build our curriculum. Partnerships with organisations like the University of Minnesota and Code Savvy have all been instrumental in supporting our curriculum development.

Over the years, through countless hours of professional development, we now have a team that actively builds and updates our PreK–8 curriculum. It's important to note that none of the educators on our CS team, either currently or at the programme's launch in 2019, had a background in computer science. Instead, they were professional educators with exceptional pedagogical skills who understood the importance of providing computer science education to our students.

This education-first background helped to embed computer science concepts seamlessly into various areas of study. This not only broadens students' exposure to computer science, but also enhances their education experience and introduces them to new interests and future career paths.

Successes and impact

One of the most significant achievements of our programme is its impact on underrepresented groups. By providing early access to computer science education, we are opening doors for students who might not otherwise consider a career in STEM fields. The programme's focus on equity and joy has led to increased engagement among girls, students of colour, and low-income students — groups that have traditionally been underrepresented in technology.

The programme's success is evident in the strong community partnerships we've built. Organisations like MnTech, Code Savvy, Amazon Future Engineers, BootUp, SICK Sensor Intelligence, and The Works Museum have played a crucial role in supporting our students and teachers. These partnerships have extended learning beyond the classroom, offering students opportunities to explore computer science through after-school programmes, summer camps, and internships.

Moreover, our high-school students have access to a range of advanced courses, including cybersecurity, AP computer science, data science, and robotics. These courses not only prepare students for high-demand tech careers, but also encourage them to apply their skills to real-world challenges in their communities.

Areas for continued growth

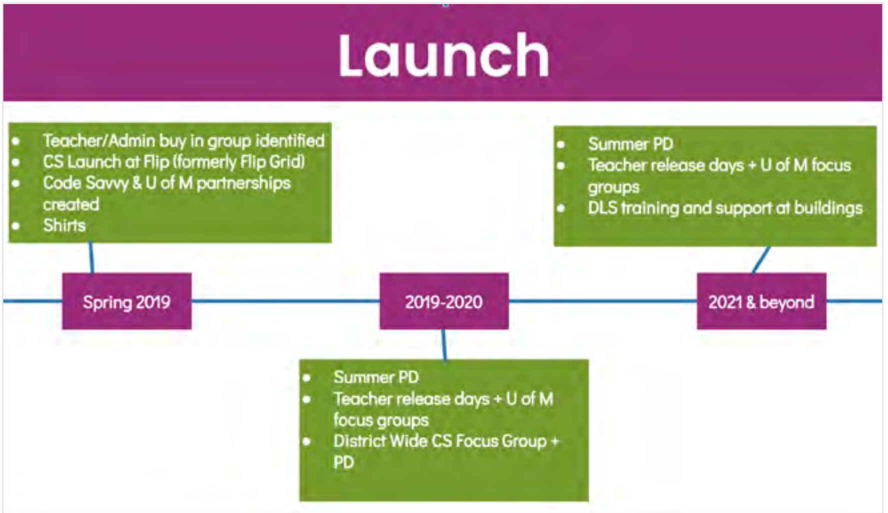
We are proud of all that we have accomplished so far. We also recognise that there is still work to be done. One of the biggest challenges we face is recruiting and retaining qualified computer science teachers. Like many districts across the state, we're struggling to find educators with the expertise and passion to teach computer science at all grade levels. The state of Minnesota is working on developing a dedicated CS teacher licensure option. This is an encouraging step, and we hope preservice teacher education programmes will start to provide all preservice teachers across the state with the knowledge of the what, why, and how of CS in a classroom.

To overcome this challenge, we are exploring innovative solutions, such as hiring passionate educators who can engage in a 'train the trainer' model, in which experienced teachers mentor their peers in computer science instruction. Our CS teachers and digital learning specialists are highly engaged professionals, attending conferences like CSTA and completing over 16 hours of professional development through Amazon Future Engineers and BootUp training last year. They collaborate continuously on lesson development, co-planning, and co-teaching with core subject teachers. This commitment to ongoing practice and iteration is what sets our educators apart. I firmly believe that Bloomington has some of the best CS educators in the nation, and this is no accident. Each educator is deeply committed to engaging and honouring the whole student and their family in the learning process. Their active participation in curriculum review, development, and adaptation, combined with their drive for continuous improvement, ensures that they embody the skills, joy, and equity that are central to our programme.

We believe every student should have the opportunity to experience the depth of our Immersion programme, but we have also worked hard to give students broad exposure to computer science through our non-immersion programme, our CS in BPS pathway. Growth is attainable with additional resources and support, and we are committed to making progress in this area.

Conclusion

The BPS Computer Science Immersion programme is more than just an educational initiative; it's a commitment to preparing every student for a technology-driven future. By focusing on skills, equity, and joy, we're not only shaping the next generation of tech leaders, but also creating a more inclusive and innovative community. As we continue to refine and expand our programme, we look forward to the day when every student in Minnesota has access to high-quality computer science education. (HW)



Supporting teachers with professional development and community partnerships was key to our launch

ALEXANDRA HOLTER

Alexandra is a computer science and career & technical education coordinator for Bloomington Public Schools in Bloomington, Minnesota. In 2015 she completed her PhD in educational leadership and policy studies from Oklahoma State University. She is the current chair of the CSTA Policy Committee and Editorial Board.

CREATING SYNERGY BETWEEN PRIMARY AND SECONDARY COMPUTING

Rob Arnold shares his thoughts on how pupils can improve their loading times and get up to speed when transitioning from a primary to a secondary computing classroom

The start of a new school year is like a relay race in which primary teachers hand off their pupils to secondary colleagues, who must quickly assess their new cohort and guide them into this new world. But as any seasoned teacher knows (and as early-career teachers will discover), this transition is often fraught with challenges, particularly in computing. So how can we ensure that students move smoothly from primary (ages 4–11) to secondary (ages 11–16) computing education? Here are some common problems, and strategies to help bridge the gap.

'I haven't learnt this before!'

Secondary schools often welcome students from a variety of primary schools, each with its own approach to teaching computing. Naturally, this leads to a mix of knowledge and skills. While this is true for many subjects, the impact seems particularly pronounced in computing, arguably due to the subject's formal introduction only in 2014. Many primary teachers have been working diligently to upskill pupils and embed the Computing at School curriculum, and more recently, integrate resources from the National Centre for Computing Education (**NCCE.io**). As many primary schools have switched on to the computing curriculum, this has without doubt shaped the declarative and procedural knowledge being taught to pupils in secondary schools.

In my view, it is crucial for secondary-school teachers to connect with their primary colleagues. This connection allows them to gain insights into the curriculum that has been taught and identify any gaps or variations that need addressing. Similarly, primary teachers should reach out to secondary specialists, who can offer valuable support with curriculum planning, delivery, and assessment. While time constraints are a reality, a simple termly meeting, or even an email introduction, can lay the groundwork for a mutually beneficial partnership. The senior leadership team at my school have ring-fenced time for this process and facilitated such meetings across multiple subjects. Having a specific focus for each meeting helps with preparation and can be built into a longer-term plan. When evaluating the impact, there are also wider benefits for the whole school to be claimed in developing good relationships with local schools.

'I can't remember my password!'

Survey a group of secondary computing teachers, and they'll tell you that one of the biggest hurdles in the first weeks of school is getting students to remember their login credentials. With the decline of homework planners and jotters, pupils are now expected to memorise usernames and passwords, on top of everything else they're learning. To ease this burden (and preserve your sanity), consider creating memory aids that balance convenience with cybersecurity.

I've created personalised support sheets for each pupil, including their username and a password hint space for them to complete. This simple tool has saved countless hours and a lot of frustration. I still anticipate some problems, but having most of the class logging in allows me to focus on the few who require further support.

A straightforward mail merge using the network manager's list of usernames can create these support sheets, which should also include clear instructions for logging in and troubleshooting. Encourage pupils to bring this sheet to every computing lesson for the first few weeks. Additionally, being upfront about the school's password policy can help minimise issues as students settle in. Some schools insist on complex password criteria, and this can confuse pupils when it 'doesn't work'.

Schools often have their own methods of generating usernames, and may either omit or include hyphens in names. There will also be procedures for managing two or more pupils with the same name. It is important for you to be aware of these issues, as they can cause problems. Speak to your network manager and establish a clear understanding so that you are best equipped to support pupils.

'The computer won't turn on!'

Schools across the country are constantly battling for space, and traditional desktop set-ups are being replaced with sleek tablets to free up precious classroom real estate. While this makes practical sense, the consequences of pupils' unfamiliarity with desktop machines become all too clear on that first day in the secondary-school computer lab.

Many colleagues have witnessed pupils looking blankly at something they incorrectly describe as a laptop (the monitor). This is closely followed by the words, 'How do you turn this thing on?' This is a stark reminder that we can't assume anything; it is often critical to ensure pupils have the basic skills they need when using the tools in your computer room.

This lack of knowledge in working with a traditional desktop computer is due to the lack of prior experiences. Tablets are fantastic; they offer access to myriad apps and can connect to physical computing resources via Bluetooth. However, they don't teach fundamental skills such as using a mouse, typing on a physical keyboard, or even the simple act of turning on a computer. During those early weeks, take the time to teach students 'your way'. Focus on the basics — logging in, logging out, and yes, even turning the computer on. Some pupils might never have used a desktop before, and that's OK — part of our role is to guide them through this learning curve.

For schools adopting a one-to-one approach with laptops or tablets, 'your way' might look different, but the goal remains the same: enhancing digital literacy from the start will pay dividends in the long run.

> WE CAN HELP STUDENTS NOT JUST SURVIVE BUT THRIVE IN THE DIGITAL AGE

Building a foundation for the future

As pupils transition from primary to secondary school, the shift in computing education requires thoughtful planning, communication, and creativity. From understanding the varied computing backgrounds of your new cohort to ensuring they can remember their passwords (and turn on a computer), these early efforts lay the foundation for a successful computing journey. By emphasising the real-world relevance of computing and making the subject accessible and engaging, we can help students not just survive but thrive in the digital age.

Ultimately, the goal is to ensure that students don't just learn 'your way', but that they're equipped for a future where technology is integral to everything they do. Consider starting with one or two of the strategies discussed here, whether it be reaching out to primary colleagues or developing the curriculum to suit the needs of all learners. By taking these steps, an effective and streamlined transition can take place; you're building the digital architects of tomorrow. With the right guidance, we can bridge the digital divide and set our students on a path to success in a tech-driven world. (HW)

ROB ARNOLD
Rob is a head of department at a secondary school in Lancashire, UK. In addition to this role, he works with local primary schools to support colleagues and develop the primary computing curriculum and assessment. He has previously also worked as a facilitator for the Barefoot computing programme.

EXPLORING THE CONNECTION BETWEEN DESIGN & TECHNOLOGY AND COMPUTING

Sam Lovatt and **Janine Pavlis** share how to enhance learning opportunities in primary education

At our institution, we give our BA Primary Education (with Qualified Teacher Status) students the opportunity to develop as a specialist in one of the primary curriculum areas. This year marked the introduction of the first-ever combined elective module in design and technology (D&T) and computing for our Year 3 undergraduate students. We were delighted to develop and implement this innovative and imaginative module.

How do computing and D&T align?

