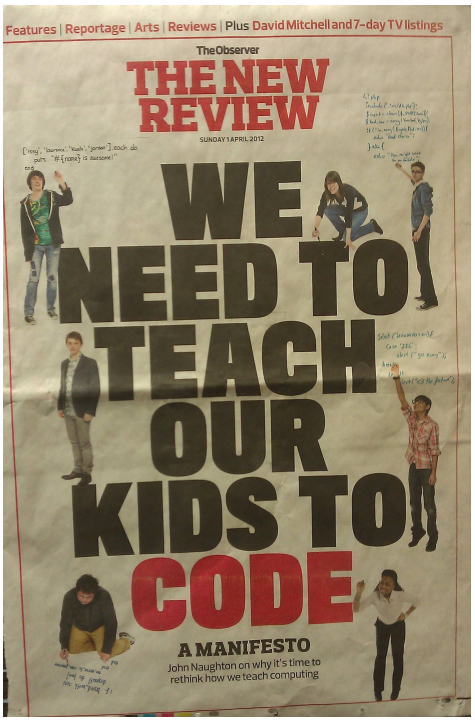
COMPUTING AT SCHOOL
EDUCATE · ENGAGE · ENCOURAGE
In collaboration with BCS, The Chartered Institute for IT

SWITCHED ON

COMPUTING AT SCHOOL NEWSLETTER AUTUMN 2012

HOW DO WE TEACH OUR KIDS TO CODE?

Last April, Observer columnist John Naughton reached a mass audience with his manifesto call to rethink how we teach computing. It followed a period of unprecedented pressure for change. Google's Eric Schmidt put the issue centre stage. The Royal Society Report and announcements from Michael Gove echoed the concerns felt by many in industry, academia and teaching that something, often for the best of reasons, had gone badly wrong in what we taught. Through the summer term, teachers have been busy rewriting schemes of work to embrace the new focus on Computing as well as ICT. Exam boards have developed new GCSE specifications so there will soon be five Computing qualifications to choose from. The first cohort the pioneering OCR GCSE Computing have just finished and hopefully many will be inspired to carry on with further study. The pipeline that is so important for providing the next generation of researchers and innovators is being repaired. It is increasingly recognised that some exposure to computing is now a vital component in many scientific disciplines.


Features | Reportage | Arts | Reviews | Plus David Mitchell and 7-day TV listings

The Observer

THE NEW REVIEW

SUNDAY 1 APRIL 2012

WE NEED TO TEACH OUR KIDS TO CODE

A MANIFESTO

John Naughton on why it's time to rethink how we teach computing

This is not about training 'code monkeys', but empowering a generation with a basic appreciation of how their digital world works. It's not about doing a bit of Java, or any other language, but developing the ability to think in a computational ways. These are exciting times but for many teachers, much will be new. CAS exists to support you on this journey. We all agree we need to teach our kids to code. The debate about how best to do so is only just getting going. The more you try, the more you'll find you can contribute and share.

LAYING FIRM CODING FOUNDATIONS IN 2012-13

Many ICT teachers will be new to the ideas embodied in Computing as a discipline. Even those with some background may well be new to ideas about how to teach that discipline. The subject has, after all, been off the school curriculum for the past twenty years or so. Everyone therefore has a part to play in rediscovering what we can do and what our kids are capable of. There is no blueprint. CAS local hub meetings are teeming with ideas and discussion. In September, in conjunction with BCS, The Chartered Institute for IT, CAS will launch the Network of Computer Science Teaching Excellence which will link teachers to Computing departments in universities. This unique development extends the communities developed in local hubs.

The CAS community is characterised by openness, mutual help and support. This issue of SWITCHED ON is brimming with suggestions of things you can do right now in your own classroom. These aren't official schemes of work passed down from on high, dictated in minute, '3 part lesson' detail. They are just cracking ideas CAS members want to share because we share a passion. We hope you'll give some of them a go and share your experiences too.

The "Computing At School" working group (CAS) is a membership association in partnership with BCS, The Chartered Institute for IT and supported by Microsoft, Google and others. It aims to support and promote the teaching of computing in UK schools.

bcs Academy of Computing

ISSN: 2050 -1277 (online) 2050 -1269 (print)

PRESTON HUB TAKES NEXT STEPS IN LEARNING PYTHON

With extra revision lessons, marking exams, writing reports, reviews, school productions and curriculum planning; the Summer Term can be one of the busiest for teachers. In that case, you may well wonder what persuaded nearly 40 teachers from around the North West to spend their Thursday evening at our school? To attend a CAS Hub meeting of course, to take their "Next Steps in Python". After our last hub in March which introduced Python, I read the evaluations and realised there was a clear appetite for more. After posting the listing on the Eventbrite website, I was amazed to see all 30 free tickets taken within 24 hours and the waitlist started filling up. Making a calculated allowance for a proportion of 'no shows', I decided to mimic the practice of airlines and hotels and oversell an additional 10 tickets. Regretably, the room designed for 32 children felt rather cramped with the 39 adults who did attend all moving around.

Referring back to the "Hello World" program from our last meeting, I presented the class with 5 variations and asked them to predict which would work. They tried the code to test their predictions, producing some interesting revelations. Our next exercise involved extending a program. Since we had a real mix of ability and experience in our class, I asked experts to add some challenging functionality. Partners were encouraged to support and challenge each other. I got a real buzz watching our learners supporting each other, not relying on the teacher too heavily.

The meeting closed with a questions session and allowed me to promote some other ventures such as The Hack Rap, Hack To The Future and the upcoming Raspberry Jams spreading around the UK. The sense of achievement meant that a day later I was still buzzing. This was our fourth event; our first one, a year ago had only 8 participants. They have grown in size each time and now we are having to turn people away because demand exceeds our capacity. If you would like to know more about setting up a hub, contact Claire Davenport. She is very welcoming and will help get you started. *Alan O'Donohoe*

BUILDING CAS LOCAL HUBS: THERE IS NO 'THEM', ONLY US

Probably the most significant development since the inception of CAS has been the development of local hubs. Bringing together a range of expertise, the enthusiasm they generate is infectious.

One of the great things about a group that shares a genuine desire is how structures can develop free from the inertia that exists within education. How many other subjects can boast the impressive mutual support that exists in CAS? It links University Computer Science departments, concerned IT professionals, parents willing to help out, school governors and, of course, teachers in ever increasing numbers. Local CAS hubs meet, usually once a term, incubating many of the initiatives you read about in SwitchedOn. None of this is planned centrally—in fact CAS has virtually no central organisation. Hubs have a friendliness borne of the common passion. They embody a humility that recognises we all bring different skills to the table and none have a blueprint for success. They

Local hub organisers gather at the CAS Teacher Conference

work because, once a need is recognised, folk roll up their sleeves and do something. Not everything will work, certainly not first time, but by trying things, everyone learns and develops new insights. If you think about it, just like children, CAS hubs learn through doing. The amount they are doing is absolutely staggering. With around forty local hubs now established and thriving, teachers up and down the nation are finding them an indispensable forum for advice, support and ideas. The two reports (left and bottom), give a flavour of the diversity. Take a look at the CAS website to find your nearest one and, if there isn't one, we have a free help manual that guides you through how to organise one.

NORTH EAST SCOTLAND HUB VISITS GAME STUDIO

Hunted Cow Studios is a games development company specialising in multi-player online role play. We met at the Elgin-based studios and were given a informative tour by one of the owners, Andrew Mulholland. Andrew was educated in Elgin and went to Abertay to study game development before returning to Elgin to start Hunted Cow Studios with a fellow Abertay graduate, Glenn Murphy. Hunted Cow work with local schools providing a number of work experience placements for local secondary students. The students are mainly involved with game testing, a vital part of the design process for all online games both when they are being launched and when new upgrades are added. We had a short meeting after the tour where we discussed ideas for future speakers and the upcoming CAS conference. *Claire Griffiths*

SPREADING THE KNOWLEDGE VIA THE NETWORK OF EXCELLENCE

For computer science education in schools to be successful over the long term the UK needs a national Network of Computer Science Teaching Excellence, which is supported by universities, employers, learned societies and professional bodies.

In September 2012 BCS, The Chartered Institute for IT with CAS are launching such a network. With support from The Department for Education, OCR, CPHC, Microsoft and Google the Network aims to provide much needed training and support to teachers of Computer Science.

The Network of Excellence will create a closely knit national federation of university-led local school networks. The central goal of the local networks is to build capacity of expert schoolteachers with the competencies and capabilities necessary to support the development of other computer science teachers. The intention is that many of these expert teachers will go on to become Master Teachers of computer science.

Universities will lead in the delivery of foundational computer science courses that help to develop expert schoolteachers within the lead schools. The national steering group will ensure the CPD courses offered throughout the network are academically rigorous, intellectually challenging and cover all the relevant computer science principles, concepts and techniques. Lead schools will in turn directly support other nearby schools with staff development, thereby disseminating innovative teaching practice and subject matter expertise.

The 30+ CAS Regional hubs are the starting point for the Network. Each is a local grouping of school teachers who meet informally to share innovative pedagogy and experiences of what works in practice. What these hubs do not do is equip schoolteachers with expert subject matter knowledge of computer science or the competencies to professionally develop colleagues' teaching abilities but by developing closer links to the universities and It professionals they can. With over 500 schools registered as part of the Network we have our work cut out but with the support of DfE, industry leaders, university academics and the wider CAS community we are looking forward to the challenge!

Simon Humphreys

COULD YOU BE A COMPUTING MASTER TEACHER?

One of the first steps for the new Network of Excellence is to recruit and train 20 Computer Science Master Teachers. The CS Master Teachers will start work with Computer Science departments, BCS, CAS, employers and interested Teaching Schools to create professional development programmes for schoolteachers. The Department for Education has provided some funding to support these teachers until March 2013. They will be pivotal in the development of the Network of Excellence. The opportunity will also provide theses teachers with experience and skills for their own career advancement.

