Hampshire Services
HIAS SCHOOL IMPROVEMENT

SUMMER 2014

ICT User Friendly

In this issue:

Hampshire County Council

www.hants.gov.uk

Contents

Editorial

Almost the end of another academic year and we are undergoing changes again.

As you will all know, Stella has now retired and moved to Houston, Texas, with her husband. She has not left Hampshire for good and is returning in July so that she avoids the very hot weather in America – a problem we are unlikely to encounter here. I am already missing her in many ways. Apart from her obvious expertise in primary education, she has always been a calm and steadying influence. I am hoping to recruit someone on a part-time basis to partially fill her shoes, but I know that she has created an indelible legacy within the county – apart from anything else this is evident from the comments made on ICTPrim.

Another significant change to the new computing programme of study is that it has a greater focus on computational thinking. This does not mean that important elements of ICT should not be taught, as it is important that pupils understand how to use ICT as a tool for learning. In this edition of *ICT user friendly* I have included a link to many of the documents that Stella and myself have created to support teachers in delivering computing.

For the rest of the term I am also looking forward to our annual conference hosted by Tim Rylands and Sarah Nield, as well as an opportunity to meet colleagues at Primary Cluster Groups and Secondary Subject Leader Meetings.

Have a restful summer break. Looking forward to working with you in the new academic year.

Sue Savory

County Inspector/Adviser for ICT

Notes for Hampshire computing curriculum materials

All the non-statutory materials that were developed by Hampshire advisory staff are available at the following website:

www.hants.gov.uk/education/hias/computing/ict-resources.

Please note: all documents are draft and subject to change.

These documents are non-statutory and should be used in conjunction with the computing programme of study.

We are working on assessment strategies and criteria so in *Document 3 – Hampshire draft ICT curriculum FS-KS2, 4 October 2013*, the capability statements are shaded and will be changed.

Schools are welcome to use the materials as they are in draft form, but should be aware that there may be some changes before the documents are finalised. If you use these materials and change them to match your school's needs, then they become your schools documents and cease to be the official Hampshire documents.

Document 2 – Key Stage 1 and 2 programme of study provision

This document gives an overview of the computing curriculum, showing what should be covered in each key stage. There are suggested curriculum links and space for schools to enter their own topics. In each year group pupils should have opportunities in the three key areas, but not necessarily in every aspect of the three key areas, eg in Year 1 you might not use a database but you might do some graph drawing. There needs to be enough coverage to ensure that pupils can build on previous experiences and that there are sufficient opportunities for progression.

There is also a list of resources that are commonly used in these areas. This does not mean that these are recommended resources or that schools need to have every item on the list. Schools should edit the resources list to reflect their own resources.

Document 4 – ICT planning or tracking grid KS1, October 2013 and

Document 5 – ICT planning or tracking grid KS2, October 2013

These documents can be used for either planning the coverage of the computing curriculum throughout the year or for tracking what has been covered, or both.

I suggest using one for each year group.

Computing should be used in a curriculum context to support and enhance learning.

When planning, I would always start with the rest of the curriculum. This is best done as a collaborative activity with the whole staff or the staff within a year group or key stage.

We would look at the whole curriculum map and discuss and identify the best matches between the computing curriculum and activities in the other curriculum areas. Some of the computing activities will provide opportunities for teaching new skills, others will provide opportunities to practise, refine and reinforce previously taught skills.

For example, in a *Myself* topic, depending on the age of the pupils, I would probably use some of these: art package; digital photography; some audio devices (talking buttons, postcards, photo albums, microphones, etc); some graph drawing; word processing; DTP; multimedia.

We would annotate the whole curriculum map with ideas, then map them on to Document 4 or 5 and look to see that all areas are covered.

You will probably have more ideas than you have time to deliver so you will need to prioritise.

When this has been done for each year group, we would then track through the progression in each aspect to check that there is enough coverage for pupils to build on their previous experiences, that there are sufficient opportunities for progression, and that there are opportunities for pupils to practise, refine and reinforce previously taught skills. We would also check that the programme of study is covered.

The next step is to identify any new skills that need to be taught. This can be done using *Document 3 – Hampshire draft ICT curriculum FS-KS2, 4 October 2013* or *Document 6 – Computing progression of skills*.