The primary national curriculum in England for D&T states the importance of children drawing on a broad range of subjects such as computing, while the computing national curriculum explicitly reinforces the subject's links with D&T (both DfE, 2013). The alignment of these two subjects affirms the significance of broader STEM (science, technology, engineering, and maths) goals, with D&T and computing highlighting maths and science in the curriculum. Before planning this module, we understood that we should aim for a strong foundation in these disciplines.

We are aware that the D&T and computing curricula encourage these connections and go as far as suggesting purposeful design in computing and control technology in the study of D&T. However, we see the connections between our subjects at a deeper level; there are connections between the pedagogy of each, and between the desired skills we want students to gain. Computational thinking is a set of skills and approaches to learning that equip pupils with a framework for solving problems (Wing, 2006). Skills such as algorithmic thinking, abstraction, and debugging are promoted through computational thinking. Despite the term 'computational thinking', it is widely argued that the skills promoted can be applied away from technology and computers (Morris et al., 2017). The approaches and concepts in computational thinking clearly link to D&T and the iterative design process (Design and Technology Association).

One approach that best encapsulates this is tinkering. Tinkering is the process of exploring and testing opportunities before final design choices are made (Resnick, 2017). The iterative nature of tinkering promotes the desired skills of the iterative design process. These connections and shared skills across D&T and computing are elements that we celebrate and highlight in our new module.

What learning opportunities did the module provide for our students?

In our module, we wanted to give the students opportunities to develop their subject knowledge, but also to experience the desired pedagogy and subject connections. To enable this, we decided to explore two projects throughout the ten weeks: designing and making a car using Crumble boards, and a textile project designed using CAD. By allowing our students to complete the projects, they were exposed to a range of experiences as learners and could reflect on the connections and skills explored above. They decomposed, tinkered, created, and debugged in an iterative nature as they created, whilst we modelled the desired pedagogy that would allow children to be brave and to tinker and create within the primary classroom. With frequent reflections, the pedagogical choices we made as lecturers were explained to our students. In this way, we are highlighting how iteration and tinkering can be utilised and supported when teaching computing and D&T, and lessons where the subjects collide.

Alongside the projects, we visited the Design Museum in London to reflect on the opportunities a trip like this may give primary-aged pupils. We also invited subject leaders for D&T and computing from two local partnership schools, to share their experiences of leading in these subjects.

Finally, our students had the chance to apply their newly acquired skills and experiences in a partnership school. Working in groups, they designed a one-hour lesson for Year 6 (aged 10–11) children with the specific aim of creating a prototype steady-hand machine using a micro:bit to develop the game. The students were clear in their teaching approach throughout the lesson, making specific vocabulary and skills links between D&T and computing. Providing children with these links allowed them to understand the importance of the subjects working together within a design process.

An important aspect of the students' lessons involved planning for deliberate errors in the steady-hand machine design, challenges that the children had to address. For instance, once the children had set up the game on the micro:bit, they had to figure out how to clear the micro:bit screen to restart the game. This underscored the collaborative and resilient nature of the lesson, with our students encouraging the children to persevere and tinker, as part of a trial-and-error approach. This allowed our students to model the pedagogy they had seen in sessions at university.

Were our intentions noted by the students and partnership school?

Towards the end of our module, our priority was to gauge the students' perceptions regarding its effectiveness in supporting their journey towards becoming teachers. We achieved this by gathering anonymous feedback through an end-of-module questionnaire. The responses we received enabled us to gain deeper insights that will help us to refine our approach in the upcoming academic year:

"I love the practical work within this module. The discussions beforehand help with the understanding of what we are doing and why we are doing it."

"From the module, I will reinforce the importance of tinkering in planning a project, explaining the positive aspect of trial and error; and the importance of making [abstract to concrete]."

"The module gave us all the ability to create a project and then reflect on it. This allowed me to learn things about myself as a trainee (such as the normality of tinkering)."

Our partnership school told us that the session presented an exceptional opportunity for their Year 6 (aged 10–11) children to explore new technology and acquire valuable skills. It was also recognised that our student teachers had made an impact. As a result, inspired by the innovative approaches observed, the school is now planning to integrate micro:bits into its future schemes of work in D&T.

What is next for our subjects and for this module?

We hope this module will continually evolve to reflect the students studying it and new developments in D&T and computing. Through our work with schools, we recognise that the areas of the national curriculum highlighted in this article can be challenging to teach, but we want to alleviate these challenges by prioritising the development of skills among both children and teachers. As an initial teacher education institution, we respond to the needs presented by our partnership schools, and aim to prepare our students to be the best teachers they can be. (HW)

FURTHER READING

Department for Education (2013). The national curriculum in England: key stages 1 and 2 framework document. (**helloworld.cc/england-curriculum**)

Design and Technology Association. The Iterative Design Process (**helloworld.cc/iterative-design**)

Morris, D., Uppal, G., and Wells, D. (2017). Teaching Computational Thinking and Coding in Primary Schools. London: Sage Publications

Resnick, M. (2017) Lifelong Kindergarten: Cultivating Creativity through Projects, Passion, Peers, and Play. London: The MIT Press.

Wing, J. M. (2006) Computational thinking. Communications of the ACM. 49(3) pp.33–35.

JANINE PAVLIS

Janine is a lecturer in primary education (initial teacher education), design and technology at St Mary's University, Twickenham, London.

SAM LOVATT

Sam is a lecturer in primary education (initial teacher education), computing at St Mary's University, Twickenham, London.

CONNECTING WITH MICRO:PEGS

Graham Hastings offers a solution to tricky croc clip circuits

A requirement to teach children about physical computing is written into the English national curriculum programmes of study for Key Stages 2 and 3 (ages 7–14):

- **Key Stage 2:** design, write, and debug programs that accomplish specific goals, including controlling or simulating physical systems

- **Key Stage 3:** design, use, and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems

The creation of excellent hardware, such as the micro:bit and the Crumble controller, specifically designed for the teaching of physical computing within Key Stage 2, has dramatically shifted the landscape. This hardware, along with the associated online resources, has removed many of the historic barriers to teaching physical computing.

That said, I have encountered one very specific problem when setting activities for my pupils involving the creation of circuits to demonstrate control technology and for data logging. These require components and wires to be temporarily connected to the 4mm terminals on the devices. The standard method of doing this is through the use of short wires with a crocodile clip at each end.

Figure 1

The problem with croc clips

This method does work, as illustrated in **Figure 1**, but when using croc clips in my lessons, I have experienced the following issues:

1. If not connected at right angles to the board, the clips slide along the edge strip, causing short circuits, or the clips slip off completely.
2. Children can find it difficult to open the clips far enough to connect them at right angles to the board, as recommended.
3. Additional connections between the component and croc clip can result in short circuits and connection failure.
4. The extra connections add to the lengths of wiring, making the circuit more difficult to understand and follow.
5. On occasions, several components need to share a common GND terminal, as with the traffic light model in **Figure 2**. This can be difficult to do using several crocodile clips.

Having recently gained access to a 3D printer, I thought it might be possible to design and manufacture an alternative method of connecting wires to the 4mm terminals, and came up with the micro:peg.

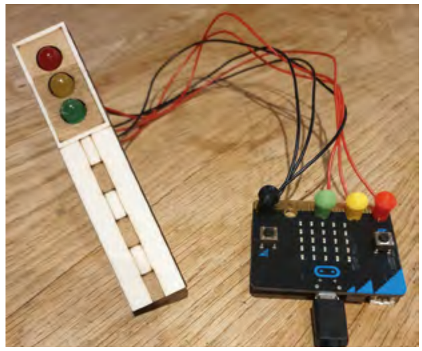
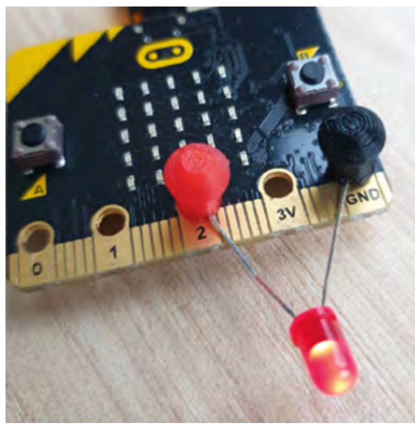


Figure 2

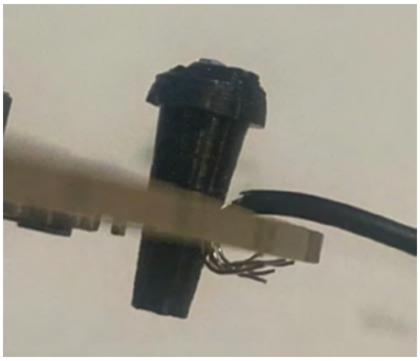


An LED connected to a micro:bit using two micro:pegs

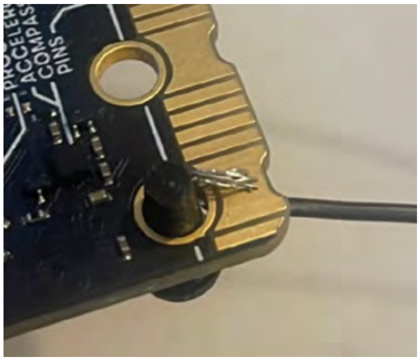
How do micro:pegs work?

About a centimetre of insulation is stripped from the end of the wire to be connected. The bare end of the wire is pushed down though the 4mm terminal hole and folded back on itself. A micro:peg is then pushed into the hole to hold the wire firmly in place.

Multi-strand wire works best, but for single-strand wire, or when connecting unwired components, the groove in the Mark 1 micro:peg (see box) can be lined up



A micro:peg holds the bare end of a wire firmly in place



Push the micro:peg from the back of the circuit board to remove

with the wire to make space for it next to the peg. Removing the micro:peg releases the wire.

The Mark 1 pegs are secure, and good for semi-permanent connections. Because they are a little fiddly to use, I also created a simpler peg, the Mark 2. These work best for a quick, simple, temporary connection.

Removing the micro:peg releases the wire. The Mark 2 micro:pegs are simply popped out by pushing on them from the back of the circuit board.

I have used micro:pegs extensively with children in Years 3 to 6 (aged 7–11). With a little bit of practice, they found it much easier to attach and detach wires from the 4mm terminals using the pegs than they did using crocodile clips.

What advantages do micro:pegs bring?

Because the connections made with micro:pegs are more secure than crocodile clips, the teacher can make up the circuits before the lesson and hand them out to the children to use.

If the children start to make a circuit but do not finish in time, their circuits can be collected and handed back to be completed during a later lesson.

Other advantages gained from using the pegs are as follows:

1. The connections are secure and do not come undone accidentally.
2. Pegs cannot slide along the edge strip.
3. The peg connections do not cause short circuits.
4. It is possible to match the colour of the peg to the colour of the wire from the component, so children can build their circuits by following a simple picture.
5. 'Bare' components such as a single LED and wired components can be connected directly to the 4mm terminals.
6. Up to three wires can be connected to the same terminal.
7. Once made, the circuit is much neater.
8. The simplified wiring is easier for the children to follow and understand. (HW)

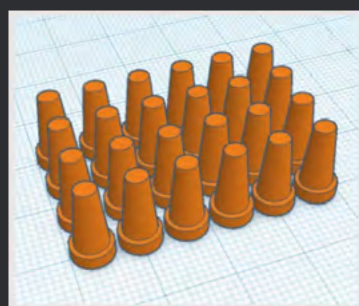
GRAHAM HASTINGS

Graham Hastings is head of computing at St John's College School, Cambridge, UK.