It will be their role to support the training and development of existing teachers to become more expert in their practice. Working closely with university computer science and education departments they will build the capacity of additional teachers for the Network and create resource materials that can be used by these teachers in their own classrooms. If you would like to apply to be a CAS Master Teacher or find out more please email Mark Dorling. Contact details are as follows:. mark.dorling@computingatschool.org.uk

CAS ROUND-UP'S SHOW IT IS GOOD TO TALK

One of the best things about CAS is the sense of community, and one of the best parts of the annual conference and the CAS Hubs is the opportunity to talk to other people, to share ideas and experiences and discuss tools and strategies. CAS Roundup is another way, with a regular, informal, online meeting. Once a month there is a Flashmeeting, hosted by Neil Brown and myself, where anyone can drop in and join the discussion. All you need is a web browser and Adobe Flashplayer, although a webcam and a microphone are beneficial too. With a mixture of video, audio and text based chat, everyone can participate. The topics are suggested at the CAS Roundup Wiki although burning questions can be introduced as we go. The finished meeting is packaged as a podcast available from wordpress and the iTunes Podcast Directory (details in the web supplement). Keep an eye out for the date and time of the next meeting. I hope to see you there. *Mark Clarkson*

CAS Online is the new website for the CAS community. Whilst the Google forum has thrived it hasn't been easy to share material or keep track of threads. CAS Online aims to make sharing ideas and resources easier. This is not always easy. The main problem is not lack of material. But that teachers think their work isn't good enough. We often hear: "Just give me time to polish it a bit, and then I'll upload it" yet we all know that time never comes. So, the main message is: Share it anyway! There is only one rule: Be nice. We're all in this together. You're amongst friends here.

CAS TEACHER CONFERENCES ARE ANOTHER HUGE SUCCESS

The fourth annual CAS Teacher Conference was held at the University of Birmingham in June, followed closely by the second CAS Wales Conference (see page 18). As we write this, preparations are being finalised for the first CAS Scotland Conference which is scheduled for Saturday 27th October.

CAS has grown rapidly in the last year and the attendance at the conferences reflects this. The level of discussion and debate, infectious enthusiasm and sheer number of workshops makes them a 'not to be missed' event. Unfortunately, demand currently outstrips supply and many teachers may have missed out. If so, you can get a flavour of what the conferences involve from the CAS website. Thanks to some sterling work by Leon Cych and others many of the workshops and keynotes were recorded. These, and some of the supporting slides are now available online. Further supplementary videos, with interviews of CAS members can also be found on the CAS YouTube and Audioboo channels. Together, these provide some excellent source material for use in schools. You will also find material from the 2011 conference workshops. Time spent checking some of them out would be well worthwhile. See the supplement for links.

COMPUTING BLOGS BEING AGGREGATED AT PLANET CAS

Several CAS members write good blogs, but newcomers don't always know about them all, and following lots of disparate blogs can be irritating. So I've created "Planet CAS", an aggregator for CAS blogs, gathering together the best relevant content into one place. There is also an RSS feed.

It's already live, and any new posts on the contributing blogs get pulled across within the hour. I'm grateful to all the contributors who agreed to let their content be syndicated. I've written a little bit about my selection criteria. You can find the links in the supplement. If you do blog about computing and think your blog is suitable, please drop me an email and I'll see about adding it. *Neil Brown*

THE RELATIONSHIP BETWEEN CODING AND COMPUTER SCIENCE

Computing and coding are not, of course, unrelated. We want our learners to develop an understanding of computer science as well as gaining some experience of writing computer programs. Can you have one without the other?

I'm fairly confident that coders develop an understanding of the most relevant aspects of computer science through the assimilation and accommodation necessitated by the day by day experience of writing software; the same, I'm sure, is the case with proficient artisans in fields where others pursue academic study.

The thing is, I don't think it's enough just to provide *training* in craft skills. I see this as one of the big failings with the approach we've taken to ICT education – we've focussed far, far too much on providing learners with the skills they need to use particular applications and not nearly enough on an all-round understanding of how technology works. I don't think it's enough just to teach children how an e-mail client works without also explaining what's happening behind the screen. Equally, I don't think it's enough just to show children how to assign variables or manipulate lists without providing some way for them to think about these rather than just using them. It's just this sort of understanding which we've come to label as computational thinking.

That said, when it comes to how children *acquire* this understanding, direct experience is going to matter more than direct instruction. For most coders, the process of cognitive accommodation necessitated when the program doesn't work as expected lies at the heart of most meaningful learning experiences. I'm far from convinced that there's enough opportunity for this in school – a spoon-fed, teaching to the test, learning objective led, waterfall-like approach rarely provides sufficient chance for this. A more problem/project/portfolio based, agile approach might well do so though, particularly if teachers give enough thought to structuring the progression and challenge that learners meet.

With very young children, we've an abundance of evidence to suggest that they learn through play, exploration and experiment: I'm thinking here of Piaget's notion of the child as lone scientist, making use of the concrete manipulatives around them to develop their own schema of how the world works. The path from Froebel to Piaget to Papert and Logo and then to Resnick and Scratch is one well worth exploring. An approach which guides learners to discovering the core ideas of computing through the practical experience of coding has much to commend it – we learn through making, but we learn more than just how to make.

There is still some place for *theory* in CS education. We aren't *merely* Piagetian lone scientists, we also can use language to draw on the experience of others, both present and past. Nevertheless, for theory to make sense to the learner, for it to become part of their schema, there ought, surely, to be some connection made with existing knowledge, and thus, sooner or later, prior experience. Thinking about other domains, it's hard to think of academic musicologists, literary critics or theoretical physicists who haven't at some stage enjoyed making music, writing or experimenting. *Miles Berry*

GETTING PRIMARY PUPILS CODING WITH THE HELP OF CODE CLUB

"We were motivated by Eric Schmidt's [2011 McTaggart] speech in Edinburgh," says Clare Sutcliffe. "We wanted to do something that helped kids understand how cool and fun coding can be," adds Linda Sandvik, founders of Code Club.

Believing that even young children can learn to program, they aimed their efforts at primary schools. The problem was, how? There was no way a pair of outsiders could push additional lessons into an already crowded curriculum. Therefore, Linda and Clare organised around after-school clubs. Another problem was the lack of programming expertise among primary teachers. To alleviate that, Linda and Clare set out to recruit volunteers who would provide the technical expertise, while the teachers dealt with class management aspects of the club. They decided the first Code Clubs would use Scratch, specifically designed for young children to quickly create exciting projects.

Code Club volunteers working on ideas for their curriculum

Code Club was launched to the world in April and was quickly picked up by places like the BBC and Wired.com. Offers of support came flooding in. At the time of writing, 135 schools and over 1600 volunteers have signed up.

Linda and Clare aren't professional educators, and realised they needed help developing material. They sent out a call for helpers and soon assembled a team of talented individuals (and me). We set to work designing a curriculum for Code Club. The after-school club setting has a strong influence on how the material is designed. It had to be immediately exciting and engaging. It had to stress fun over theory. It had to accommodate children missing a week or two for other commitments. Code Club would be based around a set of projects. Each would be self-contained. Projects include simple games (and some aren't so simple!), drawing and animation systems, and an animated "all about me!" project. We'll be developing more at a hack day in the autumn term.

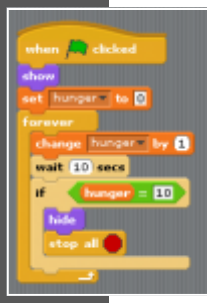
In each session, children pick the project they want to work on, individually or together. They work through it mostly on their own, with the Code Club volunteers on hand to give guidance and encouragement. The projects are split into stages. The first few have explicit, "follow along" instructions to quickly get a basic program working. Subsequent stages are more open-ended, with suggestions for extending or modifying the project. Throughout, children are encouraged to be creative and customise the projects. In the summer term twenty schools piloted Code Club. The children produced all sorts of ideas to make the projects their own. Based on this, the future of Code Club looks bright. By the time you read this, the first Code Clubs should be running and the next generation will be learning the joys of Code!

Neil Smith

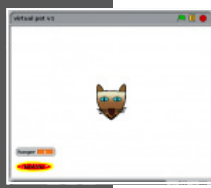
PROJECT IDEA (PART ONE): CREATING A VIRTUAL PET

I teach at Ellington and Hereson School in Ramsgate and share ideas and experiences in my blog. You can find the link in the supplement. Remember the tamagotchi? They made a reappearance a few years ago but didn't have the same impact as the original run, where school secretaries were forced to run crèches to avoid class disruption. Still, it's a fun concept, and one suggested by Nikki Maddams on her website. I have developed this a little and will continue to develop it as the project goes on.

This is what a simple pet sprite code looks like in BYOB (more or less just Scratch, here; no extra features used yet). The timing is very short at the moment to allow for efficient testing.



There is a button to click to provide food. The variable hunger is set to apply to all sprites (global variable). Another option is to make it apply only to the pet, but in that case the button sprite would not be able to change the hunger value and would have to send a message to the pet instead, asking it to change its own code. This is what the game looks like so far: very simple. The pet's hunger is displayed on-screen and gradually goes up to 10. If it gets to 10, he disappears. If the feed me button is clicked, this decreases the hunger by 1. The next development will be to have a value for health, which will gradually decrease by a random amount and will require medicine to be given. You can read about some of my ideas about possible ways to develop this further overleaf.



Lin White

PROJECT IDEA (PART TWO): MY VIRTUAL PET SPECIFICATION

Having created a simple prototype in BYOB, the next task was to create the same project using MIT App Inventor. This was quite straightforward. The details are on my blog but it says a lot for the usability of this system that I felt confident enough to try it so early in my own learning.

If I'm to build a game, I need to be very clear what I expect it to do. Then I can look for code structures to implement it. The pet will be awake three-quarters of the time and sleep the rest of the time. It will have hunger, happiness and health, which will vary from 0 to 100. All these will change over time, less so while asleep. Hunger will be controlled by feeding, health by giving medicine and play, and happiness by playing with it. Waking it up will reduce happiness, but may be necessary if it is ill or starving. There will be some interaction between attributes: being too hungry will decrease happiness and health. This can encode this with maths, rather than needing extra if statements (making the happiness directly relative to the hunger and health values in some way).

Clicking three buttons: feed, play and medicine will interact with the pet, changing the graphic and any appropriate value(s). I haven't any incentive to keep the pet alive, no way to keep any kind of score. I'm relying on the player wanting to keep the pet alive. I could add an ageing system and players can try to beat their oldest but that can wait for the extended version, I think!

