CAS NEWSLETTER

SPRING 2011

COMPUTING AT SCHOOL
EDUCATE · ENGAGE · ENCOURAGE

SWITCHEDON

COMPUTING AT SCHOOL NEWSLETTER WINTER / SPRING 2011

MOBILE APP DEVELOPMENT GOES VISUAL

There can be few contexts as motivational for pupils as developing applications for their mobile phones. Until now though, developing mobile applications has been the preserve of small numbers of older students. All that is set to change with a new development from Google. Google App Inventor is a web based application that has the potential to allow anyone to create software for Android devices. Users can drag and drop objects onto a simulated phone screen and use a graphical interface very similar to Scratch to develop the code required.

App Inventor for Android draws on a large body of research into educational computing. Open Blocks visual programming, which lies behind the App Inventor interface, is distributed by the Massachusetts Institute of Technology (MIT), Scheller Teacher Education Program (STEP), home of StarLogoTNG, and is closely related to Scratch, a product of MIT's Media Lab. Both have a growing uptake in schools. These projects have been informed by a constructionist theory of learning which traces its roots to the work of Seymour Papert and the MIT Logo Group in the 1960's. The theory highlights programming as a vehicle for developing powerful ideas through active learning. CAS will be showcasing projects that emphasise a problem solving approach in our BETT seminar. There are much broader educational benefits to computer programming than simply getting code to work. There couldn't be a better time to get your pupils involved.

A BUSY YEAR AHEAD AS SUPPORT GROWS

Computing At School is barely two years old but our membership is already over 600. We're a diverse group united by a concern that Computing, one of the most exciting disciplines in today's world, has been squeezed out of the school curriculum.

Computing is a unique discipline that we believe all pupils should have exposure to. As Nicholas Negroponte, architect of the One Laptop Per Child Project commented, "Computer programming is a powerful tool for children to 'learn learning', that is to learn the skills of thinking and problem solving… Children who engage in programming transfer that kind of learning to other things."

CAS teachers can testify to the powerful ideas pupils generate through computing and the antidote this gives to the increasing 'powerpointlessness' of much that passes for ICT. Growing numbers are getting SWITCHEDON. Inside you'll find reports of things being tried and shared, of meetings, initiatives and collaboration with colleagues in higher education. This issue is packed with ideas to take back to your classroom. Enthusiasm is infectious. If you like what you read, please join us. See the back page for more details.

The "Computing At School" group (CAS) is a membership association run by BCS, The Chartered Institute for IT, supported by Microsoft Research and other partners. Its aim is to support and promote the teaching of computing in UK schools.

bcs Academy of Computing

A COMPREHENSIVE SUITE OF VISUAL TOOLS FOR THOUGHT

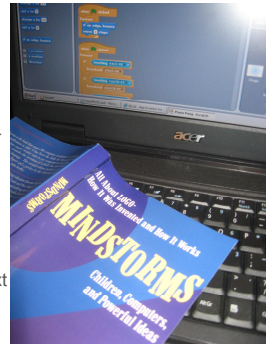
The release of Google's App Inventor for Android is just the latest development in a suite of free tools that seek to release the creative potential of pupils through programming whilst removing some of the common hurdles. Visual languages have been in development for many years and Seymour Papert and MIT's Media Lab have led developments in this field. Most recently, Scratch has taken the educational world by storm. Developed by the MIT Media Lab, it has been widely taken up by teachers and pupils. The bar to entry, the simple interface and snap together commands make programming constructs easy for children to visualise. Most importantly, the use of visual snap together blocks removes the syntax barrier. Creating programs is highly motivational, finding a missing semicolon is not.

So successful is the Scratch environment that others have developed more powerful extensions that allow the development of programs involving more complex data structures and the ability to develop procedures and pass parameters (which themselves can be structures). Best known is Build Your Own Blocks (BYOB) from Berkeley University which can be used to teach concepts up to and beyond A level.

MIT also developed StarLogo, designed to program multiple agents simultaneously and develop computer models to simulate and explore emergent patterns of mass behaviour. Recently it has been given a Scratch like makeover, StarLogoTNG, using a similar block structure. It supports the development of procedures and provides an excellent environment to develop quite complex models. If you are familiar with Scratch, why not explore StarLogoTNG as a way to introduce modelling into your curriculum? Look out for a full review in the next issue. *Roger Davies*

MINDSTORMS: COMPUTERS, KIDS AND POWERFUL IDEAS

Seymour Papert's pioneering work with Logo started a revolution. His perspective, which has motivated many subsequent developments in computing education, holds that programming can be a vehicle for engaging powerful ideas through active learning.



Papert's thinking, which he called "constructionism", is a version of constructivist thinking based on the ideas of Jean Piaget, with whom he had studied. Constructivist thinking holds that individual learners construct or modify mental models to understand the world around them. Constructionism (with an N rather than a V) holds that learning can happen most effectively in a context where the learner is consciously engaged in constructing a public entity in the real world, whether that is a sand castle on the beach, or a theory of the universe in the lab. A teacher's role is thus that of a guide and a facilitator, rather than an instructor. Constructionist learning involves students drawing their own conclusions through creative experimentation and the making of social objects. Teaching "at" students is replaced by assisting them to understand, and help each other to understand by active exploration of a problem space.

It is the notion of active learning: kids learn best by experience - learning by doing, by exploring the world. Logo was intended to help teach maths, geometry and logic by creating a "mathland" that could be explored. Turtles came later, and it became obvious that they were a powerful tool to teach programming, and new ways of thinking. Controlling a turtle drawing a line was a simple and easily understood analogy for sequential programming. Robotics, extending from controlling turtles, is a good way to teach computer science including real-time interactions, control theory, co-operating processes and modelling.