MAKING MICRO:PEGS

I have made two Tinkercad files available for export as .stl files ready for 3D printing.

Mark 1 Use for a more secure, semi-permanent connection: **helloworld.cc/micropegs-Mark1**.



Mark 2 Use for quicker, more temporary connection: **helloworld.cc/micropegs-Mark2**.

Scan the QR code to access the micro:peg website.

CODING WITH ELAN

Richard Pawson and team have created the first new programming language for British schools since BBC Basic. He explains how — and why — Elan differs from existing languages

The name Elan is a contraction of 'educational language'. It is also an English word meaning style, flair, or panache. This is convenient, because Elan isn't designed just to support the teaching of programming; it is designed to encourage the teaching of good programming style from the outset. Existing languages do the former, but most of them do nothing to support the latter, with many actively encouraging poor design and coding practices.

RICHARD PAWSON FBCS

Since starting at Commodore in 1977, Richard has spent his career in the computing industry, gaining a PhD in computer science along the way. Since 2016 he has devoted all his time to writing free resources designed to help advance the art of teaching computer science in schools. Feel free to contact him at **rpawson@nakedobjects.org**

The first time you see a program written in Elan, you might get a shock — the shock being that it isn't very shocking! A simple Elan program looks remarkably like pseudocode. If you have taught any text-based programming language, you should have no difficulty reading what the code does. We've favoured keywords over punctuation symbols, in part to make the code easier to verbalise in the classroom. You might object that this implies more typing, but it doesn't: Elan requires less typing than most existing languages, because it uses an integrated 'frame-based' editor.

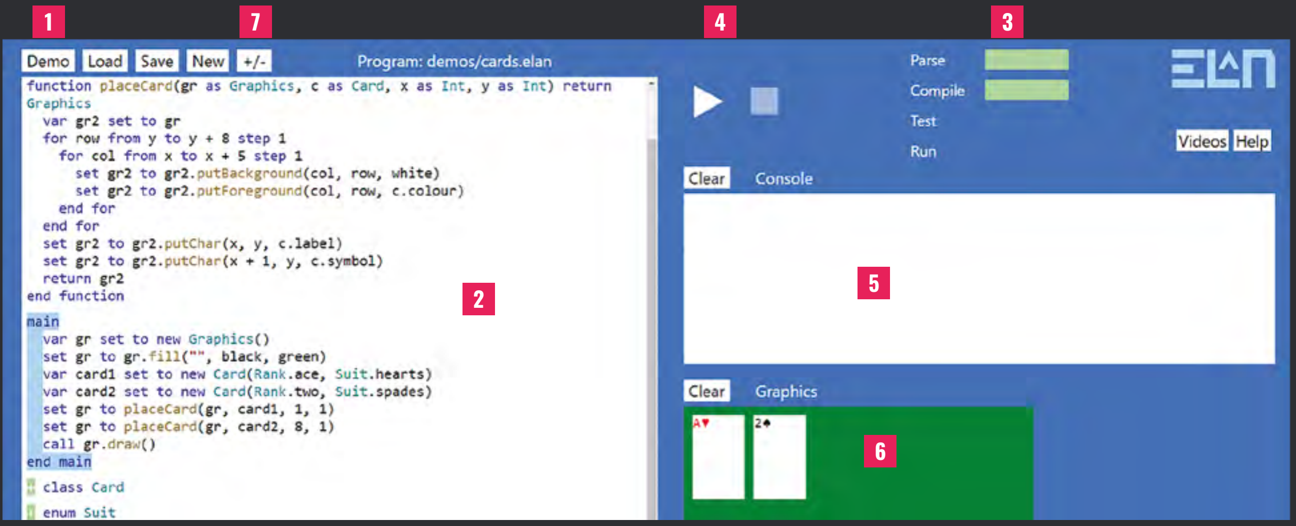
Driven in part by the recent consultation document from the Department of Education, there is currently a lot of debate about block-based versus text-based languages, and whether GCSEs should be changed to accommodate the former. Frame-based editing breaks that dichotomy. Other frame-based editors such as Stride (**stride-lang.net**) and Strype (**strype.org**) have been positioned as stepping stones between the two. Elan goes further: it reads exactly like a pure text-based language, but when entering or editing code, it feels (though does not look) like a block based-language. It is still possible to enter non-parsing code into a specific field, but the field background will change to red (see page 64) at the first wrong keystroke, and the location of the error is immediately apparent. However, in Elan you simply cannot create those all-too-common syntax errors where the point at which the error is identified may be far from the root cause.

As well as eliminating most forms of syntax error, and reducing the amount of typing needed, the frame-based approach helps in another way. Wherever you can insert a new line or multi-line construct, you are prompted with a list of the possibilities (see page 65) and you can typically select the one required by typing just the first letter (or in a few cases two letters). Teachers who have used prototypes of Elan reported that they thought this feature would also help overcome 'the fear of the blank screen'.

If this is beginning to sound like an AI-driven co-pilot, it isn't. Elan uses no AI, nor will it do so. The IDE is never guessing what the user is trying to write; it is simply applying the grammatical rules of the language. The fact that the language is tightly integrated with the IDE means that teachers and exam boards can be confident about which capabilities are available — and which are not available — to pupils programming in Elan. This integration has also made possible a unique capability: built-in anti-plagiarism (see page 65).

As well as making it easier to create the code in the first place, the Elan grammar makes it impossible to create specific common anti-patterns, for example breaking out of a loop, or returning early from a subroutine — patterns that predate 'structured programming'. Another such anti-pattern is the 'hanging function call': you might never have heard this term, but it occurs whenever your pupils write **name.upper()** or **name.ToUpper()** thinking that it will change the case of the text in the name, ▶

THE ELAN IDE



1. The beta (**elan-lang.org/beta**) includes several demo programs that you can run and modify. If you have programmed in another language, this is a great way to get started. Demos can be removed by editing the profile (see separate panel), if you don't want your pupils playing Snake all lesson!
2. The Code window has a built-in frame-based editor.
3. An overall status bar shows whether the code is parsing and compiling, whether unit tests (if written) are passing, and, if the program is run, whether it is actively running, paused, or has thrown a runtime error.
4. Simple run controls fade out when not applicable. Debugging controls will be added here soon.
5. Programs can **print** to the console, and there are several **input** methods for user input via the keyboard.
6. The integrated character-mapped graphics display is far simpler to program than most modern GUI frameworks, and is ideal for writing simple retro-style games and simulations.
7. By clicking on the +/- button, the user can show the whole code file collapsed to a summary view (shown below). This is helpful both for navigating around a large program, and also — from an educational perspective — for understanding the overall structure of a program. If the code is not parsing, or not compiling, or contains tests that are not passing, the + sign for the code region that contains the error will be rendered with an amber or red background. The user may expand the whole program, or any individual frame, by double-clicking on the line.

```
main
  print "Press any key to halt"
  var gr set to new Graphics()
  set gr to gr.fill("", black, white)
  var grid set to empty [Boolean]
  call fillRandom(grid)
  var cont set to true
  + while cont
end main
+ procedure fillRandom(starter as [Boolean])
+ function updateGraphics(grid as [Boolean], gr as Graphics) return Graphics
+ function n(c as Int) return Int
+ function s(c as Int) return Int
+ function e(c as Int) return Int
+ function w(c as Int) return Int
+ function neighbourCells(c as Int) return [Int]
+ function liveNeighbours(cells as [Boolean], c as Int) return Int
+ function willLive(currentlyAlive as Boolean, liveNeighbours as Int) return Boolean
+ function nextCellValue(cells as [Boolean], c as Int) return Boolean
```

NAVIGATING AND EDITING CODE

Elan uses a frame-based editor. By clicking on the **for** keyword, the user selects the whole of the **for** loop frame:

```
procedure fillRandom(starter as [Boolean])
  for i from 0 to 1199 step 1
    call starter.add(random() > 0.5)
  end for
end procedure
```

The whole frame (including the statements contained within it), may be moved up or down, cut and pasted, or deleted. But the user cannot damage the integrity of its syntax. For example they cannot edit, delete, or move, any keyword within the highlighted area. Frames may be multi-line or single line. The user may select any frame by clicking on its starting keyword, or by navigating to it via the cursor keys:

```
procedure fillRandom(starter as [Boolean])
  for i from 0 to 1199 step 1
    call starter.add(random() > 0.5)
  end for
end procedure
```

Fields are the parts of a frame intended to be edited by the user. Here the user has selected a field (the contents having been deleted). The grey background (all colours are configurable) indicates that this field is optional, and the white text provides a prompt for what is required. Clicking on the **?** will offer additional help on what is required:

```
procedure fillRandom(parameter definitions?)
  for i from 0 to 1199 step 1
    call starter.add(random() > 0.5)
  end for
end procedure
```

Now the user has entered the name for a parameter, but the field is amber, indicating that it is incomplete — and the white text indicates that the user must specify a type for the parameter:

```
procedure fillRandom(starter as Type?)
  for i from 0 to 1199 step 1
    call starter.add(random() > 0.5)
  end for
end procedure
```

Specifying that the type is a list of **Boolean** values, the field is now in a valid state:

```
procedure fillRandom(starter as [Boolean]?)
  for i from 0 to 1199 step 1
    call starter.add(random() > 0.5)
  end for
end procedure
```

If the user types any character that would be invalid, the field background immediately goes red. For example, all Types in Elan start with a capital letter, but here the user has typed a lower-case **b**:

```
procedure fillRandom(starter as [b?)
  for i from 0 to 1199 step 1
    call starter.add(random() > 0.5)
  end for
end procedure
```

when in fact it returns a new string, leaving the original one unchanged.

Elan is also the first multi-paradigm language intended for use in schools that enforces a proper distinction between a **function** and a **procedure**. In most languages, the only difference (if any) is that a function returns a value; a procedure doesn't. Properly, a function should behave like an operator — like +, -, *, or /. The value returned by a function should depend upon nothing except the arguments provided, and should leave no side effects. By contrast, all current school languages wrongly permit the pupil to create side effects (e.g. printing to the screen) or create external dependencies (such as getting input from the keyboard) within a function.

Is this just splitting hairs, or does the distinction matter? One reason it matters is that if you got the right idea about functions at the start of programming, you would find it much easier to transition to functional programming languages (becoming increasingly dominant in the professional programming world) at a later stage. As a teacher you might, rightly, protest that most computer science pupils don't intend to become professional programmers, and that's not why we teach the subject anyway. But making this distinction clear from the outset would help all pupils — including those with no intention of taking the subject further — to write better code and spend less time chasing down obscure bugs. Because proper functions can always be safely combined to make more advanced functions and programs, if function **foo** is correct, and function **bar** is correct, then calling **foo** on the results of **bar** is guaranteed to be correct. This is simply not true for combining procedures, because their implementations may interfere with each other.