I would suggest using a *just in time* approach to teaching new skills, so that they are taught when the pupils need to know them for a task or activity. Where it is necessary to teach pupils new skills, I would always start with a discussion around what they already know and move on from there. Some pupils may already have the skills they need for the task they are going to do, in which case they could start on the task while you teach the rest of the class the new skills.

Document 7 – Computing planning template can be used for recording the planning for individual lessons or a series of lessons. Schools often have their own planning templates or record the computing planning on a whole curriculum document. It does not matter which method of recording planning is used, as long as it is clear what computing is to be taught and where it is used to support learning in the curriculum.

I often get asked …

I often get asked *"How can we teach programming within schools?"*, but perhaps changing the question *to "How can we teach children to solve problems efficiently using computing?"* would be a more effective way to teaching programming and computing.

When I began training as an ICT teacher (**when?**) it felt like I was teaching by instruction, eg *"Do this … to achieve this …"* This approach involved very little problem-solving, as the students were never faced with an open-ended problem that required critical problem-solving skills. Now that I teach solely computing in my school, my question is now more what do I want children to learn? After that I plan experiences in which they may learn and discover for themselves. I have attended many schools and conferences this year, all with the subject of teaching and planning for computing on the agenda. Most of the events were publicised with a sweetener – free resources for attendance. This is fine, but I worry that the focus on resources and tools will shift teaching back to instruction-based learning. For example, *"Learn Python"* rather than *"Learn to problem-solve using text-based programming languages"*. To all my fellow teachers who are going to be teaching, or are currently teaching, computing, I would ask you not to worry about or feel threatened by all the tools and resources. Rather, think how you can deliver a great experience to your students. In particular, I ask you to think about how you can include the following three simple things in your lesson:

- problem-solving
- experimentation
- discovery.

Nowadays, when planning any lesson, I always try and include these three things. Some computing theory can be very dry, and I find this helps me make it more engaging (think-fetch-execute cycle). The biggest problem I have faced when teaching programming is the transition between different programs. For example, App Inventor is a visual-based programming language that is very intuitive with its drag and drop system. I find children get very frustrated when moving on to text-based languages, such as Python, C#, Ada, etc. This is usually because of the dreaded syntax errors when coding, which then prevent their work from loading or working. This usually creates a negative association with text-based programming, as the failure rate is higher than what they are used to with their visual programming counterparts, such as the aforementioned App Inventor. For example, I experienced Year 10 students, learning text-based programming, stopping at the first problem. They would either wait for me to fix the problem or just sit there quietly and give up. When I did help the ones that were stuck, it created a trend of: *"We will just wait for sir to fix it for us"*, and it quickly turned into me doing the work instead of them. I tried a variety of approaches to encourage children to independently learn and want to program in a text-based environment. However, my attempts did not go too well – and in some cases I am sure it made the children despise me even more than usual.

The solution to the problem came when I read a series of articles about teachers in colleges playing fault-finding games rather than focusing on someone's code. I tried this myself in school, and it worked. Children are far better at debugging and correcting errors in this way than they are at spotting errors as they go along.

The way it works is this:

1 Everybody starts with a script that definitely works, eg an example from a book or their own previously tested coded solutions to a problem.

2 We then test the game, play the game, etc just to make sure that we know it definitely works. Sometimes this step is not necessary.

3 The teacher then asks everyone in the class to think of examples of syntax errors that might prevent the game from working properly. They then share as many of these as they can with their partner in just one minute. Then in groups of four they see if they can come up with any more examples.

4 The teacher then compiles a list of generic examples by asking each group to suggest one typical error.

5 The teacher then explains to the class that they are going to deliberately sabotage the working code by hiding five errors in there. They do not have to hide five different types, but there must be five errors in total; no more, no less. I suggest that one should be pretty obvious and one should be very sneaky, eg using the character zero instead of the letter *O*.

6 Next the teacher asks the pupils to take it in turns to debug the errors. The debugger should be coached by the saboteur, as the aim is for the saboteur to tie hints and clues that eventually enable the debugger to spot all five errors. They then swap over.