Code-wise: I need to respond to the button presses by changing the graphics and variables. The variables also change according to a timer. There will be randomness in some variable changes, particularly illness. More complex would be variables changing at different rates according to activities eg getting hungry quickly if playing. I foresee lots of looping and if statements in the code. What I need to do now is write the algorithm for the pet's behaviour then translate it into the different languages. I aim to implement this in a variety of languages to help my learning. You can have a go yourself or follow my progress on the blog. *Lin White*

TEACHING PROGRAMMING: FIVE KEY TIPS FOR SUCCESS

Peter Donaldson is Head of the Computing Department at Crieff High School, Scotland. He has been teaching programming to secondary pupils for seven years, with a variety of successes and some failures along the way.

Based on my own experience, examining international schemes of work and reading some of the educational research into the challenges posed by teaching programming, I'd make the following recommendations:

- Don't rush! There's a lot more to pupils becoming fluent with the basic language concepts than you might think. It's particularly important to give pupils experience of the different roles that these features play and gradually build up more complex combinations of the features over time in a variety of different contexts.
- Choose motivating problems and contexts. A concrete and engaging computational context that pupils can relate to and find interesting is critical, creating an abstraction for solving problems in another abstract domain such as mathematics without anything for pupils to directly relate to is a recipe for disaster. Engaging computational contexts in rough order of complexity are computer generated media, animation and storytelling, 2D games, robotics, simple cryptography, AI such as chat bots and text adventures, physical computing, 2D simulations, 3D games and simulations.
- Focus on the big picture. Build up pupils problem solving ability, confidence and enjoyment of the process using an environment with a visual programming interface before you move on to a more traditional text based environment. Most pupils won't see the wood for the trees if they get mired in syntax errors and having to reduce the complexity of the problem they are trying to solve to compensate will also reduce their sense of accomplishment.
- Ask their opinion. Provide opportunities for them to activate their prior experience and knowledge by giving them opportunities to predict and discuss with their peers what they think the program is doing. You can do this by showing a piece of code with one or two new features and asking them to predict what it will do, explain how a faulty program should work and get them to discuss what they think the problem is or show them the behaviour of a program and not the code and then ask them to describe what the computer is doing to produce that behaviour.
- Show the problem solving process in action. Interactively modelling solving parts of a particular problem while thinking aloud and making small mistakes is very effective (Michael Köllings Joy of Code videos are a particularly fine example of this in action). Showing them how to think about a problem and then how to find and fix small mistakes really helps to demonstrate that you don't need to be a member of the black arts or Albert Einstein to be able to create a program.

10 EXCELLENT MINI PROGRAMMING PROJECTS

Often the hardest part to teaching programming is choosing fun contexts that are at an appropriate level. If you are stuck for ideas try resource posted on the new CAS Community. Written by Laura Dixon, Head of Computing at Rugby School, it suggests ten excellent projects around 2 to 6 lessons each. Each one is supported by notes, skeleton code and supporting material. Thanks Laura - link in the supplement.

DEVELOPING A NEW LITERACY: INTERVIEW WITH CODECADEMY

Since its launch a year ago, Codecademy has built a huge following for their free online coding courses. At the CAS teacher conference, SwitchedOn talked to Sasha Laundy from Codecademy about the vision behind the project.

Sasha Laundy, Codecademy Curriculum Strategist ,and speaker at the recent CAS conference.

Welcome Sasha, I hope you are finding the conference stimulating.
Thanks for having me! I'm struck by the amazing movement right now in the UK toward teaching computing in schools. As computers get more powerful and pervasive, it's ever more important for students to understand how they work. There's a huge opportunity to help shape how the next generation use technology and their ability to make new tools.

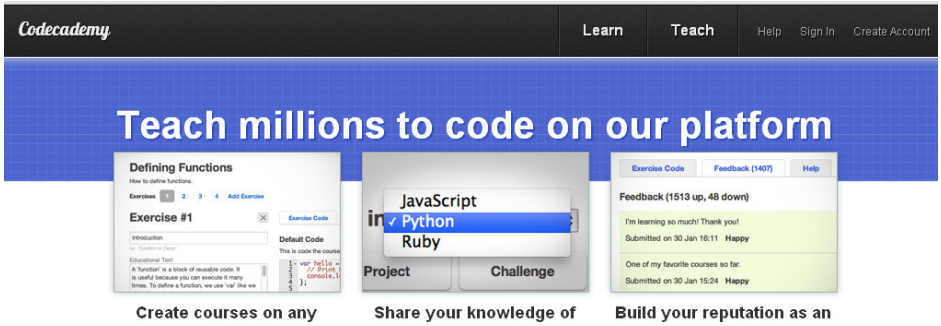
What was the central theme of your workshop session?
Digital literacy is rapidly become a critical skill in the 21st century. Software is an incredibly powerful tool that is revolutionizing every single field. Learning to program doesn't necessarily mean becoming a computer scientist or a professional software engineer. At Codecademy we're already seeing professionals like journalists, librarians, and scientists forming their own study groups to apply their new programming skills to be more effective in their respective fields. These learners are making the transition from software consumers to software producers, which gives them a voice in how that software works. This is a big power shift that we also saw with reading and writing. Before cheap paper was widely accessible, only elites could read and write, cutting off the larger population from all the benefits of those skills. With the expansion of literacy, countries saw rapid increases in economic opportunity and quality of life. The power to create new tools is transformative. It's really important for schools to prepare students for this new reality, equipping them with the skills they need to understand and build basic software.

For the unfamiliar, could you tell us a little more about Codecademy?
We offer a web platform where learners can take interactive programming lessons and teachers can create custom lessons. An online envi- ronment avoids issues with installation or permission problems. You get to the fun part right away—coding! Learners love this and we've seen tremen- dous interest – over a million regis- tered users and a half-million people signed up to CodeYear, through which they get an e-mail with a new lesson every week. We have seen users from 100 countries and in just our first year. The founders, Zach Sims and Ryan Bubinski, are passionate about teaching people to build cool things.

What languages do you support?
You can take a create custom lessons in HTML, CSS, JavaScript, and most recently, Python. More languages like Ruby and Java are coming soon. We introduced Python in response to popular demand, and your feedback can help shape future versions. Quite simply, we want as many people as possible to learn to love programming.

You mentioned the launch of Python. This is an increasingly popular language used in schools.
Yes, we heard from so many teachers who wanted Python that we added it to our site. Teachers love it because it is a great language for beginners; it's flexible and powerful with a clean, readable syntax. It's widely used, popular with scientists and academics as well as in industry players like Google. It's extremely well supported with third -party modules and a great community. I understand it comes installed on the Raspberry Pi and you can interface with Arduino as well. We'll be adding more module support soon.

For teachers, flexibility in delivering material is important. How can lessons be customised?
We offer a free tool so you can create custom lessons and projects. You can develop your own lessons and have them tested before sharing them with students. So you can choose how you use Codecademy; classes, clubs or enrichment. And because you can embed instructions into your lessons, it can free your time to support those students most in need of help. And of course, the material you create need- n't be restricted to children. As the demand for computer science teachers grows worldwide, there is a need for high-quality professional development material. You can help teach a new generation of educators..



Create courses on any programming topic

Share your knowledge of JavaScript, Python and Ruby

Build your reputation as an expert in your field

TURING CENTENARY WINNERS OF CODEBREAKER COMPETITION

The Codebreaker competition, launched by CAS in conjunction with University of Manchester's Animation 12 competition, celebrated Alan Turing's centenary by asking students to create a Greenfoot program with a theme related to code. Supplementary teaching materials for Computing Teachers was also provided to encourage teachers and students new to Greenfoot to get on board. Winners were awarded prizes for individual and group entries in the following ages.

Congratulations to Callum Harrod and Ajaimon Mathen (shown above) from Sussex Downs College, where they are studying Level 3 BTEC Software Development, and Gagan Kaur from Surbiton High who won the 17+, group and individual categories with their respective Maze Runner and Pest Control entries. The 11 to 13 group winners were Naomi Witts and Yvette Murphy with Musical Codes. The individual category was awarded to Ben Harries with Ceasar's Shift. All three students attend Aberdour School in Surrey. In the 14 to 16 group the winner was Mrinank Sharma and Isaac Golberg (both from Calday Grange Grammar, Wirral) received a special merit.

You can read more details about Animation12, which again received another huge entry on page 14.

NATIONAL CIPHER CHALLENGE

The University of Southampton Mathematics Department hosts an annual Cipher Challenge for schools. Running late September to January, the competition releases new challenges each week. Competition can be intense for getting answers submitted first! More details in the web supplement.

INTRODUCING PRINCIPLES OF ENCRYPTION VIA SPREADSHEETS

The Digital Schoolhouse and Bletchley Park teamed up to develop KS2/3 code breaking lessons. Using traditional media and spreadsheets they combined key principles with concepts from Computer Science.

All students, both past and present have at some point passed a note around the classroom to a friend. Most have had that message intercepted and experienced the subsequent embarrassment that comes with somebody reading a message for whom it was not intended. When asking students if they would like to learn techniques for saving this embarrassment they always answer "yes"! This provides an excellent opportunity for the class teacher to introduce the history of encryption, showing them the lengths that some have gone to in order to hide messages. For example the Romans used to shave a messenger's head, tattooing the message onto the scalp before letting their hair grow back. Mary Queen of Scots failure to sufficiently encrypt her messages resulted in her paying the ultimate price i.e. being beheaded for high treason.

The scheme of work, published free of charge by The Digital Schoolhouse and Bletchley Park builds on ideas published in 2006 by Mark G. Simkin entitled "Using Spreadsheets to Teach Data Encryption Techniques". The author explains how to model a transposition and substitution cipher using spreadsheet functions such as concatenate, vlookup and if statements. The scheme also models the modulo-2 addition cipher and relates how one of the world's first computers, Colossus, was used to crack the German encrypted messages, using the successor to Enigma, the Lorenz SZ42A, nicknamed Tunny by the Allies.

Getting to grips with the coding exercises at Bletchley Park

Piloting the resources in Key Stages 2, 3 and 4 at Langley Grammar School showed some ciphers are easier to model and more suitable to younger ages. However, GCSE students with little prior experience found the entire scheme (including non-computer activities investigating data compression) invaluable for covering elements of the specification. The scheme contain step-by-step presentations and videos for completing each method of encryption. It also include challenges requiring students to use the models developed to crack the code. The answers to all the challenges provide the clues to solving the Database Detectives lesson featured in a previous issue of SwitchedOn. There are further plans to provide alternative creative curriculum links relating to educational visits to Bletchley Park. Further details about the resources can be found in the supplement.