And how can a pupil know that the **foo** and **bar** functions are themselves correct? By testing them. The GCSE and A-level specifications require that we teach pupils to test code, but that is typically done by writing, running, and testing the program manually. We are also required to teach them to refine their code, but manual testing is unreliable and tedious to repeat, and refinement runs the risk of breaking your code again. This gives rise to the common aphorism, 'If it ain't broke, don't fix it!', but that is the antithesis of good programming. Professional programmers solve this conundrum by writing automated unit tests for every function, which can be repeatedly executed without effort. Unit tests aren't hard to write in any language, but they are unnecessarily fiddly. In Elan we have built unit tests into the language as first-class features and made them so easy to write that it now becomes viable, even attractive, to teach pupils to write automated tests from the start of programming (see page 66).

Unit tests are like seat belts — they make coding safer. In the 1960s, few drivers or passengers wore seatbelts. Concerted efforts in the 1970s — by governments to raise awareness of the dangers, and by car manufacturers to make seat belts easier to fit and more comfortable to wear — meant that by the early 1980s most new young drivers always wore their seatbelts, even before it became a legal requirement. Wouldn't it be great if we could do the same for the automated testing of code?

In many ways, then, Elan is a forward-looking language. In just one respect it is deliberately retrospective: graphics. I hear many complaints from teachers about how hard it is to get pupils started with

>

ENTERING NEW CODE

Staring from a **New** file, the user is prompted with all the types of frame that they may create at file (or 'global') level:

```
# Elan Beta 1
main procedure function constant test #
```

In most cases, typing just a single letter will identify the frame to be added (in some cases two characters are needed to disambiguate options). Here the user has typed **m**, and the editor has inserted a **main** frame:

```
# Elan Beta 1
main
  call each for if print repeat set switch throw try var while #
end main
```

Within the main a new 'selector' has appeared showing which types of frame may be entered at this location — all of them are statements (some single-line, some multi-line frames). The user now types **v** to define a variable, using a **var** statement:

```
# Elan Beta 1
main
  var name? set to expression
  new code
end main
```

The frame for the **var** statement has been inserted with all the required keywords (and in some cases additional punctuation), and also indicating the field(s) that the user must (amber) or may (grey) define. (See separate panel on **Navigating and editing code**.)

Notice the grey **new code** underneath the active line. When the use hits **Enter** on the last field in the **var** statement, focus will move to the line underneath, which will now show the frames that may be inserted there.

CUSTOM PROFILES & ANTI-PLAGIARISM OPTION

By editing simple JSON configuration files, a teacher may choose which capabilities to make available to specific year groups, sets, or individual pupils. This includes the ability to specify language features. Here, the teacher has written a profile for their Year 10 pupils and has chosen not to include **class** and **enum** at global level, or **try** and **throw** at statement level:

```
{
"name": "Year10",

"globals":["main", "procedure", "function", "constant",
"test", "#"],
"statements":["assert", "call", "each", "else", "for",
"if", "let", "print", "repeat", "set", "switch", "var",
"while", "#"],
```

The result is that when their Year 10 pupils focus on the corresponding 'selector' they are not given the option to add those types of frame to their code. For example, the global selector now just shows:

```
# Elan Beta 1
main procedure function constant test #
```

Other options in the config files include a unique 'anti-plagiarism' mechanism. If this option is specified by the teacher, the user may load only Elan code files that they have authored themselves — or files authored by named teachers:

```
"include_profile_name_in_header": true,
"anti_plagiarism_option": true,
"can_always_load_files_created_by": ["MsBrown", "MrGreen"],
"show_demo_button": false
```

The reason that this works is that every Elan code file (which is stored as plain text) contains a hash of the code, including the author's name and profile. The Elan editor and run environment will not load any file that does not have this hash, or where the hash does not match the code.

AUTOMATED TESTING

In Elan, unit testing is a first-class feature of the language, with very clean and simple syntax. Tests may be written anywhere within the code — for example, adjacent to the function they are testing. A **test** may optionally be given a name, but this is not required (because thinking of a unique, descriptive name for each test can be difficult).

Here, the user has written tests for a function **pythag** to calculate the hypotenuse of a right-angled triangle before even defining the function (this is an optional approach known as 'test-first'). The tests all parse, but the amber background and (white) message indicate that they do not yet compile because **pythag is not defined**:

```
test
  assert pythag(3, 4) is 5
  assert pythag(0, 0) is 0
  assert pythag(-5, -12) is 13
  assert round(pythag(3.1, 4.2), 2) is 5.22
end test
```

Now the user has defined a 'mock-up' version of the **pythag** function, which always returns **0**. This means that the code — though not yet correct — now compiles. And because it compiles, the tests are run automatically and the results displayed alongside each **assert** statement:

```
test
  assert pythag(3, 4) is 5 actual: 0
  assert pythag(0, 0) is 0 pass
  assert pythag(-5, -12) is 13 actual: 0
  assert round(pythag(3.1, 4.2), 2) is 5.22 actual: 0
end test

function pythag(b as Float, c as Float) return Float
  return 0
end function
```

Unlike in most unit test frameworks, Elan does not halt at testing at the first **assert** failure. This means that it now often makes sense to test multiple cases within a single test frame — which saves both code and screen space. When the user attempts to write a proper implementation of the **pythag** function, the tests are automatically rerun each time the code compiles, and are now all passing:

```
test
  assert pythag(3, 4) is 5 pass
  assert pythag(0, 0) is 0 pass
  assert pythag(-5, -12) is 13 pass
  assert round(pythag(3.1, 4.2), 2) is 5.22 pass
end test

function pythag(b as Float, c as Float) return Float
  return sqrt(b^2 + c^2)
end function
```

This instantaneous feedback encourages pupils to experiment with refining their implementation without the fear that they might have unwittingly broken a specific case. Additionally, if the user saves any code file containing any tests, the tests will automatically be rerun whenever the code is loaded — giving the user the instant reassurance that the code they worked on last lesson was left in a good state (or, if not, that they should fix the code before starting on further work).

GUIs, and even when they get the hang of it, they learn very little programming in the process. Elan emulates the simpler character mapped displays of the 8-bit micros of the 1980s. Writing retro games such as Snake, Blackjack, or Tetris, is not only straightforward with this approach; you learn far more programming in the process. The graphics aren't as glitzy as modern offerings, but the games and other simulations are no less engaging. The Elan Beta has a few built-in demos of this.

In conclusion:

- Elan and all associated resources are completely free and open source.
- All colours are configurable for users with CVD (colour vision deficiency).
- The language and IDE run entirely within the Chrome browser.
- Elan is currently at beta stage. We are now actively seeking feedback on the beta from teachers. A teach-ready v1.0 with supporting resources will be released in early summer 2025, ready for teaching from September 2025.
- We have engaged in discussions with OCR, AQA, Edexcel, and Eduqas throughout the development, as well as other significant stakeholders. Feedback has been almost universally positive, and we hope that by release v1.0 we will be able to announce more concrete commitments.
- You will have the option to run Elan remotely or install it on your school server (as a simple web page) or even to run it as a stand-alone HTML page on your desktop with no server access.
- Elan is a compiled language. It will be possible from v1.0 to make calls to third-party JavaScript libraries from within Elan code.
- Elan is committed to remaining focused on the needs of teaching programming in school. By next summer we will have established a governing body that clearly represents and safeguards those interests.

Please try out the Beta (**elan-lang.org/beta**) and be sure to send me your feedback. (HW)

Algorithm design is a critical component of mastering computational thinking because it teaches students how to break down complex problems into smaller, more manageable parts and design solutions for each part.

ALGORITHM DESIGN

A STEP-BY-STEP APPROACH

DESIGNED BY FAHEEMAH VACHHIAT. FVachhiat

STEP 1

INTRODUCE THE CONCEPT

Provide a clear definition of what an algorithm is and why it is important in computing and physical programming. Use visual aids, such as diagrams or infographics, to help students understand the concept of algorithm design in an engaging way. This can help students develop a deeper understanding of the concept and its practical applications.

STEP 2

TEACH PROBLEM-SOLVING SKILLS

Teach students how to break down problems into smaller parts and identify the inputs and outputs required for each part. Use of visual aids, such as flowcharts or mind maps, allows students to visualise the problem-solving process and break it down into smaller parts.

STEP 3

MODEL SEQUENCING

Teach students how to sequence the steps in an algorithm to ensure that they are completed in the correct order. Model this process using guides, such as flowcharts or diagrams, to help students organise the steps and visualise the sequence of the algorithm.

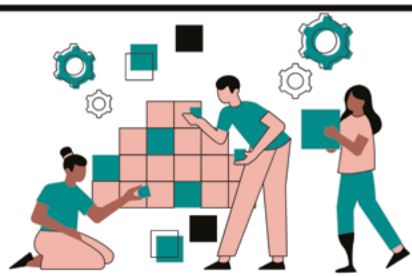
STEP 4

INTRODUCE LOOPS AND CONDITIONALS

Explain the concepts of loops and conditionals and how they can be used in algorithm design to solve more complex problems. Using interactive simulations or games will help students understand these concepts in a hands-on and engaging way.

STEP 5 PROVIDE HANDS-ON ACTIVITIES

Provide students with opportunities to apply their algorithm design skills in both computing and physical programming contexts, through hands-on activities using tools like block-based programming platforms or programmable robots. Use visual aids, scaffolds, guides and live modelling to help students understand how to use these tools and apply their algorithm design skills in a practical and engaging way.



USE PHYSICAL TOOLS

Use physical programming tools to teach algorithm design in a tangible way. Begin with simple activities that involve programming a robot to move in a straight line or turn 90 degrees. This can help students understand the basics of algorithm design.

ENCOURAGE EXPERIMENTATION

Encourage students to experiment with different programming concepts, such as loops and conditionals, and allow students to explore the effect of these concepts.

ENCOURAGE COLLABORATIVE LEARNING

Collaborative learning such as group problem-solving activities, peer review of algorithms, and co-design of algorithms with a partner can be particularly effective because it allows students to work through complex problems together and share their approaches and solutions.

HOME-HOSTING A LEARNING MANAGEMENT SYSTEM USING RASPBERRY PI

Riyad Dhuny shares his case study of running an LMS at tertiary education level

Are you an educator or academic who hasn't yet embraced a learning management system (LMS) for classroom management? If so, you're missing out on an essential tool that can significantly streamline your daily work. However, setting up and hosting an LMS comes with its own set of costs and challenges.

ENHANCING YOUR LMS WITH PLUG-INS

Moodle's extensibility through plug-ins significantly enhances classroom management capabilities. The Oauth2 plug-in simplifies access by allowing students to sign in using their Google, Microsoft, or Facebook accounts. H5P transforms content delivery by enabling interactive elements on top of YouTube videos, increasing student engagement. The attendance plug-in streamlines record-keeping by facilitating student self-check-ins during sessions. Additionally, integration plug-ins for services like Google Meet, Zoom, and Turnitin seamlessly incorporate essential third-party tools into the daily teaching workflow. These plug-ins exemplify how a home-hosted LMS can be tailored to meet specific educational needs, enhancing both teaching efficiency and student engagement.