Logo, although easy to use, is a complete programming language. It was derived from LISP, the main language used for AI programming at the time, and that in turn was derived from lambda calculus. Logo has been described as "Lisp without brackets". It is mostly text based, since displays then were uncommon and limited in function. In its turn Logo, and Papert's work have been influential both in teaching and in computer science. The group became part of the MIT Media Lab, who later developed Scratch. Papert's ideas were expanded in his book "Mindstorms: Children, Computers, and Powerful Ideas" first published in 1980.

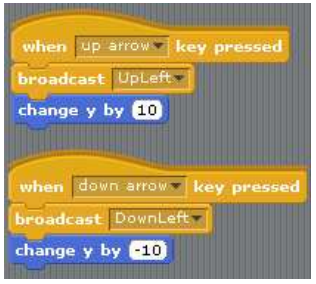
Those were days, as is sadly becoming now, when people thought of programming as highly skilled, available only to specialists. Yet Papert knew kids could program, as the children of faculty members and pupils from local schools, some as young as 9 or 10, found their way to the lab, and were indeed writing code. What Papert showed was that given the right environment almost anyone can program, and in doing so use it as a tool to explore new ideas and create new things. *Jack Lang*

BUILD YOUR OWN BLOCKS: A RECIPE TO PLAY PONG OVER A NETWORK

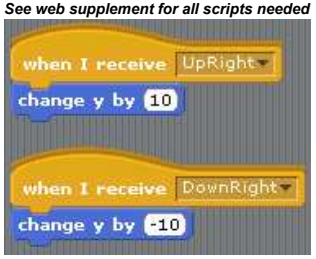
Many teachers are now using BYOB, a powerful extension to Scratch, to develop exiting projects for their classrooms. Simple games are well within the reach of young children, but a networked game can take their learning even further. John Stout shares a successful recipe.

First the required ingredients:
2 networked computers running BYOB
1 ball (a sprite)
2 bats (both sprites)
4 walls (all sprites)
2 scores (both 'global' variables, i.e. for all sprites)
2 direction changes (both 'local' variables for the ball, i.e. this sprite only)

Second the method:
From the Share menu on one machine (this will be the server) select Share - Host Mesh. BYOB will report the IP address of that machine. From the Share menu on the other machine (this will be the client) click Join Mesh and enter the other machine's IP address. The two machines are now 'meshed' - broadcast messages are received across the mesh, and global variables are shared but read through the use of sensor value blocks (from the sensing palette).



The servers LeftBat and RightBat scripts
See web supplement for all scripts needed



On the server create a Ball, and a left bat called LeftBat. Right click each sprite and select Share sprite. These sprites get copied across the mesh and appear on the other machine. The client should meanwhile create a right bat called RightBat and share that. The server now creates and shares four walls, called LeftWall, RightWall, TopWall, and BottomWall. These can be shared with the other player, then rearranged on their stage to provide the environment for Pong.

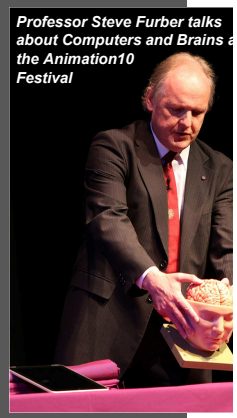
The ball resets the game (scores, positions etc) when the green flag is clicked, then starts a game by broadcasting the PlayGame message so the ball moves regularly, bouncing off bats, top and bottom walls. Scores are updated and the game restarted when bouncing off a left/right wall.

On the server the LeftBat has two scripts responding to up/down key presses by moving and broadcasting a LeftUp/LeftDown message across the mesh. The RightBat responds to RightUp/RightDown messages by moving up/down. On the client the roles are reversed in that the LeftBat will respond to broadcast messages and the RightBat to key presses.

This is a very useful game to introduce team programming. One student does the server, the other the client and they get the game working. Then they swap roles. As an extension you can add a TopBat and BottomBat, so that the mesh has four machines and four players. This raises interesting discussions on what a win does to the scores, and can lead to heated discussions about the best way to proceed. Extensions are left to the ingenuity of your pupils. *John Stout*

ANIMATION 11: NOW LET'S GET SCRATCHING!

The School of Computer Science at The University of Manchester is once again running its annual UK Schools Computer Animation Competition, now in its fourth year. The competition has, to date, involved over 4,000 schoolchildren from 1100 schools across the UK, and is funded by Google, Electronic Arts, and the Granada Foundation, in association with BBC21CC and cs4fn.



Professor Steve Furber talks about Computers and Brains at the Animation10 Festival

The competition is open to all UK school children aged between 7 and 19. The deadline for entries is 1 April 2011. Entrants can create their animations (max duration: 1 minute) using any of the following programs: Alice, Scratch, Flash, or Serif DrawPlus.

There are great prizes. Winners will receive their prizes at the Animation Festival, part of an Inspirational Computer Science Day, to be held in Manchester in July 2011. Any school may attend, free of charge, subject to capacity. Last year, over 400 pupils attended the Festival held at The Lowry, Salford Quays which combined the Awards Ceremony with a packed agenda of activities and talks. See the supplement for links to the Animation11 website, a film of last years festival and the prize winning entries. Last year saw over 1,300 entries from around 200 schools - a measure of how established the competition has already become. *Toby Howard*

NEW SCENARIOS INCLUDED IN GREENFOOT 2.0 RELEASE

Recently, a new version of Greenfoot brought new tools to the teaching of programming with Java, while three new Greenfoot scenarios demonstrate how to use Greenfoot to meet the demands of the new GCSE Computing curriculum.

Greenfoot 2.0, released in October, adds a variety of new tools – some to increase motivation of students, others explicitly to help in teaching important concepts of programming. Built-in sound recording and integrated image editing make the use of media in student projects much easier, enabling students to create games and simulations with custom made sound effects and graphics easily.

A new feature especially interesting for teachers is the new "scope colouring" in the editor, a tool that makes understanding the concept of nested scopes (and detection of related errors) much easier than ever before. The new Greenfoot version can be downloaded free from the Greenfoot website.

At the same time, three new sample projects have been posted in the 'Greenroom', a place for teachers to exchange material. These examples demonstrate how to use Greenfoot to build GUI-oriented applications matching the requirements in the new GCSE Computing curriculum. To access these examples, simply get a free Greenroom account. See the web supplement for links to both the Greenroom and the new version. *Michael Kolling*

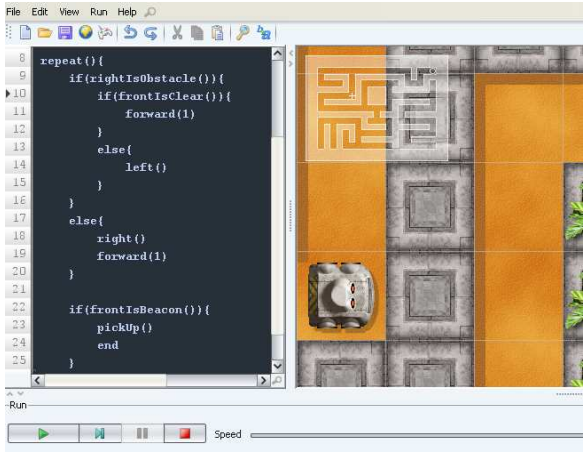
GCSE COMPUTING TAKES OFF

OCR have met their initial target of 100 centres offering the new GCSE Computing Pilot. Feedback from schools embarking on the course has been very positive and new guidance has been added to the materials on the website. New centres who wish to offer the course as an option for next year can still register their interest. Contact Karen Reid at OCR in the first instance and each application will be judged individually. The qualification has been accredited until 2015, so it is very likely that the pilot will become live in 2012.

ENGAGE YOUNG PUPILS WITH ROBOMIND

ROBO is a simple educational programming language to familiarize children with the basics of controlling their own virtual robot. Peter Marshman, an AST in West Berkshire explains.

RoboMind is a free, text-based, programming environment developed at the University of Amsterdam. It allows students to program a sequence of instructions for a robot to traverse a map or maze whilst, along the way, painting, moving obstacles and much more. Students can load a series of predefined mazes and use sequencing techniques to get the robot to relocate to the end position. The syntax used to do this is largely based on Java and introduces core concepts, debugging and dealing with annoying syntax errors, allowing them to become precise programmers. Basic sequencing problems can then be made more efficient and complicated by adding selection, repetition and procedures (which can even be called recursively). Although these may sound complex steps for young students to take, the beauty of RoboMind is the compactness of the application where boundaries are tightened so that students can apply their algorithmic thinking to problem solving without using large amounts of time considering aesthetic design.



I have been able to engage students further by incorporating group work. For example, students can create procedures to draw letters using the paint tool, upload their solutions to a wiki and then identify the importance of code re-use. RoboMind also offers a simplistic map creation option using a text file. Students can develop a map and associated routines for the robot to traverse it. These can be shared, with peers further refining each others' procedures to develop efficient robots.

Why not get your class to work as a team and create a robot that will be able to traverse any default or user defined mazes? Robomind enables students to start with basic sequences but progress quickly to develop intelligent solutions using procedural programming techniques. Ideal for Year 7 and a must for anyone who wishes to introduce the disciplines of computational thinking at an early stage. *Peter Marshman*

SUPPORTING KS2/3 TRANSITION WITH THE DIGITAL SCHOOLHOUSE

Langley Grammar School (Berkshire) is a specialist Maths and Computing school. As part of their specialism they launched a transition project, led by Mark Dorling, aimed at Year 6 pupils called The Digital Schoolhouse (DSH).

The DSH offers local primary schools the opportunity to study in a dedicated ICT environment. There is no cost for the teaching. Many local primary schools are unable to justify buying control equipment. The DSH offers them two control projects using Scratch or Probots. At the start pupils are given a brief and work to complete it through a series of tasks and rewards, made up of learning objectives from the QCA KS2 or 3 frameworks.

To ensure a solid foundation, lessons revisit previous learning, employing a range of kinaesthetic activities. For example, pupils develop instructions using a flow diagram creating a procedure to complete the Michael Jackson Moon Walk. Children act as human robots being reminded of the Repeat command. Learning then focuses on expected outcomes from Year 6 e.g. If, Then and Else. Where applicable, the teacher introduces KS3 concepts.

The Scratch brief develops a Pacman game. Programming is introduced by asking two questions; How many parents drive cars? and How many design cars?, relating the difference between ICT and computing. Pupils download and install the software. Whilst the installation completes they study example games, identifying good features. Pupils use these to produce objectives. Teacher-led modelling and independent learning videos support them. Pupils uninstall Scratch before returning home.

The second project uses Probots. Developed with the MFL department it integrates control with language learning. Pupils navigate their Probots around a town using new vocabulary, in any of four European languages. The new vocabulary focuses on directions and shops found in the map. Schools can use a predesigned town or have pupils design their layouts prior to their visit. Further worksheets integrate with the Passport to the World geography topic.

These ideas are only two suggestions from a vast range now available to schools, often for free. As computing becomes reintegrated into the curriculum, staff in both primary and secondary education need to become more familiar with them. *Mark Dorling*

SSAT / CAS SCHEME OF WORK TAKES SHAPE

The Head Teacher steering group of the Mathematics and Computing specialist schools meets termly. Last March, Simon Peyton Jones attended, describing the ambitions of CAS. A group has since been convened to write a short scheme of work, accessible to all ICT teachers, to introduce computing at KS3. On the 2nd November, 14 representatives from specialist schools, SSAT and CAS began creating a sequence of lessons. A busy morning was spent mapping out ideas and a structure of 3 pairs of lessons with ample possibility to extend was agreed. Three groups are now drafting sample lesson plans and aim to trial them during the coming term. The 3 themes currently being developed are:
What is a computer anyway? looks at the variety of applications and the logical manner in which they obey instructions.
Computing puzzles looks at ciphers, codes, encryption.
Efficient information transfer explores compression, error checking, video, sound and data.

Publication of the scheme on the CAS and SSAT sites is due after Easter 2011 to allow teachers to plan to introduce computing in the 2011/12 academic year.

Alec Titterton

SSAT LEAD PRACTITIONERS

Two Maths and Computing SSAT Lead Practitioners have organised sessions for teachers developing computing themes. Scott Tooley organised a day entitled "If you have an itch, Scratch IT! Using programming to develop mathematical qualities" in Sawtry, Huntingdon whilst Nicki Maddams, co-ordinator of the practitioners has a session planned in Kent to encourage teachers to start using "Kodu In The Klassroom".

PUPILS TO ACT AS JUDGES IN JUNIOR TURING TEST

It's a proposition straight from the realms of sci-fi – can we create machines that imitate humans so well we can no longer tell the difference?

Far from being a concern for the far-off future, however, this question is already being put to the test in an international competition which is set to come to the South West for the first time in its history.

In 2011 the Loebner Prize, named after its founder, Hugh Loebner, will be held at the University of Exeter. The event offers thousands of pounds worth of cash prizes and challenges computer programmers to create software which can provide human-like responses to questions – essentially holding a conversation.

This year, the event at Exeter will include an additional competition where four secondary school students will act as judges in what will be known as the Junior Loebner Prize (JLP). The students will select the program they think is the most human-like in its responses.

To find the four student judges, a competition will take place at the 2nd Computing Conference for Schools (CCfS 2011). Trophies will be awarded to teams placed in the top three for the competition. The school with the winning team will also be awarded a prize for use in their school.

Building on last year's successful conference, CCfS 2011, hosted by the College of Engineering, Mathematics and Physical Sciences, aims to highlight some of the other possible uses for machines within Artificial Intelligence (AI). Students will be introduced to some of the problems that AI researchers face and how they may be addressed. They will also be taught some basic programming concepts to perform a simple AI task.

See the supplement for links to more information on the JLP, CCfS 2011 and the other outreach activities that are offered by the University of Exeter. These include visiting workshops and lectures to students in local secondary schools. *Zena Wood*

COMPUTING: THE SCIENCE OF NEARLY EVERYTHING

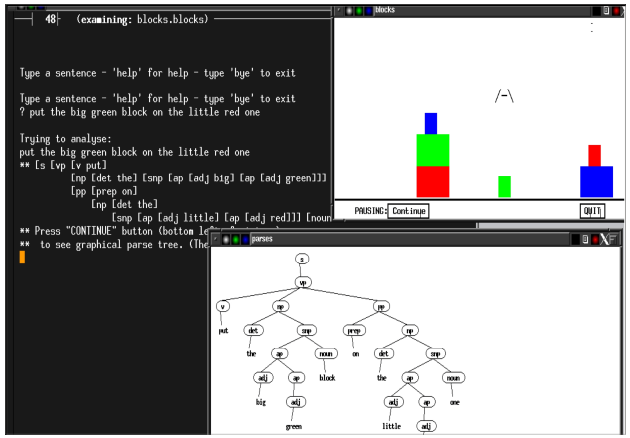
There are now powerful educational tools that enable learners to program multiple agents moving around in a simulated world where they sense other things and react accordingly. A different sort of task is to develop simulated agents that can acquire store, use, share, and later modify information obtained from the environment. Exploring concepts in school to do with Artificial Intelligence (AI) can inspire pupils to explore 'thinky' as opposed to 'bumpy' behaviour argues Aaron Sloman.

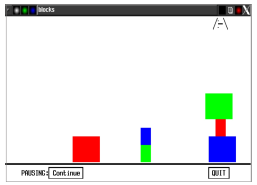
Everyone knows that developments in hardware and software technology have had and will have tremendous impacts on all our lives, in many different ways, in industrial, medical, social, educational, financial, and other applications; and that computing products include not only practical applications but also games, toys, new opportunities for artistic activities, and many kinds of collaboration. A growing number also know that developing programming languages, algorithms, and software applications can be enormous fun, and intellectually challenging. Those are all good reasons for expanding teaching of programming at all educational levels. However, not everyone knows there is another profoundly important aspect of computing that is not well understood: in addition to matter and energy, the universe includes information. Not all information processing systems use bit patterns implemented in transistors. Long before humans developed information-processing machines, biological evolution was developing natural information processing systems, in enormously varied, powerful ways, most of which we do not yet fully understand.

Micro-organisms process immediately available information in controlling their behaviour. More complex animals (and plants) have more varied ways of acquiring and using information to select options e.g. about which way to move, what to eat, how to avoid being eaten etc. Some use information collected and transformed over a period of time and stored for future use, e.g. where things are, which actions have which consequences, what things are unpleasant to eat, and how other animals behave. Some even use information about their own information processing and help their offspring to do the same.

Understanding such things is of great importance in philosophy, psychology, neuroscience, education, social science, linguistics, biology and many other fields. We still have much to learn about how the competences work, how they develop in individuals, how they evolved, and which species have which competences. We need to educate far more people so that they can think about the workings of information-processing systems, on the basis of first-hand practical experience of designing, building, testing, debugging and explaining (initially simple) examples. Our understanding is enriched by trying to model processes, often best done with programming languages and tools specially developed for the purpose. AI languages, (such as Prolog) help us produce systems that, in addition to their visible behaviours also have 'thinky' abilities: they can perceive, learn, reason, formulate questions, generate desires & feelings, make plans, build theories, and communicate.

Instead of teaching programming for its practical applications, we can also teach some learners to use computing ideas and programming experiments to illuminate other disciplines including modelling natural phenomena. For example, many learners like creating programs that manipulate fragments of natural language, initially in textual form and simple ways (as in chatbots or poem generators), then gradually adding deeper models of uses of language to talk about something. The enjoyment and learning that comes from getting a machine to manipulate language doesn't depend on producing a perfect conversationalist or poet. Analysing and reducing the imperfections of a flawed program can provide deeper learning than simply producing a bug-free program.





Above: how a machine understands a command "put the big green block on the little red one" showing the structure of the sentence diagrammatically. Having understood, it makes a plan to achieve the goal. The /-\ represents a hand! Left: The scene after executing the plan. Based on Winograd's SHRDLU , implemented in the AI development environment, Poplog.

There are now powerful tools enabling learners to program agents moving around in a simulated world where they sense things and react accordingly. A quite different sort of task is to develop simulated agents that acquire, store, use, share, and later modify information obtained from the environment. See the supplement for links to several examples, of varying sophistication. The opportunities for creative teachers are vast – if they have powerful tools and exemplars to start with. *Aaron Sloman*

PITTING MANY HUMANS AGAINST THE INTELLIGENCE

Will you be able to create a creature that out runs those designed by others on the net ... and if you can how about beating those created by machines using Artificial Intelligence. Sodarace is a joint effort between the Queen Mary Department of Computer Science and London based digital arts company Soda Creative Ltd. It allows humans worldwide to pit their wits against machine intelligence in an online Olympics. Humans, and AI programs, use the

MACHINES: SODARACE

ee Sodaplay constructor kit to reate lifelike virtual racers to ce over various terrain. Join e Sodarace community and arn how to build racers, or try iting artificial intelligence programs to beat your friends.

sodarace
www.sodarace.net

SIMPLE PROJECTS TO TEACH ABOUT THINKING MACHINES

It's a common notion that soon computers will be able to do everything. But will they? It's worth challenging your pupils to think about things computers can't do well. Can they see, for example. Simple questions can spark children's curiosity. We often hear people say computers are very clever, but what does that mean? Are computers intelligent? Can they think and can they be taught to learn?

Our friends at cs4fn (see back page) have produced some wonderful resources to help fire young minds. If you've never visited their website before, you're in for a treat. First up, use the links to browse by subject. A search for Artificial Intelligence will reveal over a dozen short articles, written in child friendly language, looking at fascinating aspects of AI.

There are also online applets to allow children to try to beat the computer at Nim, an ancient numbers game. There's links to a project called QGames that uses AI to deliver computer games to your mobile based on your skill level and SodaRace (see box below left). Most important, the cs4fn website provides ideas and resources for teachers to use in the classroom. Several of their classroom activities are ideal introductions to discussions of AI.

The Intelligent Piece Of Paper has never lost a game of noughts and crosses, and the website's supporting app allows students to understand the rules a computer uses to play the game. A natural progression is The Sweet Learning Computer, where a class can challenge a computer made of sweets to a game of hexapawn. The computer only knows 3 moves, but soon begins to learn from its experience when it starts losing sweets!

Finally, The Brain In A Bag introduces a group activity to simulate a small neural network that can play snap. Again, the classroom activity is supported by an online simulation to help reinforce the concepts encountered. All resources are free to download and provide stimulating ways to introduce computing concepts without needing a computer.

SWITCHEDON web supplement: www.computingatschool.org.uk

COMPUTING AT SCHOOL: LAYING FIRM FOUNDATIONS

Many of you will know that, following a consultation with our members, CAS has formed a strategic partnership with the BCS, The Chartered Institute for IT. The institute is the professional society for our discipline, and recent launched the BCS Academy of Computing, so it is the natural home for CAS. Better still, the BCS Academy is very, very supportive of CAS and its crusade to improve the teaching of Computing in our schools.

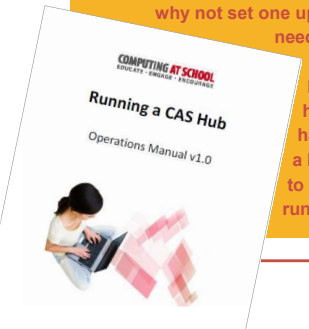
Partnering with a major professional body is immensely helpful in lots of ways.

- It gives us a base for making our case at government level. For example, our recent article in the Science in Parliament Journal was written by Bill Mitchell at BCS.
- BCS is already contributing significant resources in terms of time and money.
- As CAS grows, our informal mechanisms for managing membership (ie Simon PJ + a spreadsheet) have been creaking. As a membership organisation, BCS can handle this load.
- BCS gives us communication channels to thousands of computing professionals.
- In time, BCS would offer a natural route for professional accreditation for Computing teachers, should you want that.

CAS is, and remains, a grass-roots organisation, whose energy, creativity, and drive comes from its members. Our partnership with BCS gives us a tremendously helpful umbrella, but it absolutely does not presage a shift to a bureaucratic, top-down management style. CAS will change the world only if WE roll up our sleeves. As I keep saying: there is no "them"; there is only us. *Simon Peyton Jones*

ORGANISING A LOCAL CAS HUB

If there isn't a CAS hub yet in your area, why not set one up? Often it just needs an individual to get the ball rolling. To help you CAS have produced a helpful guide to get you up and running quickly.



HUBS PROVIDE A CHANCE TO MEET AND SWAP IDEAS

Local hubs of CAS supporters are taking off around the country. Such groups provide an excellent opportunity for teachers to meet with other computing enthusiasts.

The second half of the Autumn term was a particularly busy time with new groups launched in South Wales, Teesside and Surrey and several established hubs meeting . The following snippets give a flavour of the vibrant atmosphere they generate. The second meeting of the North West hub included a presentation and discussion lead by Dr David Rydeheard on algorithmic complexity, problem solving and big O notation; an area many teachers reported their students struggled with. The session was very well received, with teachers looking forward to trying some of David's techniques with their students. Bohdan Serednycky also presented his experiences using 80's games with A-Level students, as a good way to get students involved with programming.

The meeting in West Berks generated plenty of discussion around the themes of using Google's Android App Developer in the classroom (including a demonstration downloaded onto a phone). Vital presented their CPD offerings, different KS4 and 5 courses were debated as was the problem of teachers being overwhelmed with potential resources.

Andrew Tringham and Shahneila Saeed led valuable discussions on GCSE Computing at the East and South London hubs respectively. At both venues Michael Kölling delivered a highly successful workshop on Greenfoot. Many attendees asked, via the feedback, for 'more time' and 'longer sessions' with many staying long after the sessions had ended

The launch of the Surrey hub was opened by Tony Ho, the new Head of the Computing Department at Surrey University. Attendees also heard from Bill Mitchell, who explained the role of the BCS Academy and the importance of the Royal Society's study of Computing in Schools. The need to engage with industry was the overarching theme of the subsequent talk by Roger Peel, Senior Tutor for Professional Training at Surrey, and BCS Guildford Branch Chair. There followed a double act by Emma Mulqueeny and Isabell Long. Emma has been running 'Young Rewired State', inspiring students to use computers to delve into open Government data during so-called hackdays. Isabell was a hackday participant who created GovSpark, a website that ranks Government departments by their increase or decrease in energy use. A short input, from Fiona Henry, discussed the resources that Vital are able to make available, followed by a short workshop resulting in a set of actions and agenda for subsequent hub meetings. Throughout there was a vibrant and interactive atmosphere, which was in part responsible for the 2 hour session finally finishing after 4 hours. Many took the opportunity to form new contacts with other teachers and those offering curriculum support. Though likely tired after such an intensive session, participants left looking forward to follow-up activities.

Many thanks to Claire Davenport, Ona Sumner, Thomas Ng, Shahneila Saaed and Lee Gillam for hub reports. Contact details for all hubs are in the web supplement, but if there isn't one near you, why not consider setting one up - see the box left for more details.

VITAL ANNOUNCE EXCITING NEW CPD OPPORTUNITIES

Vital has worked with CAS to create a range of support materials for computing teachers, designed to encourage staff to introduce and develop programming activities for their students.

Because teachers sometimes lack the confidence to use programming languages in lessons, students don't always get the chance to explore the opportunities they offer. To support the early introduction of programming in schools, Vital sponsored three CAS members – Amada Wilson, Zoe Ross and Mark Clarkson – to develop initial support materials for the programming languages Scratch and Alice. The materials are free to download and provide simple interfaces which let users easily build programming sequences. There are eleven short videos about Scratch, each introducing a few key skills. These resources offer a steady, linear progression through programming commands. Alice is supported by seven videos which together help teachers build an interactive car racing world. The videos can be easily replayed, so teachers can work through them at their own pace.

Vital has also developed four videos for Microsoft Small Basic to support the programming elements of the OCR Computing (J275) course tasks. The first covers control statements and arrays, the other three look at simple collision detection and shape transformation as ways to support the course set tasks.

Log onto the Vital website to access all these programming materials. You can also go directly to the CAS teachers' space to find case studies, technology guides, resources and access online events and discussion forums.

Vital are also pleased to be hosting a series of one-day CPD courses for A Level computing teachers. Designed by CAS, these sessions are taking place at Vital's offices in Manchester, Birmingham, London, Gateshead and Bristol throughout February 2011. The courses (see right) are designed to help A Level computing teachers master some of the more complex topics on the AQA A2 computing course. *Emma Blackburn*

THE KEYBOARD IS MIGHTIER THAN THE MOUSE!

The School of Computer Science at The University of Manchester is currently running a programme of hands-on Linux command line taster workshops for sixth-formers, generously funded by CAS/BCS.

The workshop focuses on the power of the command line, and takes students through the fundamentals of shell commands and programming, demonstrating that thoughtful use of the command line can often far exceed the usefulness of graphical user interfaces.

This is the second year the workshops have been running, and to date over 270 sixth formers from over 20 schools have taken part. Feedback has been very positive from teachers and students alike, and we shall be making the workshop materials freely available to CAS colleagues. Five more workshops will be announced in early 2011. Check the links in the web supplement for dates, further details and how to book. *Toby Howard*

PLANS FOR MORE CAS 6TH FORM CONFERENCES

Following the success of last year's Sixth Form Student Conferences plans are being finalised for a second series this term. They are a great way for schools to participate in a variety of workshops and presentations. The subjects can range from topics currently being studied, to wider insights into emerging trends within the field and the opportunities higher education can offer for those wishing to study the discipline at a higher level.

Dates already confirmed are:
8 March: King Edward VI Camp Hill School for Girls, Birmingham
11 March: City of London University
16 March: Glyndwr University, Wrexham and Warwick University
18 March: Northumbria University
22 March: Oxford University
23 March: John Moore University, Liverpool
24 March: Cockermouth School, Cumbria
25 March: John Leggott College, Scunthorpe

CAN YOU HELP?

If you are a CAS supporter in industry or higher education, would you be able to spare some time to participate in these or further conferences? If so, we would love to hear from you. Please inform Claire Davenport, who is coordinating the events, on claire.davenport @ hq.bcs.org.uk

VITAL CPD FOR A LEVEL

There are still a few places left on the following full day sessions:
2nd Feb 2011 Manchester
3rd Feb 2011 Birmingham
4th Feb 2011 London
8th Feb 2011 Gateshead
9th Feb 2011 Bristol

At under £60 per head they offer excellent value. See the supplement for details of how to book.

A FORCEFUL CASE FOR CHANGE

Science In Parliament is the journal of the Parliamentary and Scientific Committee, which is an associate committee to Parliament. 'The Collapse of Computing Education in Schools' by Simon Peyton-Jones and Bill Mitchell appeared in SIP Vol 67, No 4, p39, Autumn 2010.

Copies of the journal are circulated to MPs and members of the House of Lords. It is also circulated to other learned bodies such as the Royal Academy of Engineering. We wrote the article in an attempt to raise awareness of Parliamentarians to this issue, and we are very grateful to the Parliamentary and Scientific Committee for accepting the article.

The print article is a version of the Royal Society item currently on the BCS Academy website. The article comments, "The concern is that, in too many cases, children learn only how to use office software such as word processors or spreadsheets, and miss out entirely on the excitement of learning how computers actually work. An analogy would be if classes in English consisted only of learning how to spell, but missed out how to write good prose, how to analyse literature and omitted how to articulate ideas and arguments concisely, elegantly and in a compelling way."

As mentioned in the last issue, The Royal Society is leading an 18-month project looking at the way that computing is taught in schools, with support from 24 organisations from across the computing community including learned societies, professional bodies, universities, and industry. The call for evidence, the first phase of the inquiry has now closed with over 120 responses received.

The full BCS and CAS response to the call for evidence is also available for download on the BCS Academy website. The 24 page response addresses all the questions posed by The Royal Society and makes a forceful case for curriculum change. If you are new to the ideas put forward by CAS it will prove a valuable read. See the web supplement for links to these and several other key documents that outline where we stand.

Bill Mitchell

BUILDING THE FUTURE LIES IN UNDERSTANDING THE PAST

The National Museum of Computing is dedicated to showing the development of computing in its broadest sense from the pioneering war time efforts that resulted in Colossus, to the products and systems we use today.

HG Wells said that 'Human history in essence is the history of ideas'; and as I ponder the technologies of the past in the Museum at Bletchley Park, I often look beyond the exhibits to the imagination, enterprise and ideas of the men and women that conceived, built and operated those remarkable machines.

The intriguing story of modern computing over less than a hundred years is one of ideas and creativity with world wars, secrecy and the rise of technological empires thrown in for good measure. The Museum charts much of this; from computers big enough to stand inside to those that get personal, fitting snugly into your pocket.

It is not easy to stand alongside the ghosts of those that developed those ground-breaking machines and imagine what they were thinking as they took each critical step. Dare you imagine a world without computers; the twists and turns of technology, no worldwide communications through which to shout to others and far more dead ends than we care to remember? Yet there was all to play for, all to invent and develop; how exciting that must have been?.

Primarily focused on developments in the UK, the Museum is manned by volunteers some of whom work hard to restore exhibits back to life. Based in the world's first purpose built data centre and in Bletchley Park, the Museum is within yards of where Alan Turing worked with ideas that would lead him to the concept of a computer.

Checkout the Museum's website for more information about visiting times and visits by school/college groups. We would welcome visits by teachers who would like to share their ideas for how we can make the Museum relevant to young people, inspiring them to follow in the footsteps of those in the past, enthused and driven to write the next chapters in the remarkable story of computing. One young visitor listened attentively as part of the story of storage was explained to her. At the end she turned and asked; "but what comes next?" "That's for you to decide", our volunteer said, "now it's your turn to make history".

Chris Monk

BCS CELEBRATES FIVE INFORMATION PIONEERS

Information Pioneers was a campaign by BCS, The Chartered Institute for IT, to show how the contributions of five very different people helped shape the information society of today. As part of the project, five short films, created by an award-winning production team, provide snapshots of the work of Alan Turing, Ada Lovelace, Hedy Lamarr, Clive Sinclair and Tim Berners-Lee. BCS members helped choose who to profile and a month long vote drew nearly 10,000 responses. You can find out who won, and watch the videos (which provide excellent material for lessons) on the Information Pioneers website. Links in the web supplement.

RESEARCHING THE POTENTIAL IN TEACHING PROGRAMMING

Three important research projects are currently being undertaken by CAS members; Cynthia Selby (University of Southampton), Claire Johnson (University of East Anglia) and Mara Saeli (Technical University of Eindhoven).

The renewed interest in promoting computing in the school curriculum is mirrored in formal research at doctorate level taking place in this area. Claire and Mara are now analysing their data whilst Cynthia is just starting her work.

Claire's thesis explores the potential of computer game authoring in the Key Stage 3 ICT curriculum and proposes that computer game authoring can provide a motivating and accessible introduction to basic computer programming, in the context of the 'sequencing instructions' substrand of the Secondary Framework for ICT. Authoring computer games, as an introduction to basic computer programming, is an important element in developing young peoples' digital literacy.. It provides them with an authentic, creative scenario in which to learn about how to sequence instructions to create an interactive artefact. It encourages purposeful engagement with basic programming concepts, and provides a means to strengthen this important element of the KS3 curriculum, which, according to Ofsted (2009), is one of the least well taught areas in the subject.

Mara's research focuses on the important area of the Pedagogical Content Knowledge (PCK) of programming, that is, not simply knowing how to program but knowing how to teach programming. For example, it investigates the use of metaphors to help students overcome difficulties and misconceptions. PCK is the knowledge that experienced teachers use to transform their understanding of the topics into forms of representation accessible by students. With a total of 31 participants and 7 topics explored (loops, data structures, arrays, problem solving skills, decomposition, parameters and algorithms), Mara's results will help identify the PCK of programming for secondary education. These results will then be used to produce a handbook for beginning teachers to accompany them during their first years of experience.

Cynthia asks, what are the characteristics of a good first programming language? Students, teachers, programmers, computer scientists and academics frequently debate this question. What makes this topic so important is the fundamental impact it can have on students' later development. When giving advice to students wanting to learn to program, Gupta asserts that the choice of initial language will affect "… your programming style, coding technique, and code quality in many subtle ways" (2010). This research attempts to address the issue of teaching programming to students aged from 16 to 18 years old, specifically in terms of programming language choices, the practical issues, students' and teachers' expectations and the learning objectives. It is anticipated that a framework can be constructed, not to evaluate the design of the languages, but to evaluate the efficacy of using particular languages in courses that involve introductory programming.

The CAS mail list has proved very useful for raising issues and gaining comments from a wide range of experienced and engaged colleagues. If you are requested to complete questionnaires or to be interviewed then I strongly recommend that you respond fulsomely. It is these guys that will be providing the rigorous evidence for the gut feelings many of us possess.

John Woollard

GIRLS AND GADGETS HELPS DISPEL MYTHS

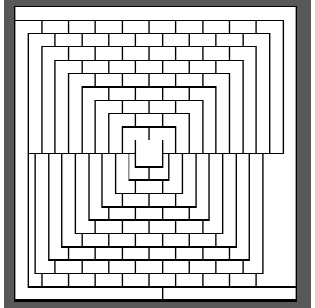
On 24th November around 110 teenage girls participated in the Girls and Gadgets Conference designed to destroy the myth that computers are for boys. The annual event was launched in 2008 and allows girls to spend the day at Teesside University engaging with all aspects of computer science. Alongside inspirational speakers, they took part in a range of activities to help them understand the creative and technical skills involved in computing. One session called Cover Girl, focused on how magazine images are manipulated by smoothing skin tone and changing body shape.

The girls came from schools across the Tees Valley, Durham and Northumberland areas. Amongst those taking part was Daniella Richardson, from Oakfields School in Middlesbrough. She said: 'It's been really interesting listening to the different talks and finding out that there are very few women working in IT.' Although girls are now using computers at rates similar to their male peers, they are still five times less likely to consider a technology related career. There has been a drop in the number of students taking computing- based subjects at degree level and historically women have been under represented in this field. Alison Brown, Senior Lecturer in Computing at the University, said: 'We hope to change this trend and to encourage more girls to consider a computer based career, whether in networks, animation, web development, mobile phone software or games development.'

A PAUSE FOR THOUGHT

The Four Colour Theorem marked the first major use of computers in mathematical proof. In 1852 Francis Guthrie was colouring a map of British counties and noted he needed at least four colours to avoid duplicate colours on adjacent counties. Guthrie asked if it was possible to draw a map requiring more colours to avoid identically coloured neighbours. In 1879 Alfred Kempe published a proof, but this was disproved in 1890. Percy Heawood demonstrated no more than five colours were needed, but the original Theorem remained unproven.

In 1976 Kenneth Appel and Wolfgang Haken proved any map could be reduced to one of 1,482 types so, if all types only needed four colours, so did every map. Checking all 1,482 maps would be unfeasible so Appel and Haken programmed a computer to do the colouring. The analysis took 1,200 hours of computer time and discovered all 1,482 maps required four colours. This helped define 'topology' which studies diverse fields like robots' movement and the shape of the universe.



It was later claimed this map of 110 regions requires five colours and constitutes a counter to the four-colour theorem. Have a go yourself. For a full sized copy and further information, see the web supplement. *Lyndsay Hope*

COMPUTING ON THE CATWALK OUT NOW!

The latest issue of cs4fn's free magazine is published in January. In this issue you'll find all sorts of ways that computer science can improve your style. You'll read about t-shirt designs made with the help of evolution, a mirror that helps you choose what to wear, and the technology behind Tony Stark's Iron Man suit. Plus they get to the bottom (so to speak) of why robots always insist on going naked.

Did you know that the team behind cs4fn do talks for schools as well? Last autumn they spoke to more than 2000 students in class sizes ranging from 30 to over 200. They give most of their talks to schools in London and the Home Counties, but they can travel further afield for larger audiences. See the web supplement if you'd like to learn more about all cs4fn activities, website, magazine and topics covered in their talks.

BCS ACADEMY HOST COMPUTER SCIENCE MAGIC SHOW

Witness an amazing magic show and sneak behind the scenes to explore the maths and computing powering the tricks. This free public lecture takes place on Tuesday 25 January 2011 from 5pm at the Manchester Museum, University of Manchester. Following a welcome by the hosts, The BCS Academy of Computing, guest speaker, Peter McOwan, will treat you to an educational talk that reveals the fun side of computer science. Peter's research interests are in visual perception, mathematical models for visual processing, in particular motion, cognitive science and biologically inspired hardware and software. He is also one of the driving forces behind cs4fn, co-authoring their two volume Magic Of Computer Science books, available from the cs4fn website. Peter was awarded a National Teaching Fellow in 2008 by the Higher Education Academy.

COMPUTING AT SCHOOL

EDUCATE · ENGAGE · ENCOURAGE

CAS...JOIN CAS...JOIN CAS...JOIN C

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. Our goal is to put the fun back into computing at school. Will you help us? Simply mail:

membership @ computingatschool.org.uk

Many thanks to the following for help with this issue of SWITCHEDON : *Jonathan Black, Emma Blackburn, Alison Brown, Tom Crick, Claire Davenport, Roger Davies, Mark Dorling, Lee Gillam, Lyndsay Hope, Toby Howard, Simon Humphreys, Michael Kolling, Jack Lang, Nicki Maddams, Peter Marshman, Bill Mitchell, Chris Monk, Thomas Ng, Simon Peyton-Jones, George Rouse, Shahneila Saeed, Aaron Sloman, John Stout, Ona Sumner, Alec Titterton, Zena Wood , John Woollard.*

SEE THE SWITCHEDON WEB SUPPLEMENT FOR MORE LINKS AND EXTRA INFORMATION

www.computingatschool.org.uk

Computing At School are supported and endorsed by:

The Chartered Institute for IT
Enabling the information society

Microsoft Research

CPHC
The Council of Professors and Heads of Computing

vital