Mark Dorling & Victoria Worpole

SENSING OUR WORLD: PROJECTS USING SCRATCH SENSORBOARDS

The Technology Volunteers project at The University of Warwick pioneered the use of Scratch, Picoboards and homemade sensors in local schools. They developed a set of resources showing how the sensors are made and giving ideas for their use.

These homemade sensors were the focus for the *Sensing Our World* workshop we led at the Scratch@MIT 2012 conference, showing how easy it really is to start making your own sensors and interfaces. Workshop participants built a variety of sensors ranging from simple tilt sensors to 'sliders' that relied on the conductivity of pencil tracks. We also demonstrated how to turn a CD case into a touchpad, and create a set of bottle top drums (amongst others).

During the conference we attended various sessions and workshops, and learnt how people were using Scratch in their classrooms along with the future plans for Scratch. The Scratch team have been hard at work with their next release, dubbed Scratch 2.0, and conference attendees had the opportunity to test it out, and ask the team about their design choices for the next version.

Scratch 2.0 will be web-based, with a downloadable standalone version, and has many new features and performance improvements. Pupils can now make their own blocks, clone sprites to create impressive displays, and use a webcam to interact with their Scratch projects. The webcam block, under the sensing tab, can detect motion and direction, and conference attendees made virtual musical instruments, a balloon pushing game and much more. The team hope a final release will be out towards the end of the year, with an open beta planned for September.

The Kinect2Scratch project is another impressive way to get pupils engaged with their projects using the Microsoft Kinect. Pupils can track the limbs of two players, and use the distance value to sense how far away they are from the Kinect. The drivers work seamlessly under Windows 7, but other versions are available, with work on fully developed version for Macintosh and Linux underway.

There were many broad themes throughout the conference, but one that stood out was introducing Scratch into the curriculum, and encouraging more pupils, especially girls, to take up Computer Science in their educational careers. The National Science Foundation is hard at work promoting their CS/10K project, which aims to bring a new CS curriculum in 10,000 high schools across the US, with 10,000 well-qualified teachers at the helm. The final keynote consisted of presentations from elementary pupils and teachers from a New York school district. The pupils showed their impressive Scratch projects and the teachers described how they had implemented the Scratch Curriculum in their classroom, which included using a comprehensive rubric for evaluating the pupils' performance.

It's fantastic to see the range of activities that people are undertaking with Scratch, and we're excited to see what the future holds for the platform. We're especially interested to see how people can continue to use the Picoboard to bring their Scratch projects into the real life. See the web supplement for links to the materials circulated at the Scratch@MIT 2012 conference.

Margaret Low, John Rendall, Marie Low & Phil How

USE SCRATCH TO CREATE YOUR OWN DATALOGGER

Many students will be familiar with using dataloggers in science, but how many have considered what goes on when using them? I like to think of myself as a scientist, so I've been writing some Scratch programs to collect and display data from one of the sensors on the Picoboard. This can allow their students to start to explore data. Collections of numbers mean more when you have collected them yourself.

Although I don't expect the students to understand all the aspects they should be able to understand much of it, and it can serve as an example of different aspects of Scratch programming (such as using list and pen code blocks). In fact the aspect that is most difficult to understand is probably the scaling involved in the plotting, which could be the subject of an interesting lesson.

I have also done a real scientific experiment by freezing a temperature sensor (a 99p thermistor from Maplin) into a block of ice and using one of the analogue inputs on the Picoboard to monitor the temperature as it melts. This might sound a bit boring, but it demonstrates something fundamental: phase change involves a change in energy. So when you take your ice block out of the freezer, it warms up rapidly to 0^0C, but then the temperature stays for quite a while at 0 while the ice melts: all the energy from the warm room is going into melting the ice, not into raising the temperature of the ice/water. In physics and chemistry students study phase change diagrams a fair bit, well, here's your chance to do the experiment yourself… All the information you need is on my web page and you can download the programs. The link is in the web supplement.

Myra VanInwegen

ENRICHMENT PUPILS EXPLORE 'AMAZING' LEGO MINDSTORMS

Once a year, all the pupils at Bay House School and Sixth Form, Gosport, Hampshire are taken off timetable for a week of enrichment activities. ICT and Computing teachers developed two activities for year 8. Outlined is a standard 60 minute lesson, although the activity could easily be customised to fit various time slots.

One group were introduced to a Lego Mindstorms robot. The robot was built on the model of an Explorer with a touch sensor, a light sensor, and 3 motors. Teams were made of 4 pupils and a robot. The objective set was to manoeuvre the robot through a simple maze into a garage. The teams explored several different methods for tackling the problem. In the first instance, they determined that they could traverse the maze by using distance travelled per second. Turning, based on angle and motor power, presented another set of problems. Some hints were provided for initial values. Using the feedback from their observations, the teams were able to park the robots. Providing their own differentiation, some teams turned the robot 180 degrees and retraced their steps back out of the maze. Other teams attempted to incorporate the touch sensor to detect bumping into walls. The room became quite lively when the ability to produce sound was identified.

Because of the limited time, complexity was added incorporating accuracy targets, such as returning exactly to the starting position, rather than programming complexity. At the end of the hour, of course, there was a competition to see who progressed furthest and extra points awarded for style. Feedback from the pupils was positive and included "challenging, but not impossible", "can't wait to do it again", and "mind blowing". In the new academic year, Mindstorms will be used in a half-term sequence of lessons based around problem solving.

Cynthia Selby

KEY STAGE 3 ROBOTICS CLUB THRIVES USING ARDUINO

Last January a robotics club started at Rodborough School in Surrey. Using an Arduino Uno starter kit to learn the basics the 24 regular attenders rapidly progressed to building their first robots.

This enthusiastic group of Key Stage 3 students have been meeting once a week to design, construct and program robots and, on occasion, plan how to use them for nefarious purposes. The students were complete beginners – no one had any programming experience beyond a term's work with Scratch – but they have learned to write programs for the Arduino board and have the eventual aim of creating robots good enough to compete in the Junior RoboCup (see below).

The first steps followed the normal 'hello world' of circuitry in making LEDs blink before writing more complex programs to control the lights with sensors. My only experience with electronics is some dabbling at home so parts of this have been a learning curve for me too. Fortunately one of our IT technicians is a dab hand so we had lots of support (and a trawl through the web returned a number of well produced tutorials, videos and books (including some excellent offerings from Make). The Arduino Uno Starter Kit by Flex comes with enough basic electronics to build rewarding, albeit small scale, projects that combine light dependent resistors, buzzers, buttons and sensors along with jumper wires and a breadboard so no soldering is needed at this stage.

Once the groups had built up enough confidence writing programs using the Arduino environment (looks like Processing and writes like C) they started to create their robots. We used the Magician Robot Chassis which is a balance between performance, durability and cost (and also within the size constraints for the RoboCup); the chassis comes as a kit which took just over an hour to piece together. The Arduino board by itself cannot power the two DC motors so each group needed a motor driver shield. These, like the Arduino board itself, can be handmade or bought already manufactured – for speed and reliability we opted for the latter. The Arduino programming environment is free and the total cost of the hardware so far has been £60 per group - not cheap but good value compared with proprietary robotics kits aimed at the education sector. For the dedicated student I'm convinced it provides a richer experience. Now over one term in, the students have started to program robots with basic manoeuvrability. The next step is to incorporate sensors to make the robots fully autonomous. Groups are investigating different strategies with some looking at ultra sound sensors, others touch sensors, IR emitter/detector pairs and so on. Over the long term we hope to create juniors and senior groups – with peer support, continuing the enthusiasm shown by the original members.

Matthew Walker

RoboCup Junior offers challenges to Primary and Secondary students up to 19 years old. There are three challenges, Soccer, Rescue and Dance which aim to emphasise co-operative problem solving. Launched 14 years ago, the annual competition was motivated by the ideas of Seymour Papert and his influential book Mindstorms. It has an international following. More details in the supplement.

BUILDING A DIGITAL CAMERA: A FIRST .NET GADGETEER PROJECT

Microsoft's .NET Gadgeteer kit provides an easy way into exploring how to control components through writing programs. Sue Sentance, shows how easy it is to get a simple application, in this case a digital camera up and running.

A .NET Gadgeteer project involves three distinct stages:

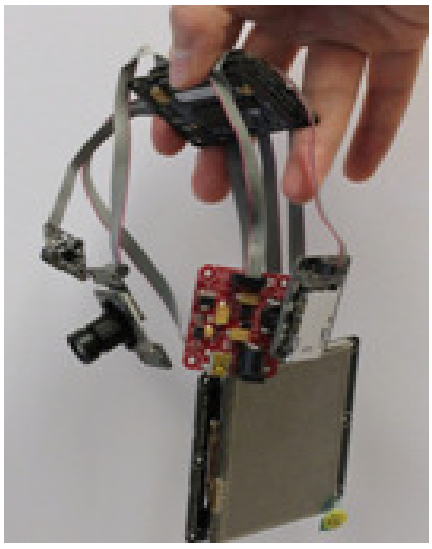
- Putting together the hardware components for the project
- Using the .NET Gadgeteer Designer to represent the configuration
- Adding lines of code in Visual C# (or VB) to make it work.

1: ASSEMBLE THE HARDWARE

Connect together these modules:

Camera
Mainboard
Power
Button
Display

Use the components letters to show you where to attach to the mainboard.



2: USE THE GADGETEER DESIGNER

Connect the modules together to match the hardware you have already assembled. To connect, either connect them manually by clicking the module in its connector, then clicking in one of the connectors that becomes green. See the image bottom left.

3: WRITE THE CODE

In the `ProgramStarted()` method, add two new event handlers for `Button.ButtonPressed` and `Camera.PictureCaptured`.

```
public partial class Program
{
    void ProgramStarted()
    {
        /*******************************************
        Access modules defined in the designer h

        e.g.  button
              camera1

        Initialize event handlers here.
        e.g. button.ButtonPressed += new GTM.HSR
        *******************************************/
        button.ButtonPressed+=

        // Do one-time tasks here
        Debug.Print("Program Started");
```

To do this very easily, enter += after the event name and then press the tab key twice. The <tab><tab> will fill in the rest of the line and generate the method itself. The result is shown in the code, below right.

