COMPUTING AT SCHOOL
EDUCATE · ENGAGE · ENCOURAGE

SWITCHEDON

COMPUTING AT SCHOOL NEWSLETTER SUMMER 2012

GETTING TO GRIPS WITH COMPUTING

What a difference a term makes! ICT departments up and down the land will remember the start of 2012 as heralding a sea change. Last summer the tide was on the turn. The remarks of Eric Schmidt, Google’s Executive Chair, at the McTaggart Lecture about the state of Computer Science education touched a nerve. In January, Michael Gove used his keynote speech at BETT to say that “Computer Science is a rigorous, fascinating and intellectually challenging subject”, and one that he wanted schools to teach. Then in quick succession AQA, Edexcel, and WJEC have all announced new GCSEs in Computer Science for first teaching in September 2012.

Mr Gove’s speech was critical of ICT, and announced a consultation on “disapplying” the National Curriculum in ICT. For ICT teachers, it is easy to become disheartened, or to ’shut down’. The Royal Society report, Shut Down or Restart (see right), published in January took a different view. It recommended restarting; recognising the discipline as Computing, within which clearly defined strands could guide students towards future progression. Out was the endless repetition of low level skills, characteristic of some qualifications and castigated by Ofsted. In was a recognition of the pivotal role Computing should play in developing the next generation of science and technologists. We are, in short, re-inventing our subject.

CAS welcomes the Royal Society’s suggestion and wholeheartedly endorses the other recommendations of the Royal Society report. Our consultation submission however (a link is in the web supplement), also highlighted the need for explicit endorsement and support of Computer Science from the DfE, lest schools misinterpret Mr Gove’s announcement as a signal to ‘shut down’ ICT. Computing is back on the agenda. Its educational value is gaining widespread recognition. This summer ICT teachers will start planning for next September. If you are new to Computing, we hope this issue will inspire you to ‘give it a go’!

ROYAL SOCIETY REPORT: SHUT DOWN OR RESTART?

“The current National Curriculum in England lumps together a range of aspects of Computing including Computer Science, Information Technology and digital literacy under the heading ‘ICT’.” So says The Royal Society Report into Computing in UK schools, the culmination of an 18 month study, published in January. It continues; “A consequence is that Computer Science is often forgotten or ignored within the heading of ICT, resulting in teaching being biased towards ‘how to use office software’ rather than the knowledge that will form a foundation for the rest of a pupil’s life.” The report acknowledges many of the advances made in developing digital literacy in recent years but goes on to say; “The UK has a proud history of contributions to Computing and especially Computer Science, but current terminology, curricula and qualifications are holding us back from being a nation of technology creators.” It makes 11 specific recommendations. CAS welcomes every one. As we’ve argued since our inception, a reboot is long overdue.

Shut down or restart?
The way forward for computing in UK schools
January 2012

EXCELLENCE IN SCIENCE

The Royal Academy of Engineering

THE ROYAL SOCIETY

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

TEACHERS REAP BENEFITS OF GETTING TOGTHER IN CAS HUBS

The explosion of interest in Computing that has followed the developments earlier in the year is demonstrated by the growth of new local CAS hubs. New hubs are forming up and down the UK as teachers gather to discuss the way ahead. With over thirty now up and running, there are, quite simply, too many to mention. They act as a wonderful counter to the isolation teachers can often feel if they perceive themselves as ploughing their own furrow, Finding like-minded teachers generates enthusiasm and breeds confidence, not least because staff realise there is no 'them', only us. It's very easy to read forum posts and think everyone knows more than you do. We don't! Meeting face to face is a great way to dispel such feelings.

A particular feature of the last term has been the plethora of CPD sessions being organised through hubs. Coventry planned a teachers day at Warwick University to assist in planning for GCSE. Three workshops in using Greenfoot were organised at Bruntscliffe School in Leeds. The Kent hub ran two sessions on using Small Basic and organised a Scratch session for Primary teachers. UWE, Bristol is offering 3 sessions in June for teachers on C, Java and Python. West London are following the lead of East London in organising Python tutorials. The list goes on ..... and on. CAS hubs have tapped a rich seam of enthusiasm and camaraderie which, for many involved, is quite unique.

SCHOOLS START PREPARING THE GROUND FOR SEPTEMBER

It's been a hectic time for teachers following Michael Gove's announcement in January that the ICT programme of study will be disapplied from September. Taken alongside his encouragement to schools to start teaching Computer Science, this gives the opportunity to radically rethink what we are teaching. CAS members have been debating how best to deliver a new curriculum within the Computing framework outlined by The Royal Society.

It's difficult not to get carried away by the tide of excitement that has developed around computing. What started as a trickle of dissenting voices has turned into a flood. Get computing back on the curriculum, they chorus. The Guardian campaign continues to highlight the need to 'teach our kids to code'. Online courses have thousands signing up to give coding a go. Ever more widespread is the recognition that we either program, or be programmed.

But amidst the heady excitement, many teachers, if they're honest, would admit to a degree of trepidation and uncertainty. CAS has come a long way in lobbying for change. Making that change a reality will now depend now on harnessing the energy, enthusiasm and commitment of those on the ground. Not just those in CAS, but colleagues in their departments and neighbouring schools. Not just teachers, but parents, governors and supporters. CAS hubs are exploding. Ever increasing numbers are embracing the challenge, and joy, of trying to get kids coding. It's a breath of fresh air, for staff and pupils alike. Such enthusiasm is a joy to see ... but it can easily dissipate. It would be foolish to deny the challenges that remain. Two tasks in particular confront those at the sharp end of curriculum change.

First off there is next year's curriculum. A new scheme of work can't be written overnight. Change has to be managed, and be manageable. One model of development might be take existing elements of the ICT curriculum and look for ways to 'computerise' them. Secondary schools that followed the old QCA scheme probably started Year 7 with 'All About Me'. Why

TEACH GCSE COURSE SUCCESS

A 10 week evening class to teach GCSE Computing, organised by Anglia Ruskin University has already had over 50 teachers attending. A third full cohort has just started, highlighting the interest shown by schools at wanting to get to grips with computing. The teachers are really enthusiastic and most are looking at introducing GCSE Computing either in September 2012 or 2013. There are online supporting materials for the theory and the programming (see supplement). Sophie Baker, who teaches GCSE Computing leads the theory part of the course. The model of the course is one hour theory, followed by one and a half hours programming.

UNIVERSITIES CAN PROVIDE COMPUTING CPD ...

On Wednesday, 21st March, twenty three teachers from Berkshire, Oxfordshire, Surrey and Sussex, took part in a Continuing Professional Development course at Reading University. The course was a half-day pilot of teaching materials that are being trialled for a CAS CPD syllabus. Peter Marshman lead a session using paper props as an alternative to computers. He followed this with programming classes in RoboMind and Build Your Own Blocks. James Anderson gave bite-size information about Computer Science and the History of Computing. He also demonstrated Second Life, both as a medium for the virtual delivery of CPD, and as a medium for whole-class lessons and self-study by pupils. Pilots of virtual classes are already under way and will soon be followed by piloting a virtual CPD course. Next term there will be a longer, one or two day CPD workshop at Reading University. For details of forthcoming events, contact Dr Yota Dimitriadi, Institute of Education, Reading University. *James Anderson*

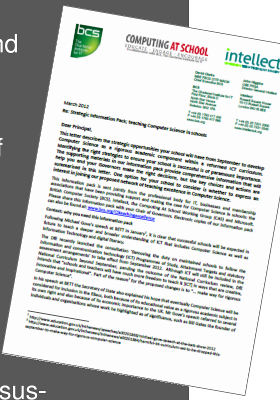
> “We aspire to an outcome where every primary school pupil has the opportunity to explore the creative side of Computing through activities such as writing computer programs (using a pupil-friendly programming environment such as Scratch). At secondary school every pupil should have the opportunity to work with microcontrollers and simple robotics, build web-based systems, and similar activities. We recognise that not all pupils will wish to seize these opportunities, but they should be able to do so if they do wish to.”
>
> *Professor Steve Furber, Royal Society Report*

not start with the same context, but use Scratch rather than PowerPoint? What better way to see what pupils are bringing from their Primary schools? Similarly, many schools already teach some basic web design using html, so maybe step it up to include some simple JavaScript. Not sure about all those squiggly brackets? Maybe introduce a unit using RoboMind. Whatever you choose, make small changes and do them well. It’s worth remembering that, in the initial phases of change, units can be used across several years. Three units introducing computing are just as valuable to Year 9, if they have had none before, as they are to Year 7, and staff can gain a lot from observing the differences in comprehension at different ages. The next year, more can be added and a scheme rolled out gradually over three years. There is, quite simply, a whole pedagogy to be rediscovered.