7 Finally, the teacher asks for feedback:

> *"Who scored five out of five?"*
>
> *"Who put really sneaky errors in?*
>
> *"What were they?"*
>
> *"Who gets the prize for being the most helpful?"*
>
> *"Who gets the prize for being the most devious?"*

Not only does this approach help children spot errors in their own code more readily, but it also encourages them to consider using their peers for assistance more frequently. The most noticeable difference for me was the students' experience of enjoyment and engagement, as they first deviously hid errors in their own code and then rejoiced as they found errors in their partners'.

Kieren Reynolds

Computing Teacher
Aldworth Secondary School

Computing – statements of progression

By the end of each key stage, pupils are expected to know, apply and understand the matters, skills and processes specified in the relevant programme of study.

For the purpose of identifying progression from Key Stage 1 (KS1) through to the end of KS4 the following statements of knowledge, skills and understanding should support teachers in establishing whether pupils are making expected progress in computing.

Computer science

Lower KS1

They investigate imaginary and virtual worlds and explore options. They explore how devices respond to commands. Understand what algorithms are. Create simple programs.

Upper KS1

They plan and give instructions to make things happen or to control devices and describe the effects. They make informed choices when using ICT to explore what happens in real and imaginary situations. Understand how algorithms are implemented as programs on digital devices and that programs execute by following precise and unambiguous instructions. Create and debug simple programs. Use logical reasoning to predict the behaviour of simple programs.

Lower KS2

They design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems. They use sequence, selection and repetition in programs and use logical reasoning to explain how some algorithms work. They answer questions when using ICT models and simulations.

Upper KS2

They develop simple ICT-based models to explore patterns and relationships, and make predictions about the consequences of their decisions. They create a range of programs and systems. They plan, test and refine sequences of instructions, including solving problems by decomposing them into smaller parts. They should be able to detect and correct errors in algorithms and programs. They work with variables and various forms of input and output. They capture data using sensors to support investigations.

Lower KS3

They create sequences of instructions and understand the need to be precise when framing and sequencing instructions. They are systematic in changing the variables in an ICT-based model and explain the impact of the changes. They can design, use and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems. They understand several key algorithms that reflect computational thinking, for example ones for sorting and searching. Use two or more programming languages, at least one of which is textual, to solve a variety of computational problems.

Upper KS3

They develop, try out and refine sequences of instructions and show efficiency in framing these instructions, using sub-routines where appropriate. They use ICT-based models to make predictions and vary the rules within the models. They assess the validity of these models by comparing their behaviour with information from other sources. They can use logical reasoning to compare the utility of alternative algorithms for the same problem They can make appropriate use of data structures (for example, lists, tables or arrays); design and develop modular programs that use procedures or functions. They can understand simple Boolean logic (for example, AND, OR and NOT) and some of its uses in circuits and programming; understand how numbers can be represented in binary, and be able to carry out simple operations on binary numbers (for example, binary addition, and conversion between binary and decimal).

More able KS3 or KS4

Pupils design and implement systems. They are able to scope the information flow required to develop an information system. They develop, test and refine sequences of instructions as part of an ICT system to solve problems. They design ICT-based models and procedures with variables to meet particular needs. They develop their capability, creativity and knowledge in computer science.

Extension

They design and implement systems for others to use. They develop and apply their analytic, problem-solving, design, and computational thinking skills.

Pupils evaluate software packages and ICT-based models, analysing the situations for which they were developed and assessing their efficiency, ease of use and appropriateness. They suggest refinements to existing systems and design, implement and document systems for others to use, predicting some of the consequences that could arise from the use of such systems.

IT

Lower KS1

They understand that work can be saved and retrieved for later use.

Upper KS1

They can save and retrieve their work for later use. They recognise common uses of information technology beyond school.

Lower KS2

They understand computer networks. They use search technologies effectively. They describe their use of ICT inside and outside school.

Upper KS2

They understand that computer networks, including the internet, can provide multiple services, such as the world wide web; and the opportunities they offer for communication and collaboration. When using search technologies they can appreciate how results are selected and ranked, and be discerning in evaluating digital content. They compare their use of ICT with other methods and with its use outside school.