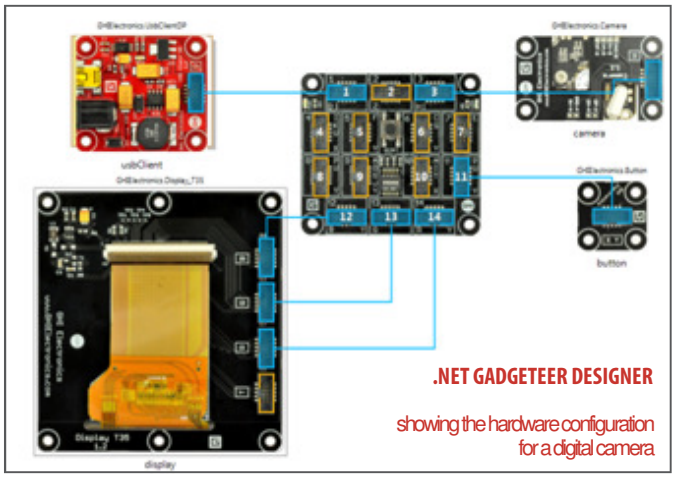
A WHOLE NEW APPROACH TO ELECTRONIC GADGETS

You can build all manner of electronic gadgets quickly and easily with Microsoft's new .NET Gadgeteer kit. It brings creating sophisticated gadgets within the realms of young children. There are an ever increasing variety of modules available to make a host of different projects. You'll find plenty of suggestions for getting started on the website. With 3D printers becoming more affordable, there is a real prospect that young pupils will soon be able to design , program and manufacture a complete electronic product!

Look carefully at the code. Inside the **button_ButtonPressed** method, delete the line to throw an exception, then add the code to take a picture. Similarly, inside the **camera_PictureCaptured** method, delete the line to throw an exception, then add the code to display the picture. Both lines of code are shown in the image at the bottom.

Press  on the toolbar to test the code works. You will need to wait a few minutes while the program loads. Then press the button on the camera and hey presto! Your picture appears on the screen. You have now put together and programmed your first digital camera!

A link to a detailed walkthrough of this project can be found in the web supplement. You'll also find a link to further lesson plans and details of hardware manufacturers from whom .NET Gadgeteer kits can be purchased.



.NET GADGETEER DESIGNER

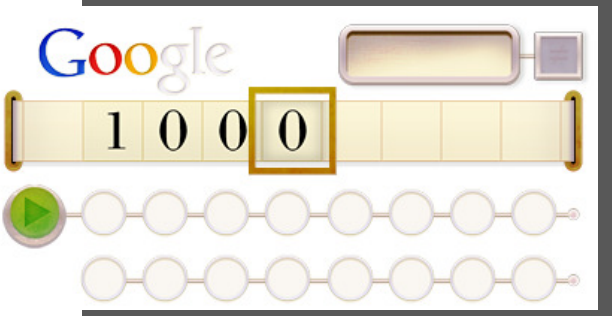
showing the hardware configuration for a digital camera

```
void camera_PictureCaptured(Camera sender, GT.Picture picture)
{
    display.SimpleGraphics.DisplayImage(picture, 0, 0);
}

void button_ButtonPressed(Button sender, Button.ButtonState state)
{
    camera.TakePicture();
}
```

ALAN TURING CENTENARY: CAN YOU SOLVE GOOGLE'S DOODLE?

The Google Doodle on Google's homepage on June 23rd celebrated Alan Turing's 100th birthday with an animated logic puzzle representing a Turing Machine. The introduction remarks "Alan Turing was a



completely original thinker who shaped the modern world, but many people have never heard of him. Before computers existed, he invented a type of theoretical machine now called a Turing Machine, which formalized what it means to compute a number. Our doodle for his 100th birthday shows a live action Turing Machine with twelve interactive programming puzzles (hint: go back and play it again after you solve the first six!)…. Turing's importance extends far beyond Turing Machines. His work deciphering secret codes drastically shortened World War II and pioneered early computer technology. He was also an early innovator in the field of artificial intelligence, and came up with a way to test if computers could think – now known as the Turing Test. Besides this abstract work, he was down to earth; he designed and built real machines, even making his own relays and wiring up circuits. This combination of pure math and computing machines was the foundation of computer science "

Once the first six puzzles are solved a further six become available. If you missed it, you can still access it via the Google archives (link in the supplement) and it provides an excellent enrichment or extension activity for students. The code itself is also available. The software engineers responsible state, "Please enjoy, and feel free to extend and improve the code. Can you think of ways to generalize the game? We have filed some starter issues to whet your appetite. If you are interested, hack away, and we'll happily review your changes!"

PROGRAMS CAN BE FAR MORE THAN JUST GAMES AND APPS

To develop pupils equipped to take up the future challenges posed in many disciplines, Aaron Sloman argues that we need to go beyond procedural programming and expose pupils to a wide range of different types of programs.

We impoverish our future if we teach only a small subset of programming languages, e.g. the ones based on sequences of instructions organised by loops and conditionals, enhanced by procedures, or even object-oriented programming. There are many other programming paradigms including rule-based programming, neural programming, evolutionary programming, constraint-programming, stochastic programming, logic programming, pattern-based programming, event-driven programming (e.g. interrupt handlers), functional programming, and programming in languages designed for particular domains. There is, I believe, a wide-spread failure to understand the kind of education required in a wide range of disciplines, including biology, neuroscience, psychology, education, linguistics, philosophy, mathematics, physics, economics, medicine, psychiatry and all branches of engineering. All need students who have experience of computational thinking - not just making entertaining 'apps' but trying to understand phenomena by building models and exploring consequences of changes to their design or parameters.

The universe contains matter, energy and information. Our understanding of matter and energy has acquired considerable depth over several centuries (especially since Newton, and the development of chemistry). But our understanding of information: what can be done with it, why it is needed, what forms it can take, what mechanisms can operate on it - is only beginning, with most of what we know having been learnt in the last 70 years, following the work of Alan Turing and others. For example, to understand how languages work, and how humans learn languages we need to be able to build and test models of language use. To understand how organisms develop from a fertilised egg, we need to be able to build and test models of the processes of development. To understand how mind and matter interact we need to be able to build and test working models of (simple at first, but increasingly complex) minds to show how the mental capabilities can relate to physical implementations.

There is a growing need for a broad-based education in computational thinking for all learners, not a computing education focused only on producing Computer Science students and IT employees. Our long term needs are deeper and broader. We may find a need for kinds of computing machinery very different from that we now take for granted. Some of these changes are already felt in the challenges of making use of multi-core CPUs, for example. But there's also neural computing, chemical computing, evolutionary computing, and other kinds to be explored in coming years. I am not the only person to express views like this. Paul Nurse, a Nobel prize winning biologist and now the president of the Royal Society has made it clear that the major unsolved problems in biology are concerned with information processing. There are many others. We need school-kids to learn a variety of types of computational thinking. See the web supplement for a link to my attempts to classify different types of computing teaching that need to be considered in this context, from primary school upwards. It's only a start. Suggestions welcome.

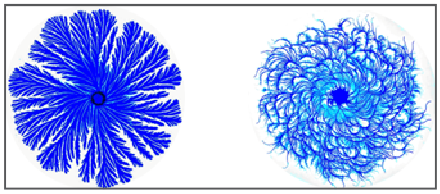
INTRODUCING THE NOTION OF NATURE INSPIRED COMPUTING

Consider these similar structures, one built in the conventional way (the entire process centrally planned), the other the end product of numerous local processes with no centralised control mechanism. Can you tell which is which?

One is the the unfinished cathedral of Barcelona by Antonia Gaudi and the other is a termite mound. The termite mound was produced by a colony in which each termite "executes" rules that specify only how it should interact locally within its environment. There is no central directing force that issues instructions according to some master plan or blueprint.

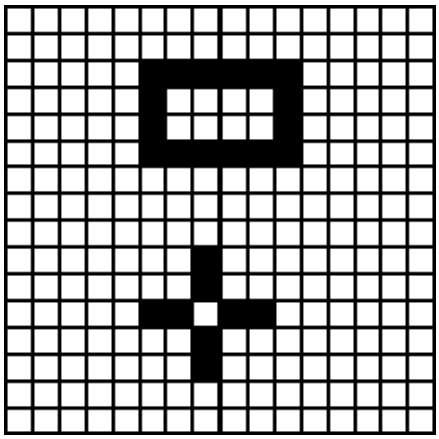
How does a bird flock keep its movements so graceful and synchronized? Most people assume that the bird in front leads and the others follow. In fact, bird flocks don't have leaders: they are organized without an organizer, coordinated without a coordinator. A surprising number of other systems, from traffic jams to economic systems, work in the same decentralized way. A pattern emerges at the global level solely from interactions among lower-level components, This process is called self-organisation.

Self-organisation can be exploited to sort a list of numbers as follows: Organise a group of students into a line. Place yellow and red cards on the ground in front of each student as shown in Figure 4. Give each student a card on which is written an integer. Give each student the rules set out in the algorithm shown below. If the list of numbers is to be placed in ascending order of size then each student must be told that they compare and swap their number cards (they don't physically move) if their neighbour to the left's card number is smaller than their own card number. Try tracing the algorithm yourself.



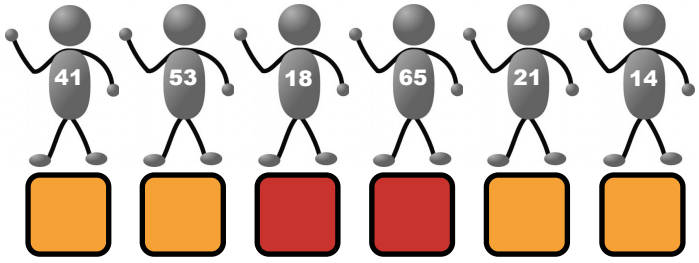
Nature is full of examples of self-organising systems. For example, colonies of bacteria are known to exhibit communal or social behaviour driven by local interactions that rely upon secreted signalling molecules for other bacteria to detect and to respond to. These signals enable the colony to exhibit communal behaviour. A colony of the same bacteria can grow into very different patterns (self-organise) under different conditions. The figure above shows two branching patterns exhibited by Paenibacillus dendritiformis bacteria for different levels of food. Different forms of self-organisation enable the colony to survive under different conditions, e.g. more food, less food. Data communication companies have also studied the living habits of bacteria colonies in order to help develop communication networks for low-powered smart devices that will self-organise and self-configure thus eliminating the need for the user to direct the configuration process of their smart device.