Therein lies the second major challenge. Computing isn’t just coding. Computing is a set of concepts that shapes a way of thinking. That way of thinking finds an expression in code, but not code alone. If you are designing web pages, look also, for example, at algorithms for search engines and page ranking. Linking the practical to explaining how the same concepts shape the everyday technology that pupils take for granted is key to grasping the power of computer science. In this task, thankfully, teachers are not alone. The pioneering work done by projects such as CS4FN, Unplugged and CS Inside means teachers have many ready made activities to draw on. Time spent exploring concepts makes teaching coding a richer experience.

EXCELLENT START TO COMPUTER SCIENCE TEACHING NETWORK

Before Easter all English secondary head teachers received a pack making the case for teaching computer science and urging them to express an interest in joining a Computer Science teaching network. Organised through the BCS Academy and sponsored by Microsoft, Google and Intellect the pack aimed to brief school leaders about the changes coming in the light of The Royal Society Report. Included is the Report’s Executive Summary, the CAS Computer Science Curriculum, a suggested framework for teaching Computer Science and IT and copies of both SwitchedOn and CS4FN. All secondary schools will now be aware of the support that exists to launch them on their curriculum journey. The letter invites schools to join an emergent Computer Science teaching network.



bcs The Chartered Institute for IT

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

intellect

David Clarke
MBE FBCS CITP MACM
Chief Executive BCS

BCS
The Chartered Institute for IT
First Floor, Block D
North Star House
North Star Avenue
Swindon SN2 1FA

T +44 (0) 1793 417 430
F +44 (0) 1793 417 [illegible]
www.bcs.org

John Higgins
CBE FRSA
Director General Intellect

Intellect
Russell Square House
10-12 Russell Square
London WC1B 5EE

T +44 (0) 20 7331 2000
F +44 (0) 20 7331 2040
www.intellectuk.org

March 2012

Re: Strategic Information Pack; teaching Computer Science in schools

Dear Principal,

This letter describes the strategic opportunities your school will have from September to develop Computer Science as a rigorous academic component within a reformed ICT curriculum. Identifying the right strategies to ensure your school is successful is of paramount importance. The supporting materials in our information pack provide comprehensive information that will help you and your Governors make the right decisions, but the key choices and actions are summarised in this letter. One option for your school to consider is whether to express an interest in joining our proposed network of teaching excellence in Computer Science.

This information pack is sent jointly from the professional body for IT, businesses and membership associations that have been providing support and making the case for Computer Science in schools: the British Computer Society (BCS), Intellect, the Computing At School Working Group (CAS) and Microsoft. Please share this information pack with your Chair of Governors. Electronic copies of our information pack can also be found at: www.bcs.org/CSteachingexcellence

Context: why you need this information pack

Following Michael Gove’s speech at BETT in January[1], it is clear that successful schools will be expected in future to teach a deeper and broader understanding of ICT that includes Computer Science as well as Information Technology and digital literacy.

The DfE recently launched the consultation ‘Removing the duty on maintained schools to follow the information and communication technology (ICT) Programmes of Study, Attainment Targets and statutory assessment arrangements’ to take effect from September 2012. Although ICT will still be included in the National Curriculum beyond September, pending the outcome of the National Curriculum review, DfE intends that “schools and teachers will have much more freedom to teach it [ICT] in ways that are creative, innovative and inspirational”. Part of the reason[2] for the proposed changes is to “... make way for rigorous Computer Science”.

In his speech at BETT the Secretary of State also explained his hope that eventually Computer Science will be considered for inclusion in the EBacc, both because of its educational value as a rigorous academic subject in its own right and also because of its economic importance to the UK. Mr Gove’s speech referred to several individuals and organisations whose work he highlighted as of significance, such as Bill Gates the founder of

[1] http://www.education.gov.uk/inthenews/speeches/a00201868/michael-gove-speech-at-the-bett-show-2012

[2] http://www.education.gov.uk/inthenews/inthenews/a00201864/harmful-ict-curriculum-set-to-be-dropped-this-september-to-make-way-for-rigorous-computer-science

The scheme aims to build a sustainable model, a key pillar being a supportive partnership with University Computer Science departments. Through this, it is hoped schools can benefit from specialist subject expertise whilst sharing their experience of making the concepts accessible and the subject teachable at school level. A measure of the willingness to step up to computing can be gleaned from the 480 schools already expressing an interest in joining the network as we went to press. Exciting times lie ahead!

... AND SIXTH FORM COLLEGES CAN DO SO TOO

In early February City and Islington Sixth Form College hosted a CPD session with ICT teachers from local schools - the aim being to meet each other and talk about Computing in Schools. We were very lucky to have William Marsh from Queen Mary's University (of CS4FN fame) as a guest speaker, talking about 'Why program', jobs in both Computing and ICT, and promoting CAS. Then there was a practical session looking at Scratch and RoboMind. One of the best things about it was how keen everyone was to talk to each other, debate the way forward and exchange ideas.

As a follow up the East London hub is planning a mini-series of 4 meetings in the remainder of the academic year at City and Islington College, with help from William Marsh. The aim is to introduce teachers to some of the concepts involved in teaching programming whilst learning to program themselves using Python.

Ceinwen Hilton

CAS COMPUTING TEACHSHARES

CAS TeachShares offer a monthly chance for colleagues to get together online and share ideas and demonstrate resources. These regular informal meetings, using Elluminate have become a regular feature thanks to Kevin Jones’s organisation. Future sessions already planned will discuss A level delivery and look at using Kodu with the youngest children. They are held at 8pm, the first Wednesday of each month.

Kevin is also portal manager of Vital’s new Computer Science Portal, where he aims to collate the very best of computing teaching resources. See the supplement for links.

EXCELLENT ONLINE RESOURCES FOR BOTH STAFF AND STUDENTS

Over the last ten years there has been an explosion in environments to introduce programming and a greater focus on how to teach these ideas in engaging ways. Here's five resources you may find useful. Links can be found in the web supplement.

Scratch Ed is the online community for teachers using Scratch in the classroom. It has a wealth of stories, discussions, resources and events. A real highlight is the new draft Scratch Curriculum which can be found in the resources section.

You can read the views of one student using YouSrc on page 6.You can create applications in the browser and share them on the site. The Teachers Info section contains a wealth of lesson ideas. Demonstration videos and links to YouSrc programs also help support staff.

Teaching Kids Programming has ideas for using Small Basic. There are invaluable videos outlining their teaching method as well as supporting materials.

Materials and example worlds for Alice, the visual 3D programming environment, can be found at Adventures in Alice Programming. The lesson plans contain detailed instructions for introducing and developing students programming skills.

Regular readers will be familiar with The Greenroom—the online community for educators using Greenfoot. They regularly upload new scenarios and discuss teaching approaches. Learn more about Greenfoot on page 10. These are just a few examples. If you are new to teaching programming you are not alone. There is lots of support that hopefully will inspire you to give it a go. *Peter Donaldson*

CODECADEMY: LEARN TO CODE

Its only been in existence a few months but Codecademy has already introduced thousands to coding via its popular free courses using JavaScript. We'll carry an in depth interview with the founders about their philosophy in the next issue of SWITCHEDON.

ONLINE COURSES TO SUPPORT PROGRAMMING FOR TEACHERS

Recent online courses introducing Computing can provide a rich source of support for teachers. Paul Revell, from The Lakes School in Windermere shares his experiences of going back to school with Udacity.

Having been alerted to Udacity by the CAS forum, I took the plunge and signed up for CS101 (Building a Search Engine). This free, 7 week online course aims to be equivalent to an introductory undergraduate course. Udacity.com has been set up by US academics on the back of other online offerings such as Coursera and Khan Academy. The name is a play on audacious, reflecting the way that this method departs from traditional higher level education. One of its founders, Sebastian Thrun bravely stumped up $200,000 to help with start-up costs!

DAVID EVANS *Professor, University of Virginia*

SEBASTIAN THRUN *Research Professor, Stanford University Fellow and VP, Google*

The course uses the notion of building a search engine, but the real content is basic coding and computer science, with Python as the chosen language. Each week there are video demonstrations interspersed with automatically marked quizzes. Some are multi-choice, others are coding problems. An online version of Python is used, though you can, of course, download Python locally if you want to. The course is led by David Evans, author of 'Introduction To Computing', a free online book. See the web supplement for the link. Auto graded homeworks are similar to the quizzes with feedback given after the deadline. The quality of the video material, quizzes and homework is excellent. To support the course, students can participate in an active online forum. In fact, the forum is nearly as useful as the 'lectures' and paid TA's are active in it, to moderate and support its use.