A cost-effective solution: Raspberry Pi

In Mauritius, where I live, a basic virtual private server (VPS) package, which rents computing hardware in a data centre, typically costs around US$ 5.50 per month. This includes four virtual CPU cores, 4–6GB of RAM, and 100GB of NVMe disk storage. Interestingly, these specifications can easily be met by a Raspberry Pi.

This raises an intriguing question: is it possible to host an open-source and free LMS application like Moodle in a production environment using a Raspberry Pi? Given that Raspberry Pis cost between US$ 35 and $70, you could break even on the initial investment in six months to a year, provided that the hardware operates continuously for that period.

While these devices typically need to remain connected online for availability purposes (usually requiring rented computing from a data centre), many areas now have readily available broadband internet connectivity. This means you can home-host an application at no extra cost, save for perhaps purchasing a domain name.

Case study: a four-year experiment

To test this idea's feasibility, I set up a Moodle LMS on a Raspberry Pi 4 with 4GB of memory in April 2020. The system was designed to support specific cohorts for my modules at the University of Technology, Mauritius. On average, the cohorts contained 15 students, with a maximum of 28 accessing the system concurrently.

Impressively, at the time of writing, the system had been running for over four years and five months. The underlying database now houses more than 450 tables with over 750,000 records and more than 2,200 submitted assignments. Most importantly, regular feedback from students throughout the experiment revealed no major issues, incidents, or perceived delays in system operation.

Performance evaluation: how many students can one Raspberry Pi support?

Among the concerns at the project's outset was the number of users a small device could support simultaneously, and the resulting user experience. Would the responses be too slow, leading to student frustration?

To address these questions, we conducted a performance evaluation of our Raspberry Pi 4 in a laboratory set-up. Our findings showed that our Raspberry Pi could support up to 30 users for a medium-sized Moodle course running from an SD card. These experiments were part of the RPI64Box project, with detailed results and disk images available at **rpi64box.com**.

Another crucial factor affecting user capacity is the bandwidth limit. Home-hosting solutions depend on the available internet broadband package, particularly the upload speed. We achieved satisfactory results for a small-sized course with a connection as low as 8 Mbps upload speed. In many areas of the world, internet service providers (ISPs) offer much higher bandwidths than this, making this solution increasingly viable.

Portability: the RPI64Box solution

Setting up an LMS like Moodle for testing and trials can be daunting for some. The RPI64Box project aims to simplify this process, supporting quality education under the UN's Sustainable Development Goals (SDGs).

RPI64Box makes it easy for educators and first-time users to try Moodle, and works well with a battery-powered casing. To get started, simply:

1. Visit the RPI64Box website. (**rpi64box.com**)
2. Download the RPI64Box image.
3. Transfer it to an SD card using Pi Imager or balenaEtcher.
4. Boot a Raspberry Pi (preferably version 4) with the image.

You'll then have a Moodle instance accessible via a new Wi-Fi hotspot labelled 'RPI64Box' (password: 'rpi64box'). Once connected, access the LMS at rpi64box.home.

Advantages of home-hosting solutions

Home hosting offers several advantages:

1. Assignments are delivered next to your router, streamlining the grading process.
2. Unlike institutional LMS deployments or standard Moodle tier plans, there are no limitations on integrating third-party plug-ins, or accessing the site administration interface.
3. Greater flexibility and control over your LMS environment.

> THIS SOLUTION PROVIDES VALUABLE LEARNING OPPORTUNITIES IN NETWORK CONFIGURATION AND SERVER MANAGEMENT

Setting up your own home-hosting solution

Getting Moodle to run on a Raspberry Pi is relatively straightforward. The OS requires the installation of a web, database, and application server, which can be done by installing the Linux, Apache, MySQL, and PHP (LAMP) stack.

The main challenge is making the web server accessible to the outside world. Here's a basic overview of the process:

1. The web server becomes accessible via the router's IP address.
2. Set up port forwarding for ports 443 and 80 on the router to the internal IP address of the home-hosted Raspberry Pi.
3. Purchase a domain from a dynamic domain name service (DDNS) provider (such as No-IP, Dynu, DynDNS, GnuDIP, or easyDNS).
4. Configure DDNS mapping in the router to update necessary information when the home router's IP changes.

By following these steps, you can create your own cost-effective, flexible LMS solution using a Raspberry Pi. This approach not only saves money, but also provides valuable learning opportunities in network configuration and server management.

As the world of education continues to evolve, innovative solutions like this home-hosted LMS demonstrate the potential for accessible, affordable technology to enhance learning experiences. Whether you're an individual educator or part of a larger institution, exploring these options can open up new possibilities for engaging with students and managing your educational content.

After years of experience in this field, I wholeheartedly encourage you to explore home-hosting an LMS. It's an adventure that can truly transform your teaching experience. Whether you have questions or ideas to exchange, or are interested in collaborating, don't hesitate to reach out. As you embark on this journey, I hope you'll find it as rewarding and exciting as I have. Here's to enhancing the quality of education, one Raspberry Pi at a time! (HW)

RIYAD DHUNY

Riyad is an academic at the University of Technology, Mauritius, with over a decade of teaching experience. He holds a postgraduate qualification in software engineering. Riyad publishes insights on **dhuny.org** and can be reached at **dhuny@utm.ac.mu**.

CLASSIFICATION WITH DECISION TREES

The 'Good Monkey, Bad Monkey' Game

Annabel Lindner and Stefan Seegerer

AGE RANGE

Target group: primary-school level, secondary-school level

REQUIREMENTS

- Monkey cards (print and cut, or use the digital version) available at **aiunplugged.org**
- Board with magnets or pinboard

CONCEPTS

- AI classifies data based on patterns
- AI uses the classification model that best fits the given data
- Classification models are not perfect
- Certain combinations of characteristics indicate a certain category

How does a computer make decisions independently? How does a computer decide whether a person is athletic, or whether they should be given a loan? Such classification processes are frequent applications of AI. In this activity, students have the opportunity to create their own classification model using a decision tree. In the end, the best of the students' models is selected for further classification tasks. (HW)

HOW THE ACTIVITY WORKS:

Students examine how a series of sample elements (training data) belong to a chosen category. To categorise these elements, students develop criteria that can be used to classify new elements. The resulting models are then tested with new examples (test data) and the accuracy of the prediction is determined.

> STUDENTS CREATE THEIR OWN CLASSIFICATION MODEL USING A DECISION TREE

CONTEXT

We are animal keepers in a zoo, and are responsible for feeding the monkeys. They look very cute, but we have to be careful, because some of the monkeys bite. We already know whether the monkeys who live in the zoo bite. However, new monkeys will be joining the group soon, and we need to consider how to find out which new monkeys bite and which don't — preferably without getting too close to their teeth!

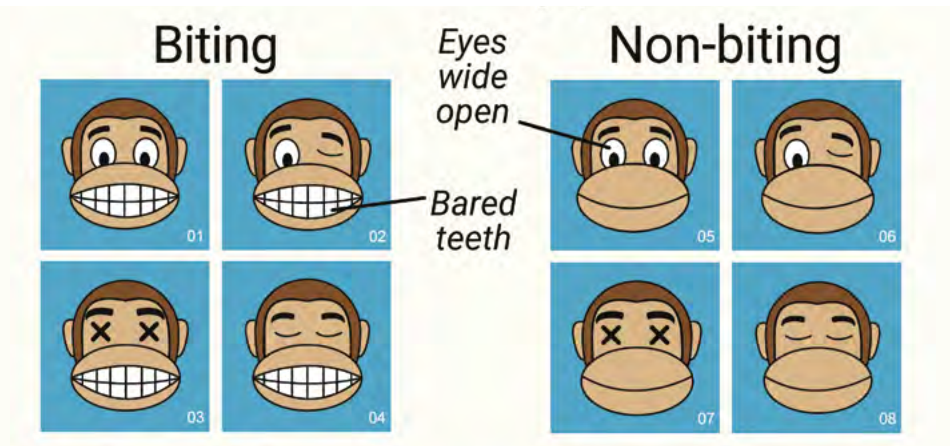
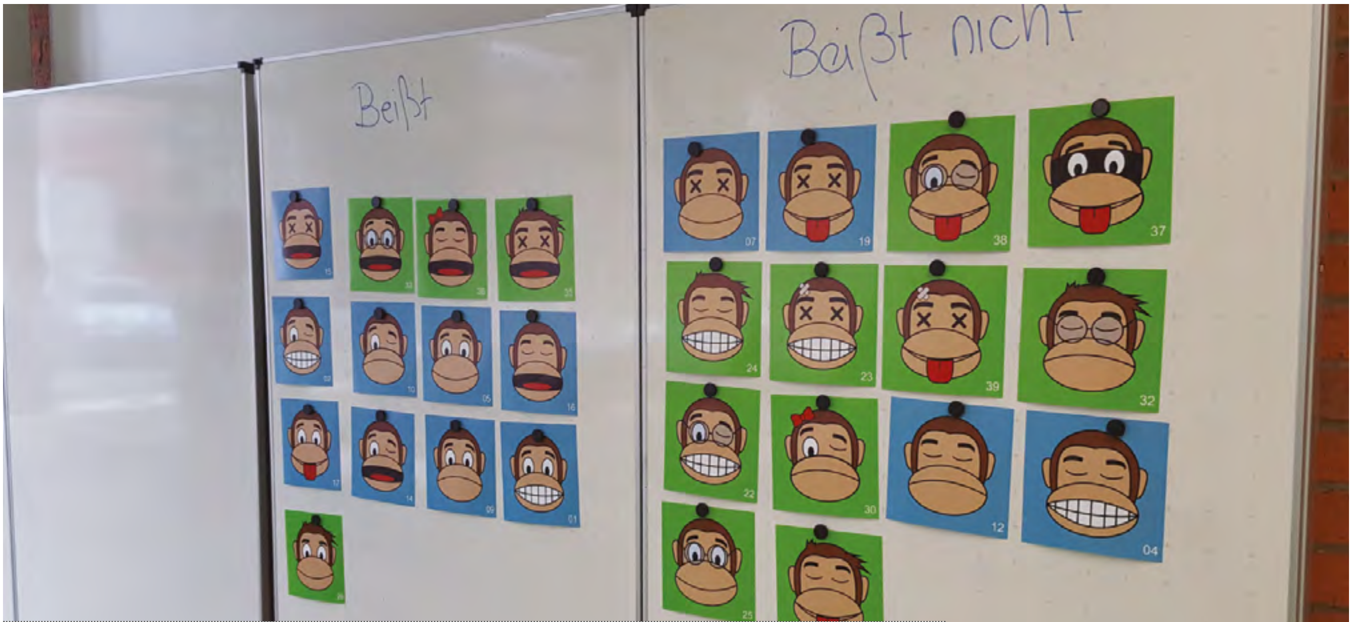