Cellular automata provide one way to investigate self-organisation. They are based on a computational model very different from the von Neumann architecture used in most computer processors today. In fact, cellular automata were invented and used by John von Neumann and Stan Ulam in 1948 to study reproduction in biology. Cellular automata like the diagram below, are made up of many independently operating cells embedded on a grid, each of which can affect only its neighbours. This resembles a



colony of simple organisms (e.g. bacteria) that can present amazingly complex behaviours or could model the structure of simple multi-cellular creatures that contain many cells working together. The Nature of Code course produced by Daniel Shiffman explores some of these ideas very successfully using the programming language Processing. Further details can be found in the web supplement. *Kevin Bond*

SORTING ACTIVITY TO ILLUSTRATE THE CONCEPT OF SELF ORGANISATION




```
Start with hands in the down position.
Repeat
  If hand down
    Then
      Do number comparison with neighbour of same colour(e.g. yellow)
         Swap if necessary
         Put hand up
    Else
      Do number comparison with neighbour of different colour
         Swap if necessary
         Put hand down
Forever or until exhausted
```

THORPE PARK INTRODUCES NEW COMPUTING WORKSHOPS

The Thorpe Park educational centre delivers over 500 presentations each season. These feature Physics, Maths, ICT and Business topics in our dedicated educational facilities at the Nation's Thrill Capital based in Surrey. Following a visit to the BETT exhibition at Olympia 2012, we were all struck by two things…

- The volume of costly and confusing re-courses available for teachers.
- The building momentum behind introducing computing into both primary and secondary education.

This informed our decision to develop materials to support teachers delivering a new computing curriculum. The aim of introducing new workshops was to explain the basics of computing from binary through to roller coaster control. We had already featured ride control using Data Harvest and K'Nex in our workshops. However, we found new uses for this equipment by teaming up with Mark Dorling and Stacey Jenkins from Langley Grammar School. With their help we developed stimulating activities to help both during the pupils visit and with free post lesson resources to support class teachers using binary and hexadecimal challenges stored on Twitter.

It was Mark's vision to combine the theory from the KS4 Computing At School Curriculum with a 'real' life hook of using how roller coasters work. Therefore, using a mixture of activities from The Digital Schoolhouse project and CS Unplugged, we have found new uses for the Computer Control equipment to demonstrate how algorithms, flow charts, programming code, and binary are relevant to a modern day roller coaster.

Thorpe Park have also invested in a small Mitsubishi Programmable Logic Controller which is the same type used to control the simpler aspects of the rides. We have asked our electrical engineering team to include this aspect of our computing sessions in their engineering workshop programme. See the web supplement for links to more details of the range of workshops available and resources to use with your classes.

Chris Chezney

INSPIRING AND INVOLVING STUDENTS IS KEY TO CHANGE

"I gasped at what Michael Gove said about my subject, but I feel he has given us an opportunity; one that I intend to use to its full." Julie Skeats, from Broadwater School in Surrey shares her thoughts on managing curriculum change.

I don't think that Broadwater School is aware or even realises that down the IT corridor a silent revolution is taking place. We are now truly on a mission embodied in our mission statement 'creating opportunities; fulfilling potential' and 'inspiring progress'. It's time for IT (or what should we call it…?) to be recognised and taken seriously. What would you do without your mobile phone or other household items, even down to the plastic we use to pay our bills.

We are starting a journey with our students to find out what they feel they should be learning in this social media age. Technology is rapidly changing the world around them. We are getting them to come up with the ideas to inspire each other, to recognise that IT is an important tool in society and that they will require considerable IT knowledge, skills and awareness if they are to be successful in the future. Social networking plays a large part in young people's lives so instead of using our usual VLE, we are trailing Edmodo as a learning platform and getting them to use their mobile phone in IT lessons. Edmodo has an App available that I am encouraging students to use in and outside of school.

The students are helping to re-write the curriculum to focus on technology they feel they need to learn about. We have started to spark their interest, making them realise that IT isn't just about using Office. There is now a curiosity for computer science as they want to learn more about creating computer games and App development. They want to learn more about animation and computer systems. We can do all this and still make sure they are confident standard software users through our scheme of work. We are going to use Scratch, Greenfoot and LiveCode to help with game and App development. We've created an area on Edmodo where they can post questions and get other students to help solve problems. We also run a computer build club for students where they build computers for the school. We will also be starting a computer programming club using Lego Mindstorms. Our aim is to help turn our users of technology into producers of technology. The best place to start is to inspire them, to make them believe in themselves. We need to change people's opinions and make them realise there's more to IT than using standard software. The Broadwater journey begins, one step at a time. It all starts by inspiring…

ANOTHER HUGE ENTRY FOR Animation12

The 5th annual UK Schools Computer Animation Competition, organised by The University of Manchester enjoyed another huge entry. There was a packed house for the Festival and Inspirational Computer Science Day held at the Martin Harris Centre. Following the awards ceremony, thirteen different workshops and activities brought Computer Science to life. Congratulations to all the winners, too numerous to mention by name. You can view their impressive work in the gallery. Registration for Animation13 opens in September. Start preparing now!

IT'S THE COMMUNITY DRIVING RASPBERRY PI DEVELOPMENTS

One only has to look at CAS to see that something very special can be created by the hard work and the goodwill of volunteers. As the saying goes, "There is no 'them'; there is only us." Such potential was not lost on Liz Upton of the Raspberry Pi Foundation.

"Building a community around the device was right at the top of my *To Do* list", she says. It was a sound investment: the Raspberry Pi community has just celebrated its first birthday (July 2012) and its members are an inspirationally eclectic mix of skills, interests and opinion. They are generous with their time too: however simple the problem (or however crackpot your latest RasPi project is!) you will get solid advice and encouragement.

Two outstanding examples of what a volunteer community can create are Raspbian and Raspberry Jams. Raspbian is an operating system which is optimised especially for the Raspberry Pi. "It just appeared out of nowhere when Mike Thompson, a fan in the US, decided he'd buy some ARM hardware and have a go at putting something together. Within a couple of months there was an enormous, self-propelling team working on it," says Upton. It has now been adopted as the official OS of the RasPi.

The Raspberry Jams are the brainchild of CAS member Alan O'Donohoe. These meet ups for Pi owners are now popping up all over the world. "It's all on the back of community work and community goodwill", says Upton. And we mustn't forget the soon to be published *Raspberry Pi User Guide* – CAS' own contribution.

Whilst the take up by the hobbyist and development community has been extraordinary, the Foundation hasn't forgotten its educational brief. The community forum has an Education section where you can discuss anything from pedagogy to classroom hardware, and share lesson resources and ideas. But it is, perhaps, the hundreds of more modest community projects that hint at the potential of devices like the Raspberry Pi to inspire creativity and to help teach computing. Recent front-page stories such as the Boreatton Scout Group and the Raspithon show what can happen when you give young people a low cost computer to mess about with.

The fact that young people are doing fantastic things in their own time shows, says Upton that there is "an enormous capacity and energy ... for learning about this stuff, and that we'd be idiots as a country not to capitalise on it by teaching them properly". With this in mind the Foundation is currently talking to UK exam boards about integration, and the UK Government about options for teaching computing in schools.

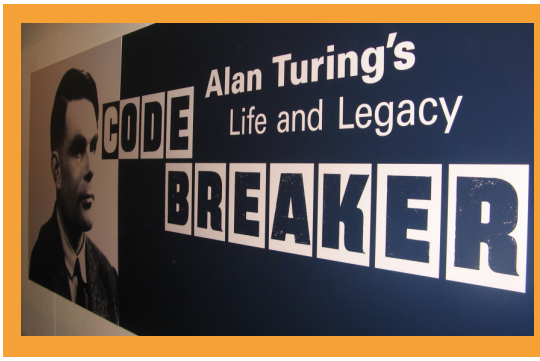
"We're just getting to a point where the distributors can make enough to supply schools and classrooms." says Upton, "Next year promises to be an interesting time." *Clive Beale*

Raspberry Jam

A global network of events for enthusiasts of the Raspberry Pi, Raspberry Jams are springing up all over the world. Some thirteen jams are currently being organised in the UK, with others further afield in the USA, Europe, Canada and Australia.

If there is no RaspberryJam running in your area, why not agree to host one. Venues range from schools, offices, museums, business parks, universities and branches of Starbucks. They cost nothing to setup but pay back huge rewards. You do not need a Raspberry Pi to attend, in fact attending a jam before you acquire one can help introduce you to people who can help you with all the questions you have. Jams attract people of all ages, from young school students to aging geeks! In Machynlleth, mid Wales, a Raspberry Jam was hosted and organised by school students from Ysgol Bro Ddyfi. You can watch their inspirational video of the launch event on YouTube.

Jams are different from CAS hubs, not being restricted to educational issues and drawing a wider cross section of the computing community. See the website link in the supplement where you will find details of all current activities. The site is the brainchild of Alan O'Donohoe, who also runs the CAS Preston hub, supported by Neil Ford from Rewired State. The informal movement of meet ups that can only grow bigger as more Pi's are distributed.



Codebreaker is an exhibition developed by the Science Museum, with support from Google. Alan Turing was not just a codebreaker though. He was also a philosopher and computing pioneer who grappled with the fundamental problems of life itself. Alongside the Pilot ACE centrepiece - a computer built to his design - are exhibits showcasing Turing's breadth of talent. Arranged in six sections, artefacts, personal recollections and historic imagery provide a succinct introduction to the pioneering work one of Britain's greatest twentieth-century thinkers. The exhibition is free and will run until July 2013.

THREE DAY SYMPOSIUM ON COMPUTATIONAL THINKING

Our "Three Counties" (Worcestershire, Herefordshire and Gloucestershire) CAS hub has had a busy year. In early July, Worcester University organised a 3-Day Symposium for some 30 teachers to reflect on teaching Computer Science at GCSE level. The symposium was funded by a generous grant from Google as part of their "Computer Science for High Schools" (CS4HS) programme.