The course soon gains pace. I was whizzing through early units but by the end I was working at it almost every evening. Early coverage of variables, assignment, loops, lists and memory would complement GCSE Computing very well. Moving on, Python dictionaries, iterative versus recursive algorithms, networks, graphs, BNF and formal language are more akin to GCE Computing. There are interviews with a wide variety of people in Computer Science. Students are graded but there are no plans as yet for these to be validated in any formal way.

My initial motivation was as a way to get started with Python, but having completed it I have learned a whole lot more and found it incredibly engaging. I shall certainly be borrowing ideas from Udacity in my own teaching. The course does not have any formal pre-requisites, but someone with no coding experience is going to find it a steep learning curve and would probably need to do it twice. Nevertheless, you have to start somewhere and this is as good a place as any. For those already teaching Computing, or with some form of coding background, it is a great refresher course and introduction to Python. Level 2 and 3 courses are in the pipeline and if they are as good as CS101, I shall be doing them all. *Paul Revell*

PROGRAMMING PEDAGOGY: KEY IS UNDERSTANDING, NOT DOING

How can we best teach programming? It's a key question as increasing numbers of teachers begin to introducing coding into their lessons. Quintin Cutts argues that students must be able to read code, not just write it.

Computing educators tend to spend more time speaking about the content of their courses than they do about the pedagogy used to deliver that content. We get exercised about the language we use - VB, Scratch, GameMaker, and so on - but we seldom deviate far from our old instructional designs. For many learners however, we know those designs don't work. They simply perpetuate the belief that some learners can't learn about programming. This is a particular worry to me as we enter an unprecedented phase of expansion of computer programming education.

I've been exploring recently the major pedagogical focus in our teaching on problem solving and program writing. For many teachers, the most important goal in an introductory course is to generate learners' engagement with the subject via the thrill of seeing their own efforts come to life. However, when the creation of a program that works is more important than understanding why it works, we are in the realm of "monkeys and typewriters." Rather than making our art better understood, we are inadvertently upholding the view of computing as magic. "I've no idea why it works, but isn't it great?!" That is why, in the early stages, the creation of artefacts should primarily be motivational only; an understanding of computation should be the fundamental learning objective.

A pre-cursor to program writing should therefore be demonstrations of program understanding. This must involve more than just the reading of code, although this can be a useful start. In some classrooms, students are taking reading a step further by using a technique called Peer Instruction to discuss multiple-choice questions concerning aspects of code fragments, programming problems and their solutions. In small groups, they discuss their reasoning for the answers they chose, and it is this effortful practice to explain their understanding that marks the transition from programming as magic to programming as logic.

If students develop their understanding using discussion practices, can they also be assessed this way? Large-scale oral examinations are expensive, but students can be asked to write about their understanding instead. Finding ways of helping teachers to help students do this accurately, and of formalising this within assessment structures, is my current focus. I hope to be able to report soon about on-going work with the Scottish Qualifications Agency. In the meanwhile, some challenges to consider for assessment:

If assessment drives learning, and a course's learning objectives drive the creation of assessments, then we must get those learning objectives right. What should they be for an understandings-oriented course? Reading and explaining code requires a language in which to express the code. Do we pick one language to be used in all assessment? Do we require assessment agencies to write exams in multiple languages?

And for readers who may be thinking that a focus on reading/understanding code over writing is crazy, consider this: without an ability to read code and to precisely interpret its behaviour in relation to the original problem, effective debugging is impossible.

Quintin Cutts

HELPING TEACHERS COPE WITH COMPUTING KIDS

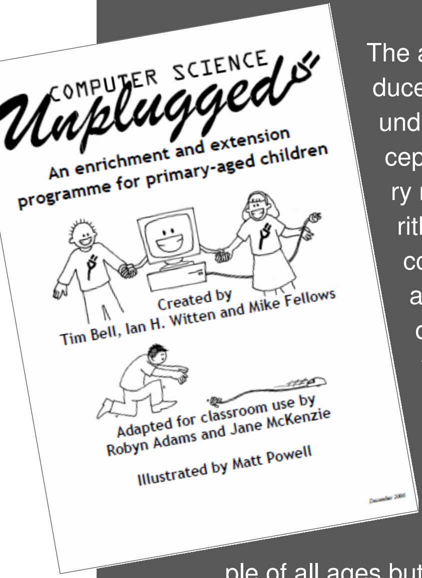
I remember very well moving from teaching maths to computer studies as a young teacher in the seventies. Whereas in maths I felt safe, computer studies was frankly alarming. Some people think being one step ahead of the kids is an essential characteristic of a 'good' teacher. But I think truly brilliant teaching lies in another direction. You cannot stay ahead of them – all those kids with their endless enthusiasm, energy and countless hours of hacking. Instead, teachers need to think about what they bring to the table: you are an expert in learning first and the subject second, so stop feeling you have to compete. Help communicate successes, focus on what they achieve, not on what you don't know, and enjoy helping them explain what they did.

Once you stop trying to master coding in the 15 minutes between your last lesson and the staff meeting, you can start to see your students as a resource not a challenge. Promote a culture where all learners can be teachers and where a teacher can admit to learning something. Pour in a little uncertainty, discomfort and risk taking and have fun! Don't be the 'expert' that is a target for a fall. Learn together, explore together, make mistakes together. A real teacher starts from the premise that they don't know everything and celebrates learning new things with students. A real teacher gets a kick out of the unexpected. I became a 'real' teacher the day a student got ahead of me. *Chris Monk*

Chris is a retired teacher and learning co-ordinator at The National Museum of Computing. This is an edited version of an article first published in The Guardian 11th April 2012

TEACHING COMPUTER SCIENCE WITHOUT A COMPUTER

Many computing concepts can be best illustrated without the need for a pc. Indeed, kinesthetic activities can help illuminate ideas and make them more concrete to pupils. This is particularly true in primary classrooms. CAS members may have been fortunate to hear co-author Tim Bell talk at the first CAS teacher conference about CS Unplugged — a collection of activities that teach Computer Science through engaging games and puzzles that use cards, string, crayons and lots of running around.


COMPUTER SCIENCE
Unplugged
An enrichment and extension programme for primary-aged children
Created by
Tim Bell, Ian H. Witten and Mike Fellows
Adapted for classroom use by
Robyn Adams and Jane McKenzie
Illustrated by Matt Powell
December 2006

The activities introduce students to underlying concepts such as binary numbers, algorithms and data compression separated from the distractions and technical details we usually see with computers. The materials are suitable for people of all ages but many of the original activities were written with the primary classroom specifically in mind. Developed in New Zealand and used for over fifteen years, such is their popularity they have been translated into eight different languages.

A book of twelve of the most used activities can be freely downloaded and includes photocopiable masters for the activities alongside clear instructions and a short explanation of ’what it’s all about’. Perhaps one of their greatest strengths is they are free of the sort of jargon that often clutters otherwise worthwhile resources. The book is supported by a YouTube channel where you can watch some of the activities being performed. If you are a Primary teacher new to computing and want to get to grips with some of the concepts, the reworked Teachers Edition is simply excellent. But if that’s not you, don’t be put off. These activities can be used right through school. *Roger Davies*

CAPTURING KS2/3 INTEREST USING DATABASE DETECTIVES

The Digital School House (DSH) transition project, based at Langley Grammar School, recently introduced ‘CSI Schools’ in collaboration with the Royal Holloway, University of London. Mark Dorling outlines his approach.

The popularity of forensic science in school continues to grow alongside the proportion of prime time scheduling devoted to CSI television shows. The understanding of how forensic tests are used in narrowing down a field of possible suspects to the point where just one can be arrested or taken to trial is far less well understood. Students may well fall for the imagined glamour and glory of mapping blood splatters but have very little awareness of the role of Computing in the logging and analysis of the data in the lab.

The CSI Schools and Certain Death Projects has grown from the realisation that computing specialists, literacy, maths and science education practitioners have a lot to offer one another. By piggy-backing on the popularity of forensic science even the most reluctant learners cannot help but be interested! Computing can also be embedded within the curriculum as a tool that enables higher cognitive skills to develop.

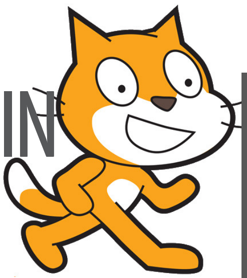
The lessons are based on the book Certain Death by Tanya Landman. The aim is to highlight the role of computing in the analysis of the data generated in the crime scene investigation lab and to illustrate how detectives use the clues to solve crimes. Before completing this series of lessons, the teacher is encouraged to either read the book (except the last chapter!) or complete a series of forensic tests and numeracy (encryption) challenges loosely based on the scenario from the book. The answers provide pupils with the clues to identify the murderer.