Lower KS3

They discuss their knowledge and experience of using ICT and their observations of its use outside school. They understand the hardware and software components that make up computer systems

Upper KS3

They understand how components of a computer system communicate with one another and with other systems. They understand how instructions are stored and executed within a computer system; understand how data of various types (including text, sounds and pictures) can be represented and manipulated digitally, in the form of binary digits. They discuss the impact of ICT on society.

More able KS3 or KS4

They take part in informed discussions about the use of ICT and its impact on society. They develop their capability, creativity and knowledge in information technology.

Extension

They take part in informed discussions about the social, economic, ethical and moral issues raised by ICT. When discussing their own and others' use of ICT, they use their knowledge and experience of information systems to inform their views on the social, economic, ethical and moral issues raised by ICT.

Digital literacy

Lower KS1

Pupils navigate on-screen resources to explore and locate information. They capture information and share their work with others. They create content by working with text, images and sound to explore and share their ideas. They talk about their use of ICT.

Upper KS1

Pupils find and use information to answer questions. They sort and organise information and present it in different forms. They store and retrieve digital content. They manipulate and develop their work by using simple editing and formatting techniques. They describe how they use ICT to develop their work.

Lower KS2

Pupils search for and use information from a range of sources and make judgements about its usefulness when following straightforward lines of enquiry. They collect, record and organise data to answer questions and present findings. They use editing and formatting techniques to develop and refine their work to improve its quality and presentation. They use communication tools to share and exchange their ideas with others, and …

Upper KS2

Pupils understand the need for collecting information in a format that is suitable for processing. They interpret their findings, question plausibility and recognise that poor-quality information leads to unreliable results. They create and combine different forms of information using a variety of software (including internet services), refining and presenting it for a particular purpose, showing an awareness of audience and the need for quality. They are able to carry this out on a range of digital devices. They communicate and exchange information and ideas with others, collaborating to develop and improve work. They understand the benefits of online communication and …

Lower KS3

Pupils combine ICT tools within the overall structure of an ICT solution. They undertake creative projects that involve selecting, using, and combining multiple applications, preferably across a range of devices, to achieve challenging goals, including collecting and analysing data and meeting the needs of known users. They use ICT to organise, store and retrieve information using logical and appropriate structures. They assess the use of ICT in their work and are able to reflect critically in order to make improvements in subsequent work. They use appropriate evaluation criteria to critically evaluate the fitness for purpose of their work as it progresses.

Upper KS3

Pupils create, re-use, revise and re-purpose digital artefacts for a given audience, with attention to trustworthiness, design and usability, as well as demonstrating increased integration and efficiency in their use of ICT tools. They develop and refine their work to enhance its quality, using a greater range and complexity of information. Where necessary, they use complex lines of enquiry to test hypotheses. They present their ideas in a variety of ways and show a clear sense of audience. They plan and review their work, creating a logically structured portfolio of digital evidence of their learning.

More able KS3 or KS4

They combine information from a variety of ICT-based and other sources for presentation to different audiences. They identify the advantages and limitations of different information-handling applications. They select and use information to develop systems suited to work in a variety of contexts, translating enquiries expressed in ordinary language into the form required by the system. They consider the benefits and limitations of ICT tools and information sources and of the results they produce, and they use these results to inform future judgements about the quality of their work. They make use of audience and user feedback to refine and enhance their ICT solutions. They develop their capability, creativity and knowledge in digital media.

Extension

Pupils independently select appropriate information sources and ICT tools for specific tasks, taking into account ease of use and suitability. They design successful ways to collect and prepare information for processing.

eSafety

Lower KS1

Use technology safely and respectfully. They use ICT safely by following instructions.

Upper KS1

Keep personal information private; identify where to go for help and support when they have concerns about content or contact on the internet or other online technologies. They use ICT to communicate with others following instructions on safe use

Lower KS2

Use technology safely, respectfully and responsibly and follow strategies for staying safe when using communication tools. They can identify a range of ways to report concerns about content and contact.

Upper KS2

They can manage some of the risks associated with the digital environment when using online communication and recognise acceptable/unacceptable behaviour.

Use technology safely, respectfully and responsibly; recognise acceptable/ unacceptable behaviour; identify a range of ways to report concerns about content and contact.