Figure 1 In this simple example, all monkeys with bared teeth bite



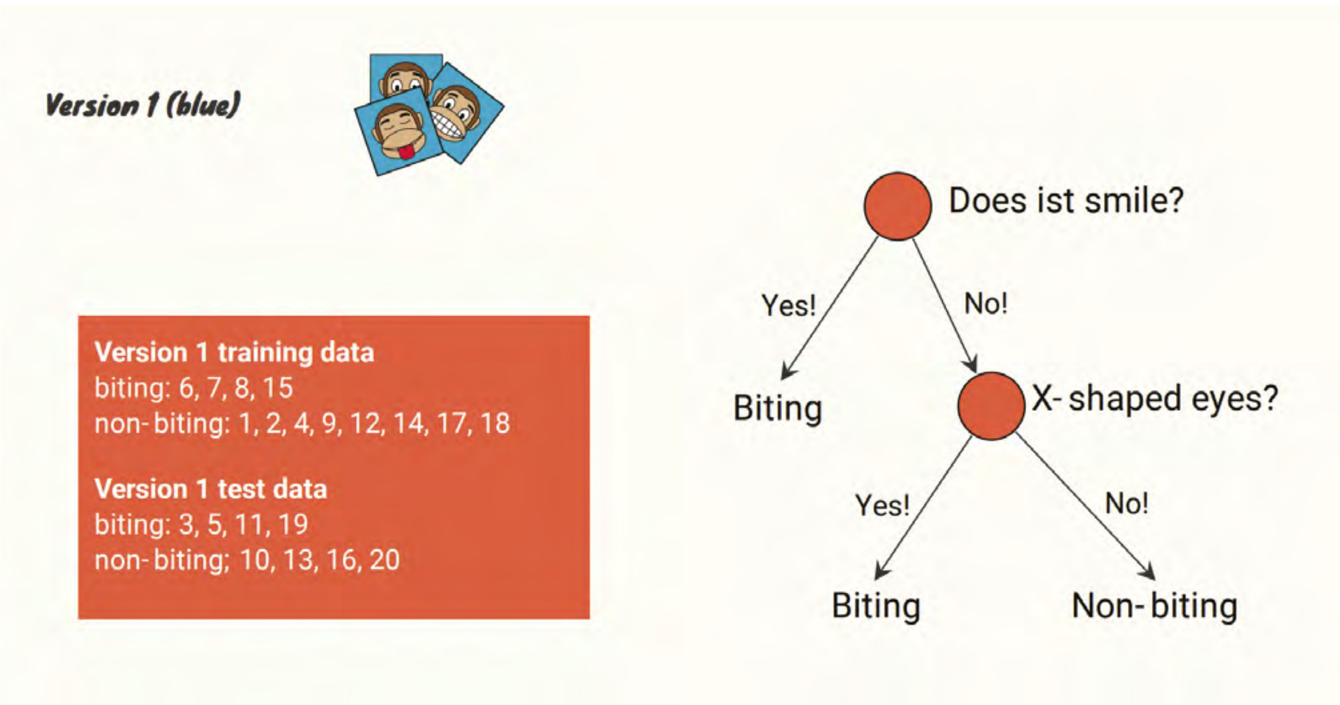
The advanced version uses both blue and green picture cards

ACTIVITY

Depending on the target group, you choose the simple game (Version 1) with 20 blue picture cards or the advanced game (Version 2) with 40 blue and green picture cards. These 20 or 40 monkeys all live in the zoo already, so we already know if they are going to bite. They are split into training and test data. Based on the training data, we think of criteria that determine whether the monkeys bite, and check their reliability based on the test data. The training data, subdivided into the two categories of 'biting' and 'not biting', is pinned on the board. The test data is not revealed at first. You can think of rules by which to distinguish the monkeys yourself, or use one of the proposals below (using reduced subsets is also possible). The rules that apply in the examples are illustrated with decision trees. First, make your students aware of the details they could focus on by illustrating the procedure with an example. For example, compare the monkey cards numbered 01 to 04 and 05 to 08. In this example, the shape of the mouth is an indication for biting monkeys, but the eyes are not (**Figure 1**). With older students, you can also use the simple version of the game (Version 1) to demonstrate the rules and the necessary procedures.

The students form teams of two, and use the training data to develop criteria for distinguishing biting from non-biting monkeys. These must be clearly noted so that they can be applied to new examples by another team afterwards. One way of recording the criteria is a decision tree. It should be the goal that the existence or absence of a particular characteristic permits a clear assignment to one of the groups. The use of decision trees is optional; it is also possible to explicitly write down decision rules. At the end of the training phase, the criteria formulated are exchanged with another team. Now, the students are shown the pictures of the remaining monkeys (the test data), one after the other. For each image, the teams decide whether the monkey will bite or not, using the scheme of rules developed by their classmates. Each team notes down their decisions. After showing all the monkeys, it is evaluated which team has best assessed the biting behaviour of the monkeys. It comes to the students' attention that many classification models categorise most monkeys correctly, but that it is difficult to properly classify all the animals. For us as animal keepers, it is therefore clever to use the most successful model when feeding the new monkeys, even if it doesn't guarantee that we will never get bitten.

The 'AI Unplugged' booklet features more activities



One way of recording the criteria is a decision tree

ADVANCED VERSION

In the advanced version, image 21 (see **Figure 2**) can be used to illustrate the problems of an AI system when the characteristic value of an element differs significantly from the training data. We have no experience with the characteristics of image 21, because this monkey has a new, unknown mouth shape. Accordingly, an appropriate assignment of the monkey is not possible. In practice, the behaviour of an AI system is very difficult to predict for this case. Instead of image 21, you could also use the image of a different animal to emphasise the different characteristics of the new element. Subsequently, these examples can be applied to reality: a bank unexpectedly does not grant credit to a specific customer; a self-propelled car sees leaves on the road as a dangerous situation and mistakenly slams on the brakes. In these situations, the AI system can also be dangerous if it is not comprehensible how these decisions were made.

> IN SO-CALLED SUPERVISED LEARNING, AI SYSTEMS OBSERVE A SERIES OF INPUT AND OUTPUT PAIRS, OR TRAINING DATA

BACKGROUND

Category formation is made possible by recognising repetitive patterns in individual elements. But how do these aspects relate to artificial intelligence? In so-called supervised learning, the AI system observes a series of input and output pairs (training data) and learns how they relate to each other, as well as which patterns are typical for which category. This knowledge is then used to classify new elements into the categories. Test data, whose categories we know, but the AI system doesn't, is used to determine the quality of the learnt classification model. The same principle is used for neural networks and other AI applications. This procedure can lead to various problems, because no model is perfect. Depending on the training data, the classification model can overweight or neglect certain characteristics of the training data, so that no general statements, and thus no correct classification of unknown elements, is possible. Having a lot of training data can help to reduce these effects, but does not always lead to more accurate results, as too much training data can also result in overfitting. In this case, the AI system learns the training data 'by heart' and is no longer able to generalise to new data. It makes sense to address these aspects of machine learning as part of the activity. When applying their rules in the test phase, let the students explain which characteristics they used to classify the monkeys. This will illustrate that the students have created different sets of rules. Point out that a classification model is unlikely to be 100% accurate and that the model that best classifies the test data will be chosen in the end. Have students describe their own learning process and then compare it to that of a computer.

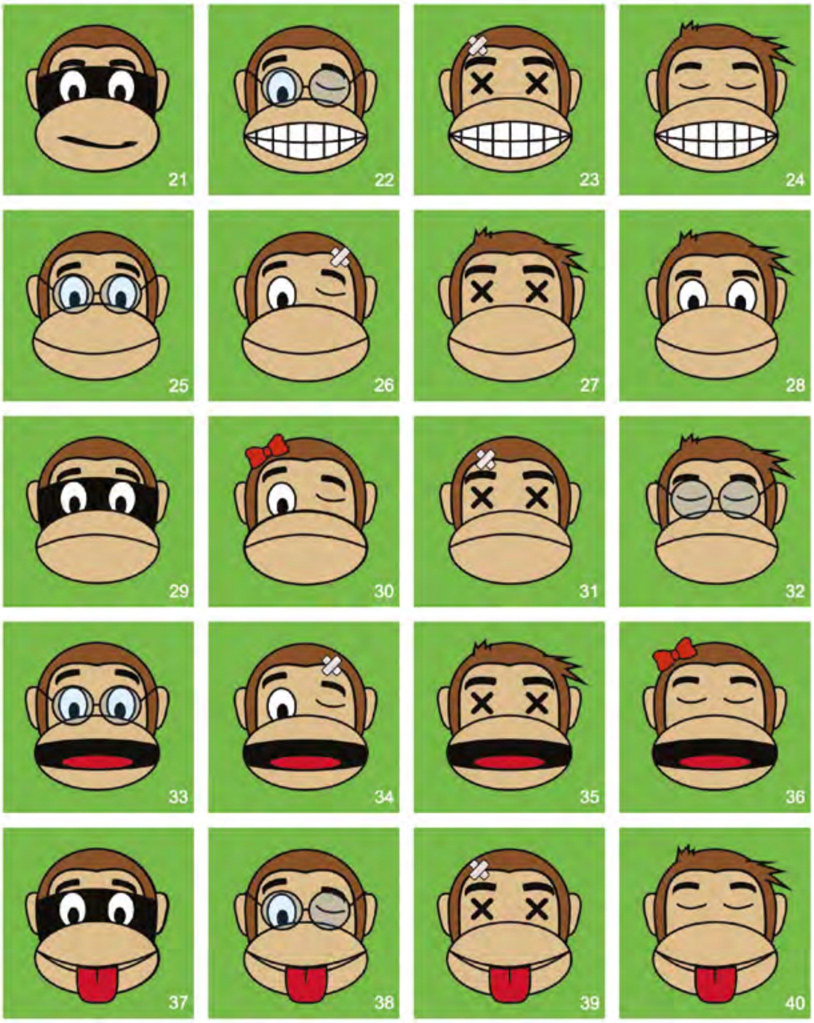


Figure 2 For monkey 21, no explicit criteria can be derived from the data

RELEVANT LINKS

This activity is from AI Unplugged (**aiunplugged.org**) by Annabel Lindner and Stefan Seegerer. You can download their brochure featuring five unplugged activities for teaching artificial intelligence ideas and concepts to learners of all ages. The brochure is available in German, English, Korean, Spanish, and Portuguese.

Raspberry Pi Foundation

Introduce the Code Editor into your school!

The Code Editor helps make learning text-based programming simple and accessible for children aged 9 and up.

"We have used it and love it, the fact that it is both for HTML/CSS and then Python is great as the students have a one-stop shop for IDEs."

- Lee Willis, Head of ICT and Computing, Newcastle High School for Girls

School accounts coming soon:

- Create engaging coding lessons and share them with your students: Encourage your learners to get creative with Python, HTML, CSS and JavaScript
- Simple and easy classroom management: Organise students into classes and help them reset passwords quickly.
- Free now, free forever: Add an unlimited number of projects, teachers, and students.
- Safe and private by design: visibility of work at all times and verified school accounts.

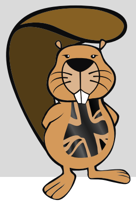
Find out more and pre-register your school: rpf.io/code-editor-hw25

THE BEBRAS PUZZLE PAGE

Each issue, **Andrew Csizmadia** shares a computational thinking problem for your students based on the work produced by the International Bebras Community

ABOUT BEBRAS

Bebras is organised in over 90 countries and aims to get students excited about computing and computational thinking. Last November, 408,469 students participated in the UK annual challenge. Our archived questions let you create your own automarking quizzes at any time during the year. To find out more and to register your school, head to **bebras.uk**.