To accommodate different learning styles, the DSH has developed a variety of audio, visual, and kinaesthetic teaching activities. For example, pupils perform a human database role play using SQL syntax to structure questions. The pupils move around the classroom to stand in groups and sort themselves into an order depending on the answer and suspect profile card they are holding. Higher order questioning is used to relate concepts to “real world” understanding enabling pupils to better grasp the concepts being taught, and thus maximizes learning.

Pupils use cloud-based spreadsheets, to collaboratively input data about the suspects from profile cards based on the book. They verify their neighbour’s data entry before downloading the spreadsheet and importing it into Microsoft Access. Pupils use the filter tool to solve the murder using the answers from the science and numeracy challenges, creating a report for the Court based on a query identifying the murderer.

The techniques used at Langley Grammar School could have future applications in teaching pupils with special educational needs or who have missed out on good computing teaching in earlier schooling. The DSH is currently extending the project to cover secondary education e.g. using relational databases including validation without the use of computers. If you would like more information on the activities of the Digital Schoolhouse contact Mark Dorling. Contact details in the web supplement.

IDEAS FOR USING SCRATCH IN THE PRIMARY CLASSROOM



It's easy getting young children working with Scratch. This visual language now has widespread uptake in many primary schools. Amanda Wilson outlines some ideas from her research working with both teachers and pupils.

I am a PhD research student at the University of the West of Scotland interested in how primary age children are able to learn about programming. I undertook my honours project two years ago looking at teaching programming in primary schools with Scratch. The lesson plan is available on the ScratchEd site.

Seeing how children responded to Scratch projects encouraged me. I wanted more primary children to get the opportunity to use it as it was thoroughly enjoyed by the children (and the class teachers who were also beginners). Since then I have undertaken projects in my local school to encourage interest in computing.

Last year one class created animations to enter into the Animation competition (see p??). I then worked with the older class to make games for the younger children to play. In the end we went with the classic 2 player pong type game and had a wonderful afternoon of knockout tournaments with the younger Primary 1 children. This was a great lesson for the pupils as they carried through the stages of designing a game for a particular audience through to making then, testing them before deciding as a class which game would be most suitable for the younger children to play. They finally witnessed their game being enjoyed by others.

I am now undertaking research into games-based learning in primary schools and working with Primary 4 to 7 classes creating their own games.

I also try to encourage the younger children in the school by giving them some lessons with Bee-bots which has gone down great – especially when we do Bee-bot races towards the end of each lesson. It has been encouraging that although I am only in schools for a short period of time the teachers are wanting to continue the work and are very positive about the effects that the projects have had on their classes. *Amanda Wilson*

CREATING COMPUTER GAMES IN YEAR EIGHT

It's not just Primary aged pupils who enjoy getting to grips with Scratch to produce games. CAS member Brian Steene sent in this report from one Year 8 pupil, who recently completed a project at St Christopher's School in Bahrain.

"Scratch allows you to make animations, applications and also (to the pleasure of many students) games! I am making a game called "Mario Maths". In this game, you answer maths questions and progress through levels to get to the boss fight.

There are many things that go into a Scratch program such as : variables, broadcasts, functions, sprites (characters that appear on the screen) and operators. My program has used variables to configure when a sprite comes or goes, how many questions have been answered, if the questions are right or wrong, etc. Broadcasts in my program act like queues for something to happen or appear or disappear. Really, there is an endless list of things to use. This unit has been very enjoyable. My thanks to our ICT department."

Rajat Chowdhury

NEW TO PROGRAMMING? LETS GET SCRATCHING!

Regional competitions can have a big impact. Norfolk CAS recently organised a 'Scratch Off Competition' for secondary schools. It certainly had an impact at Fakenham High School. The week before half-term two Year 7 boys at asked if they could have a go (there was a big poster on my wall advertising it). They came up at lunch every day. I gave them worksheets and after some initial help left them to it. By the end of the week they'd completed two worksheets (Duck Shoot and Crab games) and were feeling pretty chuffed. They thought it was very exciting to be able to make things happen, create events and decide outcomes..........great stuff!! After half-term we worked on a competition entry. I practised jumping and platforms all week!! I took three boys to the University of East Anglia to attend. We didn't win but they loved it— seeing other pupils' work inspired them to experiment and learn. The winning entry has given us all a goal for next year! We don't currently use Scratch in Year 7 so I'm very pleased the boys came along, enjoyed it and want to come back for more. That's a lovely warm feeling I'll hang onto when confronted by Year 10's who 'hate IT and don't need it anyway'. I'm hoping Norfolk CAS will agree to run another event - my Year 7s are very keen to get Scratching. *Sue Gray*

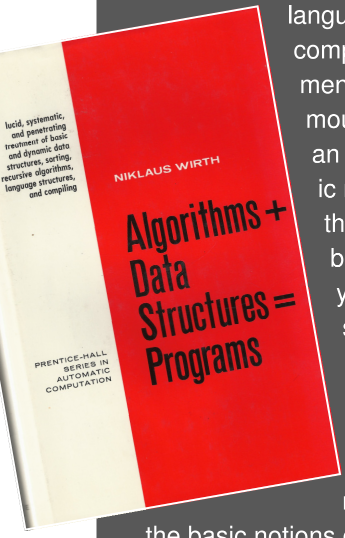
SIMPLE CONTROL PROJECTS

An easy and very effective extension for Scratch projects is to introduce some simple control projects. At last year's CAS Teacher Conference, Karen Brennan, from the ScratchEd team at MIT, demonstrated how Scratch could be connected to Pico boards or Lego WeDo kits. For more advanced users there is even software to control Arduino kit.

If you want more ideas, the team have produced a curriculum guide of twenty, one hour design based activities aimed at developing computational thinking with Scratch. More details in the supplement.

MY FAVOURITE BOOK

Algorithms + Data Structures = Programs, by Niklaus Wirth was first published in 1976. Whilst out of print now, second-hand copies can be found. The book's cover says it all: lucid, systematic and penetrating treatment of basic and dynamic data structures, sorting, recursive algorithms, language structures and compiling. They omitted mention of the wry humour: the index contains an entry for the don't panic rule. The title explains the central idea of the book: if you've chosen your algorithm and data structures (or vice versa) the program writes itself.


lucid, systematic, and penetrating treatment of basic and dynamic data structures, sorting, recursive algorithms, language structures, and compiling
NIKLAUS WIRTH
Algorithms + Data Structures = Programs
PRENTICE-HALL SERIES IN AUTOMATIC COMPUTATION

Wirth assumes the reader is familiar with the basic notions of computer programming. You don't directly learn a language (Pascal), but what you will learn is that algorithms can be analysed, their best (and worst) cases, why a selection sort is the 'opposite' of insertion sort, and why pointers (references) are like gotos.

There are lots of single variable names, programs typeset in proportionally spaced serif and few comments (but the comments that are there are models of what comments should be). The highlights? The Sorting chapter is outstanding—when and why Quicksort becomes Slowsort, and there is a complete compiler for the programming language PL/0.

I've bought everything I can that Wirth has written, or that has been written about him,. I now have 4 copies of this book in case a student loses a loan copy. Further reading? Anything by Wirth: Compiler Construction and other books can be freely downloaded from the Oberon website. The link is in the web supplement. Note that the online ‘Algorithms and Data Structures’ is not A+DS=P. While you are there take a look at Pascal's 'grandchild' Oberon. although Algorithms and Data Structures is not A+DS=P. Also recommended is Structured Programming by Dahl, Dijkstra and Hoare, frequently referenced by Wirth. *John Stout*

MASSIVE DEMAND SWAMPS SUPPLIERS OF RASPBERRY PI

A ”success disaster” was how Raspberry Pi Trustee, Jack Lang described the launch. The interest this device has generated has exceeded even the expectations of the visionary founders. Clive Beale provides an update.

If you fancy bringing down a website with a DDOS attack, forget about those messy botnets or contacting Anonymous via some seedy Internet café. Just tell people that the site will be selling Raspberry Pis the next morning–they don’t stand a chance. That is exactly what happened at 6am on Wednesday 29th February when the Raspberry Pi Foundation passed the manufacture, logistics and delivery of their tiny, Linux PCs to specialist electronics firms Premier Farnell and RS Components. Within seconds their servers were battered to death by F5 keys.

A year ago the Foundation thought that they might sell something like ten thousand RasPis - perhaps fifteen thousand in a lifetime if they did well. Current demand is for hundreds of thousands of the devices. With no formal advertising the RasPi trended higher than Lady Gaga and beat the new iPad to the top of the BBC tech stories. It became increasingly clear to the Foundation that they wouldn’t be able to handle the demand. “I'm horrified when I look back at how confident we were that we could satisfy shipping all the orders ourselves. We'd not really imagined that it might be this big”, said Liz Upton of the Foundation.