Lower KS3

They use ICT safely and responsibly. They understand a range of ways to use technology safely, respectfully, responsibly and securely, including protecting their online identity and privacy; recognise inappropriate content, contact and conduct, and know how to report concerns.

Upper KS3

When using online technologies pupils know how to identify and report a range of concerns.

More able KS3 or KS4

Understand how changes in technology affect safety, including new ways to protect their online privacy and identity.

Key Stage 1
Computing programme of study – computer science

Pupils should be taught to:

- understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions
- create and debug simple programs
- use logical reasoning to predict the behaviour of simple programs
- use technology purposefully to create, organise, store, manipulate and retrieve digital content
- recognise common uses of information technology beyond school
- use technology safely and respectfully, keeping personal information private; identify where to go for help and support when they have concerns about content or contact on the internet or other online technologies.

Lower KS1

Computer science – They investigate imaginary and virtual worlds and explore options. They explore how devices respond to commands. Understand what algorithms are. Create simple programs.

	I can:	With support	Independently
Computer science	• work collaboratively with a partner or in a group		
	• use simple adventure games or simulations		
	• follow simple instructions, eg playing at robots, country dancing (pre-Logo activities)		
	• know that many everyday devices respond to commands		
	• give instructions to a programmable robot eg BeeBot, Roamer		
	• use trial and error to create a sequence of instructions to move a programmable robot to a specified location on a grid or mat		
	• use arrow keys or click on arrows to explore a scene or backdrop in a simple on-screen Logo program		
	• use trial and error to move an object to a destination in a scene or backdrop in a simple on-screen Logo program.		

IT – They understand that work can be saved and retrieved for later use.

	I can:	With support	Independently
IT	• logon to the school system and learning platform/online learning space		
	• print my work using the Print icon with support		
	• load programs with support		
	• know that work can be saved and retrieved		
	• save work with support		
	• retrieve work with support		
	• have experience of a range of ICT hardware and software		
	• talk about what they are doing with ICT		
	• use appropriate ICT vocabulary		
	• know that digital pictures and video can be saved on a computer		
	• know that sound can be recorded and played back		
	• use arrow keys or mouse to navigate programs.		

Digital literacy – Pupils navigate on-screen resources to explore and locate information. They capture information and share their work with others. They create content by working with text, images and sound to explore and share their ideas. They talk about their use of ICT.

	I can:	With support	Independently
Digital literacy	**Finding and using information and data**		
	• develop simple classification skills based on practical sorting activities		
	• with support, use simple data plotting/graphing programs to produce pictograms and other simple graphs		
	• discuss the graphs and answer simple questions		
	• with support, use pre-selected web pages		
	Creative/productive use of ICT		
	• put text on screen		
	• use upper and lower case letters		
	• use the Space Bar, the Return key, use the Shift key to make a capital letter		
	• use word lists to enter text		
	• practise keyboard skills using both hands		
	• know that text can be saved and retrieved		
	• be able to use an art package as medium to convey their ideas, as one of a range of media available		
	• use a digital camera or digital video camera to take pictures		
	• with support, add captions or sound to digital pictures or video		
	Sound		
	• use sound recorders/players independently to listen to pre-recorded sound		
	• with support, use music software to explore, create and choose sounds in response to a range of given starting points		

Upper KS1

Computer science – They plan and give instructions to make things happen or to control devices and describe the effects. They make informed choices when using ICT to explore what happens in real and imaginary situations. Understand how algorithms are implemented as programs on digital devices and that programs execute by following precise and unambiguous instructions. Create and debug simple programs. Use logical reasoning to predict the behaviour of simple programs.