THE PROBLEM: MAGIC TREE

DOMAIN

Algorithms and programming

SKILLS

Abstraction, algorithmic thinking

AGE

8–14 years

DIFFICULTY RATING

Ages 8–10 hard
Ages 10–12 medium
Ages 12–14 easy

Bain the Beaver has a magical tree growing near their home.

Whenever a bird ( ) lands on it, the tree grows 2 apples.

Whenever a squirrel ( ) climbs up it, the tree drops 1 apple (if it has any).

Whenever a snake ( ) visits the tree), all of the apples instantly disappear!

One morning, Bain notes that the magical tree has 25 apples.

They then spend the rest of the day drawing pictures of all the animals that come to the tree.

The drawings, in order, are above.

Question

How many apples are on the tree at the end of the day?

3 7 17 31

Explanation

This task introduces the ideas behind TWO fundamental programming concepts.

The first programming concept introduced is the idea of a variable. A variable is used to store information that a computer program needs. The value of a variable can change depending on what the rest of the program's instructions are. In this task, the number of apples on the tree is a variable; its value can increase ( ), decrease ( ), or reset ( ).

In order to decide how to change the value of a variable, a computer program needs the ability to make decisions. The second fundamental programming concept introduced in this task is selection. Decision-making is accomplished by using special instructions called 'conditional statements', which allow you to select from different possible outcomes. They commonly take the form 'if this, then that'.

In this task, one conditional statement would be 'If a bird lands on the tree, increase the number of apples by 2'. Can you find two more conditional statements?

In order to solve this task, two computational thinking concepts are applied. These are abstraction and algorithmic thinking. The task is in the area of abstraction, as students need to distinguish between the different cases in the task and focus only on the information they need. To solve the problem most efficiently, it is sufficient to examine the data only from the last snake. The preceding markings are not relevant for us, as we always set the number of apples to 0 when the snake appears.

The solution to the task also requires algorithmic thinking. Students need not only to solve the task step by step, but also to choose a solution depending on the situation, just as in the case of an 'if' statement.

This Bebras puzzle was originally written by the Bebras team from Canada. Solution on page 81. (HW)

HOSTING A SUCCESSFUL STEM SUMMER CAMP

In this Insider's Guide, **Alan O'Donohoe** shares his blueprint for running an inclusive and inspiring STEM camp

Teachers of computing and other STEM subjects face the increasing challenge of inspiring their students to pursue careers in these fields. But for many young people, traditional school timetables don't allow for the deep project work that takes place in industry. Over the school break, the gap in engagement and skill development widens without access to enriching activities or continued learning opportunities, particularly for disadvantaged groups.

To address these challenges, I have organised and led holiday activities throughout my teaching career, with a strong emphasis on technology and engineering. By creating opportunities for young people to explore STEM subjects outside the constraints of a traditional school setting, we educators can ignite a passion for learning and foster creativity, problem-solving, and a deeper understanding of STEM fields.

This article will share insights and lessons learnt from running a successful STEM summer camp, providing a blueprint for how educators can replicate this experience to benefit young people.

What happened at STEM camp?

Recently, I had the privilege of running a STEM camp in Swindon, UK — a ten-day holiday programme designed to engage disadvantaged young people in STEM. We transformed a large open space in the public library into our base camp for two weeks to host three different workshops, culminating in Final Friday, a STEM showcase event for an invited audience. We received over 100 applications and were able to offer spaces to a total of 31 children who attended over ten days in August 2024. There were 17 children aged 8–11 years and 14 children aged 12–14 years. Based on camps from previous years, we had expected about a dozen children to take part in total. However, when places filled up quickly, we adapted the programme to run three separate three-day workshops, opening up the opportunity to more young people.

Workshops took place from 10am to 4pm each day, with breaks for refreshments and unplugged activities outside. Unfortunately, our space was the warmest in the building, which had no cooling system and was consistently 10 degrees above the outside temperature. The minimum temperature was 27 degrees Celsius, and on one day, it reached 37 degrees inside! To ensure the well-being and comfort of participants, we provided ample water breaks and scheduled outdoor activities in shaded areas.

We received funding from Eko (**helloworld.cc/eko**) which allowed us to provide every child with lunch and other refreshments throughout the camp. Funding also contributed to staffing, travel, and accommodation costs for the STEM camp staff, each of whom travelled more than 200 miles to Swindon. We used pi-topCEED desktops, originally released in 2016 using Raspberry Pi 3B, for the workshops. With generous equipment donations and support from Exa Networks (**exa.net.uk**), CPC (**cpc.farnell.com**), Micro:bit Educational Foundation (**microbit.org**), and Monk Makes Ltd (**monkmakes.com**), we were able to provide every attendee with all the hardware they needed to take home, allowing them to continue working on STEM projects even after the camp. This ensured that all participants, especially those from disadvantaged backgrounds, had the necessary equipment to continue working on projects at home, regardless of their home environment.

ALAN O'DONOHOE

Alan has over 20 years' experience teaching and leading technology, ICT, and computing in schools in England. He runs **exa.foundation**, delivering professional development to engage digital makers, support computing teaching, and promote the appropriate use of technology (**@teknoteacher**).

Young people could choose from three different workshops at our STEM summer camp

Engaging activities: make STEM fun

We offered three different workshops, each designed to engage and inspire. When registering, participants could choose from the following options (listed in order of popularity):

1. **Minecraft creator:** using Minecraft Pi, this workshop enabled participants to combine coding, creativity, and engineering. By modifying and creating their own Minecraft environments, young people saw the practical applications of programming in a familiar and exciting way.
2. **Software developer:** participants were introduced to the basics of software development using tools like Scratch, Python, and EduBlocks. They learnt how to create their own games and applications, developing a deeper understanding of how software works and how it can solve real-world problems. This hands-on approach, combined with algorithm design, helped to demystify programming and made it accessible to participants of all ages and skill levels.
3. **micro:bit inventor:** this micro:bit workshop allowed participants to experiment with physical computing. Participants learnt how to program micro:bit devices to complete various challenges, such as controlling a fan (a welcome relief in the heat!) or creating simple alarms supported by the Monk Makes kits. This introduction to hardware and software integration provided a tangible connection to the concepts they were learning about and sparked interest in engineering and robotics, echoing links to STEM industries.

STEM showcase

To celebrate the achievements of our participants and share their work with their parents and the wider community, we held a STEM Showcase on the last Friday of the camp. This public showcase afternoon event not only gave participants a sense of accomplishment, but also served a further dual purpose: raising awareness of the importance of STEM education in the local community, and providing an opportunity for participants to develop presentation skills. On the morning of Final Friday, participants took part in a workshop to develop their STEM presentation skills for the afternoon showcase event, giving them tips on welcoming visitors, engaging audience interest, and recommending other projects to visit.

> OUTSIDE THE CONSTRAINTS OF A SCHOOL SETTING, WE CAN IGNITE A PASSION FOR STEM

Planning and preparation: key takeaways

One of the most important lessons we learnt from running the Swindon STEM summer club was the importance of early planning and preparation. Due to the late approval of funding (just a few weeks before the camp!) and logistical constraints, we faced several challenges in securing a suitable venue and finalising arrangements in time. However, with careful planning and a proactive approach, we were able to overcome these obstacles and deliver a successful programme within the reduced time frame.

1. **Start early:** begin planning your STEM holiday event well in advance. This will give you plenty of time to secure the most suitable venue, recruit staff and volunteers, and properly promote the event.

Learners work on their STEM presentation skills in an informal camp setting

2. **Choose the right venue:** ideally, the venue should be easily accessible by public transport, have adequate facilities for both indoor and outdoor activities, and provide a safe and secure environment for participants. In our case, Swindon Central Library was a public space, so we had to regularly remind participants to guard their personal possessions and be mindful of the fact that it was a shared environment.
3. **Recruit qualified, trained staff:** having knowledgeable and enthusiastic staff is essential for creating a positive learning environment. Our team included subject matter experts who delivered engaging and informative workshops. Additionally, we had a dedicated pastoral lead to ensure the safety and well-being of all participants.
4. **Managing expectations:** clear communication with parents and participants is essential to managing expectations and ensuring a positive experience for everyone involved. In some cases, parents clearly misunderstood the commitment required for the workshops, leading to unexpected absences or lack of participation. To avoid these issues, provide detailed information about the camp's format, expectations, and requirements during the registration process, and don't just rely on email to communicate.
5. **Safeguarding and safety:** it is crucial to have comprehensive safeguarding measures in place. This includes conducting risk assessments, providing first aid training for staff, and organising emergency procedures. Ensuring the safety of all participants should be a top priority. Keep safety and security at the forefront at all times, and choose venues that provide a balance between security and accessibility for young people.
6. **Rewards and recognition:** to motivate participants and recognise their achievements, we incorporated a variety of rewards and incentives. Each participant received a brand-new Raspberry Pi computer and an SD card preloaded with the software and projects they had worked on during the camp, allowing them to continue exploring STEM at home and reinforce their learning.

Fostering diversity and inclusion

One of the key goals of the Swindon event was to promote diversity and inclusion in STEM. We made a concerted effort to engage young people from diverse backgrounds, including those from minority groups, those with special educational needs (SEN), and those from neighbourhoods with high levels of deprivation. By creating a welcoming and inclusive environment, we were able to attract a diverse group of young people, many of whom might not have considered STEM as a potential career path.

Targeted outreach: to ensure we reached a diverse audience, we promoted the opportunity among our existing networks and partnerships with local schools and community organisations (Exa Networks provides connectivity to many Swindon schools). Social

FEEDBACK FROM PARENTS AND PARTICIPANTS

A feedback survey revealed high praise for the Swindon STEM camp. It showed that 91% of parents were impressed with the event's organisation, and 82% noted their children's new-found enthusiasm for STEM subjects. Parents appreciated the hands-on activities and supportive workshop leaders. Suggestions included running more workshops and finding a better venue. Feedback underscores the camp's success in sparking interest and fostering a positive learning experience in STEM.

media advertising also played a crucial role in promoting the event to targeted groups. By using a variety of channels, we were able to reach a wide range of young people and families — perhaps too many, which created additional logistical challenges!

Inclusive activities: we designed our workshops to be accessible and engaging for young people of all abilities and to be suitable for ages 8–14 years. By offering a range of activities, from software development to physical computing, we catered to different interests and skill levels. This approach helped ensure that all young people could participate and succeed, regardless of their background or prior experience.

Emphasising role models: to inspire young people and provide real-world context, we invited guest speakers from diverse backgrounds to share their experiences and career journeys in

> ONE IMPORTANT LESSON LEARNT WAS THE IMPORTANCE OF EARLY PLANNING

STEM, some in person and others virtually. These role models helped the participants see themselves in STEM careers and understand the importance of diversity in the field. By highlighting the achievements of individuals from underrepresented groups, we aimed to break down stereotypes and encourage participants to pursue their interests in STEM.