Being too popular wasn’t the only launch problem. The first batch of boards had the wrong network sockets fitted and had to go back. Global pricing and compliance testing (for CE certification) are all still causing issues. Whilst these delays and logistical problems have frustrated many potential buyers, in the long term Farnell and RS will be able to manufacture and ship the RasPis in much greater numbers than the Foundation ever could. Real life Raspberry Pis are now in the UK and the first batch is about to ship at the time of writing. CAS have one or two to play with–and to test out the work-in-progress user guide/programming manual that we are currently writing. We will report back in the next newsletter.

Ultimately, of course, the aims of the Foundation are more important than the hardware. The Foundation have always said that they would be happy to see the hardware copied (or bettered at a similar price point) and that the ownership of a computer at this sort of price should be the norm, rather than something newsworthy. They want to see the Raspberry Pi being used in low-income and low-opportunity areas in the UK and around the world. They are also working on support for UK teachers ready for the academic year September 2012 and have just taken on their first employee, who is dealing with the charity and educational work.

Success, Upton says, would be reaching only one kid in every school “who had the potential to enjoy programming–and the sort of thinking that comes with it–but who hadn't had the opportunity before we came along.” In view of the interest and the level of community support so far I don’t think that this is too far-fetched. *Clive Beale*

FROM TECHNICIAN TO TEACHER: COULD YOU MAKE A DIFFERENCE?

A key challenge for Computer Science is the current shortage of specialist teachers. The Royal Society has urged the Government to set targets for future recruitment and training of suitable staff. Kate Glover explains why she couldn't wait.

I have been an IT Technician in a large FE college since I was 18. However in January 2009, I found myself "on the other side of the desk" for the first time after a lecturer had to drop out of delivering the Web Page Construction Level 1 course with just a week's notice. Undaunted, I later ended up taking on the Level 2 groups, then providing cover for Functional Skills ICT and others as more courses became available, and my QIP reports and observations kept coming back with high scores!

While people had been telling me for most of my life that I "should be a teacher" I had always denied that I'd have enough patience and organisation for the role. And yet there I was, much to my amusement and surprise, apparently making a pretty good job of the whole thing.

Last year, I made a decision, and chucked in everything I had become comfortable with to embark on 'the next stage'. At 26 years old, I'm now a STEM Ambassador, half way through my first year of a BSc Internet Computing degree at the University of Brighton, organising and running additional peer support sessions for my fellow students.

One of the STEM events I have helped with was judging the local final of the "BT App Contest" where teams of students were given the task of designing a mobile app for the London 2012 Olympic Games. While the students came up with lots of interesting and innovative ideas (use of GPS, eTicketing, etc), some of them had a certain naivety. We were able to provide encouragement and praise as well as constructive criticism for their less practical suggestions (such as using GPS in Underground stations!). All of the teams and their teachers seemed to come away pleased with how their efforts had been received by the panel, yet also had something to take back for further discussion.

The STEM Sussex team at the University of Brighton are really supportive and helpful, and I've found my experience so far to be very rewarding. Even though I'm now a student again myself, I have found that being able to share my industry experience with school-aged students and showing them that not all IT people are geeks, provides me with a great opportunity to inspire a younger generation to delve further into Computing for both fun and study!

Upon graduation I intend to follow my degree with a PGCE to allow me to teach at Secondary level. The latest developments with opening up the curriculum have got me really excited, and I'm looking forward to stepping back to the front of the classroom in a few years' time.

Kate Glover

STEM AMBASSADORS
ILLUMINATING
FUTURES

Seeing and hearing computer scientists and other industry specialists can inspire school students to take their studies further.

If you work in computing and feel you could spare some time to visit students in school you can find further details of the Ambassador scheme from the STEMNET website. The link is in the web supplement.

STEM AMBASSADORS ASSISTING IN SCHOOLS

Ben Croston was in for a surprise when he paid a visit to Queen Elizabeth School in Cumbria. Ben, who was educated on the Fylde bumped into his former Maths teacher! After graduating, Ben worked as a programmer for BAE Systems before becoming self employed.

Part of the team writing the Raspberry Pi for Education User Guide, Ben was spending the day in school as a STEM Ambassador, presenting a talk about career paths in Computing. Having an in depth knowledge of Python, he later demonstrated some of the libraries available to Sixth Form students as well as sharing his own insights into good coding practice. "To have a long career in computing, you must be able to look up new APIs, new programming languages and new technology on your own." said Ben. "Software engineering (applied computing) is problem solving and knowing which are the right tools for the job."

Students will remember 'Webster' for a long time, as an essential tool for debugging. If stuck, Ben encouraged them to explain their code to Webster, a cuddly toy! More often than not, the act of trying to articulate their code was enough to allow the pupils to identify the problem themselves.

Having a 'real' programmer on hand allowed them to get first hand stories of the discipline required when working as part of a team and opened their eyes to some of the projects computer scientists work on.

INTRODUCING TEACHERS TO THE BEAUTY AND JOY OF CODE

The Joy of Code is a new series of online videos designed to teach programming in Greenfoot. The series assumes no prior knowledge of programming, so complete novices can dive right in. There are several hours of video available, which are already being used as teaching aids by several CAS members. The videos are the creation of Michael Kölling, head of the Greenfoot project, who communicates his approach to teaching with Greenfoot.

A distinctive feature of the Joy of Code series is that "teacher commentaries" are interspersed with the regular episodes. Analogous to the director commentaries on DVDs, these videos explain the rationale behind the Joy of Code lessons, and behind the design of Greenfoot itself -- from the authoritative source! They explain the pedagogical principles behind the Greenfoot approach, and provide useful material targeted specifically at teachers. A link to The Joy of Code can be found in the web supplement.

THE SINEPOST FOR MATHS HELP

The Sinepost is a new blog about using computing to learn and apply mathematics. Contrary to popular belief, aptitude in computing and mathematics are not intrinsically linked — meaning that there are many teachers and pupils who are adept with computing but struggle with maths. This blog aims to help them by demonstrating and teaching the practical uses of maths that can be found in computing. The examples (which primarily use Greenfoot) demonstrate the possibilities for the cross-disciplinary links with computing, and may be of interest to colleagues in the maths department. The topics covered include applications of trigonometry (for moving and turning), geometry (for detecting collisions), mechanics (for movement and physics) but also probability, statistics and more. You'll find a link to the Sinepost blog in the web supplement too.

Neil Brown

BOOTCAMP GIVES STUDENTS FLAVOUR OF REAL ROBOTICS

Aberystwyth recently saw its inaugural Technocamps "Boot Camp" for KS3/KS4 children. With no prior experience of robots or programming, 22 kids were guided over 3 days to control live robots.

For many years, Aberystwyth University have been taking robots out to local schools, introducing the reality of robotics on serious hardware. The workshops work well but there is a limit to what you can do in 4 hours. The Boot Camp ran for longer and really immersed them. Why do it? Participants learned a real language; that robots are hard to control; that sonar and vision are useful and controllable senses; that interacting sensors give a whole greater than the sum of its parts and that simulators are a smart way of developing on expensive platforms.

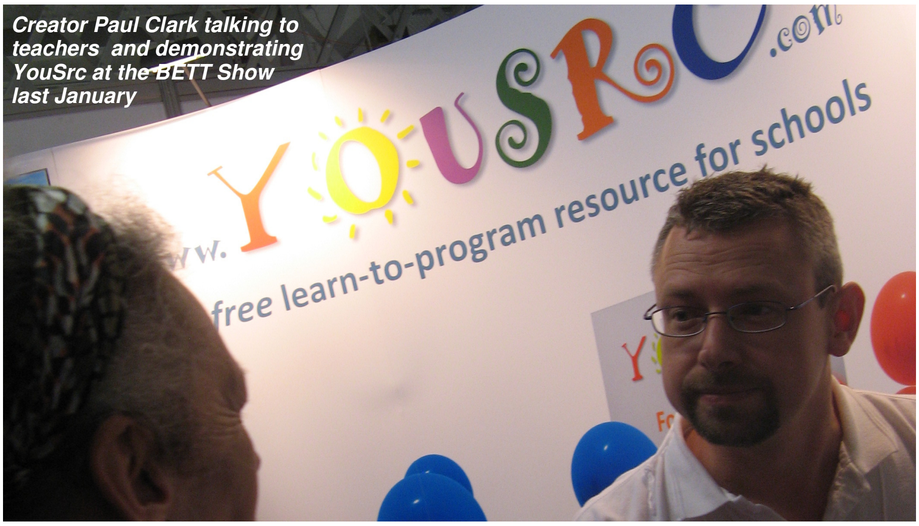
Children were dropped in at the deep end to send control instructions, and to read the sonar sensors. The simulator gives confidence that programs are "safe" before being run on live hardware via a wireless link. Activities were initiated by role-plays. A blob-tracker was programmed to move the robot toward a target. If this sounds easy, it's not! You need to learn co-ordinates, error in direction, and sensor/control feedback to close in "gently". The three days culminated with running the same program on an all terrain outdoor robot that ended up chasing the children across a field. Parents joined the class to see what had been done and were pleasantly surprised.