	I can:	With support	Independently
Computer science	• work collaboratively with a partner or in a group		
	• create a plan of the steps needed to solve a more complicated problem (an algorithm), eg how can we get the robot to the post office, then the castle, avoiding the graveyard and the lake?		
	• explore different programs logically working through one instruction at a time to predict what they will do, eg if the robot starts here and this program is executed where will it end up?		
	• use algorithms they have created to program a robot to solve a problem		
	• understand that, once programmed a programmable robot can repeat the same instructions		
	• execute their programs and identify errors		
	• talk about how to fix errors in their programs, eg it turned the wrong way after the cottage so I need to change that instruction		
	• fix their programs to achieve the original intended outcome (debug)		
	• create a plan of the steps needed to solve a problem in a simple onscreen Logo type program (create an algorithm), eg how can we get the rocket to the planet?		
	• plan and create a sequence of Logo instructions to move around a scene or backdrop in a simple on-screen Logo program, with a purpose (defined by either teacher or child)		
	• use algorithms they have developed to create programs to move an object within a simple on-screen Logo type program		
	• execute their programs and identify errors		
	• talk about how to fix errors in their programs, eg it turned the wrong way after the cottage so I need to change that instruction		
	• fix their programs to achieve the original intended outcome (debug)		
	• work with a partner or in a small group to solve problems in an adventure game or simulation		
	• use more complex adventure games or simulations.		

Upper KS1

They can save and retrieve their work for later use. They recognise common uses of information technology beyond school.

	I can:	With support	Independently
IT	• print their work using the Print icon independently		
	• save work independently		
	• retrieve work independently		
	• have experience of a range of ICT hardware and software		
	• talk about what they are doing with ICT, describe their work and how they have used ICT		
	• describe their work and how they have used ICT		
	• use appropriate ICT vocabulary		
	• be able to discuss the use of ICT in the world around us and compare to the use of ICT in the classroom		
	• be aware that sound can be recorded on the computer as a sound file		
	• use the cursor (arrow) keys for simple on-screen editing		
	• begin to annotate their work samples using prompts		
	• be aware that work can be saved in different places, eg network, PenDrive.		

Digital literacy	**Finding and using information and data**		
	• independently plot data as a pictogram, block chart or bar graph		
	• be aware that graph types can be changed • interpret the graphs – discuss the graphs and answer simple		
	• use the internet to find information for a topic		
	Creative/productive use of ICT		
	• plan what they are going to do		
	• practise keyboard skills using both hands, try to use more than two fingers, and try to use the thumb on the spacebar – possibly use typing tutor software		
	• make simple modifications to their work (edit)		
	• change the font style, the font size and the font colour		
	• with support, import graphics and add text to a document		
	• be aware of a wider range of tools in the art package		
	• use a digital camera or digital video camera to take appropriate pictures or video for a specific purpose		
	• add captions or sound to digital pictures or video independently		
	• with support, be able to do simple manipulation of images using an art package or other software, eg the digital camera's software		
	• with support, do simple editing of a sequence of digital pictures or video (presentation), eg change sequence, add transitions		
	• create a storyboard (this could be on paper) and with support, use simple animation software to create a short animated film, eg retelling a well-known story		
	Sound		
	• use a range of devices to record and play back sounds, eg voices, instrumental sounds, environmental sounds		
	• use music software to explore sounds and create and play their own compositions		
	• with support, evaluate and modify (edit) their own compositions		
	Electronic communication		
	• know that e-mail exists		
	• with support, write and send a short e-mail, eg to Santa		
	• with support, add comments to a blog or forum.		

Robot Clubs at Crofton Hammond Junior School

I work with Year 6 running Robot Clubs and find that robots can be used in a fun and challenging environment to cover many aspects of the new computing National Curriculum, in particular controlling and simulating physical systems:

"Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts."

National Curriculum in England Key Stage 2 computing programme of study (September 2013)

When I started at Crofton Hammond Junior School 11 years ago, I was studying part-time for my Masters Degree at Portsmouth University in Information Systems. It included a substantial amount of programming; the principles of object-orientated programming, Java, Java applets, SQL (databases), Authorware, Flash and HTML (web design) along with networks and analysing large systems.

At school, I found Lego Mindstorms lurking in the cupboard; it was the RCX version and I started to *play* with it. It was using similar programming principals that I was learning at university and I was really excited. The equipment came with a large manual that I had to get my head around to get to know how the different challenges and programs worked. Using Lego, the children loved building such things as a car with bumpers attached to touch sensors that could be programmed to stop and turn; the ghost ride that triggered a sensor to turn a light on. It was really fun stuff for the children to build and program.