Looking ahead: expanding the impact

The success of the Swindon STEM summer club has inspired us to consider expanding the programme to reach even more young people. Based on our experience, we believe that similar initiatives can be replicated in other locations, providing valuable opportunities for young people to engage with STEM. Here are some recommendations for educators looking to host their own STEM summer camps:

1. **Expand reach:** consider expanding your STEM camp model to reach a wider audience. This could involve hosting multiple camps in different locations, or partnering with other schools and organisations to increase your reach. By offering more opportunities for young people to participate, you can help inspire a larger number of young people to explore STEM. When I was employed by a school, I initially offered places to our own students; however, when we opened it up to other children in the area, it was much better for everyone.
2. **Evaluate long-term impact:** to measure the success of your STEM camp and its impact on participants, consider conducting follow-up surveys and interviews. Tracking continued engagement and analysing changes in attitudes, knowledge, and skills can provide valuable insights into the long-term impact of your programme. Sharing success stories and individual case studies can also help demonstrate the value of STEM education and inspire others to get involved.
3. **Foster community partnerships:** building partnerships with local businesses, universities, and community organisations can enhance the resources and support available for your STEM camp. These partnerships can provide access to expertise, equipment, and funding, as well as opportunities for students to connect with professionals in the field. By fostering strong community partnerships, you can create a sustainable model for delivering STEM education and making a lasting impact.

Conclusion

Hosting a STEM summer camp is a fantastic way to ignite passion and build skills in young people. The success of the Swindon STEM summer club proves there's a strong need for such programmes. With thoughtful planning and a focus on diversity, you can make a significant impact in your community. Ready to launch your own STEM camp? Contact the author for tips and support via the Exa Networks website at **exa.net.uk**. Inspire the next generation of STEM leaders! (HW)

An inclusive environment helps to attract a diverse group of young people

ICT IS NOT COMPUTER SCIENCE, BUT WE STILL NEED IT!

Janine Kirk shares the challenges faced when trying to separate ICT and computer science and argues that there is a need for both as distinct subjects

I have spent time in many a curriculum meeting arguing that information and communications technology (ICT) and computer science are two different subjects. Indeed, I see the difference as analogous to geography and languages: ICT/IT is the environment we are using the system in, while computer science is the language used to communicate with the system.

Student perception

You would not expect your geography teacher to teach all of the languages used in the world. Yet at Key Stage 3 (ages 11–14) we are trying to mix IT and computer science in some sort of weird hybrid. Yes, I know the curriculum should be computer science-based, but I want my students to be prepared to model data or use software, both of which are more ICT-based. I know there is more to these subjects, but because we mix the two subjects at Key Stage 3, students genuinely think they are the same.

My GCSE computer science students don't touch a computer until their second year, while I teach them all of the theory behind the system. Imagine their shock when they are told not to use a computer! My counterpart, on the other hand, has his students on computers every lesson, teaching them ICT. We spend every lesson at Key Stage 3 saying, "This is a computing lesson" or "This is an ICT lesson". Students at option evenings still ask, "What's different?" And of course, there are the next steps: "Yes, you can study computer science after studying ICT, but you could do so even if you had studied geography." Why? Because at A level, we start again with the theory, as not all schools offer computer science at GCSE. In discussions, I have heard that ICT is preferred by teachers as well as students!

JANINE KIRK

Janine is the GCSE computer science Lead, exam assessor, and teacher of both ICT and computing KS3 at the Kings Academy, Kidgsrove, in the UK. She has 15 years of experience in leading and teaching an ever-changing curriculum.

Career overlap

Then there are the jobs to be an ICT apprentice. You need experience with software and networks — we are talking data analysis and electrical engineering. For a computer science role, we are looking at programming and system maintenance. If you study ICT, you are not studying programming; and if you study computing, you are not analysing data and systems flow. Of course you can argue that you might pick up some data management when programming, and in return you might pick up some programming when creating complex data management systems. Again, to use the comparison to geography and languages, I pick up a bit of French when I am in France, but that doesn't mean I can speak French. And let's face it, who really knows what ICT and computing jobs will be available in the future? It was not that long ago (2008) that the App Store was created, and look what that did for jobs in tech!

Are we hiding behind ICT?

I started my teaching when the ICT curriculum was deemed unfit for purpose, and retrained as a computer science teacher. However, what this meant was that ICT was put on the back burner as I reskilled and retrained. What was more surprising to me was the reluctance of others to retrain. Staff chose to become business studies teachers rather than computer scientists, sending a clear message about how ICT teachers felt about the computing curriculum. This further supports my argument of the distinction between ICT and computer science, as ICT teachers seemed to be finding closer links to business than to computing. But guess what? We do teach networks in ICT and computer science. We do talk about inputs and outputs in both. We even add in an 'if' statement here or there. So it can be seen that ICT is covering some of the computer science stuff — but we are missing the nuts and bolts and the deeper understanding of the system. We can't keep hiding like this, as we need our students to be able to improve and develop technologies to take into account our changing digital landscape.

The rise of AI

AI will affect us in a similar way to the internet in the 1990s. Life will be different, and we need our students to understand the systems that are in place. Recently, we used the free Experience AI lesson series (**experience-ai.org**) to teach students about AI. What struck me was that teaching students about the specifics of search commands was very similar to teaching students how to do an internet search from the old ICT specifications. It is clear to me that we still need these ICT skills, as well as students who understand how AI is trained — leaving aside the environmental impact of using AI when a Google search will do!

At the recent Computing at School (CAS) conference in Staffordshire, it was clear that AI is having a massive impact on teaching and computing education, but it was also clear we, as educators, cannot stand still with this. We have been teaching about machine learning and using AI for years, yet we are still leaving students behind if the curriculum does not allow for a distinction between computing and ICT. I am lucky that my school is investing in this, with an AI working group and a curriculum we can adapt to take into account both computer science and ICT, to prepare our students for the choice of ICT and computer science at GCSE. But more needs to be done. Surely computer science could be a language option, and ICT could be compulsory again. You wouldn't want students to stop studying English or maths, so why stop studying ICT, when our world uses it to communicate?

> WHY STOP STUDYING ICT WHEN OUR WORLD USES IT TO COMMUNICATE?

Curricula are curated and controlled by much higher powers than schools, and maybe we need to fight a little harder for a curriculum that reflects world priorities and keeps up with a global digital marketplace. Let's be clear — AI will change things. Are we really preparing our students? Could more be done to revisit ICT and computing in the English national curriculum? I would love to see them set out as two distinct subjects. (HW)

BEBRAS PUZZLE

BEBRAS PUZZLE SOLUTION: MAGIC TREE (PAGE 75)

The answer is that there are 7 apples on the tree at the end of the day.

As all the apples instantly disappear whenever a snake visits the tree, we can ignore everything that happens before the last arrival of a snake.

After the last snake, four birds land on the tree, which means the tree will sprout 4 × 2 = 8 apples.

Then one squirrel climbs up the tree, which causes one apple to drop, leaving 8 – 1 = 7 apples.

Credit: ©Claudio Divizia/stock.adobe.com

COMPUTING KEYWORD SPOTLIGHT: ABSTRACTION

Defining everyday words and phrases in computer science

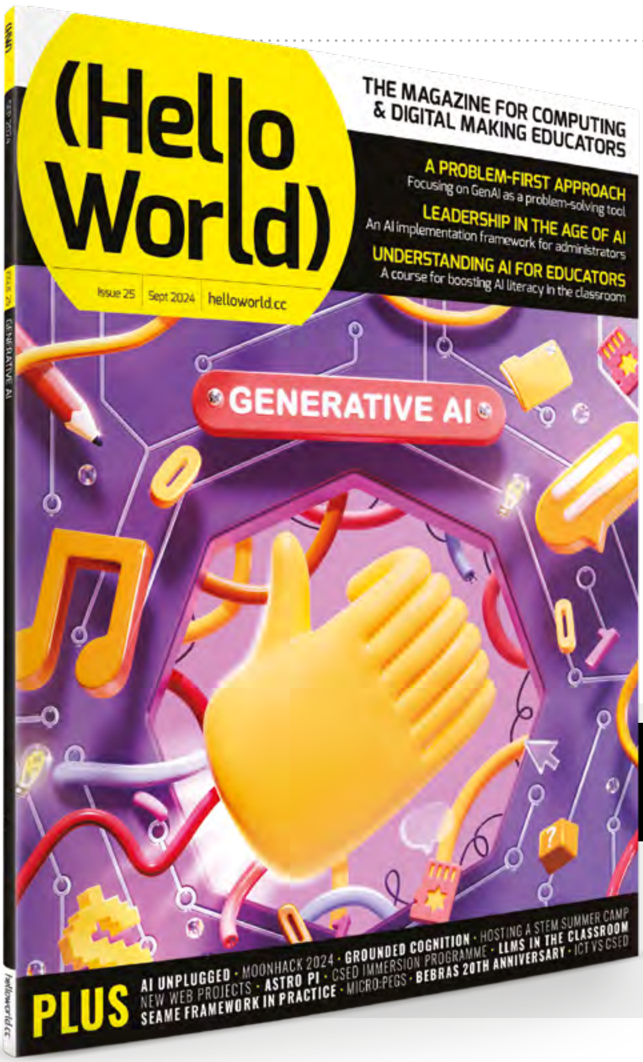
Abstraction makes problems or systems easier to think about, and is the process of making an artefact more understandable by reducing unnecessary detail. A classic example is the London Underground map. The London Underground railway system is a highly complex one. The representation of London in particular ways (usually maps or pictures) aids different users. The London Underground map is a highly refined abstraction with just enough information for a traveller to navigate the network without the unnecessary burden of information such as distance and exact geographic position. It is a representation that contains precisely the information necessary to plan a route from one station to another — and no more! The skill in abstraction lies in choosing the right detail to hide so that the problem becomes easier, without losing anything that is important. A key part of it is choosing a good representation of a system. Different representations make different things easy to do. (HW)

"HELLO, WORLD!"

Everything you need to know about our computing and digital making magazine for educators

Q WHAT IS HELLO WORLD?

A Hello World is a magazine and accompanying podcast 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 Hello World is the official magazine of the Raspberry Pi Foundation.

Q WHY DO 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, both inside and outside the classroom.

Q WHEN IS IT AVAILABLE?

A Your magazine is available three times per year. Check out our podcast (**helloworld.cc/podcast**) and monthly newsletter (**helloworld.cc/newsletter**) to get more great content between issues.

IT'S FREE!

Hello World is free now and forever as a Creative Commons PDF download. You can download every issue from **helloworld.cc**. Visit the site to see if you're entitled to a free print edition, too.

WANT TO GET INVOLVED?

There are numerous ways for you to get involved with the magazine — here are just a handful of ideas to get you started

- **Give us feedback**
  Help us make your magazine better — your feedback is greatly appreciated.
- **Ask us a question**
  Do you have a question you'd like to share? We'll feature your thoughts and ideas.
- **Tell us your story**
  Have you had a success (or failure) you think the community would benefit from hearing about?
- **Write for the magazine**
  Do you have an interesting article idea? Visit **helloworld.cc/writeforus** to submit your idea.

GET IN TOUCH

Want to talk? You can reach us at:
contact@helloworld.cc

FIND US ONLINE

www.helloworld.cc

@HelloWorld_Edu

fb.com/HelloWorldEduMag

NEXT ISSUE OUT
JANUARY 2025

THEME:
DIGITAL LITERACY

(Hello World)

helloworld.cc

Raspberry Pi
Foundation