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

SWITCHED ON

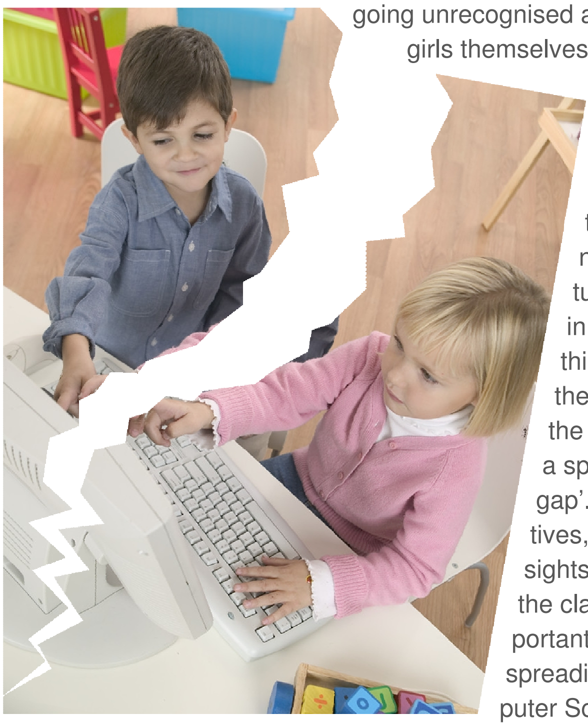
COMPUTING AT SCHOOL NEWSLETTER WINTER / SPRING 2013

MIND THE GAP

GETTING GIRLS INTO COMPUTING

Computing is back on the curriculum. The last year has seen a growing recognition of the educational importance of 'computational thought', not just for future technologists, but for all pupils. Few subjects can contribute more to the Personal, Learning and Thinking Skills required to equip young people for future employment and lifelong learning. Breaking tasks down, constructing algorithms and writing programs require children to think about thinking. When a program doesn't behave, a child must investigate. When a mental model is challenged by reality a child learns. Debugging techniques facilitate learning about learning. None of this is new. Educationalists have been saying it for years and now we have a chance to put the theory into practice.

How teachers portray our emergent subject to pupils is crucial. Computing and IT has often been viewed as a boys' thing. Negative stereotyping, project contexts, the hidden curriculum of the classroom, past contributions from women going unrecognised and peer pressure amongst girls themselves have left a gender imbalance. As the first strokes are placed on the blank canvas of a new Computing curriculum it's important to challenge these factors. A new beginning carries opportunities to portray the subject in a fresh light. Great female thinkers have always been at the heart of developments in the field. Inside this issue we've a special focus on 'minding the gap'. You'll find reports on initiatives, pointers to resources, insights into things that can work in the classroom and, most important, ways to get involved in spreading the message that Computer Science is for everyone.



LAYING FIRM COMPUTING FOUNDATIONS IN 2012-13

Twelve months ago CAS was patiently making the case, as it had for the previous three years, for Computer Science to be put back on the mainstream curriculum. Momentum had been growing and many individuals and organisations were voicing similar concerns to ours. 2012 was a tipping point. The Royal Society Report into Computing in UK Schools published a year ago set out the challenges facing education, and pointed ways forward. In his keynote at BETT last year, Michael Gove recognised Computer Science as a rigorous, intellectual discipline. Since then, CAS members have been busy laying the foundations on which to rebuild the subject. You'll find more details on what's happening inside.

Together, ICT teachers have a once in a lifetime opportunity to reshape their subject. Up and down the UK, many are seizing that chance and sharing their experiences in the CAS community, its local hubs and emerging Network of Excellence. These are exciting times. Many challenges still lie ahead, but we're hearing about more and more teachers taking their first steps in Computing and finding it to their, and their pupils liking. If you haven't yet, why not make it a New Year resolution?

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

bcs Academy of Computing

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

2012 THE YEAR SCHOOL COMPUTING CAME BACK ONTO THE AGENDA

A year ago, the numbers arguing for curriculum change were small. Things move quickly. It's been a year of change, uncertainty and excitement. Simon Peyton Jones look back on a busy year for Computing At School.

A year ago in his speech at BETT 2012 Michael Gove announced: *"Computer Science is a rigorous, fascinating and intellectually challenging subject."* He went on to say: *"Long after today's pupils leave school and enter the workplace - long after the technologies they used at school are obsolete - the principles learnt in Computer Science will still hold true".* In the strategic information pack we sent to head teachers in the Spring we launched a Network of Excellence for Teaching Computer Science. Making the Network real is a massive challenge that we are still grappling with, but it's exciting! Launch events were held in September and you can find out more details on the page opposite. In October, Michael Gove announced new £20,000 scholarships for graduates to train as Computer Science teachers. A new stream of 50 teachers/year will not solve the problem – CPD for our terrific existing ICT teachers is essential – but I am thrilled that DfE/TA have absorbed the message of the importance of Computer Science as a discipline.

The strength of CAS was evident at our annual teacher conference in June where the new CAS Online web site was launched. Nearly 300 resources, 2400 members, tripled participation.... thank you Neil Brown and Michael Kolling, who made it happen. Read more on page 4.

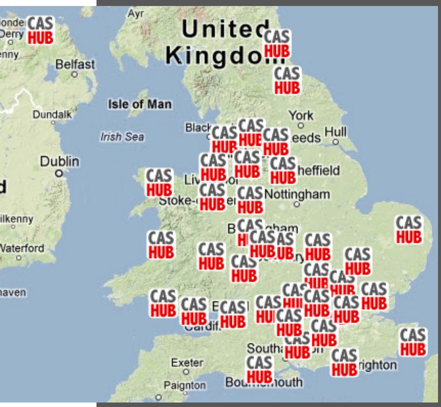
Whatever the merits of the English Baccalaureate, it's clear it is a major driver. Michael Gove indicated his willingness to consider Computer Science for the E-Bacc in his BETT speech and repeated the offer in October. Bill Mitchell coordinated a report making the case for inclusion, endorsed by a range of stakeholders. It was sent to DfE in November and we wait in hope. In August the DfE invited BCS and the Royal Academy of Engineering to coordinate drafting of a new Programme of Study for ICT. The working party, which included representation from Naace, ITTE, Vital and NextGen Skills submitted a first draft in October. DfE will publish their proposals for full public consultation in the Spring.

CAS has morphed from a guerrilla group at the bottom of a deep mine shaft shouting "computer science is important" into a group with major impact on national policy. You should feel proud of what you have all done. This is a time of change, and change brings both opportunity and danger. The opportunity is clear. We have a once-in-generation chance to make a substantial and lasting change to our children's education. The iron is hot, and we must strike it now! The dangers are equally obvious: allowing computer science to turn into coding; heightened expectations that are not met; and the insidious death of a thousand qualifications. These dangers are real but manageable, if we continue to deal with each other with graciousness, respect, and encouragement. We must deal with our growing pains without losing our grass-roots vigour. Although CAS is now quite large, we continue to draw our credibility and energy from our members; that is, from you. Our motto remains as true as ever: There is no "them", there is only us. Links to all documents mentioned are in the supplement.

THERE IS NO THEM - ONLY US CAS HUBS CONTINUE TO EXPAND

Last term over 900 people registered for 43 CAS events (mostly hub meetings) taking place all over the UK. Existing hubs have a strong following and offer targeted computing workshops for their teacher members. These often follow consultation on what is required, popular subjects being Scratch, Assembly simulators and Python.

In September Dr Rob Williams stepped down from running the CAS SW hub. Now run by Ian Johnson, he will continue all the great work that Rob achieved in their BESI outreach project to local schools. Thanks for bringing the group into existence Rob and for your contributions and energy to CAS. New hubs have sprung up throughout England and Wales: Bath, Bedford, Burnley and Pendle, East Lancashire (Rochdale & Bury) East Midlands (Notts), Hastings, Hull (East Riding), Mid-Wales (Lampeter), Mid-Wales (Aberystwyth), Somerset, South Wiltshire, South Yorkshire, Stafford and Tyneside. To find details of your nearest hub, take a look at the CAS Community map. Zoom in and click the hub to get contact details of the organiser. You can ask questions and make suggestions for hub activities in the regional forums too. *Claire Davenport*



WHERE DID CAS COME FROM?

A paper describing CAS's experiences and growth has been accepted for SIGCSE, a major international conference on Computer Science education, to be held in Denver next March. If you are new to CAS, it's well worth knowing where we came from. Links in the supplement.

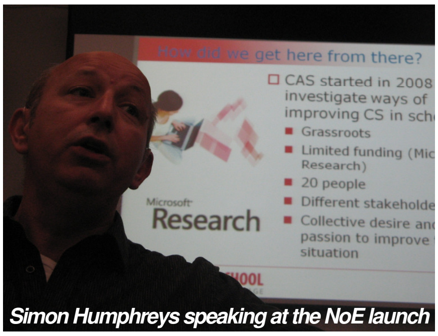
BUILDING OUR CAPABILITY VIA THE NETWORK OF EXCELLENCE

2013

The Network of Excellence, launched by CAS brings together 570 schools and 37 Universities, working together to promote and support Computer Science. We hope it will be the start of a self sustaining movement to train a new generation of teachers.

The main idea of the Network of Excellence (NoE) is to organise and deliver CPD in computer science to ICT teachers across the country. This is no small task: to scale ourselves up from a bunch of volunteers to an organisation capable of training thousands of teachers. We only have a chance of doing this because we are in partnership with a national network of institutions that know a bit about teaching computer science, namely our universities. One of the most positive aspects of the NoE is the opportunity it provides for university computer science departments to engage directly with schools. Launch events were run in September and all registered schools should have received details of Lead Schools and Master Teachers in their area by now.

So the last few months have been busy. All applications from schools wishing to become a 'Lead School' have been considered, and as a result we have designated 102. These schools have signed up to share their expertise and experience with the community by for example, partnering with a local primary or secondary school, contributing to the CAS Regional Hubs and supporting the Regional CAS Master Teacher to run CPD events. A list of these schools is available online. We have also appointed and trained 27 'Master Teachers', funded by the Department for Education (DfE) together with leading industry partners and awarding bodies. We recognise that this number isn't nearly enough to support the whole of England but it is a good start. We are putting the finishing touches to a funding model and will advertise a second phase of appointments soon.



Simon Humphreys speaking at the NoE launch

We have allocated all schools in the Network of Excellence into a geographical region, with Master Teachers and Lead Schools situated at the heart. We appreciate the inflexible nature of postcodes, for example, where a school is located on the borders on one or more region. We are current working on a system for the NoE website whereby schools can find their nearest three Lead Schools based on distance to complement the regional approach and provide the flexibility required.