If they had found the early stages tough, then the three days might have been a disaster. But they didn't, and that might be the best thing we all learned … a gang of 11-15 year-olds are quite ready for this kind of thing. Compiler error messages are less than helpful and they get frustrated by brackets and semicolons, but help "on tap" can work round this. Programs gave instant feedback, making debugging more rewarding. The real result is to note that they all managed all the tasks - there is just no substitute for the satisfaction we all have in shouting "Yay! It works". You had to be there to properly enjoy seeing a programmer trying to defeat the robot s/he has programmed by hiding the target, or fooling it with a mirror, ... Further Boot Camps are planned, participants working in serious languages on real hardware. Aberystwyth will offer "Sailing Robots", and "Digital Clothing". All the inaugural class asked to be signed up!

Roger Boyle

TWO YEARS TEACHING GCSE COMPUTING: BOTH TEACHER AND STUDENT TELL THEIR STORIES SO FAR

Stepping up to teaching GCSE Computing can seem like a big challenge, but there are great rewards for both teachers and students alike. Caroline Ghali, an AST at Tonbridge Grammar School shares her experiences and insights and singles out YouSrc as an invaluable resource. One of her students, Torty Sivill, also reflects on the journey made over the last two years.


Creator Paul Clark talking to teachers and demonstrating YouSrc at the BETT Show last January

We started OCR Computing as part of the pilot to a class of 23 girls. I was very excited, having been a software engineer in a previous life, and really liked the structure of the course. It is very hands-on and the students are exposed to both high and low-level programming. The main benefit of studying the course though has little to do with the resulting code. Over the last year and a half I have seen students learn to articulate their thought processes, break a problem into solvable chunks, work effectively in groups, peer mentor and self-reflect, learn to evaluate and refine work, have a go - knowing that they will probably get it wrong first time, persevere, screw their faces up in frustration and yelp with joy when they get 'it' to work! We have travelled together from "Oh this is so hard, I can't do it!" through the "This bit is working but this bit isn't" to "This works OK but would it better if I did it like this?" Quite amazing!

The big question at the start of the course was which language? After a motivating introduction using Scratch we chose YouSrc as our programming environment. I made this choice for a number of reasons;

- it is simple, there are only a limited number of functions and variables are either string or numeric.
- The error messages during compilation really help students figure out their mistakes for themselves and they can watch their variables changing at runtime.
- It is web-based and free – no outlay for the parents or fiddly installation,
- It provides teacher control. I can see the students work, lock their work so they have limited access and publish resources for them to use.
- There are simple tutorials and examples of how to use the functions. There is a set of lesson plans that you can use to introduce the features to the students.
- Students can publish their work, see others work and leave comments.
- The apps they create can be accessed from an Android phone.
- It looks Java like so providing a stepping stone to 6th form work.

Support is quick and friendly and the creator Paul Clarke welcomes ideas for its development. The students have made really good progress and visibly enjoy using it. I am having the time of my life as a Computing teacher and would recommend it to anyone.

OPENING MY EYES: YEAR 11 STUDENT LOOKS BACK

My ICT teacher in year 9, Mrs Ghali's faith in me and my love of solving problems led me to take a risk and choose GCSE Computing. I had no previous computing knowledge and the GCSE course was new. At first it was hard going, confused by the technicalities of arrays and looping, yet inspired by our teacher's patience.

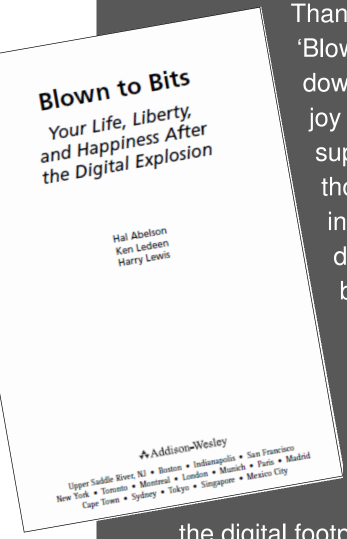
Personally, my favourite part of the YouSrc program has to be the simplicity and the multitude of different apps the language allows you to create. It was made so much easier due to the consistent support provided by Mr Clarke. He took on board our comments and requests and changed Yousrc to help our learning. He set up a forum that we used to contact him when we encountered any difficulties. He even came to visit us! He talked about his inspiration for creating YouSrc as well as informing us about the hundreds of careers that computing would help us in. As a result we were inspired to take the idea of computing more seriously and for some of us it became more than just a subject at school.

Now, after nearly two years of using YouSrc I intend to study computer science at university. Without YouSrc I would never have discovered my passion for solving problems. I look forward to the years ahead and I hope that my newly found love for code and computing continues to grow.

Torty Sivill

EXPLORING ETHICAL ISSUES IN THE DIGITAL REVOLUTION

Many ICT and Computing qualifications consider the legal, social and ethical issues involved in IT. Scratch the surface and, for many, this boils down to learning a list of relevant laws and (usually) some cursory discussion of copyright. Yet this is fertile soil for thinking how technology change social attitudes and how legal frameworks often lag far behind. A wonderful TED talk by the founder of Creative Commons, Larry Lessig is a good starting point. The point Lessig makes forcefully is that laws originating for one technology don’t readily transfer to another. Understanding how a new technology works is key to pinpointing and understanding the legal and ethical issues it raises. And that’s where so much ICT material is disappointing. An unwillingness to peer below the surface means the issues are often presented devoid of real substance.



Blown to Bits

Your Life, Liberty, and Happiness After the Digital Explosion

Hal Abelson
Ken Ledeen
Harry Lewis

Addison-Wesley

Upper Saddle River, NJ • Boston • Indianapolis • San Francisco
New York • Toronto • Montreal • London • Munich • Paris • Madrid
Cape Town • Sydney • Tokyo • Singapore • Mexico City

Thank goodness then, for ‘Blown To Bits’. This freely downloadable book is a joy to read (link in the supplement). The authors take as their starting point the fact that digitization has blown to bits many of the commonplaces of our previous analogue world. They explain how data is captured, stored and transmitted. They highlight the digital footprints left by these processes and the development of cryptography to keep data secret. They question who ‘owns’ the bits once data is digitised before going on to discuss the issues it raises such as copyright, privacy, freedom of speech and censorship.

This book roots the ethical discussions in a technical understanding that is explained in an accessible, engaging way. It should be compulsory reading for all, regardless of the course they study. My sixth form students don’t read many books. I always set Chapter 1 as a homework - almost without exception, they read the whole book. *Roger Davies*

INTRODUCING THE NOTION OF PHILOSOPHY FOR COMPUTING

Mark Dorling and Roger Sutcliffe (a leading expert in Philosophy For Children and past President of SAPERE) outline the ideas behind their exciting variation on the P4C theme: Philosophy for Computing.

Traditional philosophy, dealt with fundamental questions of life. In the 20th Century these same questions were narrowed down and applied to specific areas. There is now a developing branch of Philosophy dealing with Computer Science, with courses due to open at a number of Universities in the autumn of 2012. Whilst these new branches of philosophy, have opened up, there has been another development extending the range of people who engage with it. Children, and their practice of philosophy or philosophizing is usually referred to as 'Philosophy for Children', or P4C for short. This is a simple but powerful way of developing children's reflective and critical thinking skills. In a changing society it is vital we prepare children to not only use technology but reflect about how it works, and how it is used. Philosophy for Computing lessons have been designed and delivered by Daniel Mace, Advanced Skills Teacher at Langley Grammar School.

By collaborating with Mark and Roger, Daniel has created a series of sessions based on the accelerated learning model which incorporate critical thinking skills, high-order creativity and co-construction of ideas to address philosophical issues such as advertising on the internet, artificial intelligence, internet censorship, digital finger prints and hacking, eSafety and grooming, GIS data and privacy and viruses.

The first session focused on the digital fingerprint left by users as they select various websites and give personal information. The first half day focused on the everyday psychology behind how adverts worked, before showing how the internet magnified the issues involved. Students are shown various video adverts and asked to arrange them by appeal. This generated discussion on how adverts were designed to target personality. Paired group work focused on creating personalized adverts using a series of random words and designing a postcard of their advert.