Sessions run by University staff and CAS members followed three themes of "Which Programming language should we teach?", "Tried and tested pedagogical approaches to teaching Computing" and "What elements of theory should we teach and how?" There was significant planning between the University and the CAS hub to meet the needs of delegates. Participating schools sent at least one teacher with no experience of programming so we could build confidence in this daunting area.

Our pedagogical approach proposed teaching students to be *creative* rather than traditional problem-solving or numerical approaches which tend to be *restrictive*. Thanks to the Google grant we gave each school a Lego Mindstorms robot and a "Worcesterino" computer (left), designed by the University's Computing department to allow teachers to explore teaching programming away from a pc whilst reflecting on the nature of programming. A discussion of algorithms considered real-world activities, such as how to apply nail varnish, how to tie a shoe-lace or win a simple card game to help develop strategies for writing computer code.

Dr. Colin Price, Head of Computing at Worcester University commented, "I found the engagement of the delegates in all workshops and associated discussions truly inspiring … This is a wonderful opportunity for the University to engage with our local community". We still have much to do, and look forward to the "Network of Excellence" to further support our teachers needs. *John Palmer*

PLANNING CPD: HOW BEST TO SHARE THE KNOWLEDGE

Many ICT teachers are finding, since Michael Gove's announcement in January, that they have to quickly add "can teach Computer Science" to their CVs. Sue Sentance shares her insights organising many successful CPD sessions.

At hub meetings and the CAS forum are many requests for help with continuing professional development (CPD) in Computer Science. Teachers' needs vary but their existing commitments don't: all teachers are busy and find it hard to make time for training. Schools also seem to be less willing than ever to release teachers for courses. However, teachers are very keen for training, even giving up substantial amounts of their own time to achieve it. Adam McNicol, Sophie Baker and I initiated a series of KS4 CPD courses in our local area. These have proved very popular, generating many more enquires than places available. We are, busy schedules permitting, planning to extend our training offer to include KS3 and KS5. Many other universities are also starting to offer excellent courses in their localities.

In planning CPD opportunities, there are many variables. Should training focus on subject knowledge alone, or also address pedagogical issues? Should it be accredited (at further cost for teachers)? When should it take place – twilight sessions, evening classes, weekend classes, summer schools, residential courses, or online courses? Who is best positioned to offer Computer Science upskilling, and, of course the key variable - who funds this upskilling?

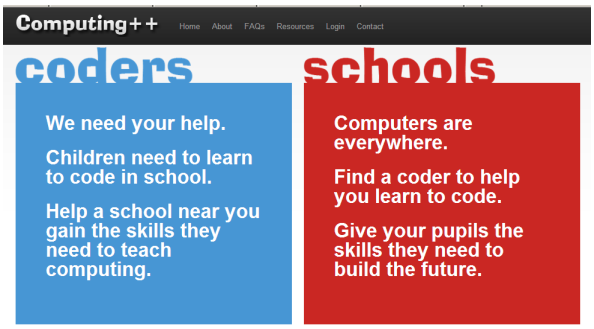
There is much research in the area of effective CPD and its impact. *'Transformative'* CPD can be achieved by a combination of types, including training, establishing communities of practice, action research initiatives, and coaching (Kennedy, 2005). At Anglia Ruskin University and in Essex, we have found it very rewarding to be involved with our local ICT teaching community. Each course becomes a community of practice of its own, with colleagues eager to share with each other and build up links that last long after the 10-week course. We are keen to include current experienced teachers in the delivery of our courses as much as possible. I would recommend other Education/CS departments get involved with CPD in this way, especially in the context of proposed changes to initial teacher training delivery. If you think you could support other teachers, then you might be able to get involved in delivering CPD in your local area. We are conducting some research into which models of CPD teachers find most useful for Computer Science. Please help us by completing a quick questionnaire via the link in the web supplement.

NORFOLK CAS HUB PRODUCES CPD RESOURCES

School governor and Norfolk hub leader, Neil Collins has produced a set of resources for use in hub meetings. "Initially these were just for forms development in Visual Basic " commented Neil, "but after trying them out with a few teachers and students, I've now added Small Basic and Visual Basic for Console/Command Line applications." These simple step by step resources are available online for any teachers to use. You'll find the link in the web supplement. Neil would welcome feedback and questions via the CAS forum.

COMPUTING++: TAPPING THE ENTHUSIASM AND EXPERTISE

Bridging the computer science skills gap that currently exists in many schools is a challenge. School governor, Martin Saunders sketches the background to an exciting initiative that aims to bring schools and IT experts together.



I am trying to help teachers gain computer science skills through local expertise via Computing++. I jumped on the band wagon when I taught a workshop at my own primary school. The day was a roaring success and more than anything, I came away with a really strong sense that kids need more of this kind of thing. As I thought about it, I couldn't get away from the idea that this was something that should go a bit further than one school. I happen to know a couple of people at Intel and was put in touch with Tim Hatch who looks after their UK educational things. Tim gave me a crash course in the world of ambassador schemes, CAS and so on. I'd already come across projects like Code Club and Young Rewired State and realised that there was a whole lot of great work going on already. What struck me though, was that it will always be down to regular teachers to deliver the vast majority of lessons if a national computer science curriculum becomes reality. So, ok, sure. Spread the word that this is important. Win the hearts and minds of the nation, great! But how many teachers know how to code? Erm…and in primary schools?

I realised there were two things I could help with. The first, providing a means for 'normal' teachers (especially primary) to gain the confidence and skills with computing to be able to deliver lessons. Secondly, a common point of contact for companies ambassador schemes would help to align everyone and help allay cynicism about big companies doing good things. I knocked together an initial prototype for the system and had a great initial response from CAS.

So, what is computing++? In a nutshell, Computing++ is a web based portal that allows schools and people who understand computing to connect with one another and suggests matches based on profile to help nudge people together. I have a load of other ideas for the site but the ability to collect people is the most important feature so once that was functional I went live, all agile like.

Computing++ is for schools (primary and secondary) that want to improve their computer science skills. It's also for schools that already teach computer science and want to share their experience. It's also for anyone who understands coding, computational thinking or how computers and the internet work and want to help nearby schools learn these things.

When people first link up the first thing is to work out what the school needs. Remember that the end goal is to enable teachers to deliver computer science lessons. Primarily the expert will be working with a teacher NOT pupils. A great place to start is to teach the teacher a programming language. Python and Java seem to get used a lot), but it should be up to the teacher to steer the topic and the expert to fill in the gaps. See the web supplement to find out more about getting involved or, if you're a teacher and stuck, get help.

Martin Saunders

GOOGLE TO HELP RECRUIT COMPUTING TEACHERS

Google Executive Chairman Eric Schmidt recently announced a partnership between Google and Teach First, to put 100 more STEM teachers in the classrooms of schools in disadvantaged areas over the next three years. 50% of these teachers will be supported in the teaching of Computer Science. To ensure that the ICT teachers, and their pupils, have access to the latest cutting-edge technology, Google will also provide each teacher with a bursary to fund the purchase of innovative teaching aides to help inspire their future classes.

GENERATING GENIUS

Google is also sponsoring and partnering with Generating Genius this year to help them extend their programme and increase the computer science content for it's students. Generating Genius works with high-achieving students from disadvantaged communities throughout their secondary school careers to help them acquire the skills they need to win places at top universities in technical fields.

Earlier this year around 200 students from 8 schools (80% of whom were girls) took part in a SketchPatch.net creative programming workshop. All of the students had no experience or exposure to coding but were keen on maths. They were nominated by their schools based on their grades in maths and science. As a result of the workshop 70% of the year 9/10 cohort said they would like to pursue Science A 'levels and 75 per cent said they would like to take up a career in Computer Science. To find out how your school can get students involved in the Generating Genius programme, get in touch via their website - you'll find further details in the web supplement.

Alison Daniel-Cutler

AS CAS HUBS GROW TEACHERS ARE RARING TO GO IN WALES

The last six months have been truly exciting for computer science education in Wales. We approach the start of the new academic year with a clear declaration from the Welsh Government, with Leighton Andrews AM, Minister for Education and Skills, stating in his keynote at the 2012 CAS Wales/Technocamps Conference in June: "Computer science touches upon all three of my education priorities: literacy, numeracy and bridging the gap. It equips learners with the problem-solving skills so important in life and work...It is therefore vitally important that every child in Wales has the opportunity to study computer science between the ages of 11-16.".

The Welsh Government have committed a further £3m investment in computer science and digital literacy (see right). This is the largest single investment in computer science education in the UK to date and certainly puts Wales in a prime position over the next couple of years.

I have been working closely with the Department for Education and Skills over the past few months, as well as the Office of the Chief Scientific Advisor, to further the CS education agenda in Wales, with hugely positive responses. While we await to see how this will affect schools and teachers in Wales over the coming academic year, there has been a clear commitment to investment in CPD and initial teacher training. Furthermore, while much of the recent policy focus has been on KS3 and GCSE, there are also initiatives to develop computer science in primary schools, particularly in South East Wales.

Finally, we now have five CAS Hubs operating across Wales (Cardiff, Swansea, Aberystwyth, Wrexham and Bangor), as well as the start of activity in Powys. These are truly exciting times, roll on 2013! As always, much of the activities of CAS Wales have been done in partnership with the Technocamps project, so a big thanks must go to Faron Moller and Stuart Toomey at Swansea University for their input. See the supplement for further information regarding the activities of CAS Wales.

Tom Crick

WALES GETS £3 MILLION TO DEVELOP COMPUTING CPD

We looked forward to a keynote by Maggie Philbin and a speech by Welsh Education Minister, Leighton Andrews at the CAS Wales / Technocamps conference in Swansea. But nobody was prepared for the announcement that followed.

There had been rumours before the conference that the Minister was going to announce something significant. A task-and-finish group put forward a 10-step plan for education in a Digital Wales last March. £7 million has been put aside for the delivery. Most excitingly, £3 million has been earmarked for "Additional professional development for teachers and other education staff to support the teaching of computer science and IT, building on the new enthusiasm around the development of products such as the Raspberry Pi and Dot Net Gadgeteer (sic) to encourage young people into future studies and careers in computing."