If you are interested in registering for the NoE or becoming a CAS Master Teacher then please see the link in the supplement. *Mark Dorling*

The first tranche of CAS Master Teachers pose for the camera at their recent training day in London

CAS MASTER TEACHERS THE FIRST GENERATION

I had a most exciting day, taking part in the initial Master Teacher training at BCS, The Chartered Institute For IT headquarters. I was met by other computing teachers, all equally intrigued to find out what we had gotten ourselves into. What we have "gotten ourselves into" is something truly worthwhile. Simon Humphreys, who co-ordinates CAS, kicked the day off with an overview and introduced Mark Dorling (a primary, and previously secondary school teacher himself) who went into more details about the specifics. He likened himself to our teacher, which made me smile since he did set us plenty of homework … !

In a nutshell, in conjunction with local CAS Hubs, Universities and interested Lead Schools we are to help co-ordinate and run CPD sessions to train non-computing specialists up in "Computing" topics, ready, and in preparation for the new 2014 ICT PoS.

No small task, which, to their credit they fully admit. What was nice is they didn't mandate at all what we "had" to deliver. In fact a fundamental part of our remit was to go out and source the local need, be it at Key Stage 1,2,3 or 4! So with all that on my plate (don't forget we are all pretty much full time teachers anyway) I am a mixture of nerves and excitement. This sort of opportunity doesn't come along every day. Computing is something I am passionate about, and I can't wait to get into local schools, find out what they want, what they need and start helping. *Craig Sargent*

CAS WHITEPAPER AIMED AT TACKLING TECHNICAL WOES

Management of school IT infrastructure can be one the biggest impediments in implementing Computer Science. The needs of teachers and learners and the requirements imposed on network managers, whether in-house or outsourced, can be a source of conflict. The apparently diametrically opposed position of service providers and their end users has always been there and it applies to the teaching of other subjects not just Computing or ICT. The introduction of Computer Science as an academic subject adds a further complication. With the right support and information the school infrastructure can deliver the processor power and networking that students need for computer science without compromising security or integrity of the other users on the system or their data.

With good training and management, in-house and third party service provider's can support the specialist needs of all teachers. This means teachers need not struggle with a heavily tied down system because the necessary protections are implemented correctly and function quietly in the background. The experience of many CAS members has shown it's possible to teach Computer Science with support from senior leadership, a constructive dialog with the service provider and co-operation from a student body supported by a well understood Acceptable Use Policy.

As we go to press CAS are finalising a whitepaper aimed at school leaders detailing the needs of teachers and infrastructure providers. It includes details of a range of technical solutions available to facilitate learning Computer Science which can accommodate all budgets and skill levels. Many will fit easily into existing support arrangements or contracts without jeopardising Service Level Agreements and the integrity of the system. The paper illustrates how a senior management led Computer Science strategy, mechanism for implementing solutions and a good change management process can all be achieved. As soon as it is published, details will appear online. Ensure you download a copy and distribute to your school leadership team. *Richard Cornell*

CAS Online
Resources / Discussions / Events

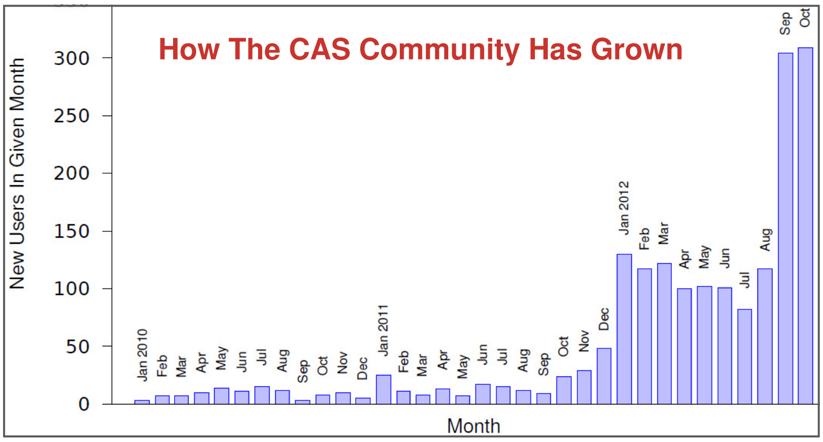
A COMMUNITY TO SHARE IDEAS AND EXPERIENCE

In August 'CAS Online' was launched. The new website provides a more featured, fully-fledged community site, to better support CAS members and their interactions. Neil Brown gives us a quick tour of all the new features.

Briefly, the new community supports participation in discussions, sharing resources, publicising events, organising hubs, contacting members and viewing a map of members, hubs and events. The most prominent feature is the ability to share resources. While there is merit in teachers developing their own resources there is a lot of wasted effort in each developing their own material. Teachers can share what they have part made, collaborating on new resources together. It saves time and effort, and the impact of great resources isn't confined to a single classroom. Sharing is built around a wiki model: resources can be viewed and edited by all CAS members. You can use the wiki mechanism to collect together useful links or collaboratively develop a resource. The idea is to encourage joint improvement: if you have something to add, please do.

Upcoming events are public. They can be imported directly from Eventbrite, making listing them quick and easy, or created manually. Each CAS hub has its own webpage. Hub events from the event feed are displayed publicly, advertising them beyond the membership. CAS Online also has a private directory with a profile for each member. The key thing linking members, hubs and events is location, location, location! All are displayed on the CAS Online map providing an easy way to find hubs, events and members near you. The discussion area is fairly straightforward with categories such as primary education, the CAS Network of Excellence, and regional forums. You can choose which categories and how frequently you wish to receive e-mail notification, to aid keeping up with pertinent discussions without getting overwhelmed.

The site is, by design, part private, part public. Discussions are private so teachers can discuss sensitive issues without worry that posts will be seen by pupils, and industry professionals can speak 'unofficially'. Resources too are private so answers can be uploaded along with resources. The private part of the site is accessible only to members, and we verify on signup that new members are teachers or professionals -- not pupils. The CAS Community will continue to develop. Our aim is to support the membership in all aspects, but especially teachers in their classrooms as they take on the challenge of teaching computing.



MUCH ADO ABOUT KNITTING: THIS WILL HAVE YOU IN STITCHES

"My name's Steve and I'm a knitter." The group of teachers looked at me strangely. "Really?" they questioned. "Yes," I said, "and I think there's a link between knitting and computer programming." So, how did this all start? Let me spin you a yarn...

A year ago, Great Uncle Norman came to visit. He had worked in Scotland in the 1950s. While he went around the isles, he learnt Gaelic, songs, stories and traditional crafts. He taught me about knitting, how important it was for communities. How men and women knitted out of necessity, rather than as a hobby. He told me of Ganseys - thick sweaters worn by the fishermen. Each village had a different pattern, so in the event of a man-overboard, the body could be identified. Talking with my Great Uncle, I realised knitting would be great in schools, for example, the physical benefits of finger strength, co-ordination and dexterity. Developing language skills like instructional writing, speaking and listening. Social skills of collaboration and communication. Links with communities too, as children could interview relatives about their experiences. So, in Summer 2011, I cast on!

Be crafty and join together

Knitting isn't easy. At first, I tried using books. Even though the pictures were clear with detailed written instructions, I couldn't understand properly. Further help was found online. YouTube is amazing; not just hundreds of enthusiasts, but also yarn companies, such as Lion Brand helping beginners to learn. What an opportunity for children? They could not only learn from videos, but create their own to teach others! Craft Club is a national campaign to develop craft in schools, a great initiative. They find volunteers to share their expertise. Hand knitting enables children to learn sequences of instructions, without complicated needles. Progress is easily seen - an example is in the supplement.

Many children have encountered knitting, but few could remember how to do it. Those who were able to knit were generally taught by a grandparent. They had an emotional bond. Their instructor had taught them on a one-to-one level and persevered to help them. Solving problems and learning together had made a lasting impression.

Tying up loose ends

So, what is the analogy of knitting and programming? Stitching it all together, let's consider three ideas.

- 'User experience' - people knit for a purpose, with a person's requirements in mind, such as size, colours, patterns and materials. This is the same as programming, where a program needs to meet the requirements of the user.
- Knitting follows a sequence of instructions - most people will say 'knit one, purl one' if you mention knitting. These instructions follow a syntax (vocabulary), including variables, loops and decisions (If...then...else). For example, if you need to knit fifty rows, you will knit one row at a time until you reach row fifty. Therefore, the program could be 'If row is fifty then stop, else keep knitting.'
- Finally, when you make a mistake you need to fix it - this is debugging the program. As I've learned to knit, I've made many mistakes. Each time, the stitches have to be undone and replaced. Debugging programs is not only necessary, but difficult. Both knitting and programming require perseverance and struggling with problems.

To infin-knity and beyarned!

When this started, I thought knitting could help social skills and new technology capture disappearing crafts. One lesson in the similarity between knitting and coding, is that future programmers need to start young. So, what next? Children love the idea of becoming hackers - and the knitting analogy continues there too! Try searching online for 'Yarn bombers' or 'Guerrilla knitting' - they've got it covered! *Steve Bunce*

KEEP CALM & CARRY YARN!

How are children currently learning to program in primary schools?

The 'BeeBot' is popular for inputting instructions to move the brightly-coloured robot around the floor. Recently released are two BeeBot apps for the iPhone and iPad, which encourage the children to instruct a virtual BeeBot through many fun games).

The Crystal Channel, from Planet Sherston, has many high-quality resources to teach sequencing instructions. These include 'The Crystal Rainforest', 'Flobot' and 'Mission Control 2'. The children I observed loved following the stories while developing their instructional skills, for example, programming a drinks machine to deliver the perfect drink.

LEGO WeDo provides the children with a simple interface to control familiar LEGO models. The supporting videos help them to build and program a variety of interesting projects.

'Scratch' teaches programming by using blocks of instructions to create the programs. The children are encouraged to share their creations back on the community website. A new web version, Scratch 2.0 is due for release in early 2013.

Find out more about Knitting and Programming at the BETT Show, Learn Live seminar where Steve Bunce will be speaking on Wednesday 30th January.

A RECURSIVE SWEATER

If you would like to explore further the analogy of knitting and Computer Science, Anna Bernasconi, Chiara Bodei and Linda Pagli from the University of Pisa, Italy jointly authored a wonderful paper looking at recursive patterns in knitting. The link can be found in the supplement.

PRIMARY SCHOOLS CAN BUILD ON THEIR EXISTING PRACTICE

There is much excellent ICT teaching undertaken in Primary schools. There is also a fear that old schemes of work will need to be replaced with new, leading to anxieties over subject knowledge and time constraints. The Digital Schoolhouse project (DSH) based at Langley Grammar School has encouraged local primary schools to review existing control units e.g. the ones using logo and Flowol against the CAS document 'A curriculum for schools'.