An 'information race' (a comprehension game based on a short piece of text) helps students learn how the internet collects information from various sources. Students are asked to 'filter out' the most important sources and weigh up which are most dangerous. The consequent discussion exposes students use of websites, their thoughts about the danger and the beginnings of a class code to encourage peers to use the internet safely. Finally, a clip from Minority Report showcases a world where adverts target a person's mind as they walk down the street. This stimulates P4C debate about pupils own role in feeding the potential power of the internet. The end product is that pupils develop their own codes of practice for more responsible use of everyday and emerging technologies. Trust is nurtured during the day between pupils and teachers to create an environment where pupils are more willing to talk freely about their own experiences and share their concerns with the class. The class establishes a shared understanding and priorities which hopefully creates a deeper and longer lasting understanding of the issues involved. *Mark Dorling, Daniel Mace and Roger Sutcliffe*

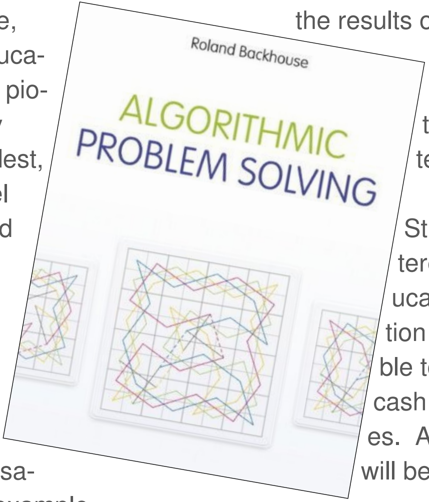
100TH ANNIVERSARY CELEBRATED WITH TURING-TAPE COMPETITION

The School of Computer Science at the University of Nottingham and the School of Computing at Teesside University have organised the Turing-Tape Competition in celebration of the anniversary of the birth of Alan M. Turing.

The competition is about finding the best solutions to a collection of one-person, solitaire-like games played on a Turing tape (a one-dimensional tape divided into squares). The solution of simple examples of Turing-tape games can be done by hand but the games quickly become too complex and are best solved using machine assistance. The games combine the mathematics of polynomial division with the algorithmics of program construction and, we believe, offer an appropriate and educational insight into the sort of pioneering work undertaken by Turing. Apart from the simplest, the games are entirely novel so solutions cannot be found on the Internet.

The competition will run in stages. It opened on April 1st with details of the game release timetable and three example games for familiarisation. You can read the first example problem in Pause For Thought on the back page. Batches of games will be released at regular intervals up until July 15th. These games will increase in difficulty and are for competitors to practise on. The competition asks for the "best" solution to each game; a table of top scores will be maintained throughout the practice period. The top scores will be divided into four categories: pre-university, undergraduate, postgraduate and other.

The competition ends with a grand final. The games in the final will be released on August 26th and the deadline for submission of entries will be September 16th. Top scores will not be published during this period; the results of the competition will be announced on the 30th September.


Roland Backhouse

ALGORITHMIC PROBLEM SOLVING

Students registered at a UK educational institution are also eligible to compete for cash and other prizes. All competitors will be entitled to a discount of 20% on the purchase price of the book "Algorithmic Problem Solving" by Roland Backhouse, courtesy of the publishers John Wiley & Sons, Inc. A link to the website giving full details of the competition can be found in the web supplement. *Roland Backhouse*

FESTIVAL OF CODE FOR YOUNG REWIRED STATE

Make sure your students take note of Young Rewired State 2012, it is going to be brilliant. The order of service is as follows:

Date: 6-11 August at various locations around the country

Who for: kids aged 18 and under who can code, they need to at least be able to edit HTML right through to that kid who sits in your libraries trying to hack into the school dinner menu

How much: It is free, however we are crowdsourcing funds. See link the supplement if you can help.

What is it? A week long coding extravaganza, where young people come together and build websites, games, widgets, apps - anything they like - using open data, mentored by volunteers from the Rewired State network.

On the Friday they all travel to the National Museum of Computing, Bletchley Park for a festival of code. They camp overnight, code more, hear lightning talks from Ben Goldacre, Dallas Campbell, Martha Lane Fox and other luminaries. There will be music, tents and pizza. On the Saturday they will present what they have built during the week to an audience of government, industry and press - as well as their peers. Prizes will be had and further speeches before we all go home.

Emma Mulqueeny

CODE BREAKER } 2 competitions { Animation12

This is the fifth year of the annual UK Animation Competition, and as usual, there have been a huge number of entries. A second competition, Codebreaker, ran alongside. To support CodeBreaker, a pack of excellent teaching resources for Greenfoot were made available. The Animation12 Awards and Inspirational Computer Science Day will take place on Friday 13 July, at the University of Manchester. For the first time we also plan to have activities on the Saturday, but details of those are not yet finalised. Such is the popularity of this event that we were fully booked within 20 hours of opening registration! There's a waiting list, so if you're interested, please do sign up. You can view photos of last years event on the Animation 12 website—link in the supplement. *Toby Howard*

Apps for Good is an award-winning course. Students create apps that make a difference and solve real life issues that matter to them and their community, giving them a launchpad in social enterprise and the exciting world of technology, design and innovation. Apps for Good can be delivered either within the curriculum or as an after school club. To find out more about the course and how to become a partner, visit the website. See supplement for details. .

CAS WALES / TECHNOCAMPS ANNUAL TEACHER CONFERENCE

I'm very pleased to announce the 2012 CAS Wales / Technocamps Conference will be held at Swansea University on Friday 22nd June. Building on last year's successful inaugural conference, CAS are continuing to work closely with Technocamps to drive forward the CS education agenda and provide an opportunity for teachers across Wales to come together to discuss the latest issues and share best practice.

The confirmed keynote speakers for the 2012 conference are:
Leighton Andrews AM, Minister for Education and Skills, Welsh Government
Simon Peyton Jones, Microsoft Research Cambridge and CAS Chair
Maggie Philbin, BBC and TeenTech

The final workshop scheduling is yet to be confirmed, but will feature: Greenfoot (University of Kent), Kodu (Microsoft), cs4fn (QMUL), Computational Modelling (Swansea), Robots (Aberystwyth), Wearable Arduinos, GCSE Computing forum and much more! There will also be a TeachMeet-style "bring and brag" event on Thursday evening before the conference. Registration for the Wales conference is online via Eventbrite— you'll also find Further details in the web supplement. We are looking forward to seeing you in Swansea. *Tom Crick*

INTRODUCING THE ARDUINO

Technocamps Aberystwyth are currently running two arduino-based workshops as school holiday clubs (TechnoBootCamps).
In the "wearable technology" workshop we've got a bunch of lilypad arduinos, some breadboards, LEDs, sensors (LDRs), bucket loads of crocodile wires and laptops running ubuntu with the arduino IDE. It's going really well and we've actually started sewing (conductive thread, battery packs and all). Kids are between the ages of 11 and 14 and seem to be loving it. The gender balance for this particular bootcamp is about 80% female! Next issue will have a full feature on 'Fabric Computing', pioneered by Leah Buechley at the influential MIT Media Lab to tackle gender imbalance in computing. *Hannah Dee*

EXCELLENT ATTENDANCE AT FIRST WELSH TECHNOTEACH

The winter months aren't known for their high temperatures, but the excellent turnout of delegates to the Technoteach event on the 17th January would have given the evenings organisers a warm glow inside.

Sixty delegates descended on the Village Hotel in Swansea to attend the first ever Technoteach. Organised by the Swansea branch of Technocamps 50 teachers, 4 students, 4 examination board staff and 2 ICT trainers were introduced to two exciting pieces of software – Microsoft Kodu and the Greenfoot Java Environment. Following a welcome by Stuart Toomey, Project Manager for Technocamps attendees were taken by Stuart Ball (@innovativeteach) into the world of simple 3D games development. Stuart manages the Microsoft UK Partners in Learning Programme, a free professional development support network for teachers and schools. His presentation introduced Kodu, an icon based 3D games programming environment. Some had seen the software before whilst others were enthused by the opportunities it offers.

A task was set for everyone to create their own simple game. The delegates present discussed with excitement the possibilities for their classrooms as they programmed their characters to eat apples of varying colours. Kodu allows users to build "code" using a visual editor. The skills are in devising the logic to make something happen.

After a coffee delegates were introduced to Dr Neil Brown of Kent University. He introduced Greenfoot – a programming environment specifically designed for novice programmers, targeted at age 14 plus. The Greenfoot environment is based on the Java language and although extremely powerful is designed to hide complexity. Dr Brown demonstrated the power of the software before creating a small application which we were able to replicate.