I enjoyed stretching myself, as I also had to learn how to use the Mindstorms software and to organise the learning into logical steps to help the children perform the challenges. It really worked as the children became very engrossed with solving the problems and showing their work off to their peer group and holding school assemblies. There was only ever enough room in the clubs for Year 6. All the lower years would pester me for when it would be their turn.

I only had two kits of the RCX and eventually it broke and I persuaded our Parent Teachers Association to purchase two and a half kits of the next generation NXT Lego Mindstorms. The half a kit came from the idea that I would buy one NXT brick without the building set. I thought that I would make another robot from all the Lego spread through the school, thinking this might save money. However, this did not work, as to make a robot you need to buy all the sensors and motors separately and that cost even more than one set.

The Lego NXT was brilliant. Again I was oversubscribed and numbers were limited, and I would ration the number of children to 12 a term – not ideal, but at least all the children interested were able to have a go. The main fault with the NXT was the main build looked very impressive but was complicated and many children gave up before they even started programming. However, we achieved great programming using the motors and sensors once the robot was built. The NXT Mindstorms program provided a tutorial plan that we worked through that was logical and I set extra challenges at each stage to ensure the children knew what they were doing.

That is the background to how I started to get involved with Lego Mindstorms.

I had always felt that I was on my own running Robot Clubs; a bit like an outpost on Mars! I had continually noticed that over the years when Crofton Hammond Junior School children went to science fairs and similar, if Lego Mindstorms was on display by university students or local companies like Astrium, I always received feedback that we were one of the only visiting schools with knowledge of Lego Mindstorms. Hence, due to the change in the curriculum towards computer programming and the launch of the new EV3, I decided to buy my own robot kits and use the knowledge from my Masters degree and Robot Club experience to run clubs and workshops in other schools as well as my current school.

Since last September I have run after-school clubs at Harrison Primary, Whiteley Primary, Alverstoke Junior and lunchtimes at Crofton Hammond Junior School, with nearly 60 children taking part in my clubs, with day workshops planned in other schools. Disappointingly, only six girls have taken part and currently I am working on more female-orientated tasks.

I run my clubs with the ratio of one robot to two or three children maximum – I have learnt my lesson that too many children to a robot is too distracting. You cannot skimp on equipment as, unless you have all the parts and motors, you cannot use the robot and lessons are delayed. This leaves children disappointed and they lose interest. The EV3 Lego equipment needs time-consuming, routine maintenance – charging the batteries, installing software updates on the laptops and the bricks, sorting the Lego and taking apart and rebuilding when necessary ready for the next session.

In my sessions, I introduce flowcharting at each stage to help co-ordinate their thoughts to systematically break the task down (abstraction). I do this even for a simple task to get them used to the concept of computational thinking. I generally get the children to literally walk through the task before they start to program to produce a step-by-step solution or an algorithm.

The following shows an outline of topics covered during my clubs or lessons.

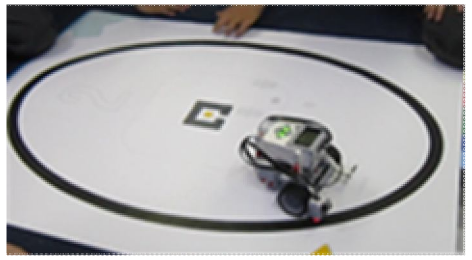
- Initially I introduce the children to the EV3 brick and the Mindstorms program on the laptop. I encourage them to investigate the basics on the software such as motors, sound and pictures and then download their instructions to the EV3 brick. The children are surprised at the immediate response of the EV3 to their instructions when sounds and pictures are displayed and the attached motors move.
- Next the children build a fairly simple robot that takes about 20 to 30 minutes. The length of time varies between seven minutes (yes, can you believe it, as there are certain children that are incredibly quick) to 50 minutes. I do not intervene unless they are desperate, but they usually get there.

Robot

Then it is basic programming to get the robot moving and turning.