Non-specialists can struggle locating the best resources. The DSH has encouraged them to use the CS4FN website. It provides excellent examples of how Computer Science links to other subjects, providing inspiration for adapting existing schemes of work. The aim is to show that, if taught properly, what was relevant last year is still relevant this year. All that may be needed is to make computer science elements more explicit i.e. focusing on Computational Thinking and using correct vocabulary.

DSH has been working with Primary teachers in Slough to give a face lift to some schemes like the year 4 unit, Repeating Patterns. Scratch's drag and drop blocks: control, motion, pen and stamp, are excellent tool for pupils to explore and apply what they have learnt in maths to draw simple shapes. Scratch enables teachers to introduce concepts such as repetition, using simple shapes to create patterns with repeats and rotation. Our experience has shown it best to separate the thinking from the coding. Once pupils have grasped the concepts using a graphical tool such as Scratch, the work can then be extended to introduce pupils to coding by using an old favourite, Logo. *Mark Dorling*

COMPUTER SCIENCE BITS & BYTES

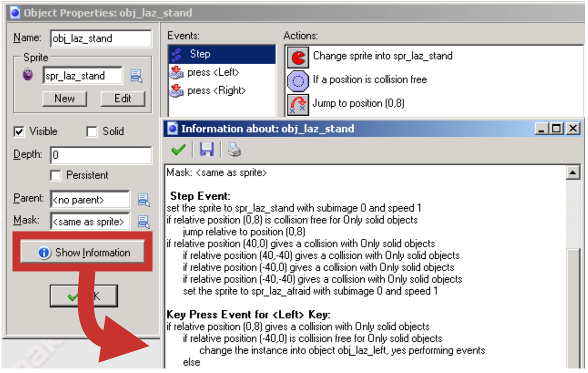
CS Bits & Bytes is a free fortnightly mailing that highlights innovative uses of computer science. Funded by the National Science Foundation in America, it was launched over a year ago as part of Computer Science Education Week. Each issue highlights a topic, written in an accessible fashion and provides suggested classroom activities. More details in the web supplement.

STARTING PROGRAMMING BY MAKING MORE OF GAMEMAKER

Many schools use GameMaker to give pupils experience of visual programming. Claire Johnson, from Thorpe St Andrew School, Norwich suggests it is also an ideal environment for introducing basic programming.

Here is a list of recommendations for those who would like to introduce pupils to basic programming concepts, but have little experience. Assuming one hour a week, it takes a term to understand the software and design, program and test a decent game. Limit time spent on creating graphics by using ready made sprites (which can be edited). Allow for a mix of exploratory learning and formal instruction too. Formal instruction is needed to deliver key game functionalities in context (e.g. the use of variables – score/health/lives; the use of conditionals – to get game objects to reappear when they go off screen; the use of the step event as a loop mechanism; how to make a game object follow the cursor; introducing the idea of a 'controller object' to control global game features etc). Pupils won't pick these things up without your input. They tend to eschew written tutorials/textbooks and prefer 'just in time' learning resources – so you will need to know how to do the things they most commonly want or need to do. Allowing pupils to work in pairs helps too.

To introduce pupils to basic programming concepts, start using correct terminology. *Game Maker* tends to hide this. The built-in variables for score, lives, health, speed etc. are not called 'variables'. The use of the 'test' or 'check' actions are really conditional statements. Likewise, the step event is often used to create a looping construct – but the word 'loop' does not exist in *Game Maker.* Use drag and drop icons to illustrate these first, then introduce *Game Maker's* textual programming language – GML – to do the same thing (using the execute code action) so pupils can experience a gradual transition to text. Use the 'show information' button (see picture) in the object properties box to introduce pupils to reading and writing *Game Maker's* pseudocode, and to help them in developing and checking the logic of their games.


Object Properties: obj_laz_stand
Sprite
New
Edit
Events:
Actions:
Show Information
OK
Information about: obj_laz_stand

Name: obj_laz_stand

Sprite: spr_laz_stand

☑ Visible ☐ Solid

Depth: 0

☐ Persistent

Parent: <no parent>

Mask: <same as sprite>

Events: Step; press <Left>; press <Right>

Actions: Change sprite into spr_laz_stand; If a position is collision free; Jump to position (0,8)

```
Mask: <same as sprite>

 Step Event:
set the sprite to spr_laz_stand with subimage 0 and speed 1
if relative position (0,8) is collision free for Only solid objects
     jump relative to position (0,8)
if relative position (40,0) gives a collision with Only solid objects
     if relative position (40,-40) gives a collision with Only solid objects
     if relative position (-40,0) gives a collision with Only solid objects
     if relative position (-40,-40) gives a collision with Only solid objects
     set the sprite to spr_laz_afraid with subimage 0 and speed 1

Key Press Event for <Left> Key:
if relative position (0,8) gives a collision with Only solid objects
     if relative position (-40,0) is collision free for Only solid objects
          change the instance into object obj_laz_left, yes performing events
     else
```

Insist on including the programming constructs you want understood. Specify that a game must feature at least one variable, conditional statement, loop etc. Get pupils to explain their functions. Although they will want to just 'get on with their game', don't let them! Incorporate a variety of structured activities – code walkthroughs, reading/debugging and scaffolded writing. Provide partially completed programs, or distribute ready made cards (like the *Scratch* cards). Modifying existing games or combining elements in a framework helps avoid incoherent or unworkable ideas. To get the most out of *Game Maker* you need to make sure time is focused on building ICT and computational thinking skills, rather than the myriad other elements which can swallow lesson time.

COMPUTATIONAL THOUGHT: THE LITERACY OF THE 21ST CENTURY

With the development of a new Programme of Study as part of the National Curriculum Review in England, we are at an exciting crossroads. There is a real opportunity to make computing and technology a key focus of our education system, argues Tom Crick.

If there's one lesson from the past 15 years it is that we must not focus on transient and superficial technology skills. Computer science is not programming (and vice versa) and we should be wary of teaching programming just for the sake of teaching programming, without thinking about why we want to get kids to program.

When Michael Bloomberg, Mayor of New York City, tweeted last January that he was going to learn how to program, critics responded by implying that programming was not for everyone. This is untrue. One of the reasons programming is increasingly perceived to be a 21st century literacy is because it is ultimately empowering, developing the ability to manipulate and control your digital world. But the key message is that learning to program is not the endpoint, but part of a journey to equip children with the necessary digital skills to solve problems. Our high-level aim should be to develop technology-independent skills and techniques, such as data literacy and computational thinking.

Computational thinking is a way of solving problems, designing systems and understanding human behaviour that draws on concepts fundamental to computer science. Computational thinking includes a range of mental tools that reflect the breadth of the field. Computational thinking means making use of different levels of abstraction, to understand and solve problems more effectively; it means thinking algorithmically and with the ability to apply mathematical concepts to develop more efficient, fair, and secure solutions; it means understanding the consequences of scale, not only for reasons of efficiency but also for economic and social reasons.

This is why computer science is important in schools: we need to embed principles and theory to develop a deeper conceptual understanding of how technology works and how it can be leveraged to solve problems.

A quote commonly misattributed to Edsger Dijkstra says: "*Computer science is no more about computers than astronomy is about telescopes.*" This is where computational thinking fits in, abstracting away the technology. There is an important balance to strike between focusing on developing practical programming skills (i.e. being able to write code for a specific task) and embedding a deeper understanding of languages and constructs: principles of programming. Technology changes quickly, so we must ensure that when "Technology X" appears, we have transferable knowledge and a deeper conceptual understanding of how it works and how it can be used.

There are still challenges ahead in changing the status quo and enthusing and engaging children in schools. Programming is a creative endeavour and offers a tangible way for children to express themselves by hacking, making and sharing. But we also have to recognise the importance of developing the deeper conceptual understanding as well as knowledge of the underpinning theoretical foundations. So let's change the focus from just writing code to developing the computational thinking skills. To quote Jeannette M. Wing, Professor of Computer Science at Carnegie Mellon University: "*Computational thinking is a fundamental skill for everyone, not just for computer scientists. To reading, writing and arithmetic, we should add computational thinking to every child's analytical ability.*"

SOME PRACTICAL WAYS TO LEARN ABOUT NETWORKS

Our first module with Year 7 is called Understanding Computers and aims to lift the lid on the technologies that we use every day. We want our students to be able to visualise how a network functions to show them the science behind the workstation in front of them. Practical examples are a great way of getting the whole class engaged so I came up with a kinaesthetic activity. Grabbing a stop watch, some pictures of network topologies and a piece of paper with the word "Animals.doc" written on it, I led the class outside to the playground. They would be acting as different network topologies to find out which are more efficient. I gave out roles; one student was a server, another a user accessing a file and the rest were workstations using their arms as network cables.

For a bus topology I put the server at one end of a line of 28 students and the user at the other. The user sent a request down the line to the server for Animal.doc. Stop watch at the ready, we timed how long it took. We repeated the process for a ring, star and mesh network. Each time, the process became faster as connectivity increased. After each we discussed the differences. We then went on to discuss costs and other factors that would affect a network administrator's decision when designing a network infrastructure.

My students now understand the different topologies and the advantages and limitations involved with each one. You can do a similar activity with servers, switches and packets. You can also make it more challenging by having a student act as a broken link. Kinaesthetic activities such as these are a fun, practical way of learning about networks that I hope others can share. *Rebecca Franks*

FREE CODECADEMY KITS FOR SCHOOL COMPUTING CLUBS

Unfortunately, many schools do not offer opportunities to learn computing. While organizations like CAS are working hard to change this, individual teachers have decided to take matters into their own hands and start after-school programming clubs at their schools.

After speaking with dozens of these teachers, Codecademy put together a kit to help others get started. It contains everything they need—a full curriculum, accounts for their students, flyers, letters to parents, and more. And because the lessons are self-paced, it works even if the club leader is not yet an expert programmer. There is nothing to download or install, and best of all, it's completely free. Check the web supplement to find out how to sign up for your After-School Programming kit. *Sasha Laundy*

Computing++

LINKING CODERS AND SCHOOLS

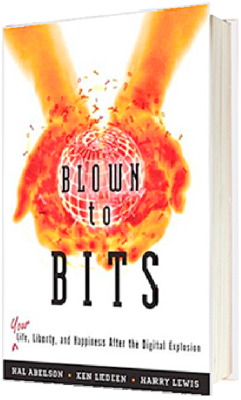
Are you worried about the changes facing ICT teachers? Do you feel ill equipped to embrace the teaching programming? Computing++ can help. Launched last year, the overriding goal is to facilitate computer science becoming a main stream subject in both primary and secondary schools.