The progression from the visual programming environment of Kodu to the more technical Greenfoot allowed teachers to see how programming could be embedded in schemes of work for secondary pupils of all ages.

The evening was brought to an end with delegates having the chance to network with colleagues and peers whilst being treated to an excellent buffet organised. *Gareth Edmondson*

NORTHERN IRELAND: RAISING AWARENESS WITH POLITICIANS

The CAS message is being heard in the corridors of power. Growing numbers perceive something is amiss and politicians are waking up to a growing movement for change. Clarke Rice provides an update on developments in Northern Ireland.

The study of Computing in Northern Ireland has declined sharply since early in the last decade. Indeed the local exam board, CCEA, withdrew Computing at A-level in 2002, citing a lack of demand. This was compounded at around the same time when the managed service provider in schools, C2k, declined to support software development tools. Only a small number of schools continue to offer AQA and OCR Computing via their own 'legacy' networks.

Concern in FHE and the IT industry has grown steadily in the past decade. The decline in A-level Computing has had several knock-on effects: students unable to make a fully informed choice regarding Computing at university level, and hence choosing other fields; students not prepared for the study of Computing, who struggle to come to terms with course content not remotely like A-level ICT; a huge shortage of Computing graduates. In any one week, a quarter of vacancies listed on the NIjobfinder.co.uk website require programming skills.

The case for Computing in schools has been taken up by the Alliance Party of Northern Ireland, and in particular Judith Cochrane MLA. Judith has been actively working with colleagues in other parties, following many separate concerns being raised by FHE and the wider ICT industry, about the skills gap that exists.

At the time of writing, Judith has tabled a motion calling on the Minister of Education to encourage the teaching of Computing as a discrete subject from ICT, and to call on C2k to support the tools needed for CS. This will be debated during the Spring (date to be confirmed). Cross-party discussion on the matter is ongoing. CCEA are also considering their position, with a Software Development A-level under development.

The issue was raised on the floor of the assembly during an Education Minister's Question Time in March. Education Minister John O'Dowd said “We have to focus now on whether [ICT] coursework and provision are adequate and meet the sector’s needs. I, too, have been approached by pupils and teachers — [Judith has] been in regular correspondence with me, as has the sector — to say that the skills base that is laid down at schools may not meet the needs of the ICT sector. The establishment of a task force by the Minister for Employment and Learning is a valuable step forward. I and my Department are happy to engage fully with that working group. I am happy to work on and move along any of its findings, which will be evidence and research based, to ensure that we have the skills base required to build the ICT sector.”

Following the sitting, Judith commented, “It is very encouraging to see the Minister and his Department begin to take notice of the alarming prospects for our computer skills sector if computer science education is neglected. During the course of my correspondence with the Minister over the past months, I feel that positive steps have been made towards placing greater emphasis upon Computer Science. I sincerely hope that yet more can be achieved going forward, particularly with regard to further exploration of GB developments and having a wider conversation on whether similar GCSE and GCE specifications should be developed for use here, with enhanced stakeholder engagement.”

CURRICULUM REVIEW ON-GOING IN SCOTLAND

It is an exciting time in Scotland. New courses are being introduced as part of the ‘Curriculum for Excellence’. Together with the Government, Education Scotland, SQA, and SISCA (Scottish Informatics and Computer Science Alliance), who represent the universities, we are working to provide CPD for teachers and promote Computer Science as a rigorous academic subject. Recently we researched the deteriorating number of Computing teachers across Scotland. Almost 10% of schools don’t have any Computing teachers now. We are working with the Scottish Government to try to resolve this.

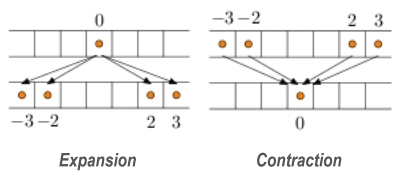
After only a year we now have a quarter of all Computing teachers in Scotland as members of CASS! We want to keep increasing our membership this year. We are planning an annual conference in September/October. We also want to set up a CASS Board with teachers as well as university and industry representatives to support the work we are doing. If you are interested in joining do get in touch. Contact details in the supplement. *Kate Farrell*

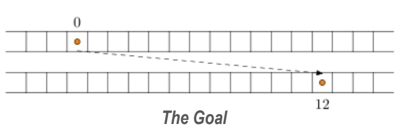
I LOVE MY SMARTPHONE

One particular activity to support Computer Science, funded by the Royal Society of Edinburgh, BCS, Education Scotland and others, has been to second a teacher, Jeremy Scott, to develop classroom materials as an exemplification project. Due out in September, they’ve just published the first batch. Although first to be published, this resource is intended as the third in a series of resources, focusing on developing mobile apps. Exercises are provided with sample answers and there are many additional activities which can be used to both broaden and deepen the topic.

A PAUSE FOR THOUGHT

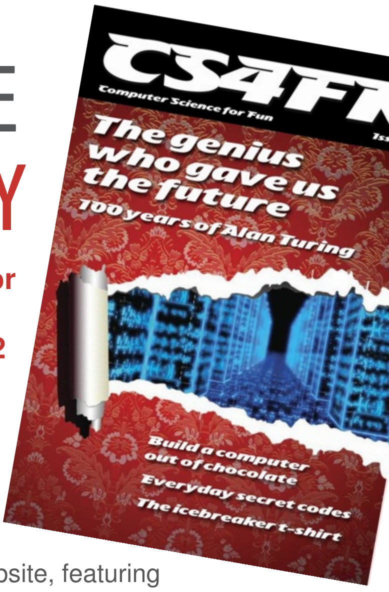
A single coin is placed on one of the squares of a Turing tape. The task is to displace the coin twelve places using a sequence of moves, which we now define. A move is one of two types: an “expansion” or a “contraction”. An expansion replaces a single coin on one square by four coins; relative to the position of the coin that is removed, the four coins are placed at positions –3, –2, 2 and 3. (Thus an expansion increases the number of coins on the strip of paper.) A contraction is the reverse: a contraction replaces four coins at relative positions –3, –2, 2 and 3 by one coin at relative position 0. (Thus a contraction reduces the number of coins on the tape.) No other moves are possible, and there are no other restrictions on moves. (The number of coins on any one square is unlimited and there are no restrictions on the positions or numbers of coins that are not involved in a move.) Construct a sequence of moves that when begun in a state with one coin on one of the squares of a Turing tape (and all other squares empty) ends with one coin at a displacement of 12 squares from the initial position (and all other squares empty).





This is the first example task of the wonderful Turing-Tape Competition (see page 13 for details). Answers can be found on the competition website.

UNDERSTANDING THE ALAN TURING LEGACY



The latest issue of the Computer Science For Fun (cs4fn) magazine is out now, and it's all about Alan Turing. In celebration of the 2012 centenary of his birth, cs4fn takes a look at current research in areas he cared about.

We explore Turing's biography and how Britain has changed since his time. We also look at some of his big ideas, like Turing machines and the Turing test, in ways that students can grasp. We check out new ideas in code breaking and unusual stories like how an animal could survive being chopped into 300 pieces. As always, cs4fn is available for free. You can order just one copy or an entire class set. Visit the cs4fn website to sign up, and to find more stories and resources about fun computing research.

cs4fn also has a new subject portal on its website, featuring stories about a project to design safer medical devices. If you have students interested in designing computers to be user-friendly, point them towards the chi- med section. Here they will find stories about hospital situations where good human computer interaction design can literally mean the difference between life and death.

Jonathan Black

COMPUTING: THE NEXT GENERATION

Computing At School Fourth Annual Teacher Conference
Birmingham University: Friday 15th June (reception 14th June)

"I learnt more and connected with more interesting people than on any course that my school has previously forked out hundreds of pounds for"

http://casconf2012.eventbrite.com to join the waiting list!

FULLY BOOKED

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: *James Anderson, Roland Backhouse, Clive Beale, Jonathan Black, Roger Boyle, Neil Brown, Rajat Chowdhury, Tom Crick, Quintin Cutts, Claire Davenport, Roger Davies, Hannah Dee, Peter Donaldson, Mark Dorling, Gareth Edmondson, Kate Farrell, Caroline Ghali, Kate Glover, Sue Gray, Ceinwen Hilton, Toby Howard, Simon Humphreys, Kevin Jones, Daniel Mace, Bill Mitchell, Chris Monk, Emma Mulqueeny, Simon Peyton-Jones, Paul Revell, Clarke Rice, Sue Sentance, Torty Sivill, Brian Steene, John Stout, Roger Sutcliffe, Amanda Wilson.*

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