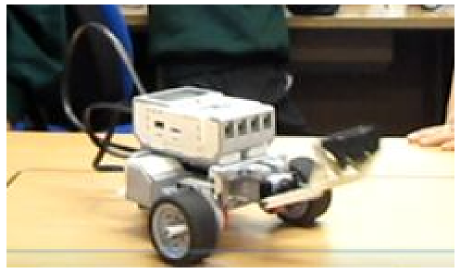
- Depending on the number of weeks for the club/ lessons, I provide a road map for the robots to navigate that requires forwards, turning and backwards. Distances have to be calculated and tested along with angles. This exercise is fairly straight forward but repetitive and starts to get the children very familiar with the software. They are very competitive to complete the task first.
- Then it is on to using sensors; the ultrasonic is always a fun session. The EV3 robots are programmed so that when you put your hand in front of the robot it will stop and possibly turn away and stop. If you put a repeat loop in, it turns away from your hand again and again (iteration). The children enjoy this as it is similar to

developing their own toy and who can blame them as it is fascinating to watch and play with the robot that they have programmed.



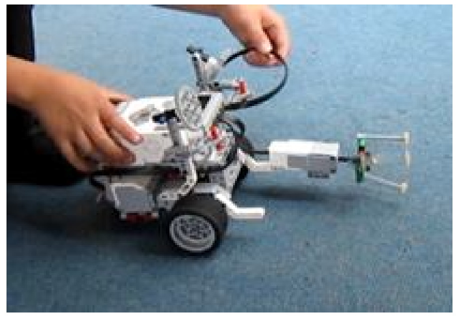
Dark circle map

- The colour sensor is another great sensor to use. You can program the robot to detect a dark line and then give it instructions to move away. Again, by introducing repeat loops it can do it continuously and if you put the EV3 inside a circle drawn with a thick dark line, successful programmers can keep their EV3 robot within the circle. When all the robots are put in the circle together, it is chaotic as the EV3s knock each other over. The last robot standing is the winner and, wow, do the children get enjoyment out of this exercise. Indeed you have to be a good programmer to win.
- There is a third motor that attaches to the EV3 brick that drives a gripper or pick-up object attachment that the children have to build. Once programmed and working, the ultrasonic sensor is then added to detect an object for the attachment to grip/pick up. Programming is becoming more complicated now and it is possibly time to consolidate and regroup.



Robot with pick-up attachment

- The final session of the term is a fun session when the children accessorize the robot. It involves designing, extensive building and working as a team to produce a good working robot.



Accessorized robot with spinning front drill

In the first term of the club/ lesson, we have already achieved much with motors and sensors. The next term involves consolidation and more advanced programming with switches and extra sensors. Recently I have purchased the EV3 Lego Space Challenge. It is perfect to reinforce the sessions that they have previously learned in an exciting futuristic space environment. Depending on the ability of the child, they have to solve challenges that can be completed at many different levels, from easy to advanced.

When I watch children programming and downloading to the EV3 brick to drive their robot, I see the delight on their faces. It reminds me of the quote:

"Children learn best when they are actively engaged in constructing something that has a personal meaning to them – be it a poem, a robot, a sandcastle, or a computer program."

Seymour Papert, Professor of Learning Research, MIT, inventor of Logo computer language and collaborator with Lego to produce Mindstorms.

Huge excitement using light sensors to find the last robot standing

Finally, I think it is important to relate where robots are used, from robotic vacuums, car park barriers and remote medical operations to searching for earthquake survivors. I play many videos to show working robots at the end of my sessions. As I have a keen interest in space and exploration, I introduce workshops with talks to link our robots with the unmanned, *robotic* missions that are currently underway and controlled by Earth to fuel the imaginations of the children. Current missions include Juno, Curiosity, Cassini, Voyager 1 and 2, Rosetta, amongst others.

I like to think that one day our young roboteers will be inspired to play a part in developing future robots to change the world for the better. But if not, at least it will, hopefully, make them think about and be aware of the increasing role of robots in our everyday lives.

Liz Stoneham

Crofton Hammond Junior School

For more information see Liz's website:

www.robotkidsandspace.co.uk

or contact her by e-mail on:

lizstoneham@yahoo.co.uk.

Contact details

Sue Savory, County Inspector/Adviser for ICT

Tel: 023 9244 1442

E-mail: sue.savory@hants.gov.uk

Website: www.hants.gov.uk/education/hias/computing

Moodle: hias.hants.gov.uk/ict