I'll state that again. £3 million for the additional professional development of teachers to support the teaching of computer science in Wales. I sat in the auditorium before the Minister came in ready to ask loads of questions, but they were answered before I'd even put my hand up. Having had some time to consider, some questions remain. For example it isn't clear yet how the £3 million will be broken down. Nor do we know yet who will deliver the CPD or how schools will access the opportunities. The Welsh Government has launched many funding streams in the past but schools usually have to bid for the money. The bid process can be time consuming and some schools / teachers may miss out if their Senior Management don't see Computer Science as a priority. Despite all secondary schools receiving the CAS information pack this may still be the case. Many of the funding streams we have had access to in the past have had restrictions on how the money can be spent. For example, it can only be spent on software, or hardware, or with specific organisations despite every school having different needs.

Here is where I think schools may need to spend the money:

- Taking teachers) off time-table for a short period on a regular basis to attend a training course or internet-based live training. Given the distances between some rural schools in Wales, weekly training at a bricks and mortar location may not work for some.
- Giving teachers time to bring their skills up. Skills to teach CS will come, not just through training courses but also through time spent programming and practising.
- Buying hardware/software and infrastructural requirements. Most schools have pretty locked-down networks as far as students are concerned. One requirements of learning computer science is that pupils will have to create and run programs. Network technicians MAY need support and guidance to make sure this happens safely and effectively on the school network. Will there be training available for them?

I'm hopeful that this funding is going to lead to an exciting time for CAS in Wales. At the time of writing, it's not entirely clear how this will all pan out but we can say for certain that we have a real opportunity to turn excellent ICT teachers into excellent Computer Science teachers. In 3 years' time Wales could be leading the rest of the UK in Computer Science education in schools.

Lucy Bunce

GREENFOOT TRAINING AND FIRST HUB MEETING LAUNCHES CAS NI

The summer term saw the official launch of the CAS Northern Ireland hub, with two events in quick succession. Clarke Rice, leader of CAS NI, reports on a highly successful Greenfoot workshop and the inaugural hub meeting.

During the summer term I hosted a one-day Greenfoot training course at my school, Loreto College in Coleraine. This built on material from the University of Kent and was a tremendous success. Thanks to word-of-mouth and Twitter advertising, a dozen teachers (pictured above) from all across Northern Ireland came along. I've used Greenfoot for the CCEA GCSE Unit 2 games task. Using Greenfoot in the gaming task has given students a good knowledge of underlying Computer Science principles, and has helped fuel student demand for further study of CS. You can read more about my experiences delivering Greenfoot at GCSE in a nine part inaugural contribution to my blog, which I hope will become a place to reflect on the challenges facing computing teachers here in Northern Ireland.

Teacher feedback for the Greenfoot day was unanimously positive, with most participants agreeing they would incorporate Greenfoot into KS3 or KS4 schemes of work. A number of teachers who were unable to attend, due to examining commitments, have expressed an interest in this course running again. Details will be posted on the CAS mailing list in due course. Thanks are due to the University of Kent, and in particular Neil Brown, for help and advice, as well as course materials.

A few days later the CAS Northern Ireland hub met for the first time. Thanks are due to our hosts, Professor Gerry McAllister and the University of Ulster at Jordanstown. Of the 35 who attended, most were teachers of ICT or Computing, with an encouraging representation from FHE, industry and exam boards. Gerry opened by outlining the case for Computer Science, and why ICT is an inadequate preparation for further study of Computing. Teachers responded by sharing their experiences, of schools moving to ICT at the expense of Computing and discussing how to run both subjects to complement each other. Representatives from OCR and AQA discussed their GCSE Computer Science courses, with CCEA discussing their forthcoming A-level in *Software and Systems Development.* It was good to see C2k (the managed network provider, to all schools) represented and willing to positively discuss how to facilitate a range of software development tools. Ian Simon's obvious enthusiasm for programming, as he discussed his *Go Berzerk!* HTML and CSS programming book has rubbed off on many who were there!

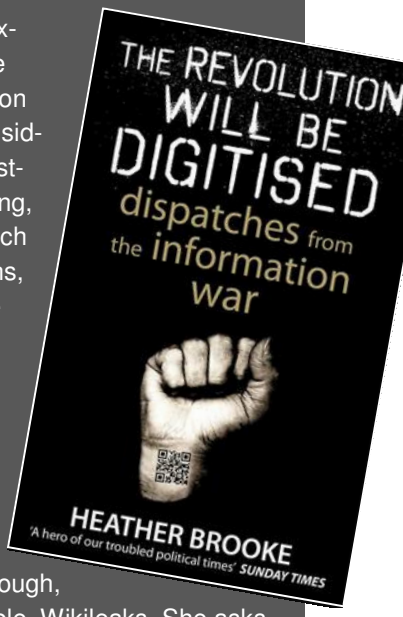
Leading on from this, a mailing list specifically for teachers in NI has been set up, to complement the main CAS mailing list. All ICT and Computing teachers in Northern Ireland, and other interested parties, are welcome to sign up - details in the web supplement.

Clarke Rice

MY FAVOURITE BOOK

The Revolution Will Be Digitized is written by Heather Brooke, an award winning freelance journalist and Freedom of Information campaigner. She is renowned for uncovering the MPs' expenses scandal, and author of 'Your Right to Know' and 'The Silent State'.

In The Revolution Will Be Digitized, Brooke examines the Establishment: governments, corporations and influential individuals who know more about us and have more access to our data than ever before. Opposing them are the hackers, activists and pro-democracy campaigners who don't accept Establishment authority and seek to protect our individual freedoms.



Brooke explores the 'Information War', considering Western hacking, free speech revolutions, corporate involvement and industrial and political espionage exemplified through, for example, Wikileaks. She asks critical questions about the conflict between freedom and security and discusses issues of privacy in an online world. She questions whether the internet will deliver freedom or censorship, surveillance and oppression?

These are questions every one should ask before posting data online. This book is an incisive antidote to the current trend in over-sharing, encouraging each of us to ask what our data is worth. Not only a book for teachers, but students too.

Lyndsay Hope

A PAUSE FOR THOUGHT

In 1978 Ron Rivest, Adi Shamir and Len Adleman - of RSA Security - introduced Alice and Bob, as fictitious characters to explain how RSA encryption worked. Using names made a complex subject easier to grasp. In 1984, security expert John Gordon explained that "Bob is a subversive stockbroker and Alice is a two-timing speculator" who have never actually met. In hundreds of papers written involving Alice and Bob, a vast array of scenarios have been constructed including defrauding insurance companies and exchanging encrypted messages over phone lines bugged by Secret Police and tax authorities.

When additional players were needed, Carole and Donna were introduced, with further characters being added when required, often with clearly defined roles. Bruce Schneier, author of *Applied Cryptography*, introduced a table of "dramatis personae". Gradually the cast expands to include Eve, the eavesdropper who passively intercepts communications through wire-tapping and capturing private emails. Then there's Mallory, a malicious attacker who can modify or substitute with her own messages, providing far greater challenges than those presented by Eve. Justin is the character used to represent justice and Plod the policeman, tracking the hackers in pursuit of Alice and Bob's communiques. Trent and Walter are frequently used to describe a trusted arbitrator and a warden whose roles vary with the protocols described; however there are many more characters called to illustrate these scenarios. Today their use has extended through computing, maths and physics into popular culture including television, cartoon, music and comics. *Lyndsay Hope*

AN INDISPENSABLE FREE RESOURCE

If you're not already a reader of Computer Science For Fun (cs4fn), now is a great time to get acquainted. It's a free magazine, published twice a year, that brings stories of cutting-edge computing research to school audiences.

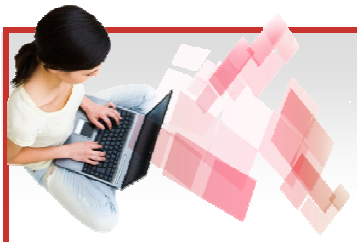
Each issue follows a theme like space or fashion, and shows how computer science plays a part in each. Students see how computing drives our everyday life and culture, not just our smartphones. (We write plenty about smartphones too, though.) You can order free copies for your school. Many teachers subscribe to get a copy for themselves, from which to pull lesson ideas. Others order enough copies to give to every student in their class. It's up to you.

We also write magazines about audio engineering and electronic engineering too. Check them out while you're on our website. Both aim to show how computing plays an important part in many technological innovations.

Our side project, The Magic of Computer Science, takes simple magic tricks and explains the maths and computing principles that make them work. You and your students can learn magic as a way of reinforcing fundamental computing knowledge. You'll find directions and explanations for loads of tricks, and our two magic books available for download.

It all began in 2005 at Queen Mary, University of London. Two computer science academics, Paul Curzon and Peter McOwan, joined forces to produce free material that excited young people about computing. Still produced at Queen Mary, today cs4fn magazine has thousands of subscribers in more than 80 countries throughout the world. Paul and Peter, along with other colleagues, also visit schools to give talks about computing and electronic engineering. Further details in the web supplement.

Jonathan Black



COMPUTING AT SCHOOL

EDUCATE · ENGAGE · ENCOURAGE

Computing at School was born out of our excitement with the discipline, combined with a serious concern that students are being turned off computing by a combination of factors. **SWITCHEDON** is published each term. We welcome comments, suggestions and items for inclusion in future issues. Our goal is to put the fun back into computing at school. Will you help us? Send contributions to **newsletter@computingatschool.org.uk**

Many thanks to the following for help and information in this issue: *Clive Beale, Miles Berry, Jonathan Black, Kevin Bond, Neil Brown, Lucy Bunce, Chris Chezney, Mark Clarkson, Neil Collins, Tom Crick, Claire Davenport, Roger Davies, Laura Dixon, Peter Donaldson, Mark Dorling, Claire Griffiths, Lyndsay Hope, Phil How, Toby Howard, Simon Humphreys, Sasha Laundy, Margaret Low, Marie Low, Alan O'Donohoe, John Palmer, Simon Peyton-Jones, John Rendall, Clarke Rice, Martin Saunders, Cynthia Selby, Sue Sentence, Julie Skeats, Aaron Sloman, Neil Smith, Myra VanInwegen, Matthew Walker, Lin White, Victoria Worpole..*

www.computingatschool.org.uk

Computing At School are supported and endorsed by:

bcs The Chartered Institute for IT
Enabling the information society

Microsoft Research

Google

CPHC
The Council of Professors and Heads of Computing