The biggest obstacle to this is the gap in skills and confidence that exists in schools and so to enable teachers to deliver computer science lessons we are calling upon our coders to lend a hand and help teachers to learn to code. The hope is that beyond teaching a few skills a great deal more can be gained from the relationship between a school and an IT professional from giving a realistic perspective to pupils to the creation of engaging and relevant lesson content. So if you want help, or can offer it, Computing++ can put you in touch. Details in the supplement.

BEGINNING TO APPRECIATE THE BEAUTY AND JOY OF COMPUTING

How could anyone resist taking a course with a title like that? The Beauty and Joy of Computing is a course developed at the University of California, Berkeley designed for non-specialists in computer science.

This wonderful resource developed as an introduction for students from high school to undergraduate freshman year (roughly sixth form to first year at university). In the words of the developers: “Computing has changed the world in profound ways. ... The real transformative and empowering experience comes when one learns how to program the computer, to translate ideas into code. … But this course is far more than just learning to program. We focus on some of the "Big Ideas" of computing, such as abstraction, design, recursion, concurrency, simulations, and the limits of computation. We show some beautiful applications of computing that have changed the world, talk about the history of computing, and where it will go in the future.”


BLOWN to BITS
Your Life, Liberty, and Happiness After the Digital Explosion
HAL ABELSON • KEN LEDEEN • HARRY LEWIS

It consists of the course itself (in Moodle), a book, **Blown to Bits**, and a language, **Snap!**, the JavaScript implementation of **BYOB** (Build Your Own Blocks), Berkeley's own extension to **Scratch**. BYOB itself will do for the majority of the course. All are available online or free to download. A new installation of Moodle 1.9 can be set up, the course downloaded and restored into it, in under 20 minutes. The course links to all of the video lectures on YouTube, the chapters of the book, and other resources, as well as quizzes on the topics covered. By setting up your own local copy you can customise it to your own specifications. You will need **Scratch** for the first 4 sections of the course. The **Scratch** interface removes the depressing syntax problems that students encounter when starting to program, but it is limited. You can switch to **BYOB**/ **Snap!** when you start on functions.

The book ’Blown To Bits’ was reviewed in SwitchedOn, Issue 8. It's not a programming book, but is more concerned with the consequences of representing 'everything' as bits. The course introduces all of the 'big' topics in computer science: Functions, Lists, Algorithms, Algorithm complexity (why solving a problem twice as large can take 4 times as long), Concurrency (Scratch makes doing concurrency so easy), Recursion (solving a problem by solving simpler versions), Higher order functions (generalising functions so that they can handle other functions) and Distributed computing (how to make *n* computers work together efficiently)

I've just been through recursion with my 10 year-old grandson, building a block that draws a tree (draw a trunk, draw a smaller tree at the top leaning to the left, draw a smaller tree at the top leaning to the right). 'Cool!' is his considered opinion, and he's currently busy programming a block to draw a dog. There are similar courses that teach computing concepts, but BJC seems to me to be aiming at something different, targeting the course at non-specialists in computer science, but at the same time covering the biggest topics whilst exploring the social and ethical implications of 'blowing things to bits' in our digital world. *John Stout*

WHERE ARE THE GIRLS? TACKLING THE GENDER GAP IN COMPUTING

Did you know the term ‘computer’ was applied to the women who performed mathematical calculations by hand for male scientists during the Second World War? Reena Pau from geekEquality looks at the changes that have occurred since.

In secondary schools, the gender gap in computing began even before computers were introduced to classrooms. Prior to the Sex Discrimination Act of 1975, it was common for boys and girls to study separate subjects e.g. cooking for girls and woodwork for boys. Once the Act had been brought in all UK comprehensive pupils, regardless of gender, race, ability and social class, had access to the same educational opportunities (Deem 1981). During this period, girls were generally underachieving (Byrne 1978), did not have the opportunity to study ‘physical sciences’ at secondary school, and, given the choice, often chose to study biology or opt out of science altogether (Whyte 1986).

A survey conducted in 1973 by the Department of Education and Science (DES), found that only 17% of girls were offered the chance to take physics in years 10 and 11 compared with 90% of boys. Of the girls who were offered the chance to take physics, only 17% took this compared with 52% of the boys. It was clear that even with a choice of subjects, there was a distinction between the types of courses boys and girls preferred to take. Ormerod conducted a study into course preferences and choices and found that maths, physics, chemistry and geography were seen as male subjects, whilst biology, languages and art were seen as female subjects (Ormerod 1975). This distinction in subject choice was due to the way in which physical sciences were taught in mixed-sex schools (Whyte 1986).

During the 1970’s and ‘80s, there was increased in funding to get computers into schools (Fothergill 1981). Maths teachers and computer enthusiasts organised their use and experimented with either the Apricot or the Commodore. These were not for mainstream use by pupils, however, as they did not have graphics or sound cards and were only suitable for word processing and spreadsheets (Forester 1985). Teachers often selected students to be part of computer clubs to learn more. These were most often boys, who were enthused by the computers (Carter and Jenkins 2001).

We have made progress in terms of technology and some aspects of inclusivity. But without the dates of the surveys and research mentioned above (references in the supplement), it would be easy to assume that this article has been talking about recent issues. How much has actually changed? Curricula can be altered but attitudes are harder to change. This is a long standing issue we need to tackle together. geekEquality is working with the University of Southampton to make computer science inclusive, regardless of gender, race or age. We want everyone to be involved in the technical revolution and to take a part in writing history.

International Journal of GENDER SCIENCE AND TECHNOLOGY

Formed in 2009, GST is a peer reviewed journal focusing on gender issues in and of science and technology, including computing. Published three times a year, it allows those outside academic institutions to have access to research data that can help inform strategies. There are often interesting articles about women and IT. Link in the supplement. *Kate Sim*

GIRLS INTO COMPUTING: YEAR EIGHT IS TOO LATE

Women have played a significant role within the development of the computing industry and more women than ever before can be found in a typical IT office. Nonetheless, there is still a clear imbalance in this particular trade. Women are recognized all over the world for their work in various fields so why is it so unusual for females to play their part within Information Technology and Computing? So how can we solve this puzzle?

Fortunately, women are not born with instinctive prejudice against Computing. Their views develop from surrounding influences ranging from parental influences to the media. The factors are too numerous to battle one by one, but we can offer alternative information and encourage women to play an important part in the field.

Are we doing enough? By Secondary School, pupils already have a vague idea of what their future holds employment wise. A study I conducted with a group of students in years 5 and 6, identified some interesting factors. “92% of the group admitted they enjoyed ICT in school and 86% of the case studies agreed that they would still continue studying ICT if it was an optional subject, but only 3 of the students commented they would like to do an ICT related job in the future” - so are we missing a trick here? It is clear that even at Primary level, pupils are already identifying ICT/Computing as “subjects for boys”. Thankfully, several high profile initiatives are challenging this view. You’ll find some of them highlighted in the next few pages. They can provide resources and inspiration so we can all work to introduce the wonders of Computing to female students at these influential young ages. *Sophie Dare-Edwards*

#include

COMPUTER SCIENCE FOR ALL

We have recently seen a small but welcome rise in students opting to study Computing. However, of the 3809 candidates who sat Computing A-Level last summer only 297 were female, just under 8%. CAS members are clearly concerned by the lack of diversity in the students we attract and want to do something about it, but most of us have no idea what we can actually do to improve the situation.

Emma Mulqueeny, founder of Young Rewired State, is often quoted as saying "Year 8 is too late" to enthuse girls before they form associations about the subject. It is clear that the problem is one we need to address at school as early as possible. As a community of educators we cannot afford to ignore our privileged position. CAS has formed a working group, **#include to** coordinate efforts and allow members to share ideas. We have a forum on the CAS community (look for "Girls in Computing") where we welcome ideas and questions, as well as a Twitter account **@casinclude** for frequent updates. Our website includes a Wiki where you can find useful reading material. We have been overwhelmed with support from members but we do need individuals who would be willing to use some of their time as working members of the group, particularly those working in the primary sector.

No one has the "silver bullet" solution to this problem; we can only challenge what we see through lot of people making a little effort wherever they can. Already, individuals are initiating activity. In the summer term **#include** will be hosting a "hack day" targeted at girls aged 11-13. We hope to provide a variety of workshops, talks and hands on experiences to give them a positive experience of Computer Science and to challenge the stereotypes associated with the subject. If you have ideas, contributions, offers of help or suggestions please do get in touch either on the forum or via Twitter. *Laura Dixon*

VIDEO STORIES THAT WILL INSPIRE GIRLS

What was it about science, technology and computer science that excited me, and how could I get that across? Carrie Anne Philbin's answer to her question was an inspirational video channel, Geek Gurl Diaries.

Women represent only 28% of all IT jobs. I noticed that uptake of ICT and Computer Science by teenage girls at KS4, KS5 was equally limited and I started to think about what I could do to improve the status quo. I knew that I wanted to develop a network of individuals who could, through their own experiences, inspire a generation to take up more traditionally geeky subjects. The theory was that by exposing students to the range of creative and exciting scientific careers in technology, they may discover an interest in a field they had previously dismissed. As an enthusiastic teacher of ICT, I set out to provide teenagers with an opportunity that I never had: access to people working in the tech industry, and resources demonstrating how to create, build, make or do different tasks using tech skills.

Working with young people, I know how powerful YouTube can be in grabbing their attention, entertaining and informing them. I make videos for lessons and put them on YouTube so students can access them outside school so it seemed like a great place to start. I created a channel for all my ideas and The Geek Gurl Diaries were born. They are a collection of video blogs and interviews with inspirational women in the fields of computing, science, technology and engineering. They include contributions from women working in IT and Science. Recently the channel has started to include 'Geek Gurl Diaries On Air'; panel discussions with graduate computer science students on various topics like computer gaming and geek culture. I have done very little marketing, but found organic growth through the teenage girls I'm targetting "liking" the Facebook page where I publicise the video episodes. The Geek Gurl Diaries is a not-for-profit project shortlisted for a Digital Heroes Award by Talk Talk, and backed by Computing At School. Most recently, through the crowd funding platform Sponsorcraft I have raised £550 for equipment and software as well as making some friends along the way who are willing to help.

I hope this will inspire teenage girls to have an interest in subject areas they may have dismissed before. To succeed I require inspiring individuals, both male and female, to contribute further ideas, videos and interviews. I also need teachers to publicise the channel and talk to students to change the stereotypes. Please visit the site to check out what I've been up to. The link is in the supplement.

BRIDGING THE GAP BETWEEN GIRLS & TECHNOLOGY

Lady Geek is a campaigning agency that aims to change the way tech companies speak to women and ending the '*pink it & shrink it'* approach. CEO, Belinda Parmar wrote 'Little Miss Geek', outlining the problems and some ways to solve it. "The aim is to do what Jamie Oliver did for school dinners: to inspire the next generation of young girls to become tech pioneers." Links to her excellent blog and book in the supplement.

TECHNOLOGY TO WEAR: CREATIVE CODING WITH LILYPAD ARDUINO

The LilyPad Arduino brings computing to a wide audience in an interdisciplinary context. It appeals to students from computing to fashion and performing arts. Sophie McDonald outlines the workshop she ran for a TechnoCamps BootCamp in Wales.

The LilyPad is wearable technology, designed to be embedded into textiles and fabrics. It has petal-like holes around its edge for connecting components using conductive thread in the place of wire. It uses the same chip as other Arduinos and is programmed using the same free development environment. Software and hardware are opensource. A perfect educational tool, encompassing design, craft, electronics and computer science.

Last April myself and Rain Ashford, with help from Hannah Dee, delivered a three-day 'wearable technology' workshop in Aberystwyth. Of our class of 11 – 16 year olds, about 80% were girls. Running at the same time was a workshop building Arduino based land yachts. It had similar numbers, but the gender ratio was reversed. Whilst unsurprising, it is important to note. It suggests that delivering technology workshops in an audience relevant context can garner interest in computer sciences and technology generally.

Day One introduced wearable technology, programming in C, LEDs, resistors, Ohm's Law, breadboards, light dependent resistors and pressure sensors. LDRs and pressure sensors change resistance based on light levels or pressure. This can be measured using the LilyPad, and used to set conditions for events. This enabled us to introduced variables and conditionals, and we found that most of the students had no problem grasping these concepts. Participants spent time over the next two days designing garments to include the components and sensors. Pupils came up with a huge range of ideas - cats with flashing eyes, floral designs, hearts on handbags, and high-tech sleeves. Each needed to draw the circuit onto their design, check they'd avoided any short circuits, and finally sew in their components. Once the sewing was done, the program could be loaded on the LilyPad to complete the piece. We now have a one-day workshop (using pre-sewed LilyPad patterns).

Soft circuits and e-textiles have become possible with the invention of conductive threads, fabrics, and surface mount electronic components. E-textiles is embedded in the maker-hacker cultures that have emerged for artists, designers and hobbyists, and you can find a wealth of material and tutorials on-line. In particular, the High-Low Tech research group at MIT, where the LilyPad Arduino was developed, has done extensive research into embedding electronics and computers into textiles, and has great resources that are easy and fun to make. Learning computational skills through creative projects provides real motivation for children and adults. In this respect, computing becomes an artistic tool in itself.

GET STARTED WITH THE HELP OF OPEN SOFTWEAR

"Once you know a couple of basic programming commands and how to connect something simple like a LED, then you can start building. Knowledge ... comes from doing, not reading." So writes co-author Tony Olssen, from the University of Malmo, Sweden of this excellent introduction to using Arduino in wearable projects. The book assumes no prior knowledge and guides the reader through the basics needed to get going. Link in the supplement.

OPEN SOFTWEAR
Fashionable prototyping and wearable computing using the Arduino

GIRLS SOFTWEAR CLUB REVEALS PEER PRESSURE

At Queen Elizabeth School, Cumbria some 25 year 9 girls attend the SoftWear Club on Wednesday and Thursday lunchtimes. Prior to launch, I targeted girls I thought might be interested (given what I knew about them) but I also asked colleagues to suggest girls that might be keen. They were all sent an email inviting them to sign up for the club one break. Girls were asked to bring along any others they thought might be interested.

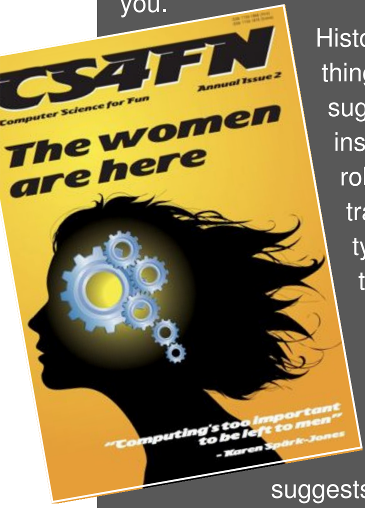
We started by looking at how to create a circuit and run the basic 'blink' sketch. We then looked at creating a sequence of lights and using tri-colour LEDs. As a group we agreed to create a mural of the sea. We worked with an Art teacher who helped us get inspired and came up with the idea to use tie-dye for fabric for the sea items. Having established the basic idea they are now trying to figure out how to get their sea grass and the arm of the Mermaid to move and how to get the yellow submarine to play a song. There will, no doubt, be several LED blinking as part of the final seascape.

One girl mailed a fortnight after starting, asking to join. When I suggested bringing a friend she revealed she had kept it quiet as they all thought it was too geeky. She even said she told her fiends a white lie about what she was doing; that she was really just asking me for some homework help. Tackling the attitudes of friends and challenging peer pressure will be the key to attracting a new generation of girls. Making tech cool for girls is a priority. Get it right and we can re-ignite many girls interest in IT and Computing.

Annelie Chambers

CELEBRATING FEMALE ROLE MODELS PAST AND PRESENT

Being able cite female role models making decisive contributions to computing is an important weapon in combating negative stereotypes. There can be no better place to start than with Ada Lovelace. She saw that if the Charles Babbage's Analytical Engine could be programmed to calculate, it could pretty much be programmed to do anything. And thus, she gave us the blueprints for computer programming as we know it. A short video, produced as part of the BCS Pioneers project, is a good starting point for pupils. Some may well know of Ada Lovelace, but how many know about Hedy Lamarr? Check out her story on the BCS pioneers website too. Ada Lovelace Day will be celebrated on 15 October. FindingAda, the supporting website has links to material and stories to inspire you.



Historical icons are one thing, but some research suggests students are inspired by present day role models who contravene the stereotypes associated with the field. A recent Google video, 'Girls in a Tech World' does just that.

Research also suggests that girls are more inspired by role models who are 'within reach'. Elite women can appear too remote for their achievements to be accomplishable. In this respect, "The Women Are Here", a special issue produced by CS4FN is required reading for all teachers. It includes a huge variety of stories about women in Computing, ranging from early pioneers such as Grace Hopper to students and researchers in higher education today. These latter stories in particular provide a range of insights into what makes computer science attractive to women. The pdf can be downloaded (or the articles accessed from their website). This is a book to be read, and used selectively—the articles are short and would make excellent case studies to drip feed into the curriculum or display in your room. All links in the supplement. *Roger Davies*

ENCOURAGING MORE GIRLS TO STUDY COMPUTING GCSE

What inspires girls to opt for studying Computing? Chris Swan, Head of Faculty at The Stourport High School and Sixth Form Centre, introduced Computing this year and talked to several of her students to find out what they enjoy.

My previous role was Head of Computing at a large FE/HE college and I joined our school in September 2011, introducing both GCSE and A level Computing this year. Although I have taught Computer Science for a number of years, I have been disappointed by the low numbers of female students. I have had the pleasure of teaching some brilliant female programmers and network engineers but unfortunately, of the hundreds of students that I have taught, females account for a very small number. This year, we recruited six girls to our Year 9 GCSE class. All were very effervescent and keen to learn. We've chatted about gender stereotyping. Although they are aware of it, they are keen to challenge it and are equally evangelical. Given the subject was not previously taught, I asked them why they had chosen to study Computing.

Laura told me she loves the challenge of learning new things and hopes it will help her get a good job in the future. She loves the Python programming that we have learned and even cited binary logic as being one of her current favourite topics. Eleanor felt that learning programming had helped her think more logically and developed her maths skills. Abi would like to be an app developer. Molli was enjoying learning new things such as programming, writing programs for the Little Man Computer simulator and using a Raspberry Pi. Ellie has also enjoyed learning Python and would like to become a programmer. Without exception, all of my girls recognised that GCSE Computing has real value in the outside world, was worth studying and was a lot of fun.

Students Molli Saull, Laura Winterburn, Eleanor Johnson, Abi Edwards-Kelly and Hannah Boit with Chris Swan

We also discussed stereotyping. They all agreed that there need to be more female developers if we are ever going to have their interests truly represented. I am planning to throw a project into the year in the summer term as a post-examination season treat. My personal interest in fashion inspired me to investigate simple wearable computers, such as the LilyPad Arduino, and how surface pattern could be designed using algorithms. The girls loved this idea. I saw the "spark" and enthusiasm that I had hoped to inspire. Of my six female students nearly all are hoping to pursue careers in Computer Science. This is music to my ears! The strength of positive female role models cannot be underestimated. In 2000 I completed my CCNA and Wireless LAN Specialist certificate. At that time, I was one of a small number of female educationalists in the field. I don't feel at all "special". Perhaps the biggest asset is my drive and determination. This is something that we strive to inspire in our students on a daily basis. Equality is the watchword. I don't think we're quite there yet but I'm more hopeful than I've ever been.

SPREADING THE WORD: W-TECH CYMRU MOTIVATES NEW INITIATIVE

It is imperative that the ICT industry becomes more inclusive to women. It is also important to make women more aware of the inspiring, challenging, wide variety of careers the sector can offer. Beti Williams OBE reports on a new initiative in Wales.

There have been many projects and events set up to address gender imbalance in IT. One such event, in 2009 was W-Tech, a career development and networking event for women interested in IT. Organised in London by BCS, The Chartered Institute for IT and Women in Technology, some 700 technical women participated.

Motivated by W-Tech, and knowing many women in Wales could benefit but were excluded through distance or cost, I got the agreement of the W-Tech organisers, in my role then as Director of ITWales, to test the concept in Wales. W-Tech Cymru was launched on 6 November 2010 at the Lampeter Campus of the University of Wales Trinity St David. At the outset, it was decided that the event would be organised not only to attract women but also, unlike the original W-Tech, to reach out to female students and schoolgirls. The presentations and workshops would not be gender specific, or 'girly', but technically solid and generically useful. It was also agreed that a female only audience may generate a more comfortable and non threatening atmosphere making the experience more enjoyable.

Presentations by academics and professionals included:- 'The use of IT in the Blood Service'; 'Computer Vision, The Next Big Brother'; 'Taking your Laptop to the South Pole'; 'The thrill of Data Mining', 'Adaptive Rollercoasters through Emotion, Detection and Affective Computing'. The workshops ranged from topics such as 'Agile Development' and 'Intelligent Open Source Research' to 'Be the Boss you would like to have'. Some 150 delegates attended the event, a huge success, particularly given the remote location of the Lampeter campus.

The success of WTech Cymru has motivated a very exciting new venture for 2013. BCS Women in Wales and ITWales will team up to organise W-Tech inspired workshops for female students and school girls throughout 2013. Keeping a similar format as W-Tech Cymru the workshops will be organised by Technocamps (a project within the ITWales portfolio) either in schools, colleges or in the Technocamps hubs at the Universities of Swansea, Aberystwyth, Bangor and Glamorgan. This collaboration will take the message to girls across Wales, especially those living in more remote rural areas. Further collaboration between the BCS Women in Wales, Technocamps and Software Alliance Wales (another project in the ITWales portfolio) will result in a celebration of International Women's Day (15 March) in each of the four venues. The events will consist of presentations from distinguished women from the world of computing with down to earth stories of the highs and lows of their career. ITWales has successfully organised a celebration of International Women's day in Swansea every year since 2000 motivated by the necessity to encourage more women to take up careers in computing and ICT.

International Women's Day celebrations of Women in Technology, followed by W-Tech workshops for students and girls across Wales, are sure to send a very strong and powerful message to encourage more women and girls to consider taking up careers in computing and ICT. These careers involve working with up to date technology or actually developing the technology of the future and can also provide a good work/life balance with opportunities for working from home and good remuneration.

CAS WALES REFLECTS ON EXCITING TIMES AHEAD

2012 has been a good year for computer science in Wales: six CAS Hubs supporting teachers, a second annual conference, huge amounts of activity and impact in schools from the Technocamps partners, as well as clear pronouncements on the importance of computer science education from Leighton Andrews AM, the Welsh Government's Minister for Education and Skills. We have also seen significant investment by the Government on digital technologies to underpin learning and teaching including the appointment of eight Digital Leaders to support schools, overseen by the National Digital Learning Council.

Nevertheless, there is much to do to continue the momentum; the next focus for the NDLC is funding CPD for computer science, ICT and digital literacy. Upskilling teachers to teach computer science has to be a priority and a model for this already exists: the Network of Computer Science Teaching Excellence. If funding is made available, we would be able to create a Welsh NoE and work with universities in Wales to provide support, resources and CPD. There is also significant activity on the policy front: the Welsh Government published its review of 14 -19 qualifications and has called for a review of ICT. The Minister is currently forming a steering group to review ICT in early 2013; more information to follow shortly. For more updates on CAS Wales activities, please see my blog. Link in the supplement. *Tom Crick*

CODECADEMY CYMRU

Codecademy is partnering with CAS Wales to build a bespoke environment to help teach programming in Wales. If you'd like to join the trial phase please complete an expression of interest on the above blog.

SPOILT FOR CHOICE: SO MANY SESSIONS AND SO LITTLE TIME

It's impossible to do justice to such a packed day, but to get a flavour of the event, here is a summary of some of the highlights. Hackasaurus lets you change existing web pages, effectively demonstrated by Charlie Love who changed Microsoft's website into a Google site! We then saw how to teach CSS positioning by using a zombie fighting project in Mozilla Thimble. Chris Martin from Dundee University kicked off the Arduino session with a bang using a party popper popping machine! What followed was a hands-on session where we were able to explore possible classroom uses. Meanwhile, Claire Griffiths from Moray Council provided an introductory talk about the gender gap. A good discussion followed suggesting reasons for under-representation including differing learning styles and computer usage. Possible solutions included promoting strong role models, using paired learning, group projects and innovative coding lessons to maintain girls interest.

In the afternoon Duncan Smeed discussed the work he has been doing with the Raspberry Pi. Kate Farrell and Tom Hendry provided an excellent range of interdisciplinary projects included using Scratch to create German Language quizzes and Lego We Do components for control boxes in CDT. We were amazed, entertained and illuminated by one of the world's foremost experts on educational programming environments, Michael Kölling, on a magical whistle stop tour of the Greenfoot programming environment. Steven Whyte gave delegates an insight into the Livecode programming environment and Sue Sentence demonstrated .Net Gadgeteer. Doug Belshaw from the Mozilla Foundation explained how Open Badges can be used to track achievement both in class and in extra-curricular settings. Jeremy Scott talked about his support material to teach the principles of computational thinking through different environments. The packs cover: introductory and intermediate programming in Scratch and mobile app development in App Inventor. And so the list goes on. Delegates were spoilt for choice and no-one could see everything available. Roll on #CASScot13 !

FIRST CONFERENCE 'SELL OUT' SUCCESS FOR CAS SCOTLAND

CAS Scotland held their first annual conference in Edinburgh last October. Mark Tennant and Kate Farrell summarise a packed day of cutting edge talks and training sessions for CS teachers from across the country.

Exploring Arduino boards at the CAS Scotland Conference with Chris Martin

The event opened with a keynote from Muffy Calder, Chief Scientific Advisor to the Scottish Government, focusing on 'computational thinking' skills in all walks of life, not just Computing Science. She defined Computational Thinking as thinking "precisely and unambiguously about data and computation". Quintin Cutts used Carol Dweck's work around mindsets as a springboard into thinking about the pedagogy around apprenticeship models of coding. Using an analogy of becoming an apprentice tailor he asked where "the button-holes, the cutting, the hemming of programming" are, pointing out that minimal guidance doesn't work. Pupils need to be equipped through worked examples and peer instruction before being asked to write whole programs. Quintin discussed breaking programming tasks into small chunks to assess learning, stressing the need for exams to have actual programming tasks. He suggested you can assess best understanding by using questions that ask the learner to move between English, pseudocode and a programming language.

Following the keynotes, some 16 workshops, seminars, talks and forum sessions were on offer in breakout rooms (see left). Delegates also heard Peter Dickman from Google speak about the range of industry-sponsored materials available for schools, and Steven Grier (Microsoft Education) gave his insight into new tools and technologies for the Computing classroom. The conference finished with 'The Conjurers' Classroom' by magician Jody Greig which demonstrated tricks to engage students, and explained how magic is a form of computational thinking!

The response from the delegates was overwhelming. When asked if the conference would affect their practice in school it looked like every hand shot up. That the room was still packed at the end of a long day was a testament to how much people valued it. Thanks are due to the many people who helped: Microsoft for letting us take over their Edinburgh offices; Oracle for providing wine; the enthusiastic helpers from Knox Academy in East Lothian, who filmed sessions, registered delegates, set up rooms and tidied up; and of course the speakers who gave up their Saturday to share their knowledge and enthusiasm! Last, but not least, it's worth remembering that the conference was organised by just 6 volunteers, all of whom have day jobs as Computing teachers: Peter Donaldson, Kate Farrell, Claire Griffiths, Charlie Love, Andy McSwan and Mark Tennant. We were all delighted by the success of the day, encouraged by the interest in CAS and impressed by the willingness to go away and get involved with organising hubs and future events.

MORE POSITIVE DEVELOPMENTS FOR CAS IN NORTHERN IRELAND

The landscape in Northern Ireland continues to move in a positive direction. Clarke Rice reports on continued collaboration from examination boards and managed service providers are helping facilitate change, which remains on the political agenda.

CAS Chair, Clarke Rice (second left) together with some of the teachers who attended the recent Greenfoot CPD workshop

Since June's Hub meeting, the provider of all our schools managed networks (C2k) have continued discussion with CAS and with CCEA, to facilitate a range of software development tools that support Computing qualifications regardless of course provider. The entire C2k infrastructure is due for a major overhaul in 2013 and the willingness to engage with teachers to support Computer Science in schools is a pleasing development.

After an enthusiastic presentation at the June meeting, Ian Simons (along with Will Moore), continue to enthuse the next generation of programmers with their 'Go Berserk' book on HTML and CSS. More books in this series are planned. It is encouraging to see local teachers getting involved in raising the profile of Computing through this initiative, thanks to sponsorship from local industry and a launch in Stormont. The need to increase the profile of Computing in NI schools continues to be discussed in the Education Committee of the NI Assembly.

CCEA's help in distributing copies of SWITCHEDON to teachers of GCSE ICT is very much appreciated. CCEA are encouraging programming-type solutions to Controlled Assessment games, using Greenfoot, Scratch, Kodu, Gamemaker and anything else reasonable.

In December, Loreto College hosted another course for teachers, facilitated by CAS since the first was oversubscribed. Teachers travelled from local schools and as far away as Co. Kildare (near Dublin!), the turnout being testament to a desire to move from a 'traditional' model of ICT towards the rigorous problem-solving techniques Computer Science develops. Various examples for teaching programming were discussed. The course used Greenfoot, an environment for teaching Java through games development. Using the friendly interface they were quickly writing code to power a game. Whilst creating virtual crabs that eat worms and are chased by scurrying lobsters might first seem like an unusual teaching method, it develops fantastic analytical skills in students. Since introducing Greenfoot in Loreto, I have noticed significantly increased student motivation. Local employers and universities have also commented on the significant difference they see in students with well-developed programming skills. Thanks are due to Neil Brown at the University of Kent for providing materials.

We hope to continue to raise the profile of CAS in Northern Ireland. All local teachers are encouraged to contribute to the NI section of the CAS forums and to make their own suggestions for training.

LATEST 'MUST READ' ISSUE OF CS4FN OUT

Schools should have received the current issue of the Computer Science For Fun magazine (cs4fn) last term. Its focus is an entire field of computer science dedicated to improving the experience of using a computer. Human computer interaction combines technical know-how with psychology, inventiveness and artistry. In the current issue we're exploring how designers make computers that feel good to use. You'll read about eating cookies in virtual reality, controlling computers by harnessing invisible airborne electricity and falling in love over video chat. Dive in!

The next free issue will be out in the spring, all about the wonders of words. Language is one of the most complicated of inventions, and it's a pretty meaty problem for computer scientists. How can computers understand, or speak human languages?

Whilst you are waiting, have you seen the cs4fn sister magazine Electronic Engineering for Fun? It's free too, and all about electronics and physical computing. In our current issue you'll find stories about implantable mobile phones, robot sign language, hacked tablecloths, and what happens when the internet is slow. You'll find links to all these and our website with more articles and activities in the web supplement. *Jonathan Black*

SECONDARY BURSARY AWARD SUPPORTS PRIMARY TRAINING

Having won a bursary from BCS, The Chartered Institute for IT, Kesgrave High School ran a training day for local primary schools. A brief recap of computing, from the lovely Lovelace herself up to the present day, was followed by exercises from the brilliant *Computer Science Unplugged*. Apart from being lots of fun, we wanted to show you don't need a computer suite full of quad core PCs to do computing. In fact you don't even need a computer. It's always good for teachers to be prodded out of their comfort zone and getting them flapping about at the front of the class as a human binary counter did just that.

Then on to Scratch: the more I use this, the more I'm convinced that it's the best tool for teaching and learning programming ever made. It was really satisfying to see people who had never programmed before realise that it wasn't as hard as they'd thought—that they could actually make things! Everyone loves physical computing and so it was PicoBoards next. These are an outstanding teaching tool when paired with Scratch and a team competition resulted in musical bananas, painting with light and Jelly Babies that screamed when you squeezed them. We rounded off with a quick look at resources and pedagogy.

When planning the day, as well as building confidence, we wanted to show that Key Stages 1 and 2 pupils are quite capable of learning computing; that non-specialist teachers can teach it capably; and that continued, long-term support is available from us. On this basis—and from the feedback we got—the day was a great success and we already plan to do more.

We also learned it can be surprisingly hard to reach some people: at times I felt like I was cold selling double glazing! I was told by one head teacher, "No thanks, we already have someone here who does ICT", which shows how far we still have to go with this Computing in schools lark. Surrounded by the energy and expertise of fellow CAS members it's easy to forget that the case for computing still has to be made in order to reach all schools. It was a useful lesson. *Clive Beale*

TOMORROWS GENERATION INSPIRED BY TODAYS CREATORS

Flushed with the success of the first Hack To The Future event, Alan O'Donohoe organised a follow up for all Year 10 pupils at Our Lady's High School in Preston. With two weeks to go and only one volunteer helper, maybe it was time to panic?

Last February we opened our school to 365 children, parents, teachers, developers, computer scientists and more for the first Hack To The Future event. The aim was to use the digital creators of today to inspire the digital creators of tomorrow. Just 8 months later, we did it all over again with an additional challenge—just 3 weeks to organise it! Our school has 5 Extended Learning Days when the timetable is suspended to allow for activities not practical in a normal school day. The ICT team hosted 180 Year 10 pupils for the day, to inspire them to see computing and digital creation as exciting and viable options for further study or career choice.

The Guardian Teacher Network offered to help promote the event but after a week I had only attracted one volunteer. This was far short of the minimum 25 I neeeded. Then, with around 10 days to go, we received an avalanche of volunteers. It started with Carl from Edge Hill University asking if he could bring 25 student teachers. After that, it just went crazy, so much so that on the day we had 60 adult volunteers in total. While this sudden surge in interest was fantastic, it brought additional challenges; parking, catering and child protection.

Constructing networks with student teachers from Edge Hill

With a few days to go I recorded a quirky message on YouTube with information for pupils. In all, our volunteers offered seven different workshops all lasting 1 hour, pupils experiencing 5 each. We kept children in registration groups moving them to designated rooms at changeover. While there was an initial panic about the lack of interest from volunteers which then turned into anxieties about the volume of offers, I would certainly consider organising more of these events in the future and I would encourage other teachers and schools to consider it. It creates many opportunities for pupils that are not normally available. I was touched that many of the pupils made a point of locating me at the end of the day to thank me for organising it. Some students were anxiously trying to secure work experience placements with the geeks they met. There is no way I would recommend the short timeframe—I'd even consider 6 weeks as too short.

The geeks and volunteers really make this whole event and in order to secure their support, you need to be publicising the event well in advance so that people can organise their diaries and request leave if necessary. Weekends have proven to be easier to gain wider community support. Another factor ensuring the smooth running of an event is to keep in contact with all involved, with regular updates so all know what to expect from the moment they arrive.

GETTING TO GRIPS WITH YOUR PI USING SOME RASPBERRY FILLING

It's been a long time in production but the Raspberry Pi User Manual for Education is now available. Editor and contributor, Andrew Hague urges you to give it a try and pass it onto your friends.

To download the manual, use Google to search for Raspberry Filling, look on the Raspberry Pi download page, or the CAS Community. The chapters are written by educationalists and programmers and aims to guide readers both young and old through some key software to support learning programming; which can then lead onto an understanding of computer science. There are chapters providing examples in Scratch, Greenfoot, Python, Interfacing with the Real World, Geogebra and Bash (a Linux command line interpreter). Our goal was for the computing student to get a Raspberry Pi, TV, keyboard, mouse, cables, and an SD card with the latest Raspberry Pi OS on it, and just sit in front of the TV typing in code and examples from the manual. No internet required.

I wrote the chapter on Python as an introduction to converting computational ideas into actual programs. I used PyGame to access to the visual and audio support in Linux. Even though Python is great for Linux on the Raspberry Pi, both it and PyGame are available on Windows and MacOS so most of the lessons learned can be transferred to school computers. While researching, I found many resources relied on understanding syntax or using modules, detracting from relating code to what it is doing. My aim was to help students begin to realise a computer is a machine that simply manipulates data. In the Raspberry Filling, these foundation concepts are covered earlier with Scratch and Greenfoot. The Python activities are 'type-in and run' experiments with some discussion on how they worked: much like a chemistry or electronics set. Just follow the pictures and see what happens. Students learn by doing, and by typing in code they will ask questions. "Why does the code have to line up in columns?" and "What happens to variables in a function when it is finished with them?" Some early experiments do not loop or have event handlers. Windowing systems and games require these to separate rendering the pictures and detecting input from the processing. They add complexity and give a false impression of a program. Instead, plenty of the experiments take some input, process it, generate some output and then terminate. This might seem trivial but it demonstrates that the computer just follows instructions.

The following chapter discusses how to wire up your Pi to the outside world and control it through Python. This is extremely useful for robot motor controls, traffic lights and so on. With a few cheap components the students could work at home with a parent or friends to make something sparkling. So where now? The manual is finished but it isn't complete. With a creative commons license we are indicating we are happy for it to grow and develop. We hope it will help get children (and teachers) taking their first steps into computing.

One year on from the general launch, the Raspberry Jamboree on 9 March, Manchester will look at how it is changing computing education. It hopes to share what a good Raspberry Jam *tastes* like, the most successful *recipes* and best *ingredients* so you can hold a wonderful Jam in your area. See the CAS Community Events section for further details and ticket information.

MANCHESTER HUB HOSTS INTRODUCTION TO PI-FACE

In collaboration with the University of Manchester and the Museum of Science and Industry (MOSI) two 'Introduction to Raspberry Pi and PiFace' workshops were held in October and November. PiFace is an interface board for the Raspberry Pi that makes it really easy to connect the Raspberry Pi to interact with the outside world. It has eight outputs that can control things like lights and motors and eight inputs for switches and sensors. It offers a degree of protection for the sensitive Raspberry Pi just in case you get your wires crossed! The board was developed by Andrew Robinson at the University.

The events attracted over 80 teachers. They spent the first half of the session setting up their Pi boards, doing some simple control with Scratch and creating a reaction timer game using Python. Some got really creative and whipped up a game in Scratch that used the PiFace board as a control pad. The second half let them play with more elaborate activities including a bird box with an infrared light beam, so the RPi can detect when a bird is coming in or out and can activate a camera or send a message, a controllable scalextric and a twitter tweeting robotic chicken! Pete Lomas (Co-founder of the Raspberry Pi Foundation) made a special appearance to show his support and a number of STEM ambassadors from a range of companies also attended. The Greater Manchester STEM team are following up by partnering schools with STEM ambassadors to enable collaboration. We are looking forward to bringing everyone back together again in the spring to share their experiences and discuss how we can develop more, better educational materials around the Raspberry Pi. *Mandi Banks*

TEEN TECH EVENTS INSPIRE TOMORROW'S INNOVATORS

In October, Martin Saunders – industry coordinator and maintainer of the Computing++ website - represented CAS at Teen Tech in Coventry. Pupils from years 7-9 of a large number of local Secondary Schools had come to learn about careers in technology and engineering, including computing and computer science. Also exhibiting were the Raspberry Pi Foundation and Active Robots, showing the NAO robot from Aldebaran Robotics. Groups of 5-6 pupils with a teacher rotated around the exhibit area giving each a ten minute window in which to learn about each topic. At the CAS stand a demonstration of Scratch interfacing a Microsoft Kinect controller introduced the fun of computing.

One group of girls initially expressed disinterest in computing as 'the games are all for boys'. But, as soon as they realised that they could learn to make games for girls they very quickly changed their view and were suddenly excited by the possibilities. There was also a clear difference in the confidence level of pupils of schools that teach computing versus those that don't. Given that computers are so prevalent in the world now, this confidence alone stands those pupils in good stead. The hope is that the schools that don't yet teach computing took something away and will start to introduce their pupils to the subject soon. *Martin Saunders*

340 AND GROWING!

Code Club is a nationwide network of after-school coding clubs for children aged 9-11. It's a volunteer led organisation launched in April this year and after a pilot scheme in the last half of the summer term they launched their first term this September. Code Club is now running in 340 schools across the country. The volunteers are developers from the tech industry, they bring with them projects that Code Club have written. Each project lasts an hour and the students will make a game or toy in each session. If your school would like to get involved and host a Code Club please visit www.codeclub.org.uk to find out how to get started. *Clare Sutcliffe*

TEACHERS MADE WELCOME AT PYCON UK COMMUNITY MEET

PyconUK is a community organised conference focussed on the Python programming language for developers and other interested parties. Former teacher Nicholas Tollervey, both an attendee and organiser, shares his views.

Given the interest in RaspberryPi and programming in schools, and as I used to be a teacher I stepped up to bring my colleagues past and present together in an "education" track at PyconUK. The resurgence of interest has caused a lot of people to look to Python. At PyconUK we simply wanted to find out how best to meet the needs of teachers and show them what it's like to be a part of a free software community.

John Pinner and I ran an "Introducing Python" workshop and there were talks on several education related projects during the conference. Sunday morning was the highlight of the weekend as teachers and developers rolled up their sleeves and took part in an "education sprint" (an intense collaboration on a specific project). We were very lucky to have Alan O'Donohoe and Carrie Anne Philbin who ran a session for discovering attendee's interests. It was a case of worlds colliding. I don't think any developers were aware of the challenges that teachers have to deal with. Once they had picked their collective jaws up from the floor, we split in to groups with a view to quickly drafting outlines of schemes of work, lesson plans and other resources. I spent an enjoyable hour creating a draft for a scheme of work about text based adventure games. At the end we came together for a show-and-tell feedback. Highlights included a project to design and deploy a dynamic website, fun with the "turtle" module and a world class game developer (Dan Pope) demonstrating simple, short and easy to understand graphical games.

The Python community are engaging with teachers and making education a central part of community events. Python groups around the world are running courses (see "Open Hatch" in the US), visiting schools (as in Brazil) and writing software (the "turtle" module is one among many such efforts). In March at Pycon in California there will be an education summit to pool international efforts. With teachers as enthusiastic as Carrie Anne, Alan and colleagues in CAS, and the Python community waking up to the fact that they can help educational efforts, it would be foolish not to collaborate and help the next generation of hackers discover their programming chops. If you are a teacher or developer interested in collaborating on Python as a vehicle for learning then you should join Carrie Anne's Python EDU mailing list or get in touch with the Python in Education Special Interest Group. Links in the supplement.

COMPUTER SCIENCE 4 HIGH SCHOOL

Improving computer science education starts long before a university Computer Science degree. The annual Google Computer Science 4 High School (CS4HS) programme provides funding for university-led Computer Science education projects such as teacher training workshops and curriculum development projects. In 2012, 50 projects across Europe, the Middle East and Africa were funded including 4 in the UK. See the link in the supplement for more details. The portal for applications closes on February16.

A GOOD THING? CODE CHECKING EXTENSIONS AND PROJECT ROSLYN

In a previous issue of SWITCHEDON John Stout discussed the Halting Problem: whether a program can, given any program and the data it operates on, decide whether it will halt. A proof by contradiction shows it's not possible to write such a program.

It is possible though to write code to detect whether **some** programs halt just by inspecting the code. Microsoft's Project Roslyn 'exposes' the inner workings of compilers, and lets you write extensions to examine the program text, warning you of potential problems. They provide a sample to detect variables never modified, warning that it can be declared as a constant.

The code below (full extension in the supplement) detects Visual Basic *Do Loop* structures that don't have an *Exit Do* within them. *GetIssues* returns an empty list of *CodeIssue*s if *node* contains an *Exit Do*, otherwise a single *CodeIssue* which 'green squiggles' the start of the loop and a tool tip explanation. This uses a recursive function (*nodeHasNoExitDo*) which checks for an *Exit Do* within the structure, or its substructures, or their substructures and so on. This is not a solution, even a partial one, to the halting problem, but it poses interesting questions.

Extensions like this can look at the code a student writes and point out potential errors, not in syntax (*Dum x As Integer*), but in the 'grammar' of the program. Potential corrections can even be suggested, and with a click, the source code altered. I still write *Do While/Do Until ... Loop*s and forget to write code to change the Boolean condition. Currently the extension doesn't check *Do Loop*s with tests at the top / bottom but it wouldn't be hard to implement the two missing *Case* clauses (*DoLoopTopTestBlock* and *DoLoopBottomTestBlock*) to detect and warn me about it. Perhaps we could use it to keep track of the errors each student makes and use that as a diagnostic tool. Well, we can do it, but should we? I'm in two minds. It could help students starting by, metaphorically, leaning over their shoulder and pointing out errors. However, the bit of me that started programming using punched-cards feels it's maybe too much of a good thing.

```
Public Function GetIssues(...,
                          node As CommonSyntaxNode,
                          ...) ...
   Dim issues As New List(Of CodeIssue)
   Select Case node.Kind
      'Currently only check Do loops WITHOUT top or bottom tests
      Case SyntaxKind.DoLoopForeverBlock
          If nodeHasNoExitDo(node)  Then
             issues.Add(New CodeIssue(CodeIssue.Severity.Warning,
                                      New TextSpan(node.Start, 2),
                                      "Your Do Loop is potentially infinite"))
          End If
      Case SyntaxKind.DoLoopTopTestBlock, SyntaxKind.DoLoopBottomTestBlock
   End Select
   Return issues
End Function

Function nodeHasNoExitDo(aNode As CommonSyntaxNode) As Boolean
    If aNode.HasChildren Then
        For Each subNode As CommonSyntaxNode In aNode.ChildNodes
            If subNode.Kind = SyntaxKind.ExitDoStatement Then
                Return False
            ElseIf subNode.Kind <> SyntaxKind.DoLoopForeverBlock AndAlso
                   nodeHasNoExitDo(subNode) Then
                Return False
            End If
        Next
        Return True
    Else
        Return aNode.Kind <> SyntaxKind.ExitDoStatement
    End If
End Function
```

WRITING SOFTWARE CHECKING SOFTWARE

Once it became obvious that programmers repeated the errors of other programmers, and also their own, they started writing software to help detect them. In the 70s C programmers used Lint, which warns about possible errors and the Interlisp system featured DWIM (Do What I Mean), which would attempt to correct errors as you wrote them.

Integrated Development Environments combining editor, compiler, and debugger provide facilities like syntax highlighting, and editors with a knowledge of program structure that allowed automatic insertion of constructs. Templates let a `For` loop, for example, be inserted with a quick key press, leaving the programmer to fill in the blanks. Scratch, BYOB and similar languages take this to its logical conclusion.

However the syntax/spelling is not usually the problem, since the interpreter or compiler will always tell you of any such errors. Where we really need help is with logical errors such as off-by-one errors, loops that don't terminate, and eventually with higher-level design errors. Language design helps: having, and using, `For Each` rather than an explicit indexing `For` loop should banish (some) off-by-one errors, automatic garbage collection means that you shouldn't get memory leaks, and explicit variable naming and typing should detect that you've written `acount` when you mean `account`.

Detecting design errors is far more challenging and strays into the realms of Artificial Intelligence, since a program that could detect errors in a program design, not just the program code must have a good representation of meaning, and knowledge of how to accomplish tasks. *John Stout*

A PAUSE FOR THOUGHT

In 'Knights and Knaves' type logic puzzles the knight always tells the truth, the knave always lies, and the spy can do either. The puzzles involve a visitor who meets small groups and is challenged to deduce which are knights, knaves and spies from their statements, for example:

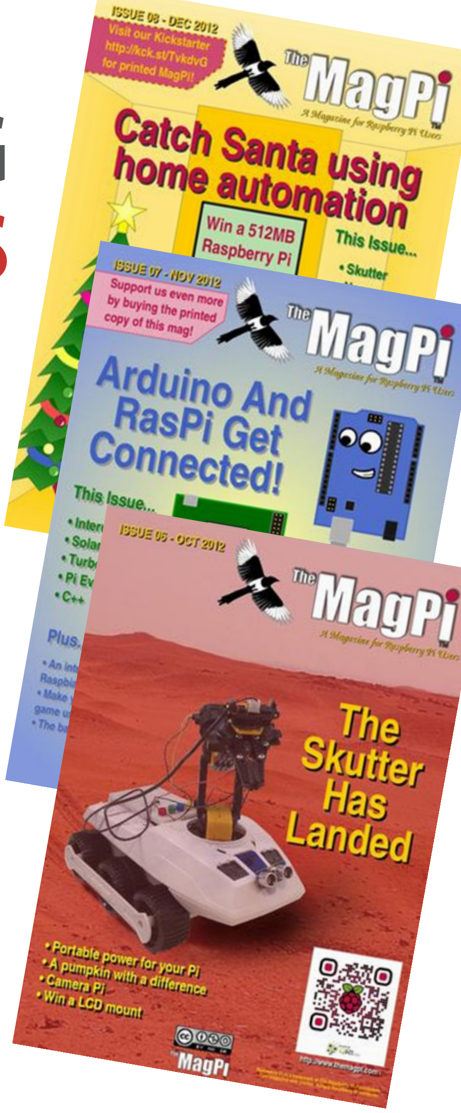
A says: "C is a knave."
B says: "A is a knight."
C says: "I am the spy."

Here, B can't be the knight as if he were A would also be the knight. C can't be the knight, since he would then be lying, therefore A must be the knight, and so C is the knave, and B is the spy. When solving the puzzle it helps to remember that no-one can say they are a knave. There are many variations including requiring other facts to be discovered, and characters using local jargon. These are particularly interesting puzzles which can be solved using laws of Boolean algebra and truth tables.

Often the scenario is an island, or a fork in a road, or doors to heaven and hell where gatekeepers are asked what lies behind their door and the questioner must determine which route is safe. This introduces the liar's paradox where the statement "this sentence is false" always produces a contradiction when tested with binary values, however solutions can be found using truth tables. Raymond Smullyan, philosopher, logician and mathematician, authored several books on logic puzzles, combinatorial logic, and problem solving, including To Mock a Mockingbird and What Is the Name of This Book?

See the supplement for a link to the Wolfram Knights and Knaves Puzzle Generator. *Lyndsay Hope*

ESSENTIAL READING FOR PI ENTHUSIASTS

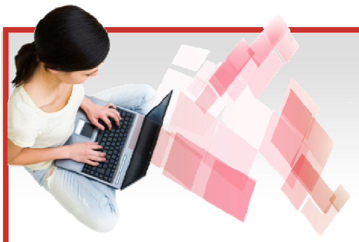

ISSUE 08 - DEC 2012
Visit our Kickstarter http://kck.st/TvkdvG for printed MagPi!
The MagPi
Catch Santa using home automation
Win a 512MB Raspberry Pi
This Issue...
ISSUE 07 - NOV 2012
Support us even more by buying the printed copy of this mag!
The MagPi
Arduino And RasPi Get Connected!
ISSUE 06 - OCT 2012
The MagPi
The Skutter Has Landed
Portable power for your Pi
A pumpkin with a difference
Camera Pi
Win a LCD mount

The MagPi is the only dedicated magazine for Raspberry Pi users. It contains basic instructions, hardware projects and more complicated programming articles.

The MagPi is aimed at children, teachers and developers. The articles are tailored to be accessible from key stage 2, introduce key programing languages and provide encouragement to those constructing more advanced robotic projects. There are regular columns on Scratch, Python and C programming and review articles covering events or important developments.

The MagPi is written by the Raspberry Pi community. The list of authors includes teachers, academics and those working in industry. The magazine is available for free in electronic form, as a PDF file, via the AppStore for the IPhone/ IPad devices and through the Pi Store (Raspberry Pi App store). The magazine is also available for purchase as a full colour printed magazine. The license allows teachers to print an unlimited number of copies of the magazine for use within academic programs. The readership of the magazine has grown dramatically since its launch. It is currently being translated into French, German and Chinese. The goal of the magazine is to reach out and encourage learning using the Raspberry Pi. We are always looking for new articles and would be very interested to hear from ICT teachers. If you would like to submit an article, help with the design of the magazine or contribute in any other manner, please get in touch. Contact details and links in the supplement.

William H. Bell



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: *Mandi Banks, Clive Beale, William H Bell, Jonathan Black, Neil Brown, Steve Bunce, Annelie Chambers, Richard Cornell, Tom Crick, Alison Daniel-Cutler, Sophie Dare-Edwards, Claire Davenport, Roger Davies, Peter Dickman, Laura Dixon, Mark Dorling, Kate Farrell, Rebecca Franks, Andrew Hague, Lyndsay Hope, Claire Johnson, Sasha Laundy, Sophie McDonald, Alan O'Donohoe, Reena Pau, Simon Peyton-Jones, Carrie Ann Philbin, Clarke Rice, Craig Sargent, Martin Saunders, Kate Sim, John Stout, Clare Sutcliffe, Chris Swan, Mark Tennant, Nicholas Tollervey, Betti Williams